Université d'Ottawa · University of Ottawa

End-to-End Quality of Service Support for Multimedia Applications in the Internet

By

Farag M. Sallabi, B.Sc., M.Sc.

A Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy
in
Electrical and Computer Engineering

Supervisor
Dr. Ahmed Karmouch

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Information Technology and Engineering

University of Ottawa

0-612-72824-2

To my Mother, my Wife
and
my Children (Sundus and Mustafa)

Acknowledgement

I would like to thank my supervisor, Dr. Ahmed Karmouch for introducing me to the exciting field of Multimedia Communications. I am most grateful to his guidance, support and encouragement during my thesis work. I really enjoyed and benefited the time spent in his laboratory and being a member of his group.

I also acknowledge my Ph.D. committee members- Dr. Luis Orozco-Barbosa (University of Ottawa), Dr. Shikharesh Majumdar (Carleton University), Dr. Raouf Boutaba (University of Waterloo) and Dr. Tet Yeap (University of Ottawa) for their comments, questions and criticisms that helped to shape this thesis into its final form.

I am grateful to my family for their support and understanding, special thanks to my wife for her encouragement, support and patience.

The work of this thesis was supported by the Secretariat of Education and Scientific Research of Libya and administered by the Canadian Bureau for International Education (CBIE). Thanks are due to the people of the Libyan Education program at the CBIE: Dr. Mohamed Farhat and Mr. Mohamed Gargoum. Special thanks go to Anna Mastelloto.

Abstract

Quality of Service (QoS) in the Internet is one of the most active research areas in the recent years. With the introduction of sophisticated multimedia applications. researchers have concentrated in finding solutions to make the Internet a suitable environment for such applications. The obvious solution is to enable applications reserve network resources according to their QoS needs. Researchers have also suggested enhancing the existing routing protocols for handling QoS. In this thesis. we present an end-to-end QoS architecture based on resource reservations. The architecture proposes the introduction of domain agencies in network domains. The domain agency is responsible for managing the resources in its domain and reserve resources according to the received requests from the users. It handles immediate and advance resource reservations and adapts the reserved resources according to user interactions. It contains a QoS routing agent to construct a QoS intra-domain path. The QoS routing agent uses a modification of Dijkstra's algorithm to find a shortest-narrowest path that accommodates the requested QoS. We present the analytical study and implementation of the main components proposed in this thesis. Then present implementation of the architecture in an agent environment. which provide a good opportunity to test and analyze the system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation

The advent of high-speed networks with powerful multimedia workstations and new storage technology provides valuable opportunities to design new applications, such as video on demand, video conferencing, distance education, Internet telephony, etc. These applications process more than one media object, such as Audio, Video, Image, Text, etc, which make the applications very sensitive to quality of service changes. Furthermore, distributed system configurations permit several powerful workstations to share resources located at several sites by communicating through the Internet [KAR96]

The present architecture of the Internet provides a very simple service, the best-effort service. Best-effort service does not provide any QoS guarantee and all traffic are treated equally. Using this service, packets compete for the network resources, and traverse the network according to the service of the underlying routing protocol algorithm. In fact, this service is not suitable for continuous real-time traffic that has stringent quality of service requirements. Therefore, new service architectures have come into existence to provide solutions to the QoS problem. The most obvious solutions are to control the number of sessions submitted to the network and enable applications reserve network resources according to their QoS needs.

To improve the existing Internet architecture, for accommodating multimedia applications, the Internet Engineering Task Force (IETF) [IETF] has been developing a number of protocols that manage the transfer of multimedia information between clients and servers. These include protocols for resource reservations, real-time transport, real-time control and real-time streaming of data [BUL97].

The first architectures for resource reservations suggest that applications could ask for resource reservations that start immediately and continue for indefinite periods of time [BRA97, CLA92]. These architectures also face the problem of the scarce of network resources, where most resource reservation requests are rejected. To overcome this problem, new schemes have been proposed to enable applications reserve resources in advance [BER98a, FER95, REI94, SCH97, WIS98, WOL95]. In fact, these schemes do not explicitly solve the problem of the scarce of network resources, but it enable users choose the periods of time in which resources are available. Once the resources are reserved, they will be available to users at the start of their applications. Meanwhile, advance reservations give chance to service providers in managing network resources and put cost schemes on using network resources.

Resource reservation protocols such as RSVP [BRA97] provide a method for requesting and reserving network resources, they do not provide a mechanism for determining a network path that has adequate resources to accommodate the requested QoS. Therefore, if resource reservation protocols are used with the existing routing protocols, which use shortest-path (least-cost) algorithms for path computation, they will load the selected path and leave other paths unloaded. This behavior raises several problems. First, bottleneck problem where all traffic is directed to one path (shortest-path). Second, fewer flows will be admitted due to the lack of alternative paths. Third, less network links utilization, where some links are free or slightly utilized. Last but not least, if the links costs change, to avoid congestion, or some links fail, the traffic will change direction and use another path in which it does not have reservation, and loose the QoS. Therefore, new schemes for quality of service routing have been developed to alleviate those problems.

1.2 The Thesis

This thesis presents a novel scalable solution to end-to-end QoS guarantee in the integrated services packet switched networks. It relies on domain agencies located in network domains to manage network resources and construct a QoS path for applications that require stringent QoS. The architecture also monitors user interactions and updates the network and end systems resources to reflect the invoked user interactions.

1.2.1 Thesis Approach

The main goals of the thesis are to provide a scalable, efficient, and flexible QoS architecture. This is because the Internet is growing very fast and performance is the main concern of the Internet users. The thesis relies on a number of recently developed network protocols and network services introduced by the IETF, such as integrated services networks [BRA94], controlled-load service [WRO97a], resource reservation protocol (RSVP) [BRA97] and Open Shortest Path First protocol (OSPF) [MOY94, MOY98]. The OSPF protocol is the most widely used routing protocol in the Internet. The other protocols have been proven to be useful for the deployment in the Internet, which provide a suitable environment for multimedia applications to be utilized in the Internet. We take advantage of these protocols and services, and propose our work to be complement for providing better scalability and provide new services for Internet users.

1.2.2 Contributions

In this thesis, we present an end-to-end QoS architecture for an integrated services network. The architecture handles each session from its start (negotiation and setup) to its end. It also handles user interactions during the passive and the active periods of the session. The thesis presents the following original contributions:

- **End-to-End Resource Reservations Architecture** [SAL99b]: We present architecture for end-to-end resource reservations with user interactions. This architecture is based on domain agencies, where in each network domain there is a domain agency. The domain agency is responsible for making immediate and advance resource reservations, and then adapts the reserved resources according to user interactions. We also present a framework for domain agencies, which facilitate the communications of domain agencies with the receivers, the senders and the routers.

- **User Interaction Admission Control** [SAL99c]: To this time, several admission control algorithms have been introduced for admitting immediate and advance resource reservations. Our work extends those algorithms by introducing user interactions, which represent one of the most supported features of multimedia applications. We implemented a simulation model to monitor the effects of user interactions on network resources, and on admitted flows.

- **Quality of Service Routing Agent** [SAL99a]: We state a different approach for QoS routing, where we suggest using a QoS Routing Agent (QoSRA) to perform the QoS route calculations for immediate and advance reservations on behalf of the network routers. Therefore, the underlying routing protocol forwards the best-effort traffic, and the QoSRA is invoked only when there is need for resource reservation and QoS guarantees. We also present a QoS route calculation algorithm, which computes the shortest-narrowest path that satisfies the desired QoS. The QoS routing agent uses a modification of Dijkstra's algorithm. We implemented a simulation model to monitor the rejection probability of the immediate and advance requests, and to monitor the utilization of each link in the simulated network.

1.2.3 Thesis Outline

In chapter 2, we provide a background of the subject topic and survey the related work. The complete architecture is discussed in chapter 3. In this chapter we define the architecture components, and present the three-stage reservation process, negotiation, reservation and adaptation. We also present the network domain agency, and define its components and state a protocol for domain agencies to adapt to network changes and how domain agencies communicate with each other and with end systems. Chapter 4 presents the user interactions admission control algorithm, where we present the equations of user interactions and then present the simulation results of the admission control algorithm with and without user interactions. QoS routing agent, and simulation results are presented in chapter 5. In chapter 6, we provide the implementation of the architecture in an agent environment, provide the results of this implementation and show an application example. The work is concluded in chapter 7.

Chapter 2

Background & Related Work

2.1 Introduction

The work on the Internet QoS continues as long as there are new multimedia applications' QoS constraints and new users' QoS requirements. The QoS provision depends on three main topics. The first is to clearly identify the Internet QoS, the QoS parameters, the type of service and the end-to-end QoS architecture that suits the Internet infrastructure. The second is to study the system and the network resources, and to provide an efficient system that manages the sharing resources and maintain their usage. The third topic is to select the necessary protocols and algorithms that support the scalability of the system, so that the Internet can accommodate as much as it can of traditional and real-time traffic with QoS guarantee. This chapter provides background information on the end-to-end QoS and compares the related work to the contributions of this thesis.

2.2 Network Quality of Service

In this section, we provide some basic concepts of quality of service. The quality of service concept is the same for all types of networks, but the way it is implemented is different for different types of networks (ATM, Telephone network, IP networks, etc.). Throughout this thesis, we will focus on IP networks.

2.2.1 Quality of Service Definition

The notion of quality of service (QoS) in communication networks has changed since the emergence of multimedia applications that contain one or more time-dependent streams. The previous definitions of QoS were concentrated on services provided by lower protocol layers and are not applicable to the application layer. Following are some definitions of QoS:

> "QoS is the collective effect of service performance which determine the degree of satisfaction of a user of the service", CCITT I.350 [CCI89].
>
> "QoS is a set of quality requirements on the collective behavior of one or more objects" [ISO95].
>
> "QoS is described in terms of a set of user-perceived characteristics of performance of a service. It is expressed in user-understandable language and manifests itself as a number of parameters, all of which have either subjective of objective values" [RAC91].
>
> "QoS represents the set of those quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application. Functionality includes both the presentation of multimedia data to the user and general user satisfaction. The QoS of a given system is expressed as a set of (parameter-value) pairs, sometimes called a tuple; we consider each

parameter as a typed variable whose values can range over a given set" [VOG95].

[ISO95] and [CCI89] definitions are too general to be meaningful, and are meant for lower layers only. [RAC91] and [VOG95] improve the previous definitions to be more specific and include all layers starting with user perception.

2.2.2 QoS Specifications

To provide an end-to-end QoS for multimedia applications, the QoS provision needs to be supported by all layers of the communication system [NAH95a]. The communication system as shown in figure 2.1, consists of three main layers [NAH95a]; application, system and devices. Hence, this implies three types of QoS; application QoS, system QoS and network QoS. Tables 2.1, show examples of common QoS parameters.

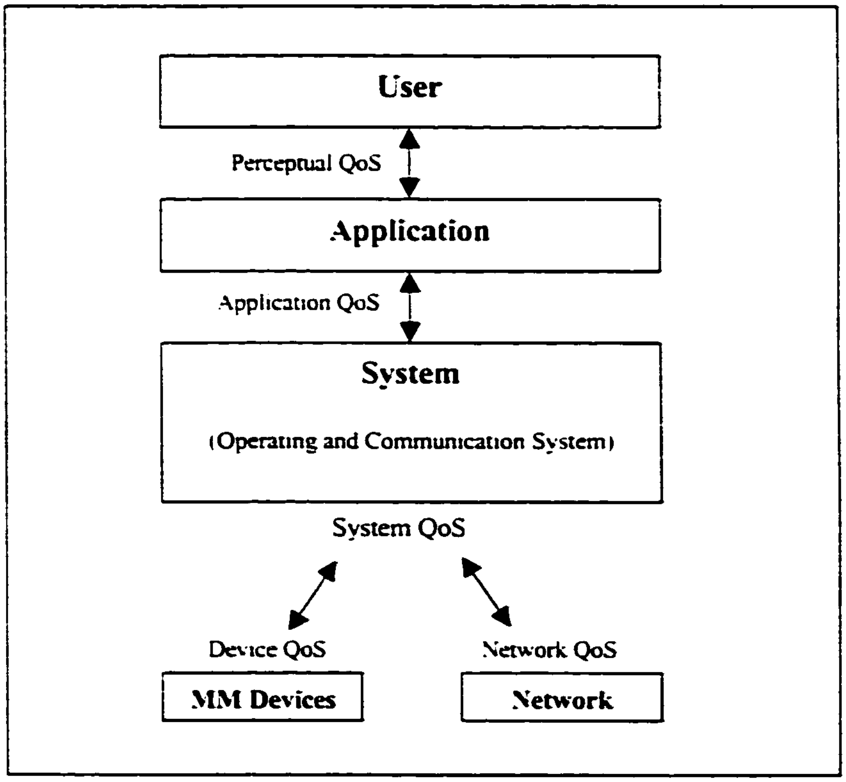


Figure 2.1: QoS-layered model for communications system

Medium	QoS Parameter	Values	Quality
Audio (Application QoS)	Sample Size	8-bit	Telephone voice
	Sample Rate	8 KHz	Intermediate delay 125 µs
	Sample Size	16-bit	CD audio
	Sample Rate	44.1 KHz	Intermediate delay 22.7 µs
Audio (Network QoS)	End-to-End delay	0 to 150 ms	Acceptable for most user application
		150 to 400 ms	May impact some applications
		Above 400 ms	Unacceptable
	Bandwidth	16 Kb/s	Telephone speech
		32 Kb/s	Audio conferencing speech
		64 Kb/s	Near-CD quality audio
		128 Kb/s	CD-quality audio
Video (Application QoS)	Frame Rate	30 f/s	NTSC format
		25 f/s	PAL format
		60 f/s	HDTV format
	Frame Width	≤ 720 pixels	Video signal MPEG
	Frame Height	≤ 576	Vertical size
Video (Network QoS)	Bandwidth	≤ 1.86 Mbit/s	MPEG encoded video
		64 Kb/s to 2 Mb/s	H.261 encoded video
		1544 Kb/s to 2 Mb/s	H.120
		140 Mb/s	TV, PCM coding
		Over 1 Gb/s	HDTV uncompressed
		Around 500 Mb/s	HDTV lossless comp.
		20 Mb/s	HDTV lossy comp.
	End-to-End delay	≈ 250 ms	
Data Network QoS	Bandwidth	0.2 to 140 Mb/s	File transfer
	End-to-End delay	≈ 1 s	

Table 2.1: Examples of audio, video and data QoS parameters

2.2.3 Type of Service

Traditionally, transmission of voice, video and data has been carried over different types of networks: telephone networks for voice, cable networks for video, and computer networks for data. Each network provides its own dedicated resource management that is optimized for its particular usage. With the increased capacity of data networks and the emergence of multimedia applications, it is desirable and feasible to have a single network over which voice, image and data streams share the same resources with appropriate end-to-end user QoS. To balance the complexity for QoS management within the network against the diversity of applications, the network should provide multiple classes of service with different QoS guarantees. Such a network is called an integrated services network [BRA94], because it provides services (e.g., real-time voice, video and best effort) that previously required separate dedicated networks.

The Integrated Services Working Group of the IETF has discussed four separate qualities of services for real-time traffic in the Internet: Guaranteed, Predictive, Controlled Delay and Controlled Load Service. From these, the Guaranteed and Controlled Load services are to become proposed standards, for further deployment and experimentation in the Internet [FLO96].

Guaranteed Service [SHE97a] provides an assured level of bandwidth that, when used by a policed flows, produces a delay-bounded service with no dropped packets. The Predictive and Controlled Delay Services [FLO96], which are not currently planned for deployment, were both intended as real-time services that provide low packet loss. Predictive service was intended for applications that require an upper bound on end-to-end delay. Controlled-Delay Service is intended for applications that can dynamically change the level of packet delay they request from the network, when their current level of delay is not adequate.

In contrast, Controlled-Load Service [WRO97a] is intentionally minimal, offering only a single class of service, and requires an admission control procedure whose goal is simply to ensure that adequate bandwidth and packet processing resources are available to handle the requested level of traffic. Controlled-load service provides the client data flow with a quality of service closely approximating the QoS that same flow would receive from an unloaded network element, but uses admission control to assure that this service is received even when the network element is overloaded. The controlled-load service is intended to support a broad class of applications, which have been developed for use in today's Internet, but are highly sensitive to overloaded conditions.

Flows in an int-serv (integrated-services) architecture are characterized by two specifications [PAR92b, SHE97b, WRO97b]: a traffic specifications, which is a specification of the traffic pattern which a flow expects to exhibit; and a service request specification, which is a specification of the QoS a flow desires from a service elements. The int-serv architecture, which is restricted to the network but applicable in the end-system, too, is comprised of four components [BRA94]:

- ***Packet scheduler***: forwards packets streams using a set of queues and timers. It should be able to implement different QoS requirements of real-time applications [DEM90].
- ***Packet Classifier***: maps each incoming packet into a set of QoS classes. State aggregation is very important at this step as it reduces the packet classifications overhead [BER98b].
- ***Admission control:*** enforces the admission control algorithm to determine whether a new flow can be admitted or denied.
- ***Reservation setup protocol (e.g. RSVP):*** necessary to create and maintain flow-specific state in the routers along the path of the flow.

2.2.4 Resources

Resources are objects that are required by processes to perform certain tasks. A resource can be a hardware device or file system stored locally at the same place of the process or remotely at different locations in the network. Generally, a resource can be anything that can only be used by a single process at any instant of time [TAN92]. Usually, when the requested resource is not available, the requesting process may wait until the resources are free again. This situation is not valid for real-time traffic, because of time delay, the request is rejected instead of being waiting.

For multimedia communication systems, resources comprise all objects required for processing and transmitting multimedia objects. These resources exist along the end-to-end path from source to destination(s). Resources are further classified to active or passive and exclusive or shared. The CPU is an active and shared resource, the memory and bandwidth are passive and shared resources, while the loudspeaker is a passive and exclusive resource. Shared resources have limited capabilities and for processes to efficiently utilize those resources, a resource management system is needed. Figure 2.2 presents the shared resources from end-to-end [WOL98].

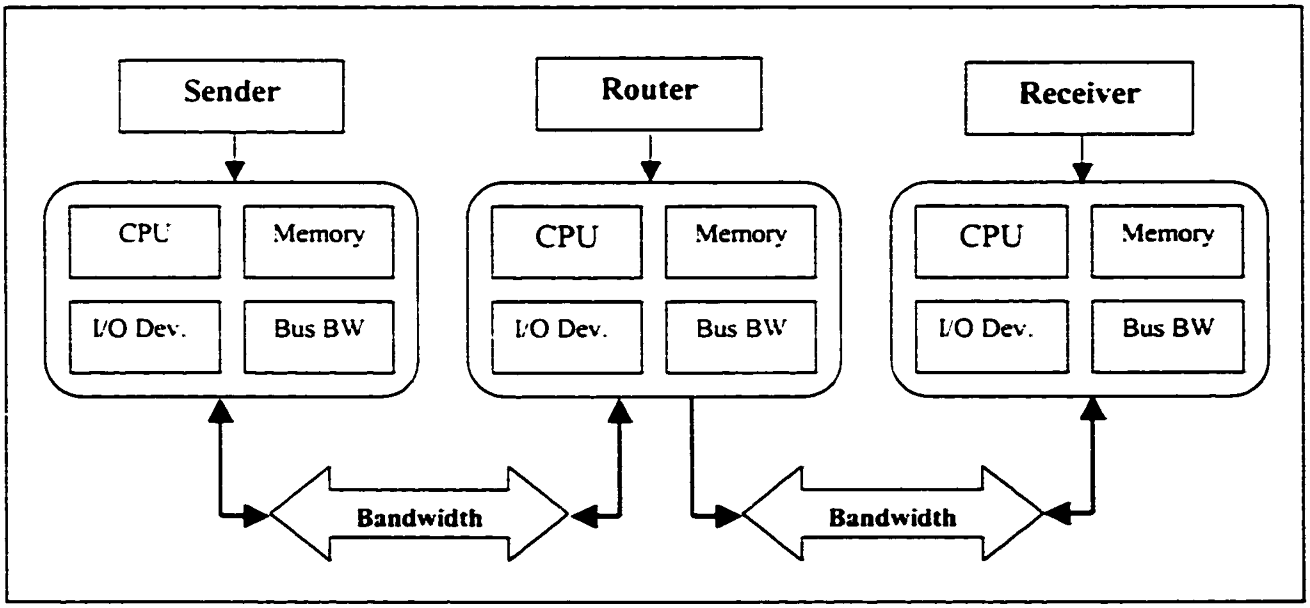


Figure 2.2: End-to-end resources

2.3 Admission Control

The resource management system depends largely on the admission control algorithm to share resources. The appropriate admission control algorithm maintains the resource usage at certain level that protects the QoS assigned to the current sessions (immediate and advance sessions). There are two approaches of admission control, the first approach is a parameter-based admission control and the second approach is measurement-based admission control. The parameter-based algorithms rely on the requested QoS requirements and the calculated QoS values of the current sessions to admit the new flow, while the measurement-based algorithms rely on the measured state of admitted flows and the QoS requirements of the new flow [JAM97].

Admission control can be applied on sessions that request immediate resources as well as sessions that request future resources. The scheduling of resources according to their capabilities and time, allow the admission control to provide better scalability. We will see later that requests that ask for resources after certain time have very high acceptance probability. In this thesis, we also apply admission control to user interactions.

2.4 QoS Routing

The current routing protocols are completely transparent to the QoS needs of real-time traffic. This implies that flows are routed over paths that are unable to support their QoS demands, while other paths that can handle the QoS needs of the real-time flows are available. This of course results in high rejection probability and low resource utilization. QoS routing is recently suggested to serve two different needs, selecting feasible paths that satisfy QoS constraints and increase the utilization of network resources.

2.4.1 QoS Routing Challenges

QoS routing protocols require the enhancement of existing traditional routing protocols or developing new protocols. In fact, QoS routing with multiple constrains renders the process of selecting a suitable path very complex, which raises a questions on deploying the QoS routing in the Internet. Following are some issues for applying QoS routing with multiple constraints [MA98a, MA98b]

- **Computation complexity:** selecting a path with multiple QoS constrains is an NP-complete [GAR79, WAN96]. This could be much complicated and impractical for large networks. For single QoS constraint, the computation may be compared to that of traditional routing algorithms, but the QoS routing algorithm is much more dynamic, as the amount of available resources change for every request.

- **Utilization efficiency:** generally, QoS routing protocols increase the utilization of using network resources. In case of multiple QoS constraints, it is not clear for which QoS parameter to optimize the network. Therefore, different QoS routing algorithms required to handle different traffic needs, in order to achieve higher utilization

- **Inaccurate routing information:** routing protocols rely on global state information (in case of link state routing protocols) or partial state information (in case of distributed routing protocols) to perform their calculations and find routes. For dynamic networks, the network state information in the routers may be not consistent, which produces false routes. Therefore, certain techniques have to be used to check the consistency of links states; this may imply increasing the frequency of exchanging the state information database, which also leads to another problem (wasting resources).

2.4.2 Agent Technology

Quality of service provisioning in computer networks requires at least three processes, QoS negotiation, resources setup and management processes. The three processes require the exchange of messages between end systems (which require the presence of the user), and between end systems and the network elements. The negotiation process continues in exchanging messages until a final decision is made. In fact, this process may take long time to finish, depending on the resource's availability.

The alternative way of exchanging messages, for QoS negotiation, resources setup and management is to use Intelligent Agents (IA). IA work on behalf of users or other entities to accomplish certain tasks. For example, in the negotiation process agents are provided with the necessary information that enable them act intelligently and perform their job without the intervention of users. There are two classes of agents: static and mobile agents. Static agents accomplish their task, in the host node, without the need to move to another node, while mobile agents have the ability to move from one node to another, in order to cooperate with other agents and/or software to finish their work. In fact, if those types of agents are used effectively in the network domains, the QoS negotiation and resources setup processes could be made much easier, scalable, flexible and fast. Following is a short description of the agent technology.

Pattie Maes, one of the pioneers of agent research, defined agents as "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed"[MAE95]. The distributed nature of the Internet has contributed in the addition of mobility to be one of the agent behaviors. In fact, agents should be able to communicate and cooperate with local and remote agents and should be able to migrate to remote hosts to operate closer to physical data stores [FAL98].

Nwana defined mobile agents as "Computational software processes capable of roaming wide area networks (WANs) such as the www, interacting with foreign hosts, gathering information on behalf of its owners and coming 'back home' having performed the duties set by its user". These duties may range from a flight reservation to managing a telecommunications network [NWA96]. The idea of a self controlled program execution near the data source has been proposed as the next wave to replace the client-server paradigm as a better, more efficient and flexible mode of communication [PHA98]. Agents exhibit some properties, which make them a best candidate for being used in QoS management in computer networks. Those properties have been defined by [NWA96] that include the following and not limited to:

- **Autonomy:** The ability to operate without the human intervention.
- **Cooperation:** The ability to interact with other agents and with humans via some communication language.
- **Learn:** The ability to learn as they interact with external environments.

2.5 Related Work

The literature on QoS contains many proposals for fulfilling the requirements of real-time applications. Most proposals agree that the best way to guarantee QoS is to provide some sort of resource reservations in the network elements. Resource reservations could be done immediately or in advance, to reflect human life in arranging and monitoring their activity schedules. To increase the admission rate of real-time applications several proposals for improving the routing protocols, to be a QoS sensitive, have been suggested. The following two sections describe the related work and the third section discusses this work with respect to our work.

2.5.1 Quality of Service and Resource Reservations

The Tenet suite [BAN94], developed by Tenet group at the University of California at Berkeley, is one of the first implemented architectures to support resource reservations. This protocol suite is capable of transferring real-time streams in packet switched networks with guaranteed quality. The architecture depends on a family of developed protocols. The first protocol is the Real-Time Channel Administration Protocol (RCAP) [BAN91], which provides a connection establishment, resource reservations and signaling to the other protocol family. The second protocol is the

Real-Time Internet Protocol (RTIP) that provides an unreliable, simplex, guaranteed-performance and in-order packet delivery service. The other family protocols, Continuous Media Transport Protocol (CMTP) and Real-Time Message Transport Protocol (RMTP), which run on top of the RTIP and works as transport protocol. An extension of the Tenet scheme [FER95] is added to allow for advance reservation. The idea behind this is to partition the network resources into two partitions one for immediate reservations and the other for advance reservations. They also discuss the issues of connection establishment and management of the advance reservation tables.

In [CAM93, CAM94] Campbell et al provide a QoS architecture, which considers QoS services and mechanisms in different layers of end-to-end transfer service. The upper layer is a platform for distributed applications combined with services for QoS configurations, and multimedia communications. Below this layer is the orchestration layer which provide inter and intra-flow synchronization. Then the transport layer, which contains a range of QoS configurable protocols. In their service contract structure they specify a clause, which allows the user to ask for fast reservation or forward reservation. However, they state that this service is a marker for further study and they do not go into detail for this service.

Reinhardt proposes in [REI94] an extension to ST-2 [DEL95] protocol to include advance reservation. He states an admission control algorithm for advance reservations and uses a calendar to store already confirmed reservations. He does not cover scalability and state explosions in the intermediate nodes.

The Heiprojects [VOL96] at IBM'S European Networking Center in Heidelberg developed a QoS model that provides end-to-end QoS guarantee. The model considers all resources through which the multimedia stream travels. The model relies on HeiTs (transport system) for transmitting multimedia streams across the network and HeiRAT (Resource Administration Technique) for providing a well-defined QoS for this transport system. The HeiRAT provides admission control, QoS calculation, resource reservation and resource scheduling.

OMEGA [NAH95a] provides an end-to-end QoS architecture for distributed applications. The main task of the OMEGA architecture is resource reservation and management of end-to-end resources. Communication is proceeded by a call setup phase where application requirements, expressed in terms of QoS parameters, are negotiated QoS guarantees are made at several logical levels, such as between applications and the network subsystems, applications and the operating systems, network subsystem and operating system. This establishes customized connections and results in the allocation of resources appropriate to meet application requirements and operating system/network capabilities.

