ADMISSION CONTROL OF MULTIMEDIA SESSIONS TO
A SET OF MULTIMEDIA SERVERS CONNECTED
THROUGH AN ENTERPRISE NETWORK

by

Md. Monirul Islam

A Thesis Submitted to the Department of Computer Science and Engineering in the Partial
Fulfillment of the Requirements for the
Degree of
Master of Science in Engineering
(Computer Science and Engineering)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DHAKA, BANGLADESH
DECEMBER 2004
The thesis “Admission Control of Multimedia Sessions to a Set of Multimedia Servers Connected through an Enterprise Network”, submitted by Md. Monirul Islam, Roll No. 040205036P, Registration No. 94245, Session April 2002, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Master of Science in Engineering (Computer Science and Engineering) and approved as to its style and contents. Examination held on December 12, 2004.

**Board of Examiners**

1. Dr. Md. Mostofa Akbar  
   Assistant Professor  
   Department of CSE  
   BUET, Dhaka-1000

   [Signature] Dec 12, 2004  
   Chairman (Supervisor)

2. Dr. Md. Shamsul Alam  
   Professor and Head  
   Department of CSE  
   BUET, Dhaka-1000

   [Signature]  
   Member (Ex-officio)

3. Dr. Md. Abul Kashem Mia  
   Professor  
   Department of CSE  
   BUET, Dhaka-1000

   [Signature]  
   Member

4. Dr. Md. Saidur Rahman  
   Assistant Professor  
   Department of CSE  
   BUET, Dhaka-1000

   [Signature] 12.12.2004  
   Member

5. Dr. M. Lutfar Rahman  
   Professor  
   Department of CSE  
   University of Dhaka  
   Dhaka-1000

   [Signature] 12.12.2004  
   Member (External)
Declaration

I, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by me under the supervision of Dr. Md. Mostofa Akbar, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka. I also declare that no part of this thesis and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Countersigned

(Dr. Md. Mostofa Akbar)
Supervisor

Signature

(Md. Monirul Islam)
Acknowledgement

I express my heartiest gratefulness to Almighty Allah for His divine blessings, which made me possible to complete this thesis successfully.

First and foremost, I must feel grateful to and wish to acknowledge me profound indebtedness to Dr. Md. Mostofa Akbar, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology. Deep knowledge and keen interest of Dr. Akbar in the field of multimedia influenced me to carry out this project and thesis. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this thesis.

I would like to thank the members of the graduate committee, Dr. Md. Shamsul Alam, Professor and Head, Dr. Md. Abul Kashem Mia, Professor, Dr. Md. Saidur Rahman, Assistant Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, and Dr. M. Lutfar Rahman, Professor, Department of Computer Science and Engineering, University of Dhaka for their valuable suggestions.

I am also very much grateful to Dr. M. Kaykobad, Professor, Department of CSE, BUET, who introduced me in the vast field of research and inspired me so many times to complete this thesis.

I would also like to express my heartiest gratitude to other faculty members and the staffs of the CSE Department of BUET.

I must acknowledge with due respect the constant support and patience of my parents for completing the thesis.
Multimedia transmission requires timely and ordered delivery of data with possibly no retransmission. As the number of users of multimedia applications is increasing day by day, overloaded servers and network congestion are also growing rapidly. These are among the main reasons for delay in accessing content from a server via a network path. Multimedia applications require a particular Quality of Service (QoS) level to be maintained to fulfill users' satisfaction. But it has another goal: maximizing the earned revenue from the users for the network owner. As resources are not unbounded, some selection criterion is essential when the number of users requesting resources for multimedia data transmission is large and their requested QoS is to be maintained. A particular QoS can be maintained only when necessary server and network resources is reserved. Current technology like best-effort service is unsuitable for above-mentioned requirements because all packets compete equally for network resources. The necessity to develop better QoS solutions to address these issues of multimedia transmission has led Internet Engineering Task Force (IETF).

In this thesis we designed an admission controller for an Enterprise Network which maximizes the earned revenue from the admitted users among many of them requesting multimedia data transmission with guaranteed QoS in terms of both server and network resources. Our admission controller considers the selection of both a server from several alternatives as a source of multimedia data and a delivery route along with reservation of resources in the selected server and on the delivery route so that the satisfaction of all admitted users is achieved. We redefine the Service Level Agreement (SLA), a contractual agreement between a user and the network owner and specifying several levels of proposal in terms of QoS. Protocols for data transmission in different stages of the admission controller have also been pointed out.