Wolf et al in [WOL95] explains a model for resource reservation in advance. They present some issues, which must be resolved in order to enable the system to make advance reservations. The model is a general overview and does not go into detail of admission control.

In [SCH97] Schelen and Pink propose an architecture that enable clients make advance reservations through agents. They show that network resources can be shared between immediate and advance reservations without being pre-partitioned.

Berson et al [BER98a] give reservation architecture similar to that given by Schelen and Pink, except that Berson et al assume the existence of an inter-domain multicast routing protocol, which gives information to construct the distribution trees.

Wischik and Greenberg propose in [WIS98] an admission control algorithm for advance reservations. The algorithm is based on the theory of effective bandwidths. At the start of the reserved flow, if there is overflow in the bandwidth the most recent advance flow is preempted.

Hafid et al [HAF96] propose a QoS negotiation approach with future reservation. The client and the server engage in the negotiation process and instead of sending a reject response to users, when there are no enough resources, they send back a proposals of degraded QoS at the same time or available future time. They also propose a solution for cost and scalability by encouraging users to use shared resources.

2.5.2 Quality of Service Routing

To further enhance the QoS requirement of the multimedia applications, new schemes for QoS routing have been developed and made feasible for deploying in the Internet. These schemes rely on finding the best path from the source to the destination that fulfils the QoS needs of the applications.

Z. Wang et al [WAN96] discuss the complexity of finding paths subject to multiple constraints, the selection of metrics for QoS routing and present three path computation algorithms for source routing and hop-by-hop routing.

A. Shaikh et al [SHA98] introduce efficient mechanisms for pre-computing one or more routes to each destination, based on the most recent link-state information. Then at the arrival of requests, the suitable path that satisfies the QoS needs is selected.

G. Apostolopoulos et al [APO99b] describe extensions to the OSPF [MOY98] protocol to support QoS routes. They focus on the algorithm used to compute QoS routes and on the necessary modifications to the OSPF to support QoS.

Zhang et al [ZHA97] describe series of extensions to OSPF and MOSPF [MOY94] that can be used to provide QoS routing in conjunction with a resource set-up protocols. Advertisements indicating the resources available and the resources used are advertised to the OSPF routing domain and paths are computed based on topology information, link resource information and the resource requirement of a particular data flow.

Crawley et al [CRA98] propose a framework for QoS-based routing in the Internet. The framework is based on extending the current Internet routing model of intra-domain and inter-domain routing to support QoS.

2.5.3 Discussion of Related Work

All the above mentioned work for QoS and resource reservations consider each node in the path from the receiver to the sender responsible for reserving the requested resources and maintaining a state for accepted sessions. This would overload the network elements and degrade their performance, and this would be even worse in case of advance resource reservations. Schelen et al [SCH97] and Berson et al [BER98a] suggest using agents or servers in network domains to solve the problem of scalability and state explosion, but they did not go into details of the architecture of the agent and how it will be deployed in the network domains. Agent technology promises to provide better negotiation and network resources management, but none of the existing QoS architectures provide a complete model, which relies on agents (static and mobile), to handle the different processes of QoS negotiation, resource setup, reservation and resource adaptation. None of the architecture discusses the effects of the user interactions on the reserved resources (changes in reserved resources and/or changes in sessions time durations), and how to deal with such interactions. These interactions should not be neglected which may introduce disruptions during the session presentation.

Providing QoS sensitive routing is very useful in enhancing the scalability of multimedia applications by providing alternative paths, but it is very complex and may overload the routers. If we introduce advance reservations the task will be much more difficult and routers will not cope with the new schemes of QoS routing. The mentioned work in QoS routing suggests enhancing the existing routing protocols for finding the QoS routes instead of the shortest routes. Therefore, each router is responsible for calculating the QoS path for each request, or calculating all possible routes and then finds the path that satisfies the received request. If we add the immediate and advance resource reservations to the router's duties, the router will be very loaded and its service is degraded. In our approach, we propose a QoSRA, which calculates the QoS path for each router. The QoSRA handles the routers that belong to a certain network domain as specified by the OSPF routing protocol.

Chapter 3

Resource Reservations Architecture

3.1 Introduction

The provision of QoS in the Internet depends on every element in the network and on the end systems. To make an admission decision for a certain session, we need a global knowledge of the available resources through which the flow is expected to travel. If admitted, we still need to monitor the services that are allocated to the session to confirm that the session is receiving its negotiated QoS values. To do all this, we suggest an end-to-end QoS architecture. The architecture is based on domain agencies, which are used to manage resources in network domains. In section 3.2, we present the complete architecture and discuss its components. Section 3.3, presents the domain agency in detail and discusses its components. Then we state a communications protocol for domain agencies.

3.2 Overall Architecture

The architecture, figure 3.1, depicts the necessary components for reserving and adapting resources in end systems and network elements. The architecture is designed to perform three different but related phases. The first phase is the negotiation process. In this phase, the end systems (receivers and senders) negotiate the required QoS parameters and reservation periods with each other, then the end systems negotiate the QoS values with the domain agencies in the network domains. The second phase is the reservation process, where the physical resources in the end systems and network elements are reserved for the specified time. The last phase is the adaptation process. In this phase, reserved resources in the end systems and network elements are adapted to reflect user interactions and link failures. Following is a description of the components in the architecture. FIPA-OS will be discussed in chapter 6.

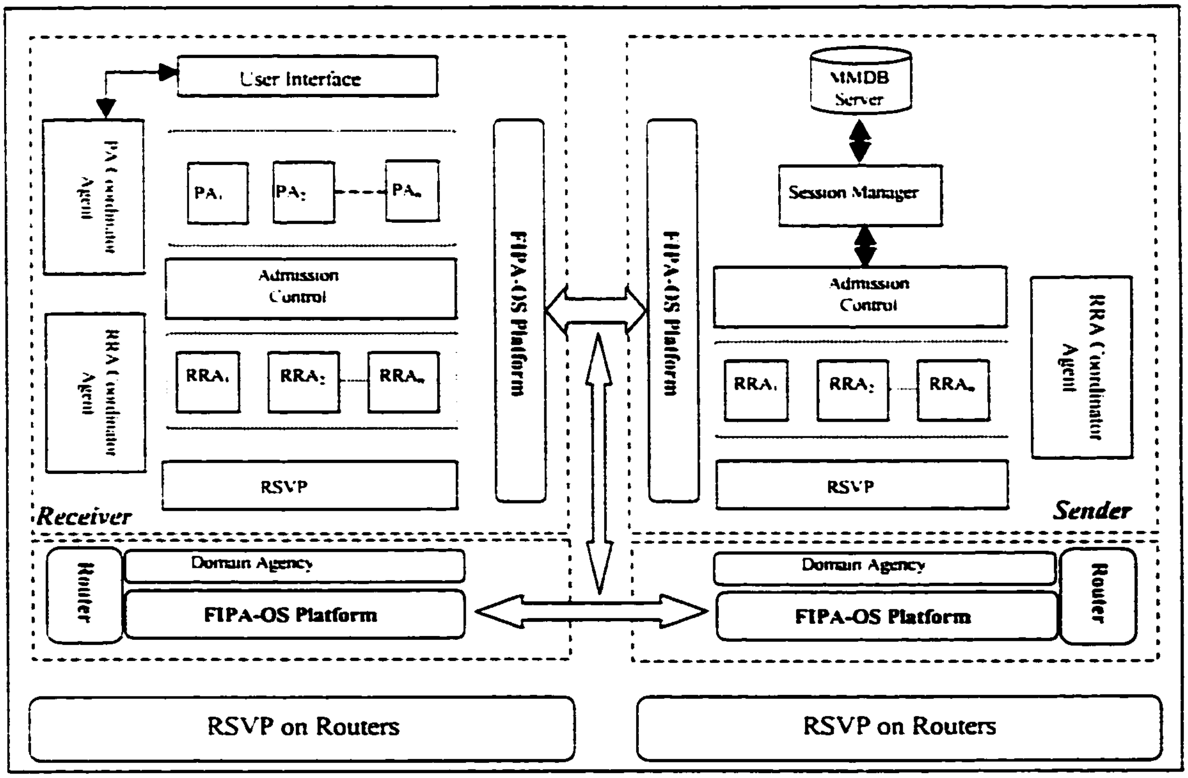


Figure 3.1: Resource reservations architecture.

User Interface (UI)

The user invokes his request through a UI module. The UI sends a message to an agent coordinator asking for a presentation agent. Users have many options to state their requests, to make the negotiation process as flexible as possible. The general format of the request is as follows:

User_Request (Application Address, Type of Reservation, Starting Time(s), Time Restriction, [QoS Parameters])

Application Address: Is the location and name of the requested application.

Type of Reservation: Can be either advance or immediate. If advance, the user should specify Starting Time(s) and Time Restriction. If immediate, those values are set to zero.

Starting Time(s): The user can specify one start time or multiple starting times for negotiation flexibility.

Time Restriction: Is concerned with the start time of the session. User can choose *Exact* or *Relax* commands. In *Exact*, the reservation should be made at the exact times as stated in the starting times. For example, if the user chooses *Exact* and set starting times to 8 and 9, the resources should be reserved at exactly 8 or 9 only. The *Relax* option allows the user to state a time interval. The reservation agent can reserve resources at any time available in this interval. So, if the user chooses *Relax* and gives 8 and 9 as starting times, resources can be reserved between 8 and 9.

QoS Parameters: These are application level QoS parameters, such as video color, frame rate, sound quality, etc. If users do not explicitly specify QoS parameters, the presentation agents will set the default values (capabilities of the system).

Presentation Agent (PA)

The PA is a mobile agent that works on behalf of the user to perform the negotiation process and monitor any user updates. For every user request, a PA is created (by PA coordinator agent) and is maintained during the passive and active periods of the session. The passive period is the period before the playback of the session, and the active period is the duration of the session. When the PA receives the user request, it first consults the admission control unit in the receiver side for available system resources. Then the PA travels to the sender(s) with the common QoS parameters (between user QoS and available system resources).

Admission Control (AC)

The AC is applied at both the end systems (receivers and senders) and intermediate network elements (routers). The admission control should be applied for all resources that are shared between reservation requests. The resources may include but not limited to, link bandwidth, memory (buffer space) and CPU processing power. Therefore, three types of admission tests are needed [NAH95a]; 1) Schedulability test of shared resources such as CPU schedulability and packet schedulability; 2) Spatial test for buffer allocation for delay and reliability guarantees; 3) Link bandwidth test for throughput guarantees. Meanwhile, the AC performs admission test at the negotiation stage and during the passive and active periods of the session for any resource updates. The AC is explained in detail in chapter (4).

Resource Reservation Agents (RRA)

For every successfully admitted scenario, the AC initiates an RRA (through the RRA coordinator agent) and submits to it the negotiated scenario. During the passive and active periods of the session, the RRA monitors the PA for any changes in the scenario, and reports them to the RRA in the sender side.

Session Manager (SM)

The SM coordinates the admission of reservation requests at the sender side. It negotiates with the PAs the user requirements until a final decision is reached. It has access to the multimedia database server from which it obtains the requested scenario. Then, it submits the selected scenario together with the start time of the session to the AC in the sender to perform admission test.

Resource Reservation Protocol (RSVP)

The resource reservation architecture relies on the RSVP for implementing the requested QoS. The RSVP should contain as minimum the signaling daemon, packet classifier, packet scheduler and packet forwarding daemon. It should be installed in the end systems as well as the routers.

Domain Agency (DA)

In each network domain there is a Domain Agency (DA) that manages domain resources and establishes QoS paths for sessions that have requested resource reservations. The DA is located in one of the routers (except the area boarder routers) in each domain, and has interface to the routing protocol and the signaling process (RSVP).

The resource reservation architecture requires certain types of messages to convey the information during the negotiation and the playback processes. Below is a description of the messages. The messages are extensions to the messages used by the RSVP protocol [BRA97].

Path Message: This message is generated by the RSVP in the sender and is carried by the PA to the domain agency in the sender domain. The *Path Message* contains the traffic and time specifications, in addition to other information. After the agency finishes its job, it forwards the PA to the next domain agency in the way to the receiver until it gets to the receiver.

Reserve Message: The receiver sends this message after it receives the *Path Message*. The message follows the reverse path of the *Path Message* and carries the information collected by the PA. The *Reserve Message* is treated as a reserve confirmation message.

Start Message: Just before the playback of every session, the domain agency sends a start message to the RSVP in the routers involved in this session, to set some parameters in packet classifiers and packet schedulers.

Refresh Message: This message is sent from the receiver to maintain the reservations. The message is sent at certain time intervals to the domain agencies, and the agencies are responsible for sending refresh messages to the RSVP in the routers. Time-out of refresh messages will cause the domain agency to teardown the session.

User Interaction Message: User interactions are explained in chapter 4. The interactions may include VCR-like buttons and other interactions that may change the QoS values. The message may be sent during the passive or active state of the session. If there is any user interaction, the action is first negotiated between the receiver and the sender. If it is accepted, the user interaction is then forwarded by the receiver to the domain agencies, to update the reserved resources. The interaction request may come at a time where all resources are exploited, at this time the best-effort service can be used.

Session Teardown Message: This message can be initiated by the sender or by the receiver. It ends a specific session and releases the resources used by this session, including utility agents.

Figure 3.2, shows an autonomous network that has been divided into domains. According to the resource reservation architecture, there should be a domain agency in each domain. The domain agencies are classified into two types, intra-domain agencies and inter-domain agencies. Intra-domain agencies are used to manage resources and find QoS paths in the domain while inter-domain agencies are used to manage resources and find QoS paths in the backbone area. The architecture provides two services, the best-effort service and the QoS controlled-load service. The best-effort service is forwarded by the underlying routing protocol. The following sub-sections describe the three phases of the resource reservation process.

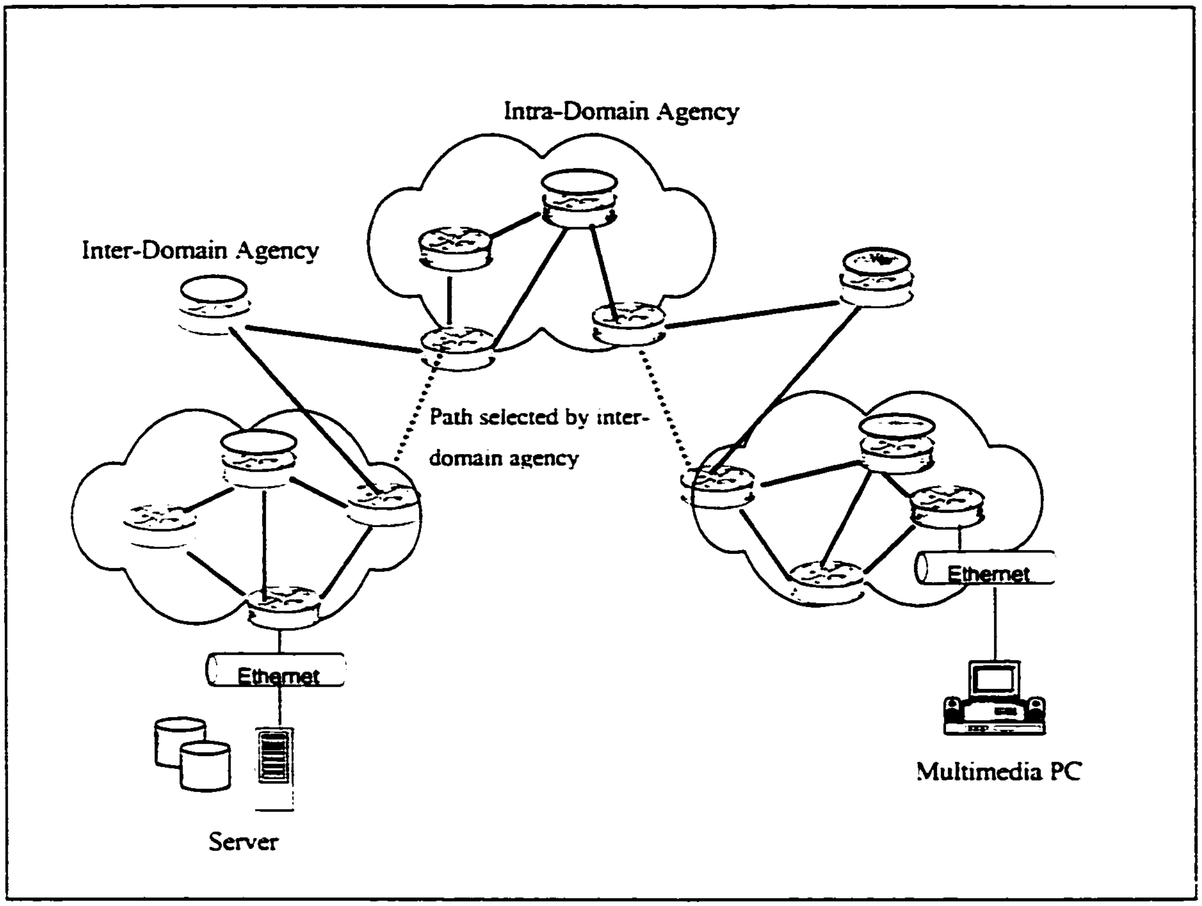


Figure 3.2: Sample network domains

3.2.1 Negotiation Phase

The negotiation process is divided into two stages. The first is between the receiver and the sender, and the second is between the end systems and the network. The reason for this division is that after the receiver and the sender agree on a certain QoS and start time, the sender sends a *Path Message* (carried by PA) to the domain agency in its domain, which starts establishing the routes from the sender to the receiver(s). Then, the receiver confirms the QoS and start time with the domain agencies.

Receiver-Sender Negotiation:

The negotiation process starts as soon as the user invokes his/her request. The *User Request* launches a Presentation Agent (PA), which receives the user information, and then works on behalf of the user to finish the negotiation process. Before the PA goes to the sender(s) site, it consults the admission control unit in the receiver side to make sure that the system can handle the user requirements. After accomplishing this step the PA travels to the sender(s) carrying the information that is common between the user requirements and the available system resources. When the PA arrives at the sender side, it negotiates with the sender the requested scenario and the QoS parameters. The result of the negotiation will be as follows:

$$PA_{load} = \mathit{User\ Requirements} \cap \mathit{Available\ System\ Resources} \qquad [3.1]$$

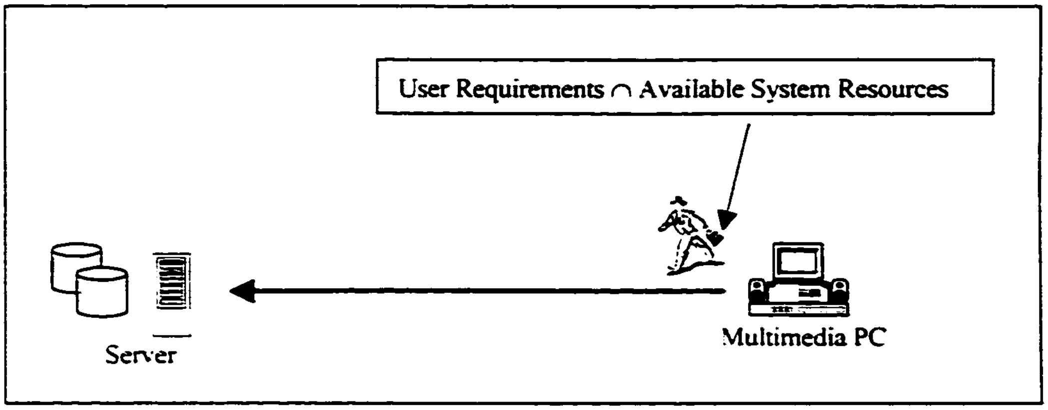
$$\mathit{Scenario}\ (\mathit{Tspec}) = [\mathit{Scenario}_i\,(\mathit{Tspec}) \mid \mathit{Scenario}_i\,(\mathit{Tspec}) \leq PA_{load}] \qquad [3.2]$$

$$i = 1, \ldots, n.$$

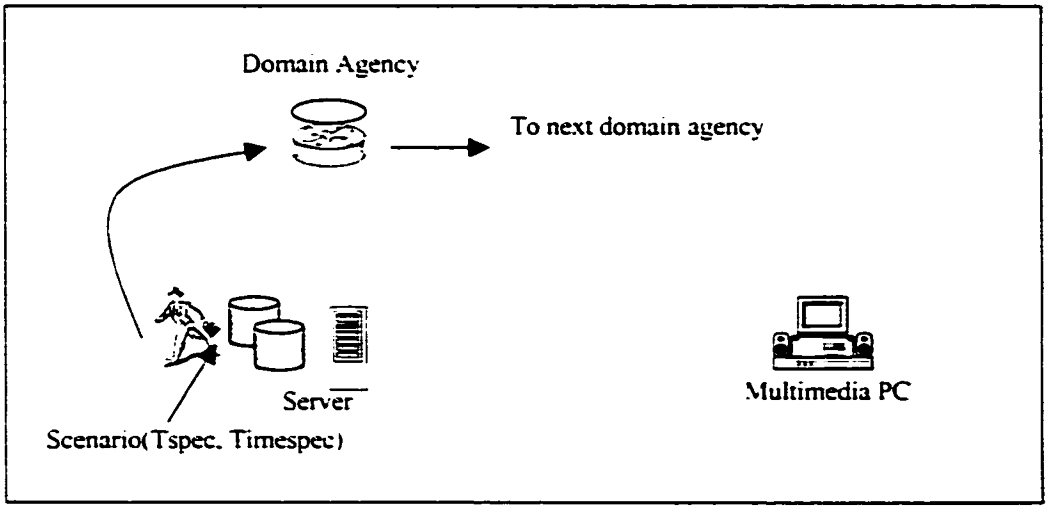
Where $\mathrm{Scenario}_1(\mathrm{Tspec})$ has the highest QoS parameters.

$$\mathit{Scenario}(\mathit{Tspec},\ \mathit{Timespec}) = [(\mathit{Scenario}(\mathit{Tspec}) + \mathit{User\ start\ time}) \mid (\mathit{Scenario}(\mathit{Tspec}) + \mathit{User\ start\ time}) \leq \mathit{Available\ Server\ Resources\ at\ specified\ time\ duration}\ ] \qquad [3.3]$$

Equation [3.1] is a result of applying admission control at the receiver side, the result should satisfy the user requirements in terms of QoS and presentation time, and otherwise the user is instructed to select other parameters. Then in equation [3.2] the SM in the sender retrieves, from the database, the scenario that has the best traffic specifications (Tspec) and suites the PA's requirements. Then in equation [3.3] the SM submits the selected scenario together with the start time (supplied by the PA) to the admission control in the sender to admit the new scenario. If the scenario is accepted, the sender starts negotiating with the network. If the scenario is not accepted, the sender re-negotiates with the PA alternative QoS parameters and starting time. If all alternatives are not accepted the PA returns back carrying a reject message. Figure 3.3 presents the steps of the negotiation.



a. Agent goes to the Sender.



b. Agent carries the *Path Message* and goes to the domain agency in the sender domain

Figure 3.3: Receiver-Sender negotiations

End system-Network Negotiation

After finishing the receiver-sender negotiation, the PA carries the *Path Message* to the domain agency in the sender domain. The *Path Message* contains the session's traffic specifications TSpec, time specification (TimeSpec) and session addresses. The Tspec is used by the routing agent and admission control in the Domain Agency (DA) to prevent over-reservation, and admission failure. The DA establishes a QoS route (according to the TSpec and TimeSpec) in the domain towards the receiver(s), and reserves resources temporarily. Then it forwards the *PA* to the next DA in the way to the receiver. If any domain agency cannot support the QoS parameters carried by the *Path Message*, the PA negotiates alternative QoS parameters according to the information it carries. This process continues until the *PA* reaches the receiver. At this time, the receiver RSVP generates a *R*eserve *M*essage, which carries the QoS parameters supplied by the PA and sends it to the DA in its area. The DA handles the *R*eserve *M*essage as a reserve confirmation, and makes the necessary updates in the reservation state. Then the *R*eserve *M*essage is forwarded to the next DA towards the sender and sends confirmation back to the previous DA or receiver. The process is repeated until the *R*eserve *M*essage reaches the sender. Figure 3.4 shows the flow of messages.

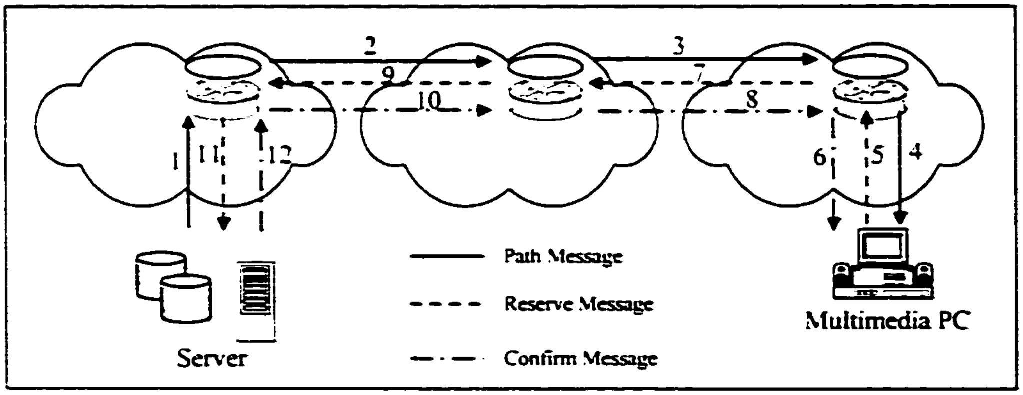


Figure 3.4: End systems-Network negotiations

3.2.2 Resource Reservation Phase

This phase is concerned with making the actual resource reservations in routers and end systems, and maintaining them. Each domain agency and end system checks its reservation state for due sessions. If there are sessions ready to start, the DA sends messages to the RSVP on routers in its domain that are involved in the session. The messages set some parameters in packet classifiers and packet schedulers to obtain the desired QoS. The domain DA is also responsible of maintaining the reserved resources by sending refresh messages at specified time intervals, see figure 3.5. In the end systems, the RRA maintains the reservation states, and sets parameters in its node to provide the due sessions with its requested QoS. It also sends a refresh messages to the DA. As shown in the figure, there is no direct contact between the end systems and the routers.

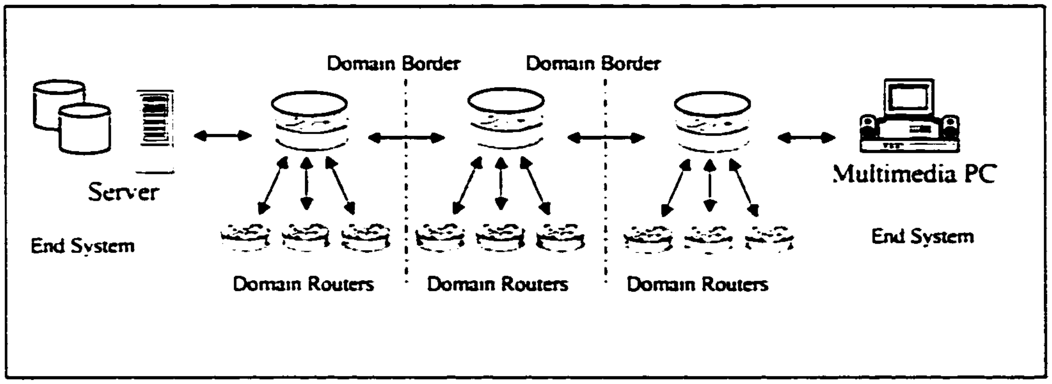


Figure 3.5: Messages flow

3.2.3 Adaptation Phase

Adaptation is concerned with adjusting reserved resources according to user interactions. The user can interactively participate in the presentation of the multimedia application, where he/she can use VCR like control buttons, or link to other scenarios from the current scenario; the user may also make changes to reserved resources not yet started. These interactions will certainly introduce a change in the reserved resources, either in QoS parameters or in duration. Any user interaction is received by the PA that makes the necessary calculations to find the change. Then the PA consults the admission control. If the change is accepted, the PA sends the user interaction request to the receiver RRA, which sends the request to the sender RRA to admit the user interaction request. This step is necessary to check first, if the sender can handle the user interaction request. The sender RRA responds to the receiver RRA by either accept or reject. If accepted the receiver RRA informs the domain agencies with the change for adapting the reserved resources. If one or more network domains reject the interaction request, a best effort service could be used.

3.3 Domain Agency

The best way to manage network resources and provide better QoS in the rapidly growing Internet is to divide the large network domains into smaller manageable network domains. In previous section we have developed end-to-end resource reservations architecture, which provides QoS guarantee for multimedia applications. The architecture is based on domain agencies for managing domain resources and updating them. In this section, we state a framework for domain agencies, and how they work in OSPF configured networks.

3.3.1 OSPF Domains

The Open Shortest Path First (OSPF) routing protocol [MOY98] is an interior gateway protocol where it distributes routing information between routers belonging to a certain area. Each router maintains an identical database describing the area's topology. This database is referred to as the link-state database. OSPF allows collection of contiguous networks and hosts in an Autonomous System (AS) to be grouped together. Such a group, together with the routers having interfaces to any of the included networks is called an area. Each OSPF area is identified by a 32-bit Area ID and runs a separate copy of the basic link-state routing algorithm.

Routing in the AS takes place on two levels, depending on whether the source and destination of a packet reside in the same area (intra-area routing) or different areas (inter-area routing). The OSPF has two useful properties:

- The link-state database gives a complete description of the network; the routers and the network segments and how they are interconnected.
- Since all routers have the same database, any router can calculate the routing table of any other router in its area;

The above two properties have been taken into consideration in designing the domain agency. From the link-state database, the domain agency knows all links and routers in the domain and then manages them locally. The design of QoS routing agent is based on the second property by finding the path from any source to any destination in the domain. Figure 3.6. presents an AS that has been divided into areas.

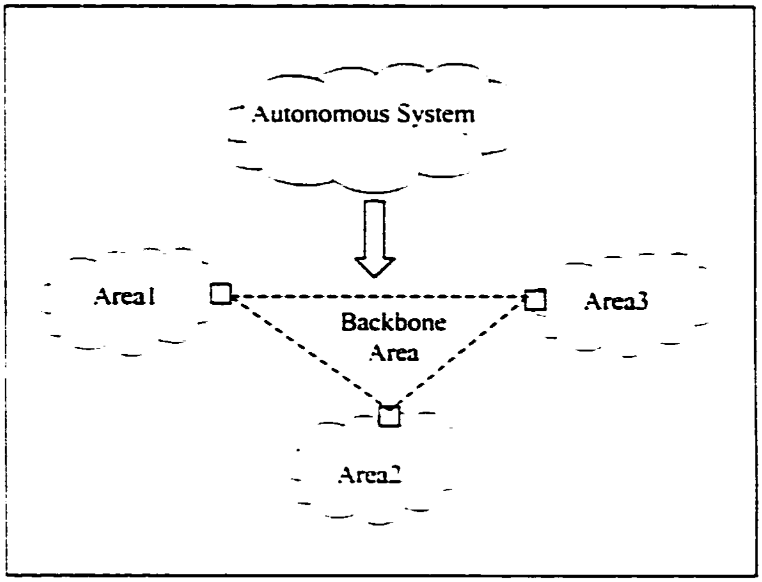


Figure 3.6: OSPF areas.

3.3.2 Domain Agency Architecture

The resource reservations architecture takes advantage of the OSPF domains (areas) and introduces a Domain Agency (DA) in each domain, where it manages all reservations in the domain. The DA receives resource reservation requests from end systems and agencies in other domains. This architecture has been adopted for several reasons. The first reason is to provide better advance and immediate resource reservations scalability. In previous resource reservation architectures, every network element is responsible for reserving resources and maintaining them. Due to the high volume of resource reservations, this would overload the routers and affect their performance. The problem is even worse when advance reservations are permitted for users, which encourage many users to reserve resources in advance. Therefore, by this architecture the agency will take care of every resource reservations and maintain their states. The other reason for using this architecture is the QoS route calculations. Sessions that need specific QoS send their requests to the domain agency, which calculates the best-path in the domain and reserves the necessary resources in this path. The best-effort traffic is forwarded by the underlying routing protocol. This relieves routers from finding the QoS routes on demand, as other proposals suggest. Other reason for using domain agencies is that the domain agency takes care of any problems in the domain and hides them from the end systems. For example, route failure, the domain agency takes care of it without informing the end systems. The agency takes care of refresh messages as well, which allow us to control the flow of the refresh messages from end systems to the domain agencies and from the domain agencies to the routers within the domain. Figure 3.7 shows the components of the domain agency. The agency should reside in any router except the area border routers, which are attached to multiple domains.

The introduction of domain agencies in an AS is not an easy task. Areas in OSPF protocol are dynamic, where system administrators increase or decrease the number of areas in an AS as needed. Therefore, the domain agencies should adapt to this configuration change and react quickly. The following sub-sections describe in some detail the components of the domain agency.

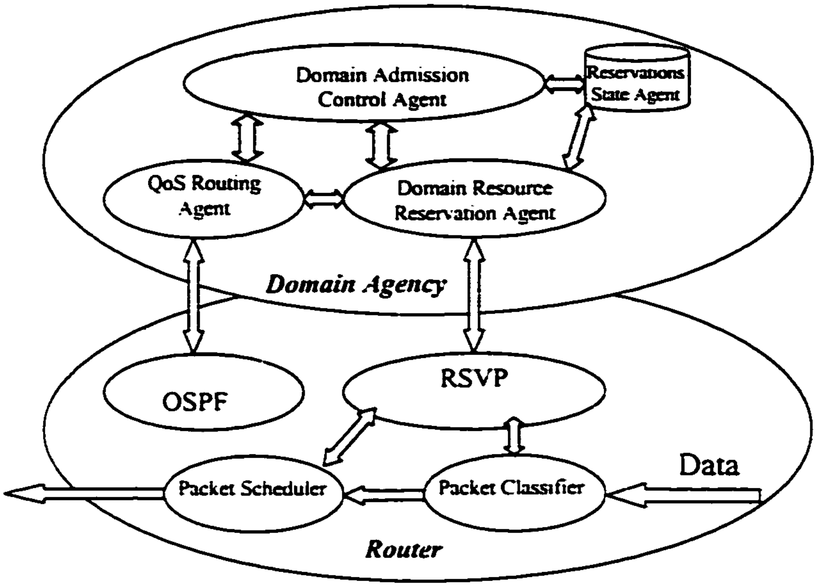


Figure 3.7: Domain agency architecture

3.3.3 Domain Resource Reservation Agent

The Domain Resource Reservation Agent (DRRA) serves as a coordinator in the domain agency. It receives reservation requests, and then controls the operation of the other agents in the domain. Upon receiving the reservation request, the DRRA submits the TimeSpec of the request to the domain admission control agent that calculates the available link resources and returns them back to the DRRA. Then the DRRA submits the Tspec together with the available link resources to the QoS routing agent to find the best path. The DRRA is also responsible for updating the reservation database. This update can be either adding accepted sessions to the database or removing finished sessions. The DRRA forwards user interaction messages to the domain admission control, for admission and then adapts the reserved resources according to these interactions. The DRRA has interface to the RSVP, through which it sends the reservation messages, to set parameters in packet classifiers and schedulers, and refresh messages. In addition to this, the DRRA monitors the configuration change of the domain and sends new domain agencies to new domains.

3.3.4 Quality of Service Routing Agent

The Quality of Service Routing Agent (QoSRA) is introduced to find the best-path that satisfies the QoS needs of the multimedia applications. The QoSRA runs a modified Dijkstra algorithm based on domain topology obtained from the OSPF. and the available link resources obtained from the DRRA. The QoSRA is based on the OSPF property. which enables any router to calculate the routing table of any other router. Therefore. all resource reservation requests are sent to the domain agency. which instructs the QoSRA to construct the QoS path. In fact. this is a new departure from other proposals that suggest enhancing the existing routing protocols to support QoS. In those proposals the calculation is done by every router. which overloads the routers and makes the advance reservations impractical. because of scalability problem.

Two QoS routing calculation approaches are considered in the literature so far [APO99a]. The first is the on-demand calculation. In this approach the QoS path is calculated at the arrival of every request (requests that demand constraint QoS). this calculation is done by every router receives the request. This approach is not scalable for the fast growing Internet and multimedia applications. This situation is even worse if the QoS requests include advance reservations. The second approach is to pre-compute paths to all destinations for each node. Then at the arrival of every request. the path that satisfies the requested QoS is selected to forward the session's packets. This approach has many disadvantages. The QoS paths should be frequently pre-computed to reflect the current available resources. For every arrived request. the QoS paths need to be searched for the suitable route. which adds extra processing. Due to the high demand on QoS. the pre-computed paths are out dated and don't reflect the currently existing resources.

3.3.5 Domain Admission Control Agent

The Domain Admission Control Agent (DACA) has two functions. The first one is to find the available link resources of all links in the domain at specified time intervals and submit this information to the DRRA (to be used by the QoSRA). The second function is to receive user interaction requests from the DRRA and apply admission control algorithm.

The idea of monitoring the available link capacities is as follows. At the beginning the QoS routing agent receives the domain topology from the OSPF and submits it to the DRRA. The DRRA inquiries about each link capacity and submits the complete information to the reservations state agent. At each resource reservation request, the admission control finds the available link resources, for the specified time intervals, then gives this information to the DRRA. The QoSRA constructs the best-path for this request and submits the resulted path to the DRRA. Then the DRRA reserves the resources temporarily until it receives a confirmation message from the receiver.

3.3.6 Reservations State Agent

The Reservations State Agent (RSA) stores all accepted reservations states. For better performance, the RSA is divided into two parts. The first part keeps track of links reservations states, while the second part keeps track of aggregated links reservations states. Figure 3.8. shows a snapshot of resource reservations made in a single link. The numbers indicate session's id, while α indicates a new session asking for admission. As we can see in the figure the duration as well as the bandwidth of each session is clearly defined. Therefore, if we aggregate the reservations states of every link, we get a summary database as described in figure 3.9. This summary database enables the domain admission control agent to get quick values of the available link resources.

The RSA periodically checks the reservation database for due sessions, if there are any sessions ready to start, it sends the session's traffic specification to the

DRRA, which forwards them to the RSVP in the routers. The RSVP then set some parameters in the packet classifier and packet scheduler to achieve the requested QoS. The time of the playback of the scenario is controlled by the RSA, where it sends a start and finish messages to the DRRA.

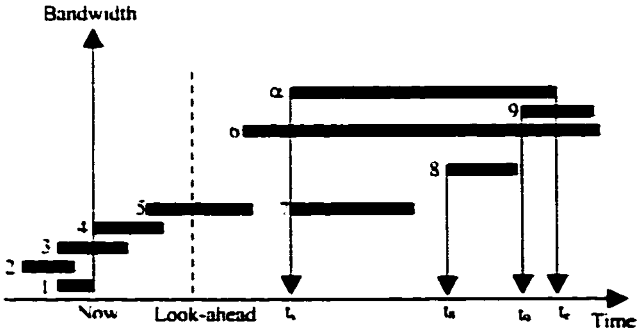


Figure 3.8: Snapshot of reservation state of a single link

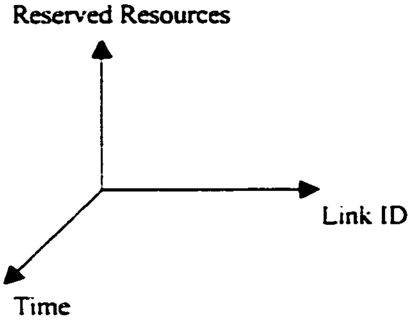


Figure 3.9: Dimensions of the reservation database

3.4 Domain Agency Communications Protocol

In previous sections, we stated how domain agencies receive and send QoS management messages. In this section we state a communications protocol that enable domain agencies exchange messages about domain changes and policies. In fact, as the OSPF domains change, by the network administrators, the domain agencies should react quickly to these changes in order not to loose any information and to adapt to the new configuration. The following sub-sections describe the communications protocol.

3.4.1 Interactions between Domain Agencies

Assume that there is one Autonomous System (AS) and one DA that manages the whole AS resources. As soon as the AS is split into areas, the QoSRA will be aware of this separation from the OSPF. The QoSRA informs the DRRA with the new AS configuration. From this configuration, the DRRA knows the addresses and the number of the new areas. Then it duplicates and sends one DA to each new area including the backbone (the backbone is an area with address 0.0.0.0). Each DA resides in a router other than the area border router.

After each DA settles in its domain, it communicates back with the parent DA to get its information. The DRRA in the child DA gets the topology data from the QoSRA, and sends this information to the DRRA in the parent DA. The DRRA in the parent DA gets the information related to each child DA from the RSA, according to topology database of each child DA, and then sends it back to its child DA. At the same time the parent DA constructs addresses table for the new DAs, and sends this table to every DA. This step is important to enable the DAs to communicate with each other. Figure 3.10, shows the parent DA in the backbone and other DAs communicate with it to fetch their information.

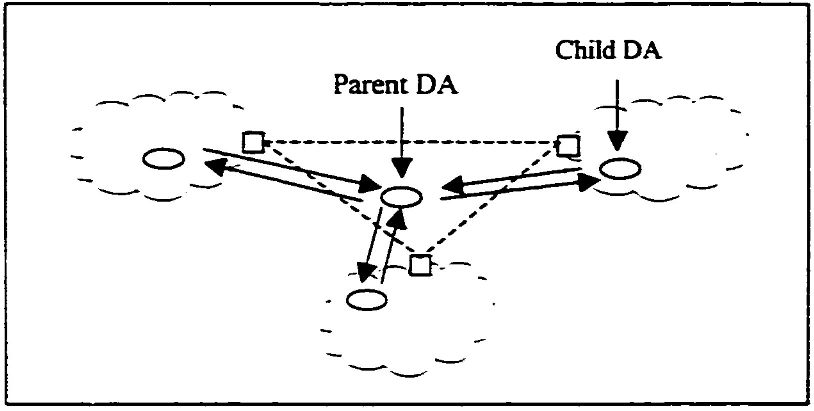


Figure 3.10: Interactions between domain agencies and the parent DA

There are five packets types that are used to convey the information between domain agencies.

Database description packet: This packet is sent by the child DA to the parent DA. It carries the topology database of the child's domain.

Reservations State: For each child DA, the parent DA collects the reservations states from the RSA and sends them back to the child DA. The message also contains the addresses table of the other DAs.

Policy and Service Agreement: Domain agencies need to cooperate with each other to setup policies that each DA should follow. DA could also send a request to another DA to reserve a certain path with certain bandwidth.

Hello Packet: Every DA should send Hello packets to its neighboring DAs, to maintain relationships and check reachability. The DAs should also send Hello packets to end systems, to identify themselves and maintain reachability.

Acknowledgment Packet: This packet is sent between the parent DA and the child DA. It has different interpretation as shown in figure 3.11.

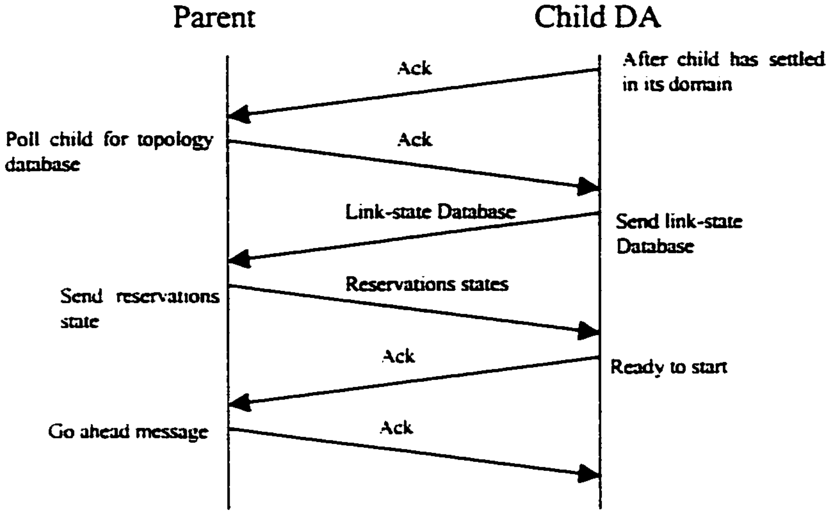
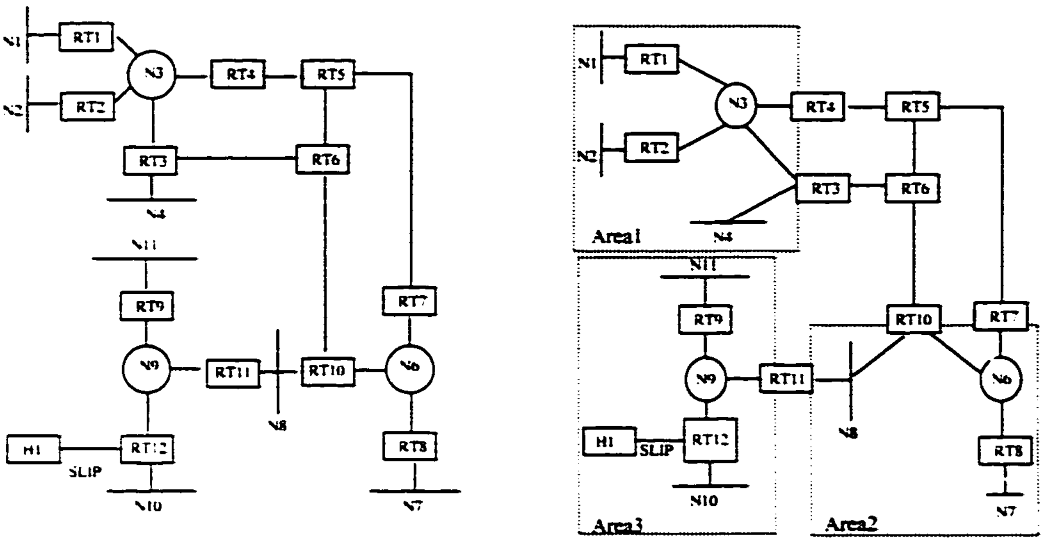


Figure 3.11: Parent DA and child DA setup

3.4.2 Status of Current (Immediate and Advance) Reservations

The domain partitioning and the exchange of configuration messages should not make any disruptions to the currently running sessions, and the current reservations should be valid for the new configuration as well, otherwise the architecture would be useless. Figure 3.12 [MOY98], presents an AS before and after partition. In figure 3.12.a all network elements belong to the same area (AS). In this configuration, only one DA manages the AS resources. If this AS is split into several areas as shown in figure 3.12.b, the DA will react quickly and dispatches DA for each new area, and provide them with the reservations states according to their new topology. All borders that have been introduced to create areas are virtual, the location and interfaces of the network elements have not been changed. Therefore, the current resource reservations will also be applicable for the new configuration, and the active sessions will not be affected.



3.12.a: Before partition

3.12.b: After partition

Figure 3.12: OSPF AS domain

3.5 Summary

Resource reservation is a very effective way of guaranteeing QoS. Multimedia applications can state their QoS demands and negotiate with the network to set aside some resources for handling their data flows. To facilitate the negotiation, we present an architecture, which makes an end-to-end reservation. The architecture also handles user interactions, and updates the reserved resources with any change. To handle the scalability problem and to isolate the receivers and senders form network problems, a domain agency is assigned to each network domain, and makes it responsible for making resource reservations in its area and establishes a QoS route for transferring the data.

Network domains provide a better chance for network protocol developers to scale their protocols and get control of the resources in the domain. The OSPF routing protocol has the advantage of dividing an autonomous system into manageable areas, where each node in the domain knows only the nodes in its domain. We have considered this advantage in designing our resource reservation architecture and assigned a domain agency to every domain. In this chapter, we have provided a framework, which enable domain agencies to react to area splitting and communicate with each other and the end systems.

Chapter 4

Admission Control

4.1 Introduction

Admission control is an essential component in the advance and immediate resource reservation process and QoS guarantee. Applying admission control assures that the network resources are maintained at certain level that enable real-time traffic to perform well in the network. Many admission control schemes have been suggested so far. Admission control schemes belong to two broad categories. parameter-based and measurement-based algorithms. Admission is applied to requests that ask for immediate and future resources. During the playback of the session or the time before the start of the advance sessions the user may invoke some interactions. which may change the state of the reserved resources. Therefore. admission control is also applied to user interactions to ensure that the admitted sessions will not be affected by these interactions. This chapter discusses the admission control algorithms and introduces the algorithm that fulfills the resource reservation architecture needs. Then we present the extensions to the selected algorithm that enable it to handle user interactions. The algorithm is then implemented and tested in two scenarios, one without user interactions and the other with user interactions.

4.2 Admission Control Algorithms Overview

In this section, we discuss four admission control schemes [JAM97], one is parameter-based and three are measurement-based. Parameter-based schemes compute the amount of network resources required to support a set of flows given a priori flow characteristics. The measurement-based schemes, relies on the measurement of actual traffic load in making admission control decisions.

4.2.1 Simple Sum Parameter-based Algorithm

This is the simplest admission control algorithm and is widely implemented in routers and switches. The policy of the algorithm is that the bandwidth of the new flow when added to the total of the reserved bandwidth should not exceed the link bandwidth. Therefore, if v is the sum of the reserved bandwidth, b^{α} the requested bandwidth of new flow α and μ is the link bandwidth (it may be the total link capacity or the capacity assigned for real-time traffic). The algorithm will be as follows:

$$v + b^{a} \leq \mu$$

4.2.2 Measured Sum Measurement-based Algorithm

This algorithm is based on the current state of the admitted flows, where it adds the bandwidth of the request to the measured sum of the existing reserved bandwidth. Therefore, the algorithm admits a new flow if the following equation holds:

$$\hat{v} + b^{a} < \beta\mu$$

$\hat{v}$ is the measured bandwidth of the existing traffic. Because of bursty traffic and delay variations, the measurement-based approach may fail at certain times, so the factor β has been introduced to ensure that the total reserved bandwidth does not exceed certain level.

4.2.3 Acceptance Region Measurement-based Algorithm

The algorithm is proposed by [GIB95], where it computes the acceptance region that maximizes the reward of utilization against the penalty of packet loss. The acceptance region for a specific set of flow types is calculated based on the following parameters:

- Link bandwidth
- Switch buffer space
- Flow's token bucket filter parameters
- Flow's burstness
- Desired probability of actual load exceeding bound

The measurement-based version of this algorithm ensures that the measured instantaneous load plus the peak rate of a new flow is below the acceptance region.

4.2.4 Equivalent Capacity Measurement-based Algorithm

According to [GUE91] the equivalent capacity of a class of traffic is defined as the bandwidth C(ε) such that the stationary arrival rate for the class exceeds C(ε) with probability at most ε. The algorithm proceeds as follows: a new flow is admitted to a class if and only if equivalent capacity for the class (new and existing flows) is less than the allocated bandwidth for that class. The reader is referred to [FLO96] for full details of the algorithm.

4.3 Discussion of Admission Control Algorithms

In this section, we discuss the properties of the admission control algorithm and then choose the algorithm that best suits the advance and immediate resource reservation architecture. Following are the properties:

- Provide session level, not packet level, admission control
- The algorithm should be able to handle immediate and advance resource reservations
- The algorithm should be able to apply admission for active and passive user interactions
- Should be able to look-ahead in time to admit immediate flows
- Ensure that service commitments are not violated
- Provide moderate to high network utilization, while still meeting service commitments
- Provide low implementation and operational cost

The first property can be handled by both parameter-based and measurement-based algorithms. In fact admission control at the session level simplifies the implementation and computational costs. The second, third and fourth properties can only be implemented by parameter-based schemes. Fifth property can be handled by parameter-based measurement-based.

Sugih Jamin [JAM96] studied the algorithms, mentioned in section 4.2, and provided simulation results for the last two points. The simple sum gives the lowest utilization, but he stated that the simple sum algorithm is the only one that can be implemented easily, given the current hardware technology.

Therefore, we conclude that the simple sum parameter-based algorithm is the best candidate to be used in the resource reservation architecture. It is still possible to use a combination of parameter-based and measurement-based, for simplicity we will use the parameter-based algorithm only.

4.4 Effects of Booking Ahead Shared Resources

In this section, we present the effects of booking ahead resources on rejection (blocking) probability. The study was done by Wischik at el [WIS98], where they conducted some simulation experiments to monitor those effects.

In these experiments the authors tested two models of advance requests. The first is Poisson process of mean .9 and the second is a clumped process in which a Poisson number of calls (of mean .45) arrive every .5 time units. The link capacity is 5.5 and all requests have bandwidth 1. the advance request holding times are Uniform(.8, 1.2). The immediate requests arrive as a Poisson process with mean 2.9 and holding times are exponentially distributed with mean 2.7. The result of this experiment is shown in figure 4.1, adopted from [WIS98].

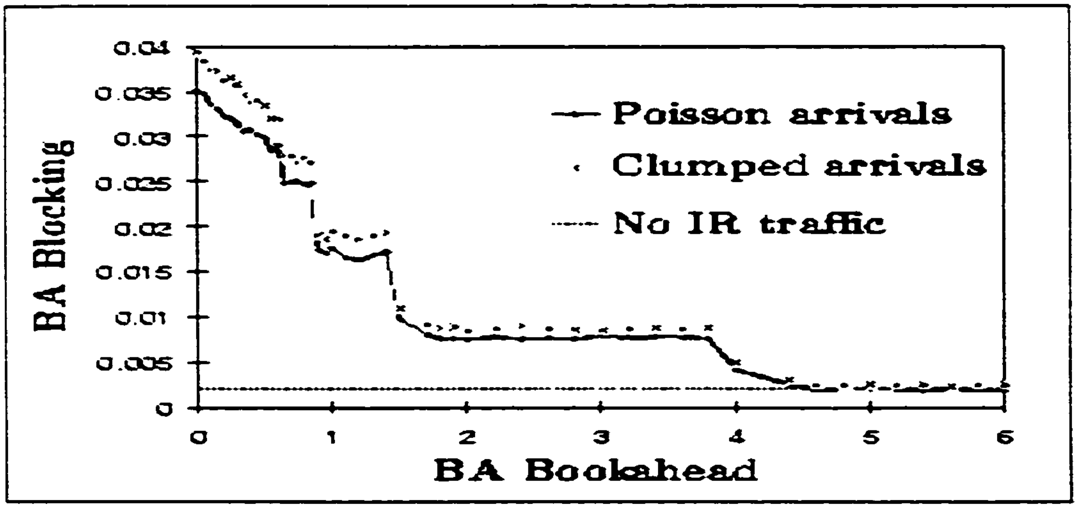


Figure 4.1: Effects of booking ahead on rejection probability

The figure shows that the rejection probability decreases as the book-ahead time increases. The clumped arrivals are used to enforce many booking requests at the same time. There is little difference between the two models. In both source models the rejection probability can be eliminated if the book-ahead time is too far (5 time units as shown in the figure). The authors defined the time at which the rejection probability falls dramatically as a critical book-ahead time.

4.5 Admission Control For Resource Reservation Architecture

This section presents the admission control algorithm used in the resource reservation architecture. The algorithm not only admits flows, but it also maintains the reserved resources of the active sessions to their negotiated values. Figure 4.2 depicts a snapshot of a bandwidth-time reservation state of admitted immediate and advance flows. The numbers in the figure represent the active and passive sessions, while α represents a new session asking for admission. The look-ahead time is optional and is usually tuned to give an acceptable rejection and preemption probabilities of the immediate sessions. As mentioned previously in this chapter the admission control algorithm is based on a simple sum parameter-based scheme. So we follow this in the admission of immediate and advance requests and provide (based on this scheme) a set of equations for admitting user interactions.