We designed necessary admission control algorithm and analyzed its complexity. Finally we simulated the admission controller, and performance data are collected extensively from a Java simulation program. We have also analyzed the performance
data with proper reasoning. The validation of the experiment has been presented by a graph theoretical analysis of the network and servers used for the simulation.
Table of Contents

Acknowledgement ........................................................................................................... i
Abstract ............................................................................................................................ ii
Table of Contents ............................................................................................................. iv
List of Figures .................................................................................................................. vii
List of Tables ................................................................................................................... ix
Glossary of Terms ............................................................................................................. x

CHAPTER ONE ................................................................................................................... 1
INTRODUCTION ............................................................................................................. 1
  1.1 MOTIVATION ........................................................................................................... 1
  1.2 PROBLEM DEFINITION AND PREVIOUS WORK ........................................... 2
  1.3 SCOPE AND FOCUS OF THE THESIS ............................................................... 6
  1.4 OUTLINE OF THE THESIS ................................................................................. 7

CHAPTER TWO ............................................................................................................... 8
PRELIMINARIES ............................................................................................................. 8
  2.1 INTRODUCTION .................................................................................................... 8
  2.2 ENTERPRISE NETWORK ...................................................................................... 8
  2.3 SERVICE LEVEL AGREEMENT .......................................................................... 9
  2.4 THE UTILITY MODEL ......................................................................................... 11
  2.5 FINDING THE SHORTEST PATHS ....................................................................... 13
  2.6 KNAPSACK PROBLEM AND ITS VARIANTS ................................................... 14
  2.7 LITERATURE REVIEW OF ADMISSION CONTROL ALGORITHMS ............. 16
  2.8 LITERATURE REVIEW OF ADMISSION CONTROL ARCHITECTURES ........... 19
  2.9 CHAPTER SUMMARY .......................................................................................... 21

CHAPTER THREE ......................................................................................................... 22
ADMISSION CONTROL METHODOLOGY ................................................................. 22
  3.1 INTRODUCTION ................................................................................................... 22
  3.2 PROPOSED ARCHITECTURE FOR A SET OF MULTIMEDIA SERVERS OF AN ENTERPRISE NETWORK ........................................................................................................ 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1 Motivation for a New Architecture</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2 The Architecture</td>
<td>23</td>
</tr>
<tr>
<td>3.2.3 Work Flow of the Admission Controller</td>
<td>24</td>
</tr>
<tr>
<td>3.2.4 A Practical Description of the Architecture</td>
<td>26</td>
</tr>
<tr>
<td>3.2.5 Protocols Used in Different Stages of the Admission Controller</td>
<td>27</td>
</tr>
<tr>
<td>3.3 Admission Control Methodology</td>
<td>28</td>
</tr>
<tr>
<td>3.3.1 The Algorithm</td>
<td>28</td>
</tr>
<tr>
<td>3.3.2 Detailed Description of the Algorithm</td>
<td>29</td>
</tr>
<tr>
<td>3.3.3 An Example Explaining the Algorithm</td>
<td>31</td>
</tr>
<tr>
<td>3.4 Complexity Analysis</td>
<td>33</td>
</tr>
<tr>
<td>3.5 Chapter Summary</td>
<td>34</td>
</tr>
<tr>
<td>CHAPTER FOUR</td>
<td>35</td>
</tr>
<tr>
<td>SIMULATION RESULTS</td>
<td>35</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>35</td>
</tr>
<tr>
<td>4.2 Simulation Steps of the Admission Controller</td>
<td>35</td>
</tr>
<tr>
<td>4.2.1 Imitating the Real Environment</td>
<td>36</td>
</tr>
<tr>
<td>4.3 General Simulation Parameters</td>
<td>37</td>
</tr>
<tr>
<td>4.3.1 Network Used in the simulation</td>
<td>40</td>
</tr>
<tr>
<td>4.3.2 Simulation Events</td>
<td>41</td>
</tr>
<tr>
<td>4.4 Simulation Results</td>
<td>41</td>
</tr>
<tr>
<td>4.4.1 Effect of the System Size</td>
<td>41</td>
</tr>
<tr>
<td>4.4.2 Effect of Epoch</td>
<td>44</td>
</tr>
<tr>
<td>4.4.3 Effect of the Number of Shortest Paths</td>
<td>46</td>
</tr>
<tr>
<td>4.4.4 Effect of the Percentage of Replica Considered</td>
<td>49</td>
</tr>
<tr>
<td>4.4.5 Effect of the Number of Servers</td>
<td>52</td>
</tr>
<tr>
<td>4.4.5.1 Keeping per Server Resource Fixed</td>
<td>52</td>
</tr>
<tr>
<td>4.4.5.2 Keeping Total Server Resource Fixed</td>
<td>55</td>
</tr>
<tr>
<td>4.4.6 Effect of the Expected Number of Users</td>
<td>57</td>
</tr>
<tr>
<td>4.5 Validation of Simulation Results</td>
<td>60</td>
</tr>
<tr>
<td>4.6 Chapter Summary</td>
<td>61</td>
</tr>
<tr>
<td>CHAPTER FIVE</td>
<td>63</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1: Media Server Farms proposed by Akbar ........................................ 4
Figure 1.2: Single EN based admission control architecture .......................... 5
Figure 2.1: A simple EN with three SLAs ................................................... 9
Figure 2.2: Relations between system and session utilities, and between resource mappings and constraints ..................................................... 12
Figure 2.3: Multiple routes between two nodes in a small network .................. 13
Figure 3.1: Architecture of the Multimedia Server System ............................ 24
Figure 3.2: The admission control algorithm ............................................ 30
Figure 3.3: Showing multiple paths from servers to users .............................. 32
Figure 4.1: Simulation Steps of the admission Controller ............................... 36
Figure 4.2: A sample Harary graph, $H_{4,9}$ ........................................... 40
Figure 4.3: Effect of system size on revenue and the number of accepted SLAs .... 42
Figure 4.4: Effect of system size on average admission time .......................... 42
Figure 4.5: Effect of system size on the average utilization ............................ 43
Figure 4.6: Effect of epoch on revenue and the number of SLAs ...................... 44
Figure 4.7: Effect of epoch on average time ............................................ 45
Figure 4.8: Effect of epoch on average utilization ....................................... 46
Figure 4.9: Effect of the number of shortest paths on the number of accepted SLAs and revenue ......................................................... 47
Figure 4.10: Effect of the number of shortest paths on average admission time .... 48
Figure 4.11: Effect of the number of shortest paths on the average utilization .... 49
Figure 4.12: Effect of the percentage of movie replica considered on the number of accepted SLAs and revenue ........................................ 50
Figure 4.13: Effect of the percentage of movie replica considered on average admission time ................................................................. 51
Figure 4.14: Effect of the percentage of movie replica considered on the average utilization ................................................................. 51
Figure 4.15: Effect of the number of servers on the number of accepted SLAs and revenue ................................................................. 53
Figure 4.16: Effect of the number of servers on average admission time .......... 53
Figure 4.17: Effect of the number of servers on the average utilization .......... 54
Figure 4.18: Effect of the number of servers on the number of accepted SLAs and revenue ............................................................... 55
Figure 4.19: Effect of the number of servers on average admission time ....... 56
Figure 4.20: Effect of the number of servers on the average utilization .......... 57
Figure 4.21: Effect of the expected number of users on the number of admitted users and revenue .............................................................. 58
Figure 4.22: Effect of the expected number of users on average admission time ..... 58
Figure 4.23: Effect of the expected number of users on average utilization .......... 59
List of Tables