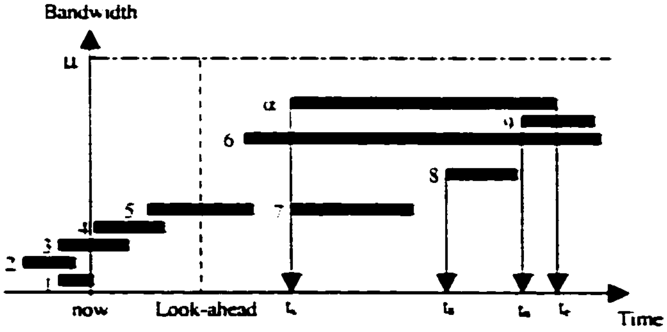


Figure 4.2: Bandwidth-time resource reservations

4.5.1 Immediate Reservations

Immediate reservation is defined as the process of reserving resources for using at the time of the request. This type of reservation is preferred by users who do not have fixed schedules or users who accept service degradation or service interruption during session's playback, and also willing to repeat the reservation requests until they get the service. For each immediate reservation request, the admission control algorithm adds, for each link, the aggregated immediate flows, the active advance flows and the passive advance flows during the look-ahead time. The maximum value is subtracted from the link capacity. The result is the available link capacity. Therefore, the algorithm is as follows:

$$L_i = \mu - Max\,(I + A\,(t)) \qquad now \leq t \leq look\text{-}ahead$$

L_i = Available capacity of link i
μ = Link capacity (bandwidth)
I =Aggregate bandwidth of immediate flows
A (t) = Aggregate bandwidth of advance flows.
Look-ahead = Time used for immediate admission.

The L_i values for all links in the network domain are then used by the QoS routing algorithm to find the path that can handle the reservation request. The request is rejected if the bottleneck bandwidth (the lowest bandwidth in the calculated path) is smaller than the requested bandwidth. From this algorithm the immediate request is guaranteed service in the look-ahead time period, and then the request proceeds if there are still available resources, otherwise it is preempted to give priority to advance sessions. In this case immediate session may proceed with best-effort service.

4.5.2 Advance Reservations

Advance reservation is defined as the process of reserving resources for using after a specific time in the future. This type reflects the nature of human life, where people may prefer to schedule events according to their time availability. So they can instruct their agents to schedule some events to be seen later. This is useful if there are many people involved in a session (video conferencing). For each new request, the algorithm calculates, for each link, the maximum value of the aggregated advance reservation, from the start to the end of the new flow and subtracts this value from the link capacity. Therefore, the algorithm for admitting advance requests is as follows:

$$L_t = \mu - Max\,(A\,(t)) \qquad t_s \leq t \leq t_e$$

The QoS routing algorithm finds the best path according to the value of L_t. The request is rejected if the bottleneck value does not satisfy the reservation request. According to this algorithm the advance reservation requests are guaranteed service during the playback of the session. This in fact comes at the cost of immediate sessions, where at the start of every advance session the admission control algorithm recalculates the reserved bandwidth. If it is greater than the total link capacity, one or more immediate sessions are preempted. To avoid abuse of reserving resources, and differentiate between immediate and advance reservation requests, some policies have to be enforced. For example, the start of the advance reservation (Book-ahead time) should be selected to give chance to immediate requests, also the book-ahead value should not be too far, to limit the reservation state.

4.5.3 User Interaction

User interactions are operations on admitted flows. To process user interactions in light of system resources, we need to translate user interactions into parameters known to system resources and associated control process. Therefore, we need to find all possibilities (with respect to resource reservations) that might happen due to these interactions. Table 4.1, summarizes all these possibilities, and states those that need admission control. The interactions in the table are based on four primitive actions:

- Inc_Dur = Increase duration
- Dec_Dur = Decrease duration
- Inc_Bw = Increase bandwidth
- Dec_Bw = Decrease bandwidth

Inc_Dur	Dec_Dur	Inc_Bw	Dec_Bw	Action	AC
0	0	0	0	Terminate	No AC
0	0	0	1	Dec_Bw	No AC
0	0	1	0	Inc_Bw	AC
0	1	0	0	Dec_Dur	No AC
1	0	0	0	Inc_Dur	AC
1	0	0	1	Inc_Dur and Dec_Bw	AC
1	0	1	0	Inc_Dur and Inc_Bw	AC
0	1	0	1	Dec_Dur and Dec_Bw	No AC
0	1	1	0	Dec_Dur and Inc_Bw	AC

Table 4.1: User interactions

User interaction requests that do not need admission control should not be rejected. These requests will free some resources, which are useful in admitting immediate requests, or in admitting advance requests, if the user interaction is for sessions that have not started yet. The other thing to be considered in user interactions is the time the interaction request is invoked. The interaction request may be invoked at the following times.

- During the passive period of the advance session:
- During the active period of the immediate and advance sessions.

If the interaction request is for passive sessions, the admission algorithm is applied as if it is a new advance reservation request. If the interaction request is for active sessions, the time of the interaction is further divided into two types; 1) interaction requests for immediate action and 2) interaction requests for later time during the playback of the session. The following are the admission control equations for user interactions that need admission control, illustrated with the bandwidth-time resource reservation snapshot.

Inc_Bw

If the increase starts from the time of the request:

$$\mu \geq Max\,[(N^{\alpha} - I + A\,(now)),\ (N^{\alpha} + A\,(t))] \qquad now < t \leq t_e$$

N^{α} = The increase in bandwidth for flow α

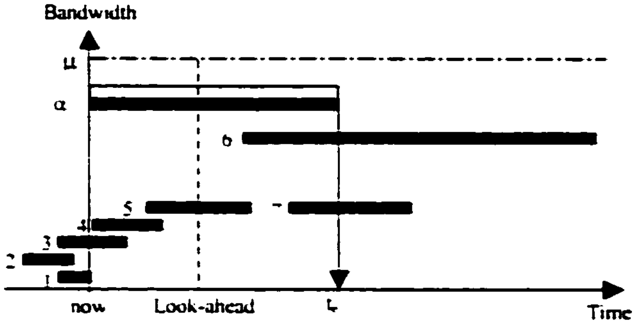


Figure 4.3: Increase bandwidth immediately

If the increase is for later time in the active session

$$\mu \geq Max\,(N^{\alpha} + A\,(t)) \qquad t_a \leq t \leq t_e$$

t_a = Start of the action

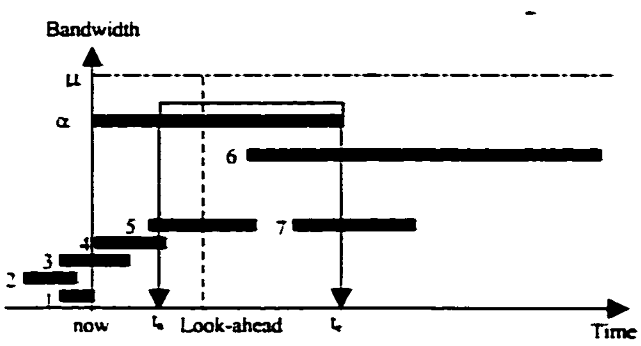


Figure 4.4: Increase bandwidth after some time

In both cases, the increase in the bandwidth do not affect the admitted advance reservations, but it definitely affects (increases the preemption probability) the admitted immediate reservations. Therefore, to alleviate this problem, certain policies should be enforced on user interactions in order not to abuse the resources and severely increase the preemption probability of the immediate reservations.

Inc_Dur

$$\mu \geq Max\,(b^x - A\,(t)) \qquad t_e \leq t \leq t_{e+y}$$

b^x = Session's reserved bandwidth

y = The increased time.

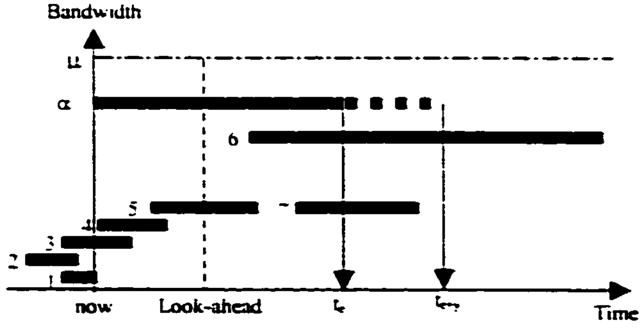


Figure 4.5: Increase duration

This interaction takes advantage of the gaps between the advance reservations, which increases the probability of accepting this type of requests. But it increases the rejection probability of immediate requests.

Inc_Dur and Dec_Bw

We assume that the Inc_Dur and Dec_Bw happen at the same time.

$$\mu \geq Max\ ((b^{x} - N^{\alpha}) + A\ (t)) \qquad t_e \leq t \leq t_{e+y}$$

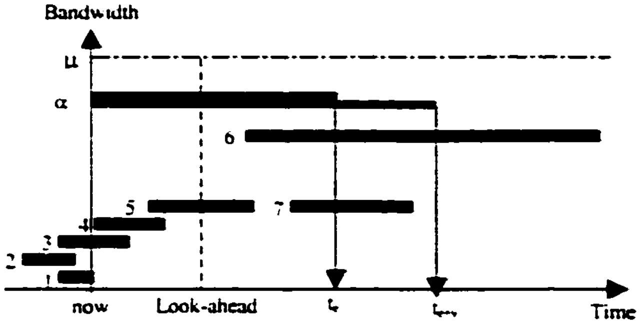


Figure 4.6: Increase duration and decrease bandwidth

This is like a new request with new bandwidth value. In fact this is a short-term advance resource reservation. It still affects the rejection and preemption probabilities of immediate requests.

Inc_Dur and Inc_Bw:

If the increase start from the time of the request

$$\mu \geq Max\ [(N^{\alpha} + I + A\ (now)), (N^{\alpha} + A\ (t))] \qquad now \leq t < t_e$$

And

$$\mu \geq Max\ ((b^{x} + N^{\alpha}) + A\ (t)) \qquad t_e \leq t \leq t_{e+y}$$

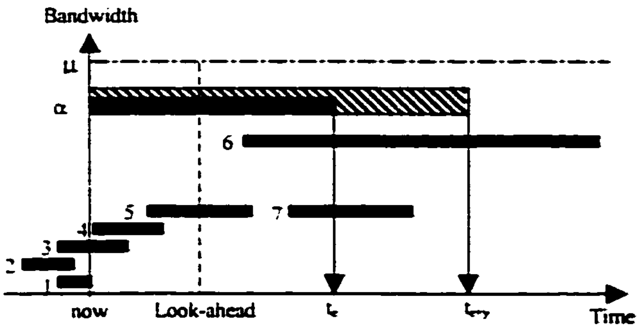


Figure 4.7: Increase duration and increase bandwidth immediately

If the increase is for later time in the active session:

$$\mu \geq Max\,(N^x - A\,(t)) \qquad t_a \leq t < t_e$$

And

$$\mu \geq Max\,((b^x - N^x) - A\,(t)) \qquad t_e \leq t \leq t_{e+y}$$

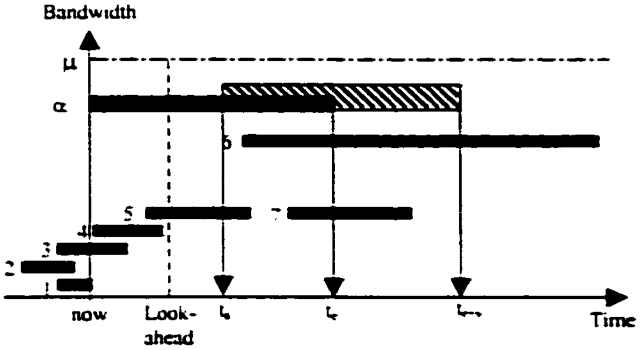


Figure 4.8: Increase duration and increase bandwidth after some time

This is the most aggressive user request. Therefore, it should undergo certain policies to limit its effects on immediate sessions.

Dec_Dur and Inc_Bw

If the increase start from the time of the request:

$$\mu \geq Max\ [(N^{\alpha} + I + A\ (now)),\ (N^{\alpha} + A\ (t))] \qquad now < t \leq t_{e-z}$$

z = The decreased time

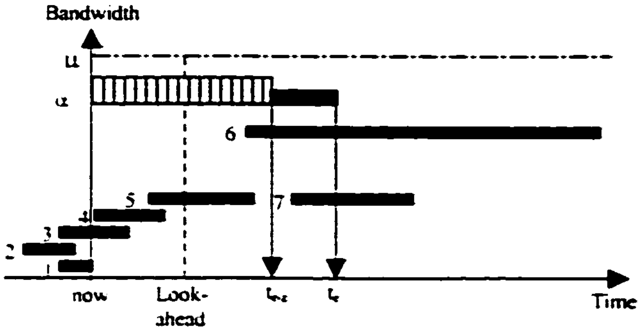


Figure 4.9: Decrease duration and increase bandwidth immediately

If the increase is for later time in the active session:

$$\mu \geq Max\ (N^{\alpha} + A\ (t)) \qquad t_a \leq t \leq t_{e-z}$$

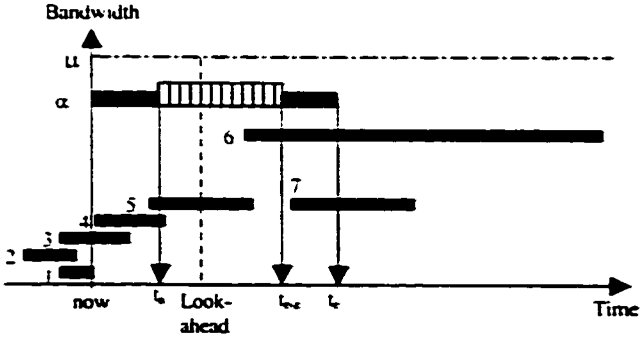


Figure 4.10: Decrease duration and increase bandwidth after some time

This is the same as the increase bandwidth request, but it has better chance to be accepted by applying admission control in shorter period.

4.6 Admission Control Simulation

Our simulation aim to show the effects of user interactions on admitted immediate and advance flows, and on the total utilization of the link. We have done simulation on scenarios with immediate and advance reservations only, and then introduced user interactions to these scenarios.

The simulated topology is a bottleneck link, which consists of three sources; one for sending advance reservations, one for sending immediate reservations and the other for sending user interactions. The advance and immediate sources are connected to the admission control module through a queue. The user interactions source receives some admitted advance flows from the admission control module, and then applies user interactions to the received flows, then sends them back to the admission control.

4.6.1 Simulation Parameter Choices

There are general parameters for the operation of the overall model, and specific parameters for each source.

- The duration of the requests is exponentially distributed with a mean of 500 simulation seconds;
- The resources requested are uniformly distributed with a mean of 3 resource units;
- The inter-arrival rate is exponentially distributed with a mean of 11.76;
- Book-ahead time is exponentially distributed with a mean of 20000;
- Minimum booking time is equal to 1000;
- Maximum booking time is equal to 200000.

We suggest some policies on user interactions, in order not to abuse the system resources. These policies are:

- The duration of the flow on which user can apply interactions. In our simulations we choose this duration to be above the average of admitted advance requests.
- The time after which the user can start sending the interactions. We choose this time to be above the middle of the average duration.

The total static link capacity is 150, and the simulation time is 2 x 10^6 simulation seconds. The last parameter is the Look-ahead time, which is varying during the simulation; the values used are 0, 250, 500 and 1000. From the above parameters, we can calculate the average offered load produced by the immediate and advance sources. According to the Little's formula, which states that "for systems that reach steady state, the average number of customers in a system is equal to the product of the average arrival rate and the average time spent in the system".

$$\text{Average Offered load} = \frac{\text{Average arrival rate} \times \text{Average duration} \times \text{Average resources}}{\text{Total link capacity}} \qquad [4.1]$$

Therefore, by choosing the average arrival rate of 11.76, the average offered load for immediate and advance sources is 170%.

4.6.2 Simulation Results

Several experiments were conducted to monitor the following performance measures:

- Rejection probability: Used for monitoring the rejection probability of immediate, advance and user interaction requests;

- Utilization: To check the overall utilization of the link capacity;
- Advance Duration: To check the duration of the admitted advance flows before and after user interactions;
- Preemption probability: To monitor the preemption probability of the admitted immediate flows.

Simulation without User Interactions

Table 4.2, shows the results without user interactions. The maximum link capacity is available to the advance requests. With 85% offered load we get very low rejection probability. As time elapses we get a steady value, which allow us to control the rejection probability by varying the offered load.

For immediate requests, the rejection probability increases as we increase the look-ahead time. At zero look-ahead time the rejection probability is at its minimum value. The reason for this is that the admission control algorithm accounts only for currently active flows. By increasing the look-ahead time more advance flows are included in the admission control algorithm. The offered load for immediate reservation is also 85%. Therefore, the total offered load is 170%, which explains the high rejection probability of immediate requests.

The advance reservation requests have higher priority to acquire resources than immediate requests. So, at the start of every admitted advance flow, the total used resources is calculated, if it exceeds the total link capacity, one or more admitted immediate flows are preempted. The preemption probability can be changed by changing the look-ahead time. At zero look-ahead, more immediate flows are admitted, which results in high preemption probability. In fact, the look-ahead time is a key factor for choosing the rejection and preemption probability for immediate requests. The value of look-ahead time should be chosen to get optimum values of preemption and rejection probability.

	Look-ahead Time			
	0	250	500	1000
Advance Rej. Prob.	.07	.07	.07	.07
Immediate Rej. Prob.	.42	.75	.78	.83
Preemption Prob.	.54	.13	.06	.02

Table 4.2: Simulation results without user interactions

Figure 4.11, shows the duration of admitted advance flows. The duration is centered at the average duration of the requests. Requests with larger duration have higher rejection probability; this encourages users to apply for moderate duration sessions.

Finally, the total utilization of the active flows is quite high as shown in figure 4.12. For zero look-ahead time, the utilization may drop up to 0.7, but it increases as new requests are coming. As we increase the look-ahead time more immediate requests are rejected, which results in decreasing the utilization, sometimes it drops to 0.6. The reason for the sudden drop in utilization is that admitted advance reservations introduces fragmentation in time, new flows may not fit into these fragments.

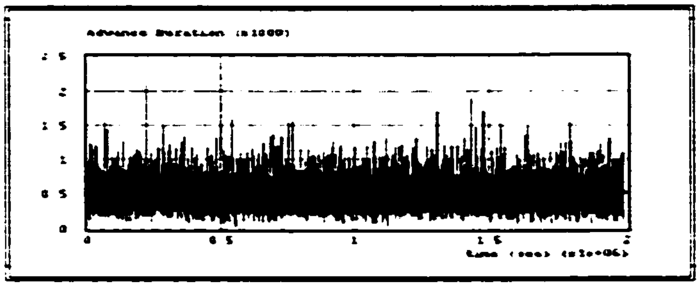


Figure 4.11: Admitted advance duration

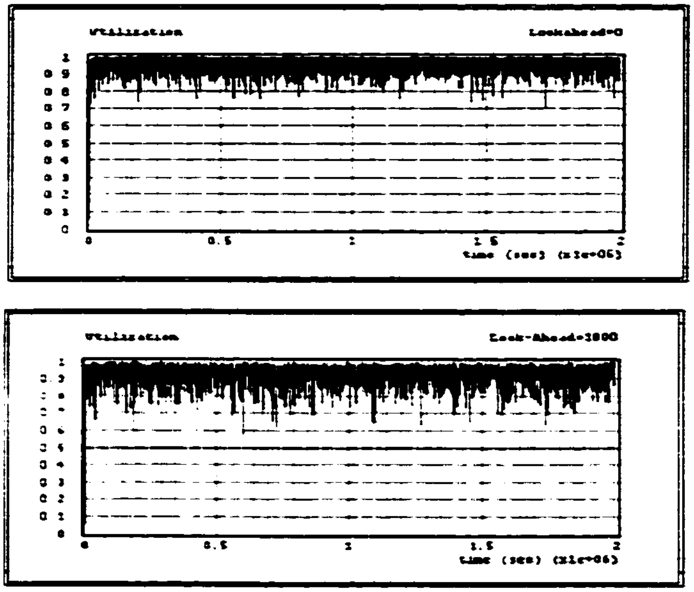


Figure 4.12: Utilization

Simulation With user interactions

Table 4.3. shows the simulation results. The advance rejection probability is not affected, because the reservation start time is much more than the average duration. For immediate requests, we can notice a slight decrease in rejection probability; this is because some user interactions free some resources, while others increase the reserved resources.

The preemption probability decreases slightly at zero to 300 look-ahead times, then it increases as we increase the look-ahead time. The reason for this is that, in our experiments we only accept user interactions for duration above 600, and users start sending interactions above the middle of this duration 300. Therefore, the more interactions the users send the more preemption we get. The duration of admitted advance flows increase due to user requests for increasing the duration of active flows, figure 4.13. The utilization as shown in figure 4.14, increases from zero to 300 look-ahead times, and then it decreases as we increase the look-ahead time. This change in utilization forces us to put some regulations on user interactions in order not to abuse the resources and waste them.

We performed several experiments for user interactions, and we came up with the results shown in table 4.3. The lowest rejection probability we got is for Inc_Dur and Dec_Bw requests, while the highest rejection probability is for Inc_Bw requests. User interaction requests come when the resources have already been reserved by immediate and advance requests. So, user requests for increasing the bandwidth of admitted flows are more prone of being rejected. Requests for increasing the duration have less rejection probability than increasing the bandwidth, because fragmentation in time produced by admitted advance flows gives more chance for increasing duration requests. Generally the rejection probability for user interactions is low.

	Look-ahead Time			
	0	250	500	1000
Advance Rej. Prob.	.07	.07	.07	.07
Immediate Rej. Prob.	.41	.72	.76	.80
Preemption Prob.	.52	.12	.065	.045
Inc_Bw Rej. Prob	.170	.170	.170	.170
Inc_Dur Rej Prob.	.115	.115	.115	.155
Inc_Dur and Dec_Bw Rej. Prob.	.06	.06	.06	.06
Inc_Dur and Inc_Bw Rej. Prob.	.165	.165	.165	.165
Dec_Dur and Inc_Bw Rej. Prob.	.085	.085	.085	.085

Table 4.3: Simulation results with user interactions

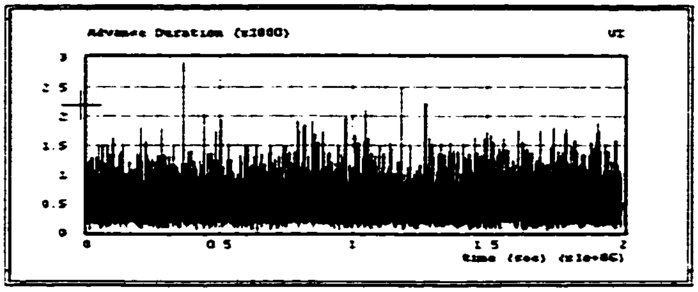


Figure 4.13: Advance duration with user interactions

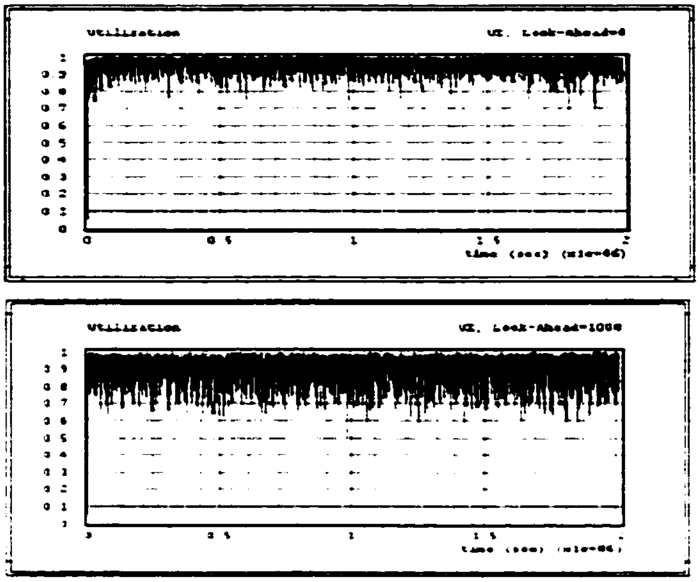


Figure 4.14: Utilization with user interactions

4.7 Summary

Resource reservation and QoS guarantee are usually associated with admission control. Admission control algorithms are available in different schemes under two classes, parameter-based and measurement-based. In this chapter we stated that parameter-based schemes is best suited for immediate and advance reservation architecture with user interactions. The simulation tests should promising results for user interactions, which promote the deployment of interactive multimedia applications.

Chapter 5

Quality of Service Path Selection

5.1 Introduction

Associating QoS routing with resource reservations is very important in achieving a better scalability for multimedia applications in the Internet. In chapter 3 we have introduced architecture for domain agency, which manages domain resources and maintains them. The domain agency contains a QoS routing agent that is used to construct a QoS path that accommodates the requested resources. In this chapter, we present an algorithm for QoS routing agent. We introduce a path calculation algorithm that best suits the immediate and advance resource reservation architecture. This algorithm is simulated and compared with other existing QoS routing algorithms. We also present a simulation model, which has been implemented to monitor the operation of the QoS routing agent.

5.2 Path Selection Schemes

Selecting a QoS path, in network domains, depends solely on the flow's QoS requirements and the state of the resources in the network domain. The flow is assigned a certain path if and only if the available resources can handle the flow's QoS. This section discusses several path selection algorithms in light of the following metrics.

- ***Link bandwidth:*** The algorithm finds a feasible path that accommodates the requested bandwidth.
- ***Hop-count:*** The feasible path should have the least-cost (least hop-count). This will minimize the amount of consumed resources.

In addition to these metrics, the path selection algorithm may deploy some policies that regulate the use of resources, and this applies to advance and immediate requests.

Meanwhile, the real-time applications may require more QoS guarantees other than the bandwidth and the hop-count (cost), those guarantees may include delay, delay jitter and loss probability. But Wang et al [WAN96] proofed that any two or more of delay, delay jitter, cost and loss probability in any combination as metrics are NP-Complete [GAR79]. The only feasible combination are bandwidth and one of those metrics. The delay metrics consists of two types of delays; queuing delay and propagation delay. The queuing delay is associated with the link bandwidth, while the propagation delay is determined by the hop-count. Therefore, the bandwidth and the hop-count constitute the most important metrics in determining the QoS path. According to these metrics we present three path selection criterions.

Shortest-widest path

The shortest-widest path, is the path between the source and the destination that has the maximum bottleneck bandwidth and minimum number of hops [WAN96]. Therefore, if the algorithm searches multiple paths to the destination, the selected path is the one that has the maximum bottleneck bandwidth that satisfies the requested bandwidth. To define this in mathematical notations, assume that a path $p = (i, j, k, \ldots, l, m)$, where i, j, k, ..., l, and m are nodes of the path. The bottleneck bandwidth $d(p)$ is the $min[b(i, j), b(j, k), \ldots, b(l, m)]$, and this bandwidth should be the $max[d(p_1), d(p_2), d(p_3), \ldots, d(p_n)]$.

Shortest-Eliminated path

This criterion [WAN96] operates by first eliminating all links that has a bandwidth less than the requested bandwidth and then find the shortest path to the destination.

Shortest-Narrowest path

We have proposed this criterion to serve certain needs of the resource reservation architecture. The detail of this criterion is explained later in this chapter. The scheme is opposite to shortest-widest path, where instead of finding the widest bottleneck bandwidth of all possible paths from the source to the destination we find the narrowest bottleneck bandwidth that satisfies the requested bandwidth. Consequently, if the algorithm searches multiple paths to the destination, the selected path is the one that has the minimum bottleneck bandwidth that satisfies the requested bandwidth. So, The bottleneck bandwidth $d(p)$ is the $min[b(i, j), b(j, k), \ldots, b(l, m)]$, and this bandwidth should be the $min[d(p_1), d(p_2), d(p_3), \ldots, d(p_n)]$.

QoS routing algorithms are classified in terms of the network state they are using and the time of the calculation. The state may be a global state (all nodes in the network domain) that lead to a source routing, or partial state (the source node, the neighbors and the destination) and lead to a distributed routing. The path may be calculated prior to the routing request (pre-computation) or at the time of the request (on-demand). Therefore, any QoS routing algorithm fall into one of the following schemes.

- On-demand source routing
- On-demand distributed routing
- Pre-computed source routing
- Pre-computed distributed routing

5.2.1 On-demand Source Routing

In this scheme each node maintains a global state of the network domain (network topology and state information of every link). Based on this state, the source node finds a feasible path to the destination. The calculation algorithm is triggered by the arrival of requests. Therefore, for each request the algorithm should calculate a new route to the destination. This scheme has advantages and disadvantages in terms of the computation overload and the collected global state.

Advantages

- Guarantees loop-free paths.
- Avoid problems associated with distribution calculations: consistency, deadlock, etc.
- Simple to implement, evaluate, debug and upgrade.
- Provide better routes by using the most recent link state.
- Low memory usage as compared to pre-computation.

Disadvantages

- Updating the global state frequently enough to reflect the dynamics of network parameters.
- The imprecision of the global state due to the propagation delay of state messages. This may cause the QoS routing algorithm to fail in finding a path [CHE98b].

- Computation overhead at the source.
- Scalability problem.

5.2.2 On-demand Distributed Routing

In this scheme, the path calculation is distributed among the intermediate nodes between the source and the destination(s). The calculation is triggered for every arrived request.

Advantages

- Better scalability.
- Better success rate, because multiple paths can be searched.
- Provide better paths, by using the most recent state.
- Low memory usage.

Disadvantages

- Is not loop-free, because of state inconsistency.
- Difficult to implement, evaluate and debug.
- There is not enough information to avoid loops.

5.2.3 Pre-Computation Routing

Pre-computation of paths with source routing or distributed routing share some advantage and disadvantages of on-demand source routing and on-demand distributed routing. The differences are:

- Pre-computations provide less accurate paths than on-demand.
- Lower computation overhead.

- Pre-computation routing introduces extra processing overhead (In addition to the path calculation) by searching for the feasible path at the arrival of the request.

5.3 The Rationale Behind the QoS Routing Agent

The QoS routing protocols have come into existence to support the provision of QoS for the emerging multimedia applications. The work until now is still under extensive research and is not clear if the deployment of those protocols will perform and scale well in the existing architecture of the Internet. As mentioned previously, QoS routing use enhanced route calculation algorithms to find a path with multiple constraints. Those algorithms, as suggested by the current proposals, should be implemented in each router. Furthermore, QoS routing is associated with resource reservations or some sort of admission control. This association, will certainly put extra load on the routers that are supposed to process and forward traffic with minimum processing delay. In this thesis, we argue the use of QoS routing agent to calculate a QoS path on behalf of the routing protocol. The following points explain our argument. Figure 5.1, is repeated here for illustration.

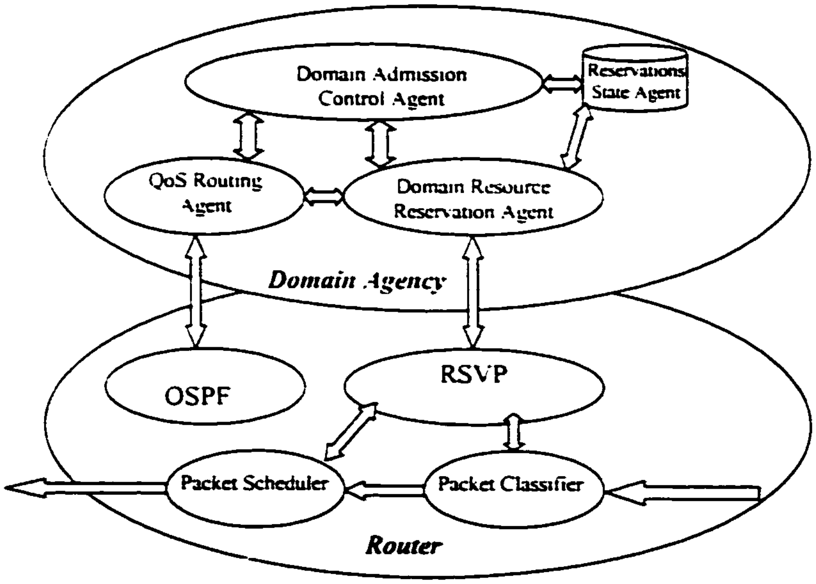


Figure 5.1: Domain agency architecture.

Argument 1

QoS provisioning in the current and near future architecture of the Internet is available to limited number of users, and cannot scale to accommodate all users of the Internet. Therefore, by using QoS routing agents, the underlying routing protocol performs its packet forwarding for the best-effort service, and only QoS requests trigger the QoS routing agent to calculate the QoS path.

Argument 2

The proposed QoS routing protocols are extensions to the traditional routing protocols, and are supported with certain path selection algorithms to find the QoS path. These algorithms may not satisfy the requirements of different multimedia applications. Therefore, by isolating the QoS calculation from the routing protocol, the algorithms can be implemented as flexible as possible to fulfill the different applications needs.

Argument 3

Implementing the QoS routing agent inside the domain agency, which controls the domain resources, allow calculations of accurate paths that reflect the current state of the domain.

Argument 4

Network domains (OSPF domains) may split and merge according to the size of the domain (number of nodes). By using QoS routing agent, it is possible to clone new agents that interact with the OSPF routing protocol to determine the topology of the new domain, and then interact with the parent agent in the original domain to retrieve the current reservations that belong to the new domain.

Argument 5

The resource reservation architecture that we have proposed handles immediate and advance reservations that rely on the QoS routing agent to find future routes with the help of the domain agency.

Argument 6

Flow aggregation is one of the important issues in providing better scalability for packet classifiers and packet schedulers. The QoS routing agent employs a calculation algorithm that allows the aggregation of flows that use the same path within certain time period.

Argument 7

Policies are important in regulating the resource usage. the QoS routing agent is flexible in enforcing local policies in a certain domain. and global policies spanning multiple domains or policies enforced on the whole autonomous system.

With these arguments. we proceed in the following sections to explain the QoS routing agent and the calculation algorithm that is used by this agent.

5.4 Quality of Service Routing Agent

The Quality of Service Routing Agent (QoSRA) plays a very important role in the domain agency architecture. It decides if the new session is admitted or rejected. according to the links state obtained from the admission control algorithm. For simplicity. the QoSRA is not involved in updating routes for user interactions. These are handled by the admission control agent. which grants admission according to the existing selected path of the session. The request is rejected if resources in this path cannot accommodate the requested interaction. The best path is the shortest-narrowest path.

Definition

The best path is the shortest, in terms of the hop-count (propagation delay), and the narrowest, in terms of the bottleneck bandwidth, that satisfies the requested bandwidth.

As stated earlier, the QoS routing is closely linked to immediate and advance resource reservations. The QoS routing algorithm therefore needs to apply a mechanism that maximizes the utilization of the resources and facilitates the prediction of the next routes to be selected. This helps in flow aggregation and saves processing time by caching the most recent routes. The main reasons for using the shortest-narrowest algorithm rather than the shortest-widest algorithm are as follows.

- **Route fluctuation:** The shortest-narrowest path algorithm allows the next selected route, with the same ingress and egress points in a certain domain, to be the same as the previous route as long as the route has sufficient resources to support the new session. Therefore, by caching the most recent routes, calculating a QoS path involves only searching the cached routes for the one with the same ingress and egress points and checking its bottleneck bandwidth.
- **Resource fragmentation:** This is a condition in which there are many scattered amounts of resources that are too small to be used productively. This condition is especially common in advance reservations. An advance reservation algorithm reserves resources at different, overlapping time slots. This increases the resource fragmentation and, as a result, increases the probability that immediate and advance reservation requests will be rejected.

As an illustration, consider the following example. In figure 5.2, assume each link has a capacity of 10 units, and three new flows arrive at the ingress point asking for five units each. Using the shortest-widest path, The QoSRA will operate as follows; choose link 1-2 for the first request, link 1-3 for the second and link 1-4 for the last. If another request arrives at the same ingress point asking for six units of resources, it will be definitely rejected. Resources have been fragmented and there are now not enough resources in the outgoing links, even though the total capacity of the links (30 units) is enough for all requests (21 units). Each time a new request arrives, it will use a different path even if the ingress and egress of the previous path are the same and the path has enough resources. This requires unnecessary calculations and will affect the performance of the algorithm.

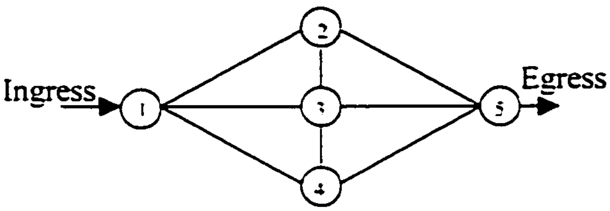


Figure 5.2: Resource fragmentation problem

5.4.1 The Shortest-Narrowest Path Calculation Algorithm

The algorithm is a modification to Dijkstra's algorithm [DIJ59, STA97], where we target the minimum bandwidth with the minimum number of hops that accommodate the desired bandwidth instead of the least-cost path. The algorithm computes the complete path from source to destination if they are in the same domain, from the source to the egress router, from the ingress router to the egress router and from the ingress to the receiver.

The routing table includes only the transit nodes, stub nodes are checked against the destination address if it is not the destination it is discarded, otherwise the

The routing table includes only the transit nodes, stub nodes are checked against the destination address if it is not the destination it is discarded, otherwise the destination has been reached and the algorithm is terminated. For simplicity, the QoSRA applies only to unicast sessions.

Consider a directed graph $G=(V, A)$, where V is the set of vertices (nodes) and A is the set of edges. Each $Edge(i, j)$ is assigned a real number b_{ij}, the available bandwidth. The algorithm proceeds in stages. By the k^{th} stage, the best paths to the k nodes closest to the source node have been determined; these nodes are in a set M. At stage $(k+1)$, the node not in M that has the best path and has the minimum number of hops (in case of multiple nodes with the same minimum bandwidth) from the source node is added to M. As each node is added to M, its path from the source is defined. The algorithm can be formally described as follows. Using the following notations:

b^{α} = Requested bandwidth of flow α

V= set of nodes in the network

s = Source node

M = Set of nodes so far incorporated in the algorithm

b_{ij} = Link bandwidth from node i to node j. $b_{ij} = 0$ if the two nodes are not directly connected. $b_{ij} \geq 0$ if the two nodes are directly connected.

B_n = Minimum bandwidth of the best path from node s to node n

Therefore, given any path $p = (i, j, k, ..., l, m)$, the bottleneck of the path is defined as

$$width(p) = min[b_{ij}, b_{jk}, ... b_{lm}]$$

and the cost of the path is defined as:

$$cost(p) = c_{ij} + c_{jk} + ... + c_{lm}$$

The shortest-narrowest path algorithm calculates a path by applying the modified Dijkstra's algorithm, which finds a path with minimum *width(p)* and least cost *cost(p)*.

Step 1: Initialize $M = \{s\}$, the set of nodes so far incorporated consists of only the source node.

$B_n = b_{sn}$ for all $n \neq s$; the initial path bandwidth to neighboring nodes

Step 2: Find the neighboring node not in M that has the minimum bandwidth from node s and equal to or greater than the requested bandwidth, incorporate that node into M; this can be expressed as:

Find $w \notin M$ such that:

$B_w = min\ [B_n] \geq b^{\alpha} \qquad n \notin M$

If w has more than one value, find w that has the minimum number of hops from the source. Add w to M

$M = M \cup \{w\}$, if all nodes are incorporated into M, terminate.

Step 3: Update the best bandwidth paths:

- If $B_n < b^{\alpha}$: then

  $B_n = min[B_w, b_{wn}] \qquad$ for all $n \notin M$

If the new value of B_n is smaller than or equal to the previous value of B_n, restore the previous value of B_n.

- If $B_n \geq b^{\alpha}$

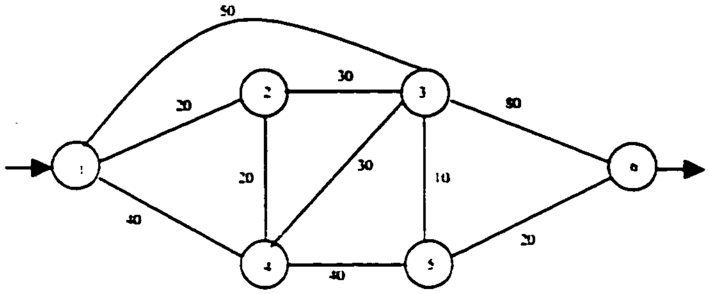
  $B_n = min[B_n, min[B_w, b_{wn}]] \qquad$ for all $n \notin M$

If $min[B_w, b_{wn}] < b^{\alpha}$, then the new value of $B_n < b^{\alpha}$, restore the previous value of B_n. If $b^{\alpha} \leq min[B_w, b_{wn}] < B_n$, the path from s to n is now the path from s to w concatenated with the link from w to n.

One iteration of steps 2 and 3 adds one new node to ***M*** and defines the best-bandwidth path from *s* to that node. That path passes only through nodes that are in ***M***.

5.4.2 Example Network

The following is a network example to illustrate the operation of the shortest-narrowest path algorithm. The result of the algorithm is shown in table 5.1.



Assume $b^\alpha = 30$, source node = 1 and destination node = 6.

Figure 5.3: Network example

K	M	B2	PATH	B3	PATH	B4	PATH	B5	PATH	B6	PATH
1	{1}	20	1-2	50	1-3	40	1-4	0	-	0	-
2	{1,4}	20	1-2	30	1-4-3	40	1-4	40	1-4-5	0	-
3	{1,4,3}	30	1-4-3-2	30	1-4-3	40	1-4	40	1-4-5	30	1-4-3-6
4	{1,4,3,2}	30	1-4-3-2	30	1-4-3	40	1-4	40	1-4-5	30	1-4-3-6
5	{1,4,3,2,6}	30	1-4-3-2	30	1-4-3	40	1-4	40	1-4-5	30	1-4-3-6
6	{1,4,3,2,6,5}	30	1-4-3-2	30	1-4-3	40	1-4	40	1-4-5	30	1-4-3-6

Table 5.1: Shortest-narrowest path calculation

5.5 QoS Routing Algorithms Performance Evaluation

This section evaluates the performance of the shortest-narrowest path algorithm and compares it with the shortest-widest and the shortest-reduced path algorithms (a shortest-reduced path is selected after eliminating the links that do not have enough bandwidth to support a new session). Dijkstra's shortest path algorithm is used as a benchmark, and is also used to test connectivity

5.5.1 Network Topology

The network topologies used are randomly generated graphs. Each node is identified by selecting its coordinates (x and y) from the interval [0, 100). The residual bandwidth of each link is randomly selected from the interval [0, 100]. To control the connectivity degree of the generated topology we apply the following algorithm.

- Each node is assigned a certain radius;
- The node is connected to other node if and only if the Euclidian distance between the two nodes is less than the minimum of the radii of the two nodes.

To determine the radius of the nodes, we follow the following procedure:

- Each node is assigned a random number x_i in the interval (0, 1);
- The radius for node i is $R \times x_i$, where R is selected so that the average node connectivity (number of neighboring nodes) degree is d.

The simulation program was completely written in Java. We have conducted several experiments with this type of random graphs and concluded that this type of graphs is very suitable to simulate network domains with certain average connectivity degrees.

5.5.2 Simulation Results

We conducted experiments with network sizes 20, 40, 60, 80 and 100. For each network size, 100 random graphs are generated. Then for each graph, we produced a connected graph with the degrees 5, 6, 7, 8. The shortest path algorithm is used to test connectivity. The routing scenario is as follows; each node selects a random bandwidth (requested bandwidth) in the interval (10, 30), then each node calculates a QoS path to all other nodes, by using the on-demand source routing algorithm. There are no reservations in this simulation, the task is to test the following performance measures:

Success rate (delivery rate): is the ratio of the successfully received messages to the total of the sent messages.

Hop-count (cost): is the ratio of the successful hop counts to the total delivery.

Success Rate

Figure 5.4 shows the success rates of the tested QoS routing algorithms and the shortest path algorithm. The shortest path has the lowest success rate. This is because the shortest path may not satisfy the requested bandwidth. The success rate for the shortest path decreases as the network size increases. This is because, for large networks, the average cost of the routes increases, also increasing the probability that the bottleneck link is smaller than the requested bandwidth. The shortest path is therefore not suitable for handling QoS routing requests. The success rate for the other algorithms is identical and is much better than the shortest path. The success rate is also sensitive to the connectivity degree, increasing and becoming more stable for large degrees.

Hop-Count

Figure 5.5 presents the results of the average hop-counts. As expected, the shortest path gives the lowest values. The second best algorithm is the shortest-reduced algorithm. The shortest-widest path provides slightly better results than the shortest-narrowest path the average difference is one hop-count.

Discussion

The results of the experiments raise two important issues. which have to be taken into consideration when deciding the size of the network domain that employs the QoS routing and QoS provision in general. The first issue is the domain size: the results show that larger domains produce larger hop-counts. This is not acceptable in some cases (such as networks with scarce resources) because more resources are consumed. Average-sized domains are easy to manage and reduce the QoS path selection overhead. The second issue is the degree size: it has two effects, one is on the success rate and the other is on the hop-count. The success rate increases as we increase the connectivity degree, justifying the usefulness of the QoS routing. The success rate also tends to be stable for higher degree rates, giving a good idea of the behavior of the resources in the domain. For large network sizes, the hop-count decreases as the degree of connectivity increases.

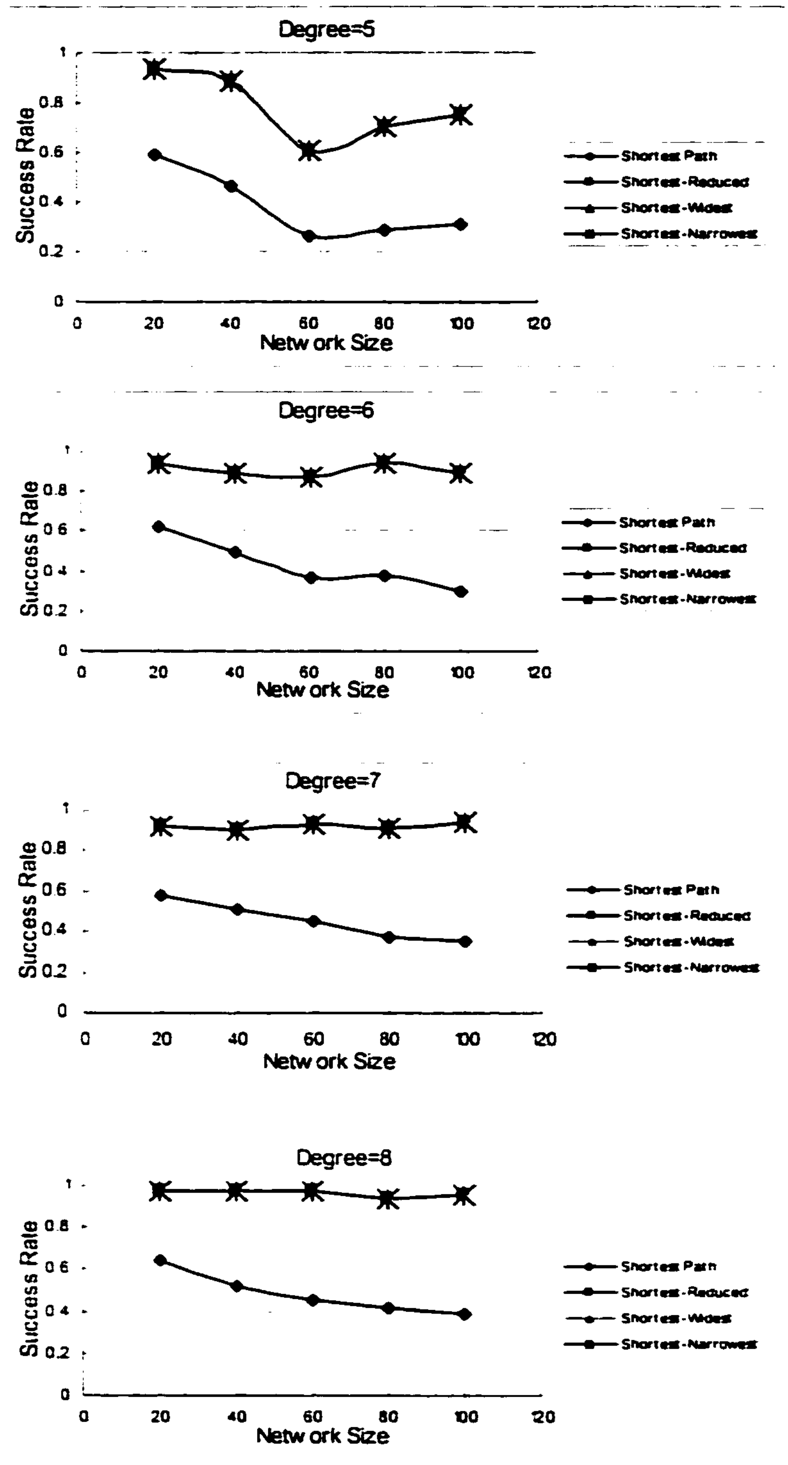


Figure 5.4: Success rate

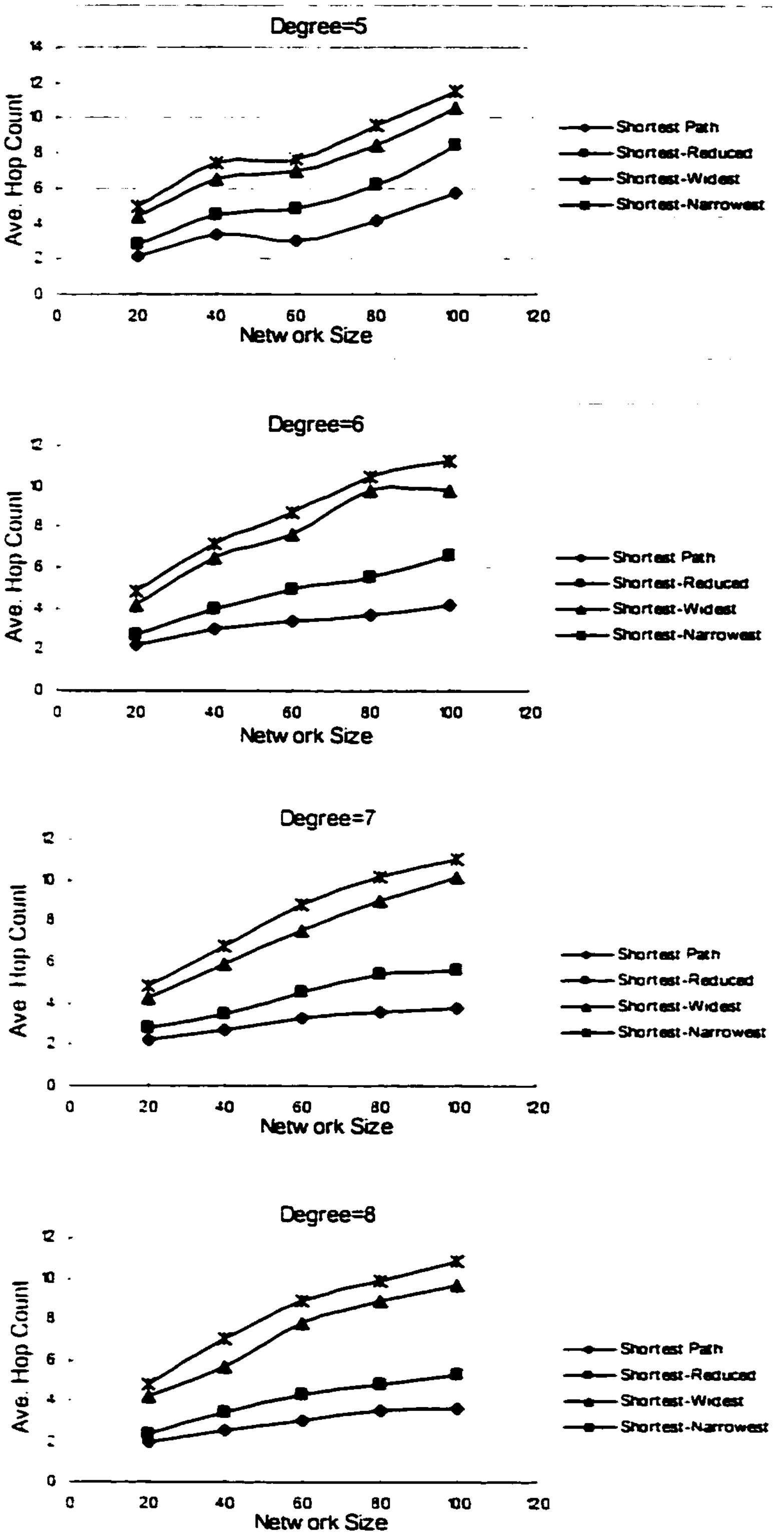
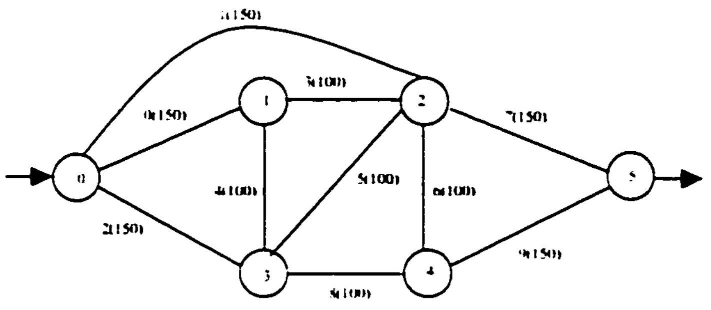


Figure 5.5: Average hop count

5.6 QoS Routing Agent Simulations

Our simulation aim to show that QoS routing is an effective way of achieving high acceptance rate of immediate and advance resource reservation requests. The simulated topology is a bottleneck link, where immediate and advance requests are directed to a QoS routing agent. The QoS routing agent applies admission control and selects the narrowest-path that satisfies the request. The network topology is pre-installed in the QoS routing agent, and is assumed constant during the simulation periods. In real implementation the QoS routing agent gets the network topology and network updates from the routing process. The network used is as follows:



Where: X(Y) are link Id and Link capacity, respectively

Figure 5.6: Simulated network topology

5.6.1 Simulation parameter Selection

There are general parameters for the operation of the overall model, and there are specific parameters for each source.

- The duration of the requests is exponentially distributed with a mean of 500 simulation seconds;
- The resources requested are uniformly distributed with a mean of 3 resource units;

- The inter-arrival rate is exponentially distributed with a mean of 5.0;
- Book-ahead time is exponentially distributed with a mean of 20000;
- Minimum Booking time is equal to 1000;
- Maximum booking time is equal to 200000.

The maximum static link capacity that can be provided by the used network is 300 resource units. The simulation time is 3×10^5 simulation seconds. The last parameter is the Look-ahead time, which is varying during the simulation; the values used are 0, 250, 500 and 1000. From the above parameters, the average offered load for immediate and advance sources, according to equation 4.1, is 200%.

5.6.2 Simulation Results

Several experiments were conducted to monitor the following performance measures:

- Rejection probability: To monitor the probability of the immediate and the advance requests;
- Utilization: To check the overall utilization of each link capacity;
- Advance Duration: To check the duration of the admitted advance flows;
- Preemption probability: To monitor the probability of the admitted immediate flows.

Rejection and Preemption Probabilities

Table 5.2, shows the rejection and preemption probabilities. The maximum link capacity is available to the advance requests. With 100% offered load, we get a rejection probability of almost zero. As time elapses, we get a steady value, which allow us to control the rejection probability by varying the offered load.

For immediate requests, the rejection probability increases as we increase the look-ahead time. At zero look-ahead time, the rejection probability is at its minimum value. The reason for this is that the admission algorithm accounts only for currently active flows. By increasing the look-ahead time, more advance flows are included in the admission control algorithm.