Table 4.1: Different Simulation Parameters .......................................................... 37
Table 4.2: Average QoS requirements and revenues ............................................. 39
Table 4.3: Initialization of different QoS requirements and revenues .................... 39
Table 4.4: Expected number of accepted SLAs and revenue for different QoS levels ...
.............................................................................................................................. 61
### Glossary of Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000baseF</td>
<td>Fiber optics with 1000 Mbps bandwidth</td>
</tr>
<tr>
<td>1000baseT</td>
<td>Copper wire 1000 Mbps bandwidth</td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
</tr>
<tr>
<td>BB</td>
<td>Bandwidth Broker</td>
</tr>
<tr>
<td>BBLP</td>
<td>Branch and Bound Linear Programming</td>
</tr>
<tr>
<td>CoS</td>
<td>Class of Service</td>
</tr>
<tr>
<td>EN</td>
<td>Enterprise Network</td>
</tr>
<tr>
<td>HCA</td>
<td>Host Channel Adapter</td>
</tr>
<tr>
<td>HEU</td>
<td>Heuristic Solution</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>I-HEU</td>
<td>Heuristic Solution to solve MMKP</td>
</tr>
<tr>
<td>k</td>
<td>Number of shortest paths between source and destination</td>
</tr>
<tr>
<td>l</td>
<td>Number of QoS levels in each SLA</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MMKP</td>
<td>MMKP (Multiple Choice Multi-Constraint Knapsack Problem)</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multi Protocol Level Switching</td>
</tr>
<tr>
<td>n</td>
<td>Number of admissible SLAs</td>
</tr>
<tr>
<td>N</td>
<td>Number of nodes in the network</td>
</tr>
<tr>
<td>NAS</td>
<td>Network Attached Storage</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>OCI</td>
<td>Optical Carrier Level 1</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Data Protocol</td>
</tr>
<tr>
<td>VoD</td>
<td>Video on Demand</td>
</tr>
</tbody>
</table>
Chapter One

Introduction

1.1 Motivation

Overloaded servers and congested paths in a network connecting users to servers are among the main reasons for delay in accessing content from a server via a network path. Delay in data transmission is annoying to any user but is definitely unacceptable to the customers paying real money to enjoy multimedia sessions such as interactive video, Video on Demand