The advance reservation requests have higher priority to acquire resources than immediate requests. Therefore, at the start of every admitted advance flow, the total used resources are calculated. If this total exceeds the total link capacity, one or more admitted immediate flows are preempted. The preemption probability can be changed by changing the look-ahead time. At zero look-ahead, more immediate flows are admitted, which results in high preemption probability. In fact, the look-ahead time is an essential factor for choosing the rejection and preemption probability for immediate requests. The value of look-ahead time should be chosen to get optimum values of preemption and rejection probability.

	Look-ahead Time			
	0	250	500	1000
Advance Rej. Prob.	.0	.0	.0	.0
Immediate Rej. Prob.	.0	.07	.1	.125
Preemption Prob.	.56	.08	.04	.02

Table 5.2: Rejection and preemption probabilities.

Link Utilization

Table 5.3, shows the average utilization of each link in the network. All packets enter at node 0 and leave at node 5. Therefore, according to the network topology and the QoS routing algorithm, links 0,3,7 have higher utilization. When there are not enough resources in link 3, the algorithm will change the path to 0-4-5-7, which is the next best bath. Once links 0 and 7 have finished their resources, the algorithm will choose the path 1-5-8-9, which explains the high utilization of link 5. As we can see from table 5 and the network in figure 18, the links are utilized according to the operation of the QoS routing algorithm.

Meanwhile, the results show that as the look-ahead time increases the utilization of each link increases. At zero look-ahead time all immediate requests are admitted, but whenever a new advance flow starts using its reserved resources, one or more immediate flows are preempted, if the used resources are more than the total link capacity. This results in a high preemption probability and lower link utilization. As we increase the look-ahead time, admitted immediate flows guarantee their operation in the look-ahead period without any preemption. Using the QoSRA, the rejection probability increases by small values as the look-ahead time increases.

Link Id	Look-ahead Time			
	0	250	500	1000
Link 0 Ave. Util.	.959	.984	.988	.990
Link 1 Ave. Util.	.369	.767	.782	.787
Link 2 Ave. Util.	.0	.0098	.015	.0279
Link 3 Ave. Util.	.968	.993	.995	.996
Link 4 Ave. Util.	.495	.542	.543	.543
Link 5 Ave. Util.	.830	.934	.940	.947
Link 6 Ave. Util.	.129	.646	.683	.705
Link 7 Ave. Util.	.956	.982	.987	.989
Link 8 Ave. Util.	.424	.540	.538	.545
Link 9 Ave. Util.	.367	.778	.8	.817

Table 5.3: Average link utilization.

Comparison with Single Path

For comparison, we show in table 5.4 the results of earlier simulations we have done for single link admission control. The total offered load for this simulation is 170%. By comparing the results in table 5.2 and table 5.4, we can see the improvements in the immediate and advance rejection probabilities. The preemption probability is also slightly improved. The admitted advance duration, figure 5.7 (for single path) and figure 5.8 (for QoS routing), is significantly improved using QoS routing, this is a very promising result where sessions with long time duration will have higher acceptance probability than sessions in single link admission control.

	Look-ahead Time			
	0	250	500	1000
Advance Rej. Prob.	.07	.07	.07	.07
Immediate Rej. Prob.	.42	.75	.78	.83
Preemption Prob.	.54	.13	.06	.02

Table 5.4: Simulation results for single path.

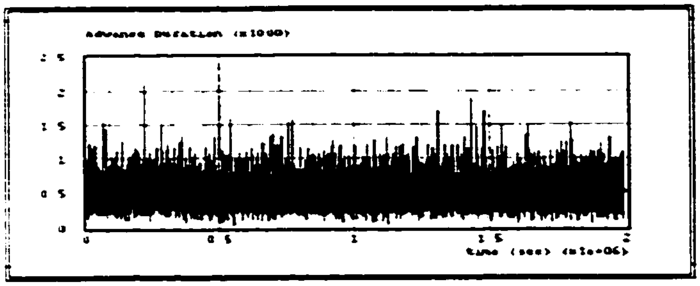


Figure 5.7: Admitted advance duration for single link

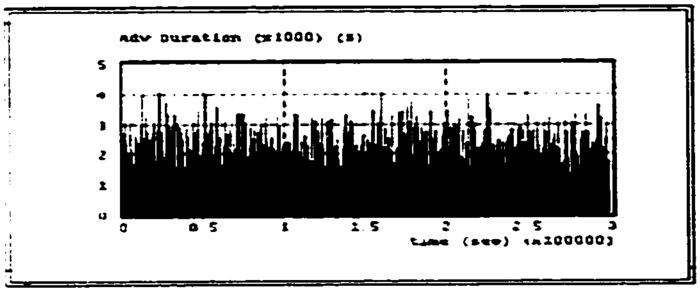


Figure 5.8: Admitted advance duration using QoS routing agent

5.7 Summary

This chapter explained in some detail the QoS routing algorithms and presented the simulation results. The simulations showed the effectiveness of QoS routing in supporting of network QoS and improving the utilization of the networks. It also presented a new method for QoS routing by using agents that make the necessary calculations on behalf of the routers.

Chapter 6

Implementation

6.1 Introduction

In previous chapters we have stated architecture for immediate and advance resource reservations with the ability to adapt the reserved resources according to user interactions. The architecture supports resource reservations at both, the network elements and the end systems. The simulation of admission control with user interactions and the QoS routing agent have showed a promising results. In this chapter, we advance a further step in testing the performance and scalability of the domain agency. We present an implementation of the domain agency and the resource reservation agents, and then apply extensive experiments. Then we integrate this work with an emerging multimedia multiparty enabling work application. This application provides an environment for distributed users through the Internet to collaborate and hold meetings. In the implementation, we used an agent management platform viz. FIPA. The implementation was completely done in Java programming language.

6.2 Limitations

The implementation has some limitations due to time and resource constraints. Furthermore, there is no implementation of RSVP that supports time specifications. In spite of these limitations we performed the implementation with the following assumptions:

- Substitute the RSVP messages with a regular messages that convey the same information as RSVP in addition to time specification.
- The OSPF protocol is used to provide the QoS routing agent with the network topology and topology updates. Therefore, in this implementation, we provide the network topology as a priori to the QoS routing agent, and assume that the network is stable during the course of the experiment.
- The aim of the experiment is to test the performance and the scalability of the domain agency. So, the traffic will consist of only the reservation requests and there is no need for packet classifiers and packet schedulers.

6.3 Overview of FIPA and FIPA-OS

FIPA [FIP00] stands for Foundation for Intelligent Physical Agents. It is an agent management specification. The agent management provides the normative framework within which FIPA agents exist and operate. It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agents. Figure 6.1, shows the different components of the agent platform, followed with a brief description of its components [FIP00].

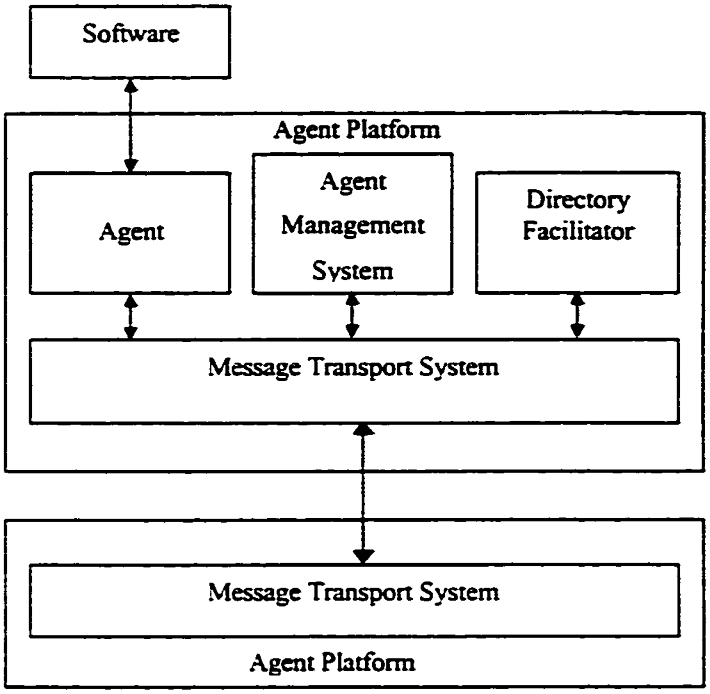


Figure 6.1: Agent Management Reference Model

- **Agent:** is the fundamental actor on an Agent Platform (AP), which combines one or more service capabilities into a unified and integrated execution model that may include access to external software, human users and communications facilities. An agent must have at least one owner. An Agent Identifier (AID) labels an agent so that it may be distinguished within the Agent Universe. An agent may be registered at a number of transport addresses at which it can be contacted and it may have certain resource brokering capabilities for accessing software.

- **Directory Facilitator (DF):** is a mandatory component of the AP. The DF provides yellow pages services to other agents. Agents may register their services with the DF or query the DF to find out what services are offered by other agents. Multiple DFs may exist within an AP and may be federated.

- **Agent Management System (AMS):** is a mandatory component of the AP. The AMS exerts supervisory control over access to and use of the AP. Only one AMS will exist in a single AP. The AMS maintains a directory of AIDs, which contain transport addresses (amongst other things) for agents registered with the AP. The AMS offers white pages services to other agents. Each agent must register with an AMS in order to get a valid AID.

- **Message Transport Service (MTS):** is the default communication method between agents on different APs

- **Agent Platform (AP):** provides the physical infrastructure in which agents can be deployed. The AP consists of the machine(s), operating system, agent support software, FIPA agent management components (DF, AMS and MTS) and agents.

- **Software:** describes all non-agent, executable collections of instructions accessible through an agent. Agents may access software, for example, to add new services, acquire new communications protocols, acquire new security protocols/algorithms, acquire new negotiation protocols, access tools, which support migration, etc.

FIPA-OS supports the majority of the FIPA specifications and is being continuously improved as a managed open source project. FIPA-OS is formerly implemented by Nortel Networks [NORTEL], then Nortel has transferred the management responsibility of FIPA-OS to Emorphia [EMO00]. FIPA-OS is implemented in 100% pure Java and it is available for public at [EMO00]. Two alternative distributions are available, one supporting JDK1.1, which is recommended for agents intended for deployment on small footprint devices; and the other supporting JDK1.2 (and 1.3) for all other uses.

6.4 Implementing the Domain Agency on Top of FIPA-OS

Figure 6.2, depicts the complete architecture of a router that hosts the domain agency. As stated previously, the domain agency and the FIPA-OS platform will be used in the experiments. For sake of clarity, we discuss the operation by assuming the presence of other protocols. Agents in the DA communicate with each other through internal platform message transport. For the third party software, which sometimes does not support agents, the communications is done with the help of agent wrappers. For example, the OSPF agent wrapper gets information from the OSPF protocol and submits them to QoS routing agent. The RSVP agent wrapper relays commands from the DRRA to the RSVP protocol and vice versa.

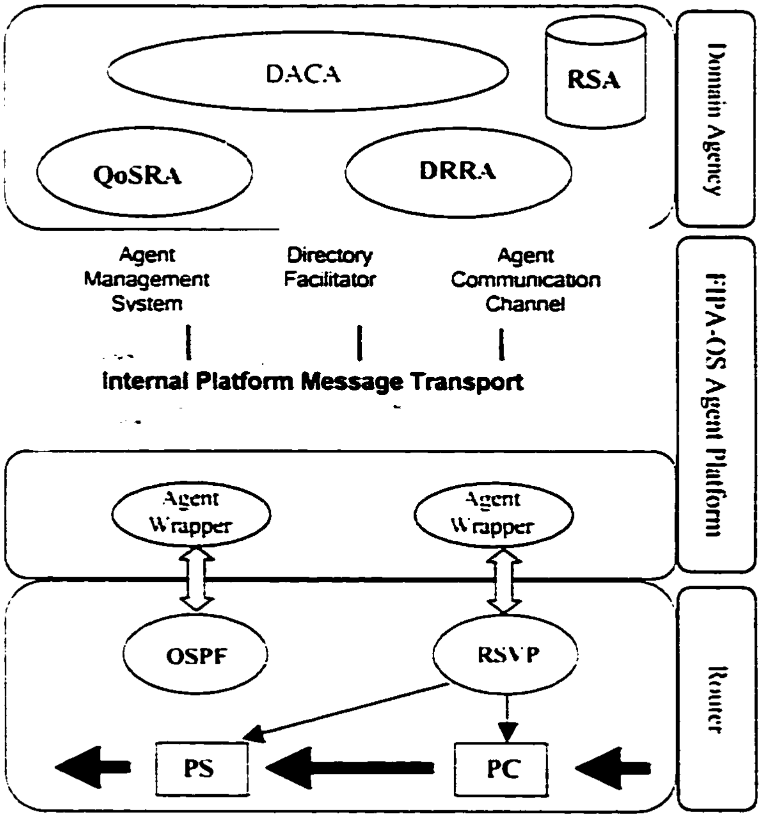


Figure 6.2: Domain agency on top of FIPA-OS

6.5 Experiment

To test the performance of the domain agency on top of FIPA-OS, we performed several experiments, which consist of the components shown in figure 6.3. The experiments were run in real-time, and parameters were selected to approximate the values used in real applications. For example, the duration of the session, the reserved bandwidth, etc. The network topology understudy was predefined to the domain agency and the topology is assumed stable during the experiments. The resource reservations agents (RRA1 and RRA2) reside in the same computer and send reservation requests to the DA through TCP/IP network.

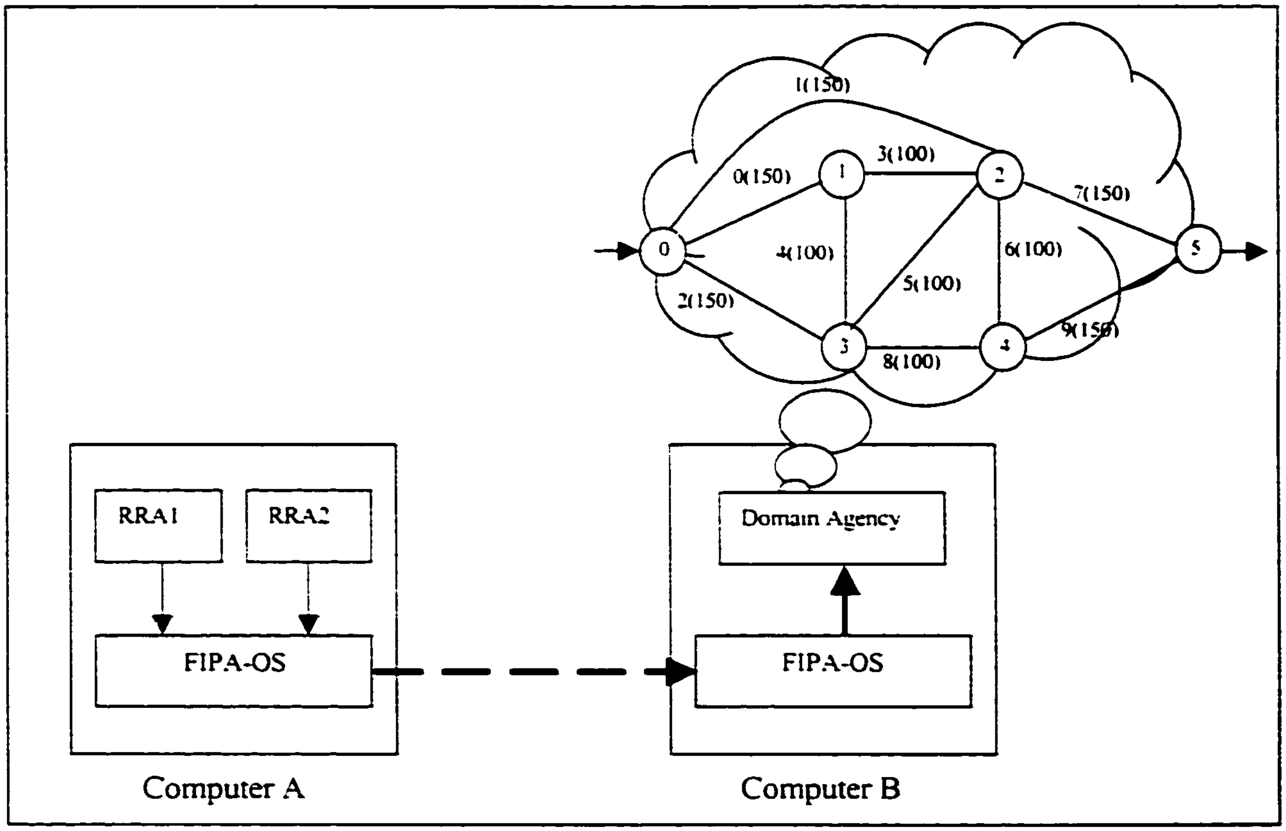


Figure 6.3: Experiment setup

6.5.1 Parameter Choices

The RRA1 sends advance reservation requests, while the RRA2 sends immediate reservation requests. The RRA agents send reservation requests to the domain agency that processes the requests according to certain policies. Following are the parameters selected for the agents and the domain agency:

RRA1

- Start time of the reservation is exponentially distributed with mean 2160Min (1.5 days);
- Inter-Arrival of the reservation requests is exponentially distributed with mean 5Sec;
- Duration of the reservation is exponentially distributed with mean 90Min (1.5 Hours);
- Amount of the bandwidth is uniformly distributed from 1 to 5.

RRA2

- Inter-Arrival of the reservation requests is exponentially distributed with mean 5Sec;
- Duration of the reservation is exponentially distributed with mean 90Min (1.5 Hours);
- Amount of the bandwidth is uniformly distributed from 1 to 5.

Domain Agency

Performs reservations according to the following policies

- Each user should make advance reservations within time greater than or equal to 60Min and less than or equal to 4320Mins (3 days);
- The advance and immediate reservation periods should be not less that 15Min and not more than 180Mins;
- The look-ahead times for immediate reservations are 0, 15, 30 and 60Mins.

6.5.2 Experimental Results

With these parameters, we ran the experiments in real-time, and monitored the following performance measures:

- Advance rejection probability;
- Immediate rejection probability;
- Immediate preemption probability;
- Duration of admitted immediate sessions;
- The utilization of each link in the network domain.

The rejection probability is the key measure for the performance of the algorithm. The goal is to minimize this probability for advance reservation requests and obtain a reasonable rejection probability for immediate reservation requests. Advance reservation requests have a higher priority than immediate requests. Therefore, at the start of every advance session, some of the immediate sessions are preempted if the total reserved bandwidth exceeds the link capacity. The duration performance measure is used to track the duration of the admitted immediate flows, which are expected to decrease as the rejection probability increases. The link utilization performance measure monitors the usage of each link according to the QoS routing algorithm.

Rejection Probability

Figure 6.4 shows the immediate rejection probability. The advance rejection probability is almost zero. The immediate rejection probability increases as we increase the look-ahead time. The rejection probability at look-ahead times 30 and 60 are almost the same and is slightly lower for look-ahead time 15. We prefer to use look-ahead time 15, as calculation overhead is low. In fact, the admission control algorithm is very sensitive to the duration of the reserved sessions. One session may be rejected because of the lack of resources available after a few seconds. This will introduce gaps between reserved sessions.

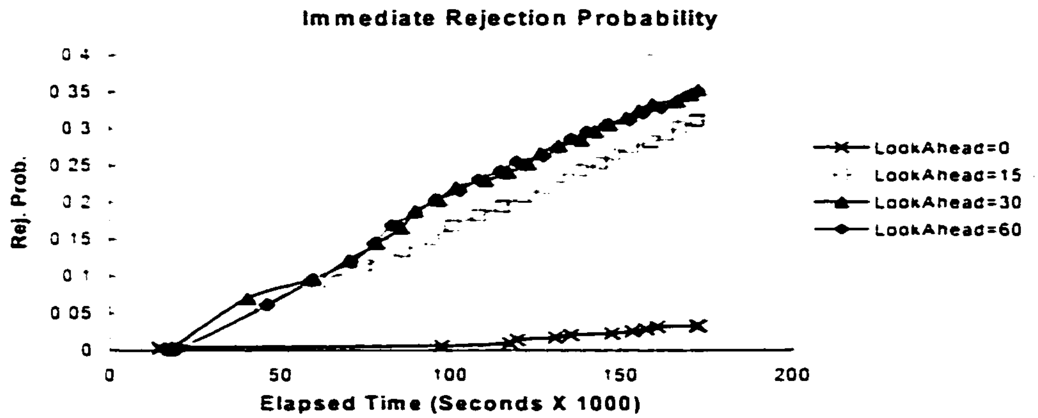


Figure 6.4: Immediate rejection probability.

Preemption Probability

The preemption probability for immediate requests is shown in figure 6.5. As can be seen, the preemption probability is inversely proportional to the rejection probability, decreasing as the look-ahead time increases. There is actually a trade-off between the rejection probability and the preemption probability. Some users may prefer not to be interrupted during the playback of the session, and may therefore accept a high rejection probability (repeating connection requests, using agents). Other users may prefer to have a low rejection probability (starting as soon as possible), and may therefore accept service degradations and interruptions during playback. Rejection and preemption can therefore be adjusted to user preference.

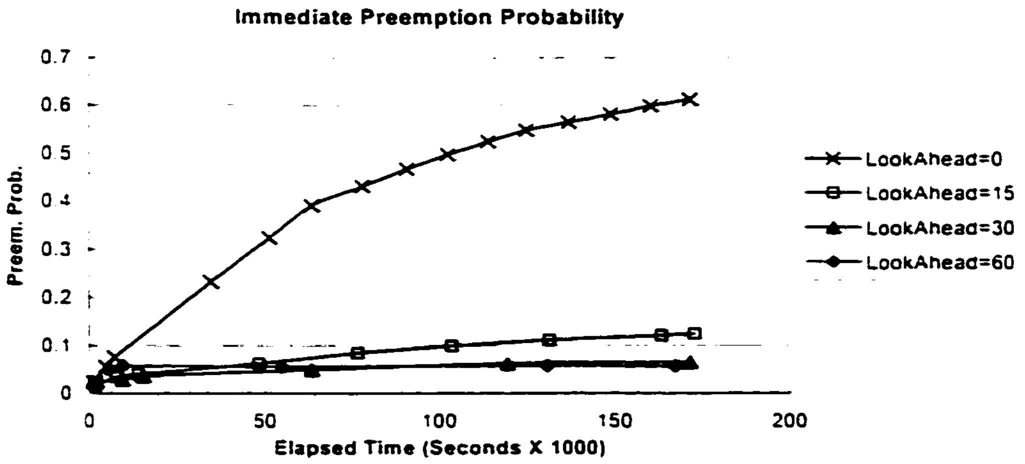


Figure 6.5: Immediate preemption probability.

Duration of Admitted Immediate Sessions

Because of the increasing rejection probability of immediate flows, we set up a performance measure to monitor the duration of admitted flows. This gives us a good idea of which sessions are most likely to be granted admission. In figure 14, we show the duration of admitted immediate sessions. As the look-ahead time increases, the sessions with smaller durations have a better chance of admission than sessions with larger durations. At higher look-ahead times, the rejection probability for immediate reservation requests increases, as can be seen from the gaps between the reserved sessions.

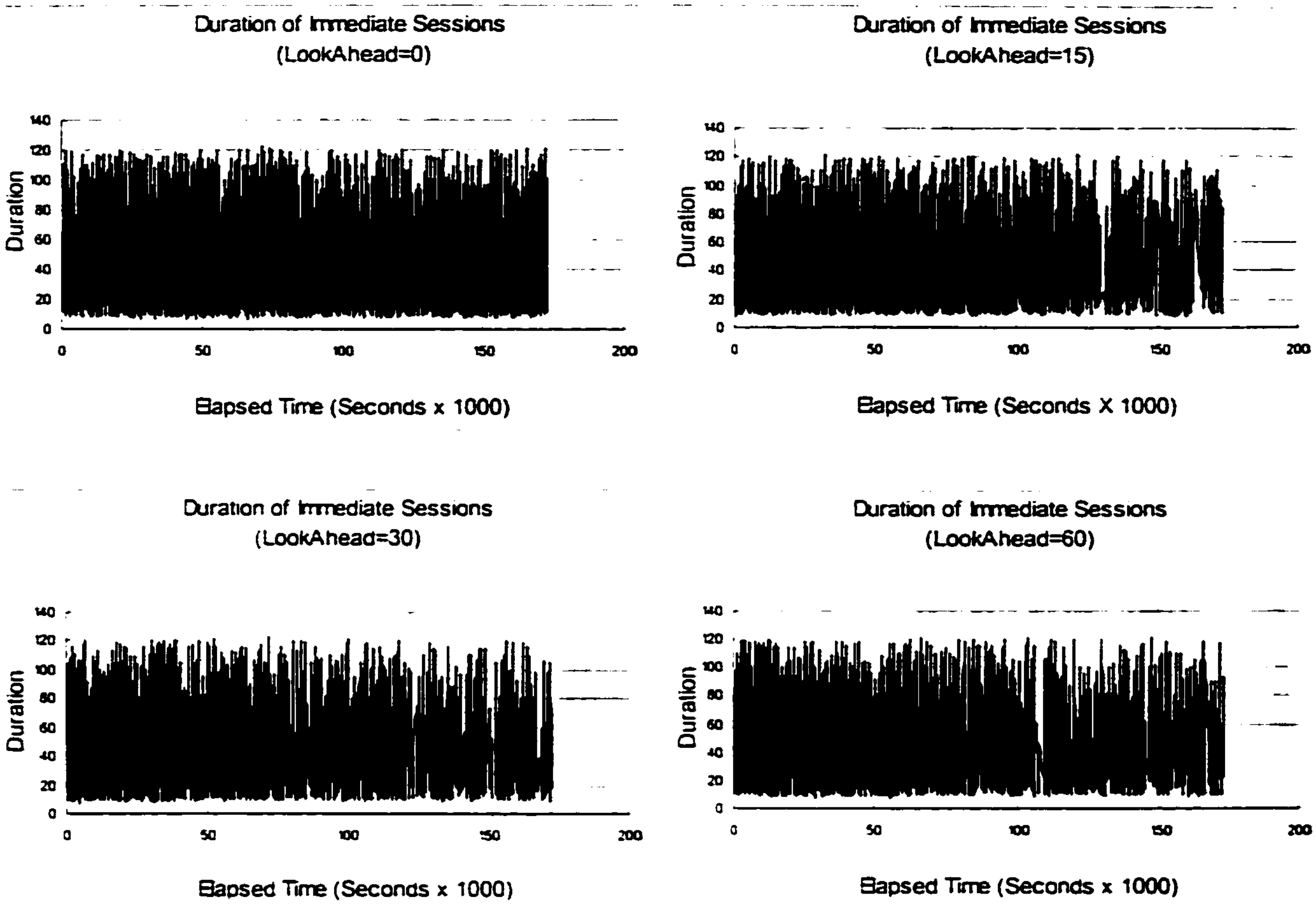
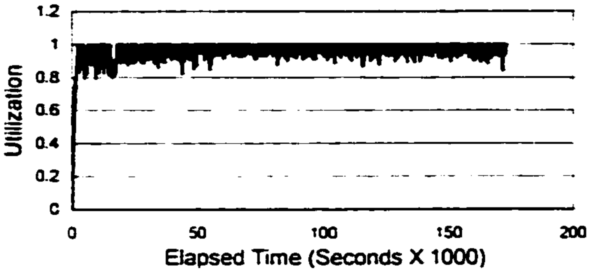


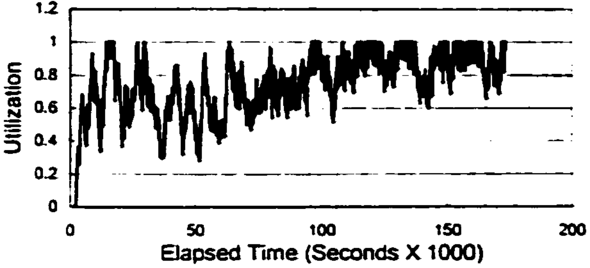
Figure 6.6: Duration of admitted immediate sessions.

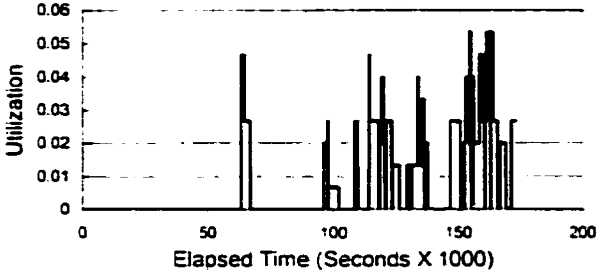
Link Utilization

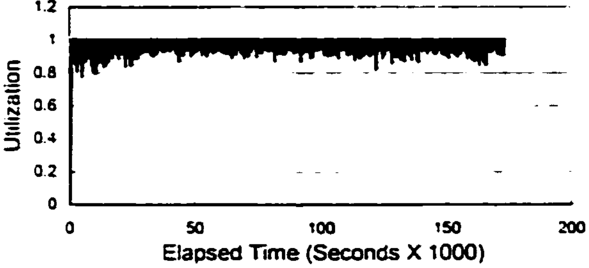
Figure 6.7 in the following pages , show the utilization of each link in the network for each look-ahead time and table 6.1. shows the average value for each link. All packets enter at node 0 and leave at node 5 (see figure 6.3). Therefore, according to the network topology and the QoS routing algorithm. links 0,3,7 have higher utilization. When there are not enough resources in link 3. the algorithm will change the path to 0-4-5-7. which is the next best bath. Once links 0 and 7 have finished their resources. the algorithm will choose the path 1-5-8-9. which explains the high utilization of link 5. As we can see from the figures and the network topology used, the links are used according to the operation of the QoS routing algorithm.

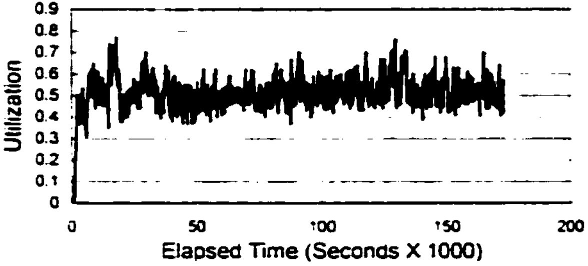
Meanwhile, the results also show that as the look-ahead time increases, the utilization of each link increases. At zero look-ahead time, all immediate requests are admitted. But whenever a new advance flow starts using its reserved resources, one or more immediate flows are preempted if the used resources exceed the total link capacity. This results in a high preemption probability and lower link utilization. As we increase the look-ahead time, admitted immediate flows guarantee their operation in the look-ahead period without any preemption. Using the QoSRA, the rejection probability increases by small values as the look-ahead time increases.

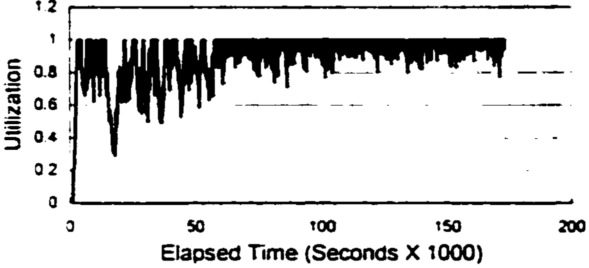

Utilization of Link_0
(LookAhead=0)

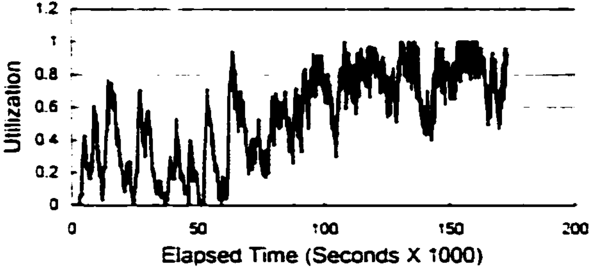

Utilization of Link_1
(LookAhead=0)

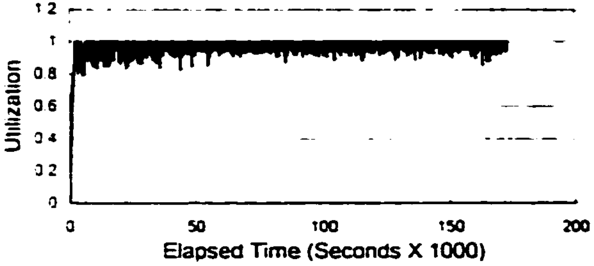

Utilization of Link_2
(LookAhead=0)

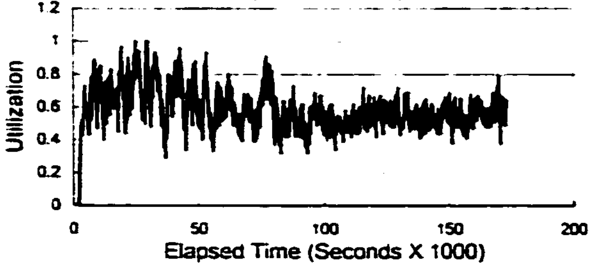

Utilization of Link_3
(LookAhead=0)

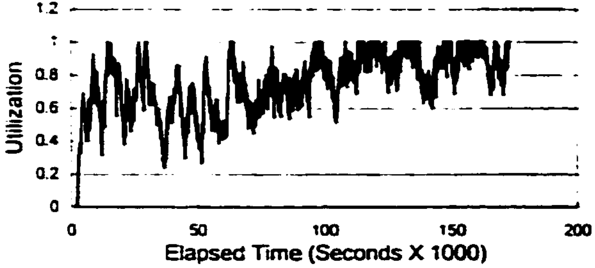

Utilization of Link_4
(LookAhead=0)

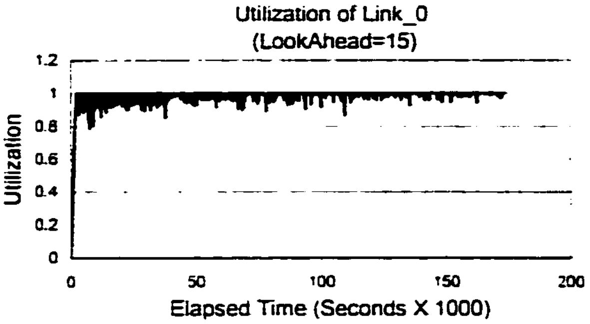

Utilization of Link_5
(LookAhead=0)

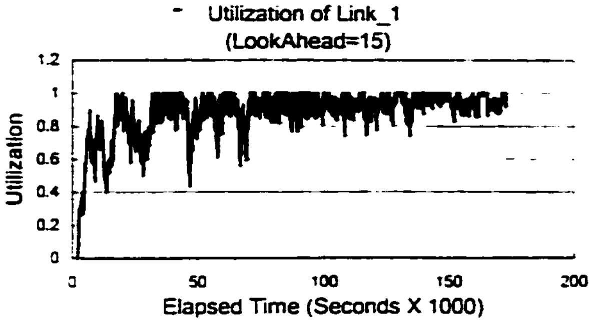

Utilization of Link_6
(LookAhead=0)

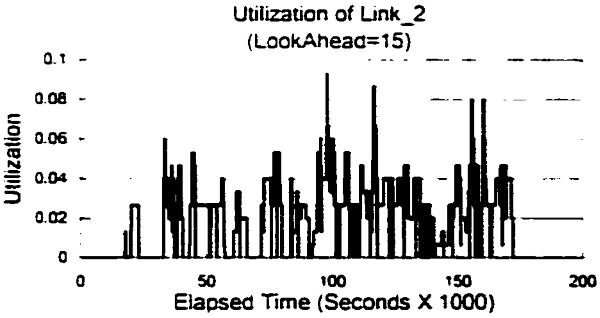

Utilization of Link_7
(LookAhead=0)

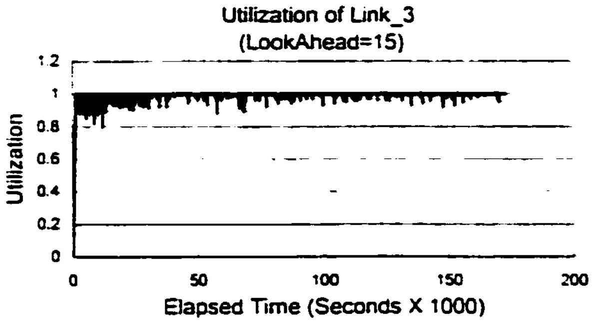

Utilization of Link_8
(LookAhead=0)

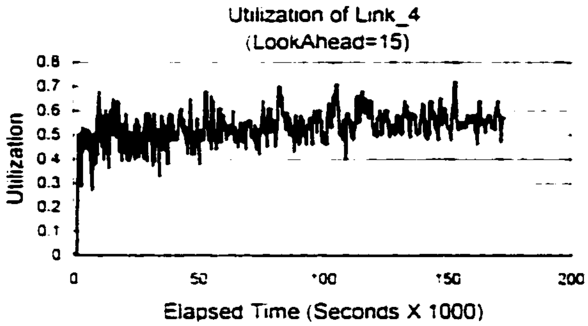

Utilization of Link_9
(LookAhead=0)

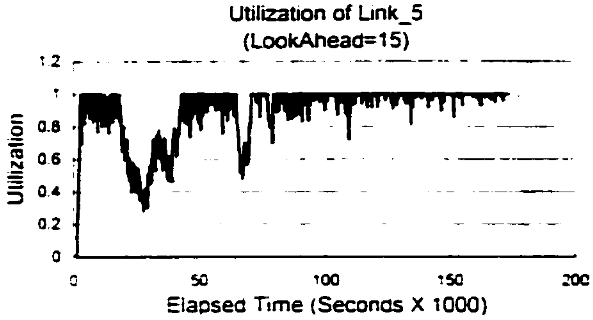

Utilization of Link_0 (LookAhead=15)

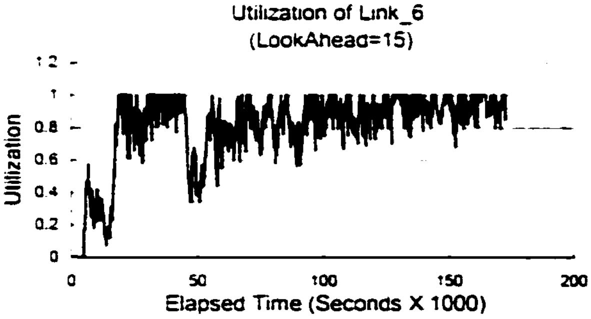

Utilization of Link_1 (LookAhead=15)

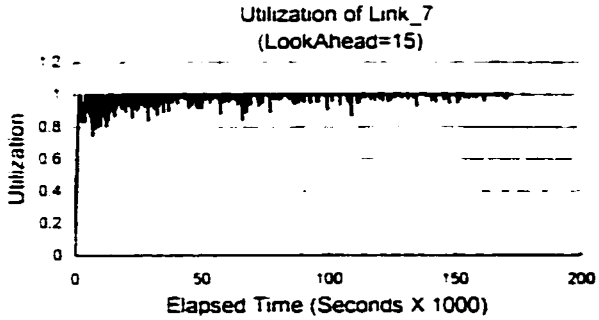

Utilization of Link_2 (LookAhead=15)

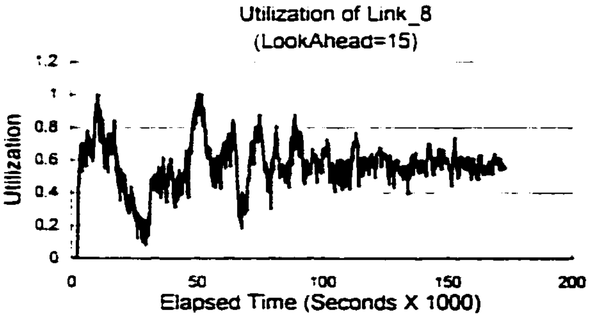

Utilization of Link_3 (LookAhead=15)

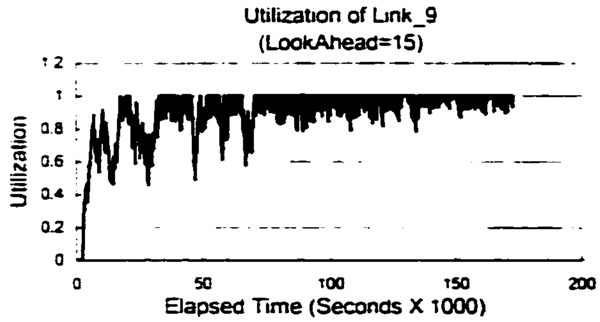

Utilization of Link_4 (LookAhead=15)

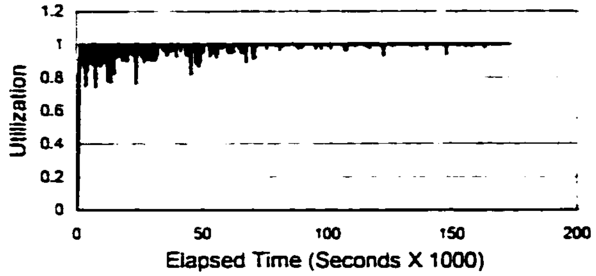

Utilization of Link_5 (LookAhead=15)

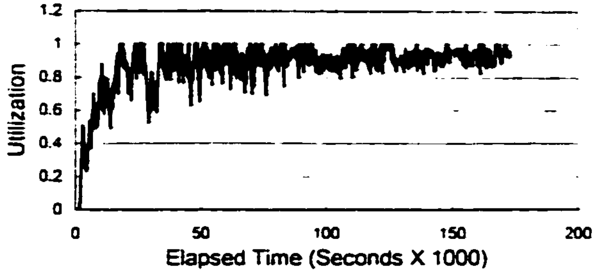

Utilization of Link_6 (LookAhead=15)

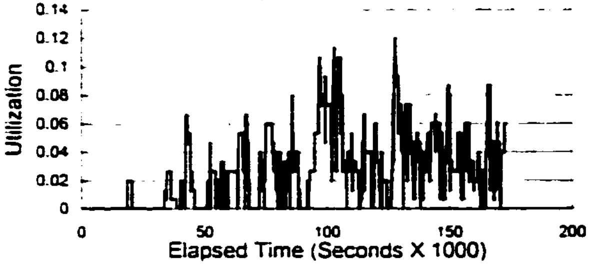

Utilization of Link_7 (LookAhead=15)

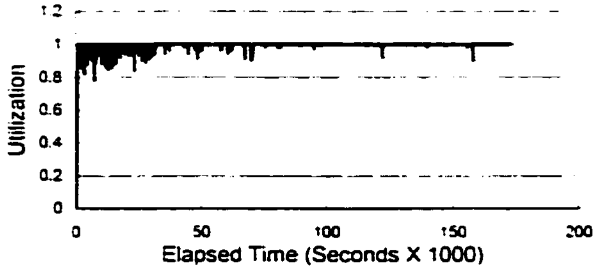

Utilization of Link_8 (LookAhead=15)

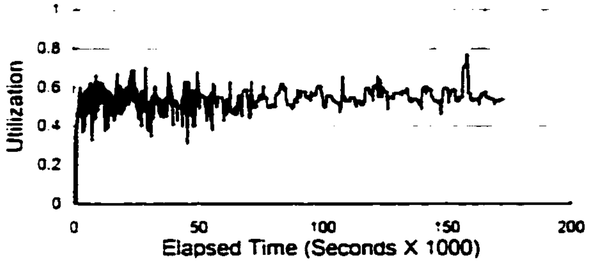

Utilization of Link_9 (LookAhead=15)

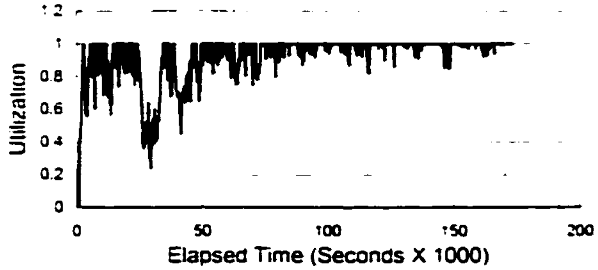

Utilization of Link_0
(LookAhead=30)

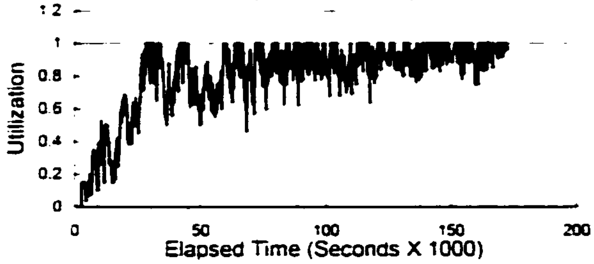

Utilization of Link_1
(LookAhead=30)

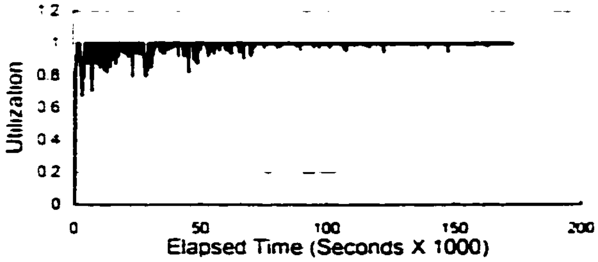

Utilization of Link_2
(LookAhead=30)

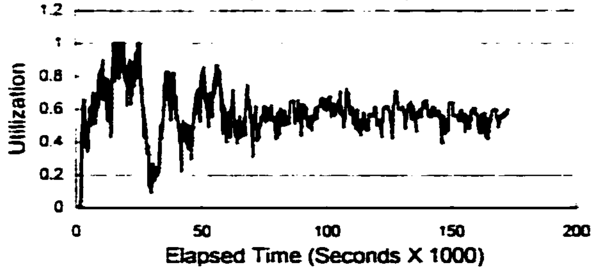

Utilization of Link_3
(LookAhead=30)

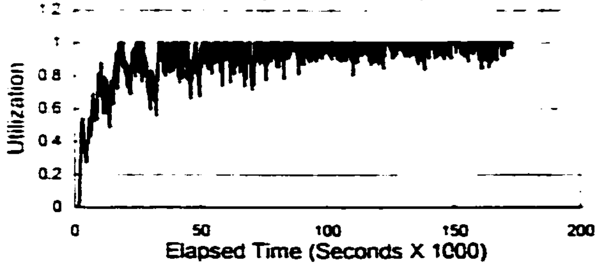

Utilization of Link_4
(LookAhead=30)


Utilization of Link_5
(LookAhead=30)


Utilization of Link_6
(LookAhead=30)


Utilization of Link_7
(LookAhead=30)


Utilization of Link_8
(LookAhead=30)


Utilization of Link_9
(LookAhead=30)

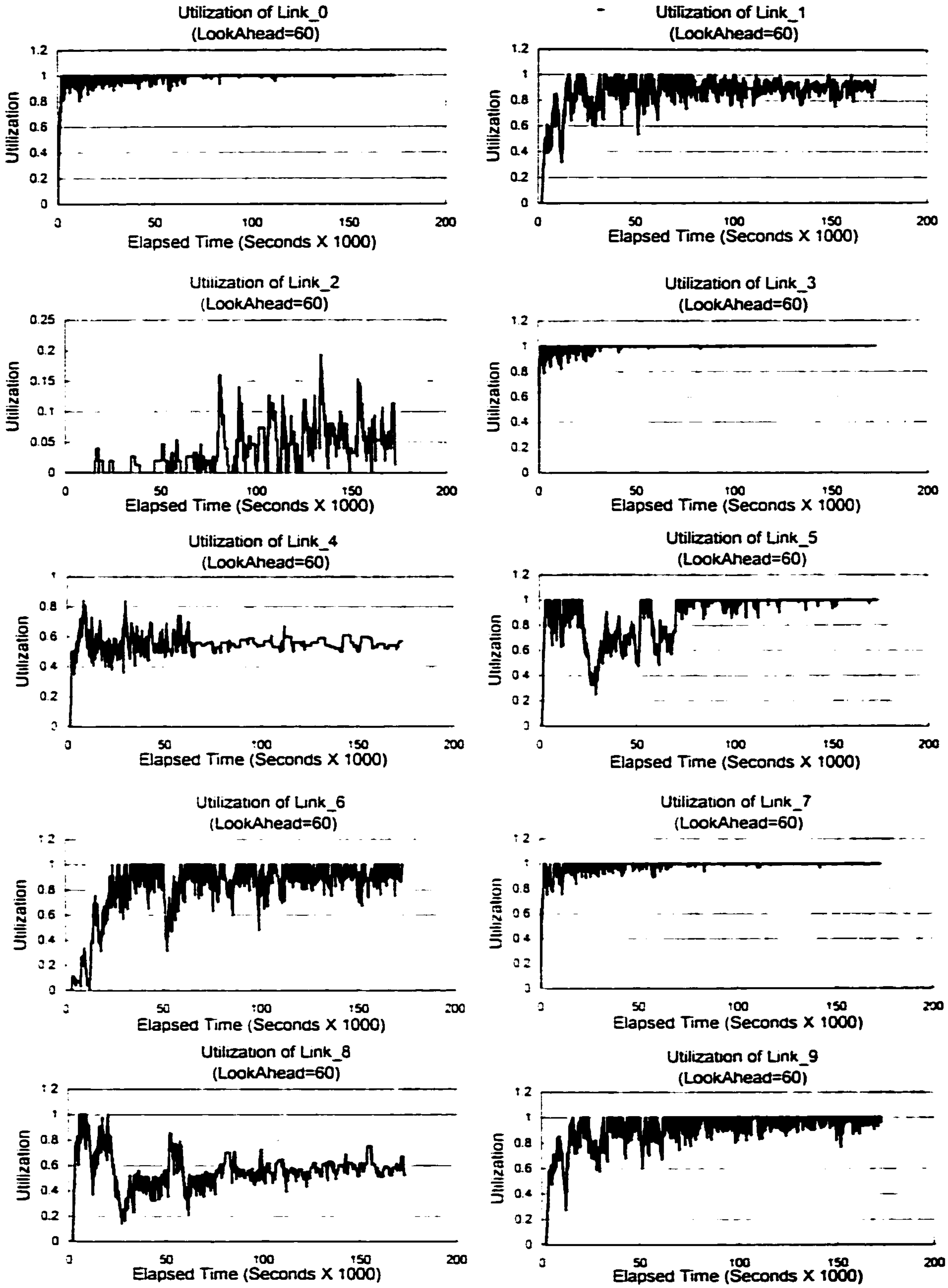


Figure 6.7: Links utilization

	Look-Ahead 0	Look-Ahead 15	Look-Ahead 30	Look-Ahead 60
Link 0 Util.	.972	.986	.987	.989
Link 1 Util.	.755	.882	.882	.871
Link 2 Util.	.007	.022	.029	.039
Link 3 Util.	.979	.992	.993	.994
Link 4 Util.	.521	.540	.545	.557
Link 5 Util.	.906	.902	.928	.890
Link 6 Util.	.546	.810	.806	.830
Link 7 Util.	.974	.983	.986	.987
Link 8 Util.	.593	.572	.588	.564
Link 9 Util.	.755	.904	.911	.912

Table 6.1: Average values of link utilization

6.6 Collaborative Environment for Virtual Teams Application

At the Multimedia and Mobile Agent Research Laboratory (MMARL) a work has been established to provide virtual teams with an environment for collaborative work. This environment provides an efficient and easy mechanism to support multimedia and multiparty applications such as video conferencing and remote consultation.

The collaborative-enabling environment is based on the use of agent technology. Agents communicate and negotiate with one another, in a cooperative manner, to allow virtual teams collaborating via multimedia applications. The virtual team invitation and QoS negotiation are done by a session initiation protocol (SIP), Then the RSVP is used to reserve resources for the negotiated QoS and time intervals. The media delivery is supported by Real-Time protocol (RTP). The transfer of media streams is managed by Media Agents, which contain audio and video tools necessary for achieving inter-media and intra-media synchronization, supporting different compression algorithms, multicasting and handling variable number of video/audio streams. The mechanism of establishing a virtual team meeting is summarized in the following steps:

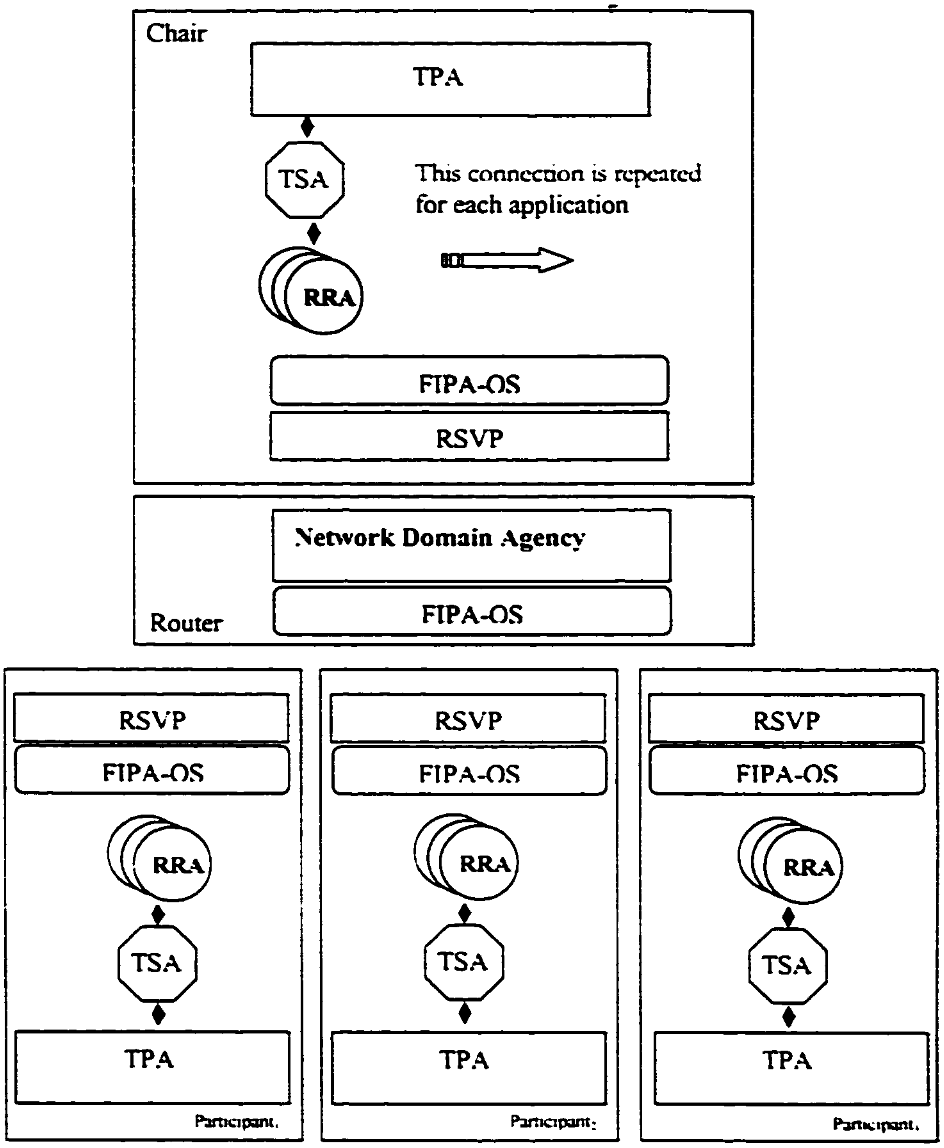
- At the beginning, a virtual team is defined and roles are assigned to participants.
- Then a meeting is scheduled. At this step the purpose of the meeting, time, duration and frequency are all specified.
- At the chair side there is a Team Control Agent (TCA), this agent generates a Control Agent for each participant. Each Control Agent negotiates with the corresponding participant, through SIP protocol, the time specification and QoS specification in addition to other parameters.
- After finishing the negotiation process, the TCA dispatches a Team Participant Agent (TPA) and *a* Team Service Agent (TSA) to every participant. The TPA maintains all participant's profiles. Participant profile contains information about the participant and the capability of his system (QoS parameters). The TSA is created for each service (application) and is responsible for providing the suitable environment for the application in the end systems and the network.
- The next step is the resource reservation process. This step is handled by a Resource Reservation Agent (RRA), which communicates with the RSVP protocol to reserve resources in the network.
- The participant becomes active as soon as the session time is due, at this time Media Agents become active and control the flow of media streams between the

chair and participants and between participants themselves. The chair controls the floor of the meeting.

- Each participant is allowed to update his profile and leave or join at any time.

6.6.1 Integration Plan

To integrate the work described in this thesis with the V-Team/CE, we suggest the architecture in figure 6.8. As we mentioned earlier that the team invitation and the QoS negotiation are done by SIP protocol. The result of this negotiation is maintained by the TPA as a user profile. For each application a TSA is activated which interacts with the TPA to provide the suitable environment for the application. This is the same as the work done by the Presentation Agent in the resource reservation architecture. So it will not be repeated here. Our contribution to V-Team/CE work starts from the Resource Reservation Agent, which is activated by the TSA.



TPA: Team Participant Agent
TSP: Team Service Agent
RRA: Resource Reservation Agent
RSVP: Resource Reservation Protocol

Figure 6.8: Integration with V-Team/CE

6.6.2 Design Considerations

- Heterogeneous Receivers: To cope with different users needs and different systems capabilities, receivers are assumed heterogeneous;

- Symmetric Receivers: The type of media and QoS values received by a participant's workstation are the same as that delivered by the same workstation;

- Resource Reservation Process: The reservation process starts concurrently between the chair and the participants, and between participants. The resources are reserved in one way and are used by both connected participants. The flow of the reservation messages, emerged from the chair to all participants and at the same time from each participant to the other participants that are in the participant's profile and have higher sequence number as shown in figure 6.9. The amount of the reserved resources is the minimum between any two participants;

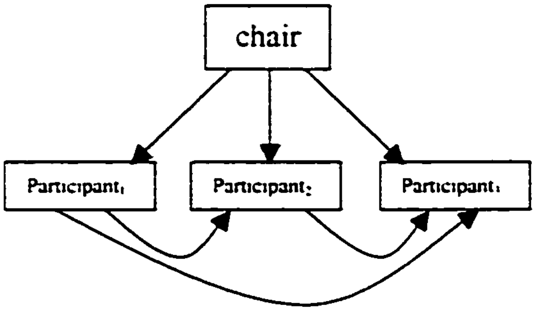


Figure 6.9: Flow of reservation messages.

- Resource Reservation Agent RRA: the RRA mentioned in the resource reservation architecture is activated by a TSA, and receives session specification from it.

6.6.3 Experiment

A connection was setup between the RRA in the chair side and the DRRA in the domain agency. The RRA sent a reservation request for each participant to the DRRA. The request contains the traffic and time specifications that each participant can handle. The DRRA made reservation and maintained a reservation state for all requests. In fact, this was a light weight reservations and is used to test the integration of the resource reservation system with the virtual team work. The performance evaluation was done in the previous section.

6.7 Summary

This chapter presents the implementation of the domain agency and the RRA in an agent environment and tested their performance. The results are very promising and encourage the deployment of agents in QoS management and resource reservations. We have also presented an integration with a multimedia multiparty application and showed how it can benefit from this architecture.

Chapter 7

Conclusions & Future Work

7.1 Conclusions

With the increase in Internet popularity and use, more services and applications are emerging to exploit the network resources. New types of applications are very sensitive to delay, and can only perform well with guaranteed QoS. This presents us with three possible directions. The first is to continue using the existing architecture of the Internet and simply to increase the overall capacity of the networks. The second is to enhance the existing architecture to cope with the new applications that require stringent QoS guarantees. The last direction is to build a new infrastructure that supports best-effort and guaranteed services.

The first choice provides fair access to all users; no users are blocked, but the degree of service depends on the load conditions of the Internet. Users receive best-effort service and must learn to tolerate QoS degradation. The question remains whether technology will provide us with a very high bandwidth compared to todays.

Will this high bandwidth be able to accommodate all Internet users of current and future sophisticated multimedia applications?

The second choice provides QoS guarantees, but some users will be denied access because resources are used up by current applications. This choice requires the enhancement of current network protocols to provide a suitable environment for real-time traffic. Most researchers agree on this as the preferred choice, since a QoS guarantee is becoming a requirement for the optimal performance of certain multimedia applications.

The last choice is not generally recommended, as it requires existing network protocols to be discarded as new ones are built. The existing Internet architecture has been in service for decades. It has been thoroughly analyzed and is well understood, with proven reliability and scalability for traditional traffic.

In our work we opted for the second choice. We believe that even if the capacity of the network increases, there is still a need for QoS management and guarantee. Our approach is to provide simple, efficient, scaleable and flexible end-to-end QoS resource reservation architecture. The present Internet is a collection of autonomous network domains. The proposed architecture therefore relies on domain agencies, which manage their resources in network domains. We proposed an end-to-end negotiation approach relying on agents that work on behalf of users to negotiate the requested QoS with the sender(s) and domain agencies. The agent has the ability to select the QoS values that it knows will satisfy the users. The domain agency is proposed to provide a better scalability of network resource use, relieving routers from the burden of performing immediate and advance resource reservations and complicated QoS routing calculations. The domain agency consists of four agents. The proposed QoS routing agent would perform all QoS routing tasks. The proposed admission control agent would perform immediate and advance admission control and would adapt the resources to reflect user interactions. Finally, there is a resource management agent and a reservation state agent. To test its performance, this domain agency was implemented on top of a FIPA-OS agent platform.

7.2 Future Work

This thesis proposes an end-to-end QoS architecture and has mainly focused on the components that make it scalable and efficient. Although it is a first step to provide a whole architecture that supports immediate and advance resource reservations with user interactions, some issues need to be further explored:

- Test the architecture in a large-scale network with the co-existence of QoS traffic and best-effort traffic, and apply the actual traffic that uses the reserved resources. This includes the deployment of the packet classifiers and packet schedulers, to implement the QoS values.
- The near future Internet will comprise wireless networks connected to wired networks, this imposes the enhancement of the architecture to handle mobile users as well as fixed users.
- Use this architecture in conjunction with the Diff-Serv architecture.
- Use this architecture with MPLS architecture.
- Enforce policies at the router, the domain and the autonomous system levels.
- Enhance the architecture to support multicast reservations and routing.

Bibliography

[ANA00] Anastasi G., La Corte A., Puliafito A. and Tomarchio O., "An Agent-based Approach for QoS Provisioning to Mobile Users in the Internet" In the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI2000), Orlando, Florida, USA, July 2000.

[APO99a] Apostolopoulos G., Guerin R., Kamat S., Orda A., Tripathi S., "Intra-Domain QoS Routing in IP Networks: A Feasibility and Cost/Benefit Analysis." Special Issue of IEEE Networks on Integrated and Differentiated Services for the Internet, September 1999.

[APO99b] Apostolopoulos G., Guerin R., Kamat S., Orda A., Przygienda T., Williams D., "QoS Routing Mechanisms and OSPF Extensions." RFC 2676, Experimental RFC, Internet Engineering Task Force, August 1999.

[AUR98] Aurrecoechea C., Campbell A., Hauw L., "A Survey of QoS Architectures", ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6 No. 3, pg. 138-151, May 1998.

[BAN91] Banerjea A. and Mah B. "The Real-Time Channel Administration Protocol" In Proceedings of 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany, November 1991.

[BAN94] Banerjea A., Ferrari D., Mah B. and Moran M., "The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences," TR-94-059, International Computer Science Institute, Berkeley, CA, November 1994.

[BAR98] Barzilai T., Kandlur D., Mehra A. and Saha D., "Design and Implementation of an RSVP-Based Quality of Service Architecture for an Integrated Services Internet", IEEE Journal on Selected Areas in Communications, Vol. 16, No. 3, April 1998.

[BAS99] Basturk E., Birman A., Delp G., Guerin R., Haas R., Kamat S., Kandlur D., Pan P., Pendarakis D., Peris V., Rajan R., Saha D. and Williams D., "Design and Implementation of a QoS Capable Switch-Router. Computer Networks, Vol. 31, No. 1-2, pp. 17-30, January 1999.

[BEC98] Beckmann C., Moin A. and Nog S., "Bandwidth Reservation with Selectable Bit-Rate Streams", Multimedia Systems, Springer-Verlag, 1998.

[BER82] Bertsekas D. "Dynamic Behavior for Shortest Path Routing Algorithms for Communication Networks" IEEE Transactions on Auto. Control, AC-27, 1982.

[BER96] Berson S., "Classical RSVP and IP over ATM", USC Information Science Institute, April 10, 1996.

[BER98a] Berson S., Lindell R., Braden R., "An Architecture for Advance Reservations in the Internet." Technical Report, USC Information Sciences Institute, July 16, 1998.

[BER98b] Berson S. and Vincent S., "Aggregation of Internet Integrated Services State", Internet Draft (draft-berson-rsvp-aggregation-00.ps), August 1998.

[BET95] Bettati R. Ferrari D. Gupta A. Heffiner W. Howe W. Moran M. Nguyen Q. and Yavatkar R. "Connection Establishment for Multi-Party Real-Time Communication", Proceedings of the Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, April 1995.

[BOL94] Bolotin V. "Modeling Call Holding Time Distributions for CCS Network Design and Performance Analysis" IEEE JSAC, 12(3), p 433-438, April 1994.

[BRA94] Braden R., Clark D. and Shenker S., "Integrated Services in The Internet Architecture: an Overview" RFC 1633, June 1994.

[BRA97] Braden R., Zhang L., Berson S., Herzog S. Jamin S., "Resource Reservation Protocol (RSVP)—Version1 Functional specification." RFC 2205, September 1997. ftp://ftp.ietf.org/internet-drafts-ietf-rsvp-spec-16.ps.

[BUL97] Bulterman D., "Models, Media And Motion: Using the Web to Support Multimedia Document," Multimedia Modeling (MMM'97), pp. 227-246, Singapore, November 1997.

[CAM93] Campbell A. Coulson G. and Hutchison D. " A Multimedia Enhanced Transport Service in a Quality of Service Architecture" In Workshop on Network and Operating System Support for Digital Audio and Video '93, Lancaster, England, November 1993.

[CAM94] Campbell A. Coulson G. and Hutchison D., "A Quality of Service Architecture," ACM Computer Communications Review, Vol.24, No.2, pp.6-27, April 1994.

[CAN00] Canonico R., Romano S., Sellitto M. and Ventre G., "A Scheme for Time-Dependent Resource Reservation in QoS-Enabled IP Networks", Proceedings of NETWORKING 2000, IFIP-TC6/European Union International Conference, Paris, France, 14-19 May, 2000.

[CCI89] CCITT Rec. I.350, General Aspects of Quality of Service and Network Performance in Digital Networks, International Telecommunication Union, Geneva, 1989.

[CHE98a] Chen S. and Nahrstedt K. "An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions. IEEE Network, Special Issue on Transmission and Distribution of Digital Video, Nov./Dec. 1998.

[CHE98b] Chen S. and Nahrstedt K. "Distributed QoS Routing with Imprecise State Information", in Proceedings of 7th IEEE International Conference on Computer, Communications and Networks (ICCCN'98), Lafayette, LA, pp. 614-621, October 1998.

[CHE98c] Chen S. and Nahrstedt K. "Maxmin Fair Routing in Connection-Oriented Networks", In Proceedings of Euro-Parallel and Distributed Systems Conference (Euro-PDS '98), pp.163-168, Vienna, Austria, July 1998.

[CLA92] Clark D., Shenker S. and Zhang L., "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," Proc. Of ACM SIGCOMM'92, pp. 14-26, Aug. 1992.

[COM95] Comer D. Internetworking with TCP/IP, Volume I, Principles, Protocols, and Architecture, Prentice Hall, 1995.

[CRA98] Crawley E., Nair R., Rajagopolan B. and Sãndick H., "A Framework for QoS-Based Routing in the Internet" RFC 2386, August 1998.

[DEL95] Delgrossi L. and Berger L., "Internet Stream Protocol (ST2)", RFC 1819, Version 2, August 1995.

[DEM90] Demers A., Keshav S. and Shenker S., "Analysis and Simulation of a Fair Queueing Algorithm", Internetworking: Research and Experience, Vol. 1, p. 3-26, 1990.

[DIJ59] Dijkstra E., "A Note on Two Problems in connection with Graphs" Numerical Mathematics, Vol. 1, 269-271, October 1959.

[EMO00] http://www.emorphia.com , Established in 2000. Source Code available at http://fipa-os.sourceforge.net .

[FAL98] Falchuk B., "A Platform for Mobile Agent-Based Data, Access, Retrieval, and Integration." Ph. D. Thesis, University of Ottawa, December 1998.

[FEL95] Feldmann A., "On-line Call Admission for High-Speed Networks", Ph. D. Thesis, Carnegie Mellon University, School of Computer Science, September 1995.

[FER90] Ferrari D. and Verma D. " A Scheme for Real-Time Channel Establishment in Wide-Area Networks. IEEE JSAC, 8(3), 368-379, April 1990.

[FER95] Ferrari D., Gupta A., Vertre G., "Distributed Advance Reservation of Real-Time connections," Fifth International Workshop on Network and operating system Support for Digital Audio and Video, Durham, NH, USA, April 19-21,1995

[FIP00] FIPA Agent Management Specification, Document Number XC00023F, July 2000, http://www.fipa.org/ .

[FLO95] Floyd S. and Jacobson V. "Link-sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking, 3(4), 365-386, August 1995.

[FLO96] Floyd S., "Comments on Measurement-based Admissions Control for Controlled-Load Service," Draft, Lawrence Berkeley National Laboratory, July 9, 1996.

[FRA96] Franklin S. and Graesser A. "Is it an Agênt, or just a Program?: A Taxonomy for Autonomous Agents" Proceedings of the Third International Workshop on Agent Theories, Architecture, and Languages, Springer-Verlag. 1996.

[GAR79] Garey M. and Johnson D., Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York. 1979.

[GEM95] Gemmell D., Vin H., Kandlur D., Rangan P. and Rowe L., "Multimedia Storage Servers" A Tutorial", "., IEEE Computer, 28(5):40--49, May 1995.

[GIB95] Gibbens R., Kelly F. and Key P., "A Decision Theoretic Approach to Call Admission Control in ATM Netowks", IEEE Journal on Selected Areas in Communications, 13(6), p. 1101-1114, August 1995.

[GUE00] Guerin R. and Orda A., "Networks with Advance Reservations: The Routing Perspective", Proceedings of INFOCOM'2000, Tel-Aviv, Israel, March 26-30, 2000.

[GUE91] Guerin R., Ahmadi H. and Naghshineh M., "Equivalent Capacity and its Applications to Bandwidth Allocation in High-Speed Networks", IEEE Journal on Selected Areas in Communications, 9(7), p. 968-981, September 1991.

[HAF96] Hafid A., Bochmann G., Dssouli R., "A Quality of Service Negotiation Approach with Future Reservation (NAFUR): A Detailed Study." Technical Report, University of Montreal, Montreal, 1996.

[HAF98] Hafid A., Bochmann G. and Dssouli R., "Distributed Multimedia Application and Quality of Service: A Review", Electronic Journal on Networks and Distributed Processing, No. 6, p. 1-50, 1998.

[HAF99] Hafid A. and Bochmann G., "An Approach to Quality of Service Management in Distributed Multimedia Application: Design and an Implementation". Multimedia Tools and Applications 9, p. 167-191, 1999.

[HAY99] Hayzelden A. and Bigham J., "Agent technology in communications systems: an overview" The Knowledge Engineering Review Journal. Cambridge University Press 12/99 Vol. 14, no. 4 pp 341-375 {193}

[HER00] Herzog S., Boyle J., Cohen R., Durham D., Rajan R. and Sastry A., "COPS usage for RSVP", RFC 2749, January 2000.

[HER99a] Herzog S., "RSVP Extensions for Policy Control", Internet Draft (draft-ietf-rap-rsvp-ext-06.txt), April 8, 1999.

[HER99b] Herzog S., "Signaled Preemption Priority Policy Element", Internet Draft (draft-ietf-rap-signaled-priority-03.txt), February 26, 1999.

[IETF] The Internet Engineering Task Force, http://www.ietf.org

[ISO95] QoS in distributed systems, SC21/WG7/N1034, 1995.

[JAF81] Jaffe J. "Bottleneck Flow Control" IEEE Transactions on Communications, COM-29(7); 954-962, July 1981. Correspondence.

[JAF84] Jaffe J. "Algorithms for Finding Paths with Multiple Constraints", Networks, 14, p 95-116, 1984.

[JAM97] Jamin S., Shenker S. and Danzig P., "Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service", IEEE/ACM Transactions on Networking, 5(1):56-70, February 1997.

[KAR96] Karmouch A. and Emery J., "A Playback Schedule Model for Multimedia Documents," IEEE Multimedia, pp 50-61, Spring 1996.

[KES97] Keshav S., An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network, Addison Wesley, 1997.

[KUN95] Kunii T., Shinagawa Y., Paul R., Khan M. and Khokhar A., "Issues in Storage and Retrieval of Multimedia Data", Multimedia Systems, Springer-Verlag, Vol. 3, p 298-304, 1995.

[LOR98] Lorenz D. and Orda A. "QoS Routing in Networks with Uncertain Parameters" in Proceedings of IEEE Infocom'98, San Francisco, CA, USA, March 1998.

[MA98a] Ma Q., and Steenkiste P., "On Path Selection for Traffic with Bandwidth Guarantees", In IEEE International Conference on Network Protocols, Atlanta Georgia, October 1997.

[MA98b] Ma Q., and Steenkiste P., "Routing Traffic with Quality-of-Service Guarantee in Integrated Services Networks", In the 8th International

Workshop on Networks and Operating Systems Support for Digital Audio and Video (NOSSDAV'98), Cambridge, England, July 1998.

[MAE95] Maes P., "Artificial Life meets Entertainment: Interacting with Lifelike Autonomous Agents." Special Issue on New Horizons of Commercial and Industrial AI, Vol. 38, No. 11, pp. 108-114, Communications of the ACM, ACM Press, November 1995.

[MEE00] Meer H., Corte A., Puliafito A. and Tomarchio O., "Programmable Agents for Flexible QoS Management in IP Networks", IEEE Journal on Selected Areas in Communication (JASC), 18(2), February 2000.

[MIT96] Mitzel D., Estrin D., Shenker S. and Zhang L., "A Study of Reservation Dynamics in Integrated Services Packet Networks", in Proceedings of the Conference on Computer Communications (IEEE Infocom), (San Fransisco, California), pp. 871-879, Mar. 1996.

[MOY94] Moy J., "Multicast Extensions to OSFP " RFC 1584, March 1994.

[MOY98] Moy J., "OSFP Version 2" RFC 2328, April 1998.

[NAH95a] Nahrstedt K. and Smith J., "Design, Implementation and Experiences of the OMEGA End-Point Architecture", Technical Report (MS-CIS-95-22), University of Pennsylvania, May 1995.

[NAH95b] Nahrstedt K. and Smith J. "The QoS Broker" IEEE Multimedia, 2(1), 53-67, Spring 1995.

[NAH98] Nahrstedt K. and Chen S. "Coexistence of QoS and Best Effort Flows – Routing and Scheduling" 10th Tyrrhenian International Workshop on Digital Communications: Multimedia Communications, Ischia, Italy, September 1998.

[NORTEL] http://www.nortelnetworks.com/products/announcements/fipa

[NWA96] Nwana H., "Software Agents: An Overview," Knowledge Engineering Review, Vol. 11, No. 3, pp. 205-244, October/November 1996.

[OPNET] Mil3 company, www.opnet.com.

[OWE98] Owezarski P., Diaz M. and Chassot C., "A Time-Efficient Architecture for Multimedia Applications", IEEE Journal on Selected Areas in Communications (JSAC), Vol. 16, No. 3, April 1998.

[PAR92] Parris C. Zhang H. and Ferrari D. "Dynamic Management of Guaranteed Performance Multimedia Connections. Multimedia Systems, 1(6), 1994.

[PAR92] Partridge C., "A Proposed Flow Specification" RFC 1363, September 1992.

[PHA98] Pham V. and Karmouch A., "Mobile Software Agents: An Overview," IEEE Communications Magazine, pp. 26-37, July 1998.

[PUL00] Puliafito A. and Tomarchio O., "Using Mobile Agents to Implement Flexible Network Management Strategies" Computer Communication Journal, 23(8), 708-719, April 2000.

[PUL97] Puliafito A. and Tomarchio O. and de Meer H., "An Agent-based Framework for QoS Management" In First World Congress on System Simulation WCSS'97: 4th ANMT on QoS, Singapore, September 1997.

[RAC91] RACE IBC CFS D510, General Aspects of Quality of Service and System Performance in IBC, RIC, Brussels, 1991.

[REI94] Reinhardt W., "Advance Reservation of Network Resources for Multimedia Applications," Proceedings of the Second International Workshop on Advanced Teleservices and High Speed Communication Architectures, Heidelberg, Germany, 26-28 September 1994.

[ROS00] Rosen E., Viswanathan A. Callon R., "Multiprotocol Label Switching Architecture", Internet Draft (draft-ietf-mpls-arch-07.txt), July 2000.

[SAL99a] Sallabi F. and Karmouch A., "Immediate and Advance Resource Reservations Architecture with Quality of Service Routing Agents" Proceedings of Multimedia Modeling (MMM'99), Ottawa, Canada, pp. 283-298, October 4-6, 1999.

[SAL99b] Sallabi F. and Karmouch A., "New Resource Reservation Architecture with User Interactions" Proceedings of IEEE Pacific Rim Conference on Communications, Computer and Signal Processing, Victoria, B.C. Canada, pp. 456-459, August 22-24, 1999.

[SAL99c] Sallabi F. and Karmouch A., " Resource Reservation Admission Control Algorithm with User Interactions " Proceedings of the Globecom'99, Rio de Janeiro, Brazil, December 5-9 1999.

[SCH92] Schulzrinne H., "Voice Communication Across the Internet: A Network Voice Terminal", Dept. of Computer Science, University of Massachusetts, Amherst, Massachusetts, Technical Report TR 92-50, July 29, 1992.

[SCH97] Schelen O. and Pink S., "Sharing Resources through Advance Reservation Agents. In Proceedings of IFIP Fifth International Workshop on Quality of Service (IWQoS'97), New York, pp. 265-276, May 1997.

[SHA98] Shaikh A., Rexford J. and Shin K., "Efficient Precomputation of Quality-of-Service Routes," Proc. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '98), Cambridge, England, pp. 15-27, July 1998.

[SHE95] Shenker S., "Fundamental design issues for the future Internet", IEEE JSAC, Vol. 13, pp. 1176-1188, 1995.

[SHE97a] Shenker S., Partridge C. and Guerin R., " Specification of Guaranteed Quality of Service" RFC 2212, September 1997.

[SHE97b] Shenker S. and Wroclawaski J., "Network Element Service Specification Template" RFC 2216, September 1997.

[STA95] Staehli R., Walpole J. and Maier D., "A Quality-of-Service Specification for Multimedia Presentations", Multimedia Systems, Springer-Verlag, Vol. 3, p. 251-263, 1995.

[STA97] Stallings W., Data and Computer Communications, Fifth edition, Prentice-Hall, 1997.

[STE95] Steinmetz R. and Nahrstedt K., Multimedia: Computing, Communications and Applications, Prentice Hall, 1995.

[STU95] Stuttgen H., "Network Evolution and Multimedia Communication", IEEE Multimedia, Vol. 2 No. 3, Fall 1995.

[SUN97] Sun Q. and Langendorfer H., "A Distributed Delay-Constrained Dynamic Multicast Routing Algorithm", Proceedings of IDMS'97, Darmstadt, Germany, September 1997.

[THA99] Thang P., Rueda J. and Alfa A., "Fast Restoration of IP Networks: Improving OSPF Restoration Performance", Proceedings of the 3rd

Canadian Conference on Broadband Research, IEEE, Ottawa, Canada, November 7-9, 1999.

[TOM99] Tommasi F. and Molendini S., "Some Extensions to Enhance the Scalability of the RSVP Protocol", Internet draft (draft-tommasi-rsvp-enhan-scalab-01.txt), October 1999.

[TOU00] Toukourou M. and Orozco-Barbosa L., "Experimental Study of an MPEG-2 Video-on-Demand Application over RSVP", Proc. IEEE/IEE International Conference on Telecommunications ICT'2000, Acapulco, pp. 835-839, May 2000.

[VOG95] Vogel A., Kerherve B., Bochmann G. and Gecsei J., "Distributed Multimedia and QoS: A Survey", IEEE Multimedia Vol. 2, No. 2, Summer 1995.

[VOL96] Volg C., Wolf L., Herrtwich R. and Wittig H., "HeiRAT-Quality of Service Management for Distributed Multimedia Systems", Special Issue on QoS Systems of ACM Multimedia Systems Journal, Vol. 6, No. 3, pp. 152-166, May 1998.

[WAN96] Wang Z. Crowcroft J., "Quality of Service Routing for Supporting Multimedia Applications" IEEE Journal on Selected Areas in Communications, 14(7): 1228-1234, September 1996.

[WIS98] Wischnik D. and Greenberg A., "Admission Control for Booking Ahead Shared Resources," In Proceedings of IEEE Infocom, San Francisco, California, March 1998.

[WOL95] Wolf L., Delgrossi L., Steinmetz R., Schaller S. and Witting H., "Issues of Reserving Resources in Advance," In Proceedings of NOSSDAV, Lecture Notes in Computer Science, Pages 27-37, Durham, New Hampshire, April 1995, Springer.

[WOL98] Wolf L., "Resource Management Multimedia Systems", Handbook of Multimedia Computing, CRC Press, Chapter 41, Boca Raton, Fl, USA, 1998.

[WRO97a] Wroclawski J., " Specification of the Controlled-Load Network Element Service, RFC 2211, September 1997.

[WRO97b] Wroclawski J., "The Use of RSVP with IETF Integrated Service," RFC 2210, September 1997.

[YAV99] Yavatkar R., Pendarakis D. and Guerin R., "A Framework for Policy-based Admission Control", Internet Draft (draft-ietf-rap-framework-03.txt), April 1999.

[ZAP98] Zappala D. and Kann J.,"RSRR: A Routing Interface For RSVP", Internet draft (draft-ietf-rsvp-routing-02.ps), June 1998.

[ZHA97] Zhang Z., Sonchez C., Salkewicz B., Crawley E., "Quality of Service Extensions to OSPF or Quality of Service Path first Routing (QOSPF)" Internet draft (draft-zhang-qos-ospf-01.txt", work in progress, September 1997.

Appendix A

Publications Based on This Research

1. F. Sallabi. A. Karmouch. "New Resource Reservation Architecture with User Interactions" Proceedings of IEEE Pacific Rim Conference on Communications. Computer and Signal Processing. August 22-24. 1999. Victoria. B.C. Canada.
2. F. Sallabi. A. Karmouch. "Immediate and Advance Resource Reservations Architecture with Quality of Service Routing Agents" Proceedings of Multimedia Modeling (MMM'99). October 4-6. 1999. Ottawa. Canada.
3. F. Sallabi. A. Karmouch. " Quality of Service Routing Agents for Supporting Immediate and Advance Resource Reservations " Proceedings of the 3rd Canadian Conference on Broadband Research. November 7-9. 1999. Ottawa. Canada.
4. F. Sallabi. A. Karmouch. " Resource Reservation Admission Control Algorithm with User Interactions " Proceedings of the Globecom'99. December 5-9. 1999. Rio de Janeiro. Brazil.
5. F. Sallabi. A. Karmouch. "Mobile Network Domain Agency for Managing Network Resources" Proceedings of Second International Workshop on Mobile Agents for Telecommunication Applications (mata'00). September 18-20. 2000. Paris. France.
6. F. Sallabi. A. Karmouch. "Network Domain Agency for QoS Management in OSPF Configured Networks". Networking and Information Systems Journal. No. 3, 2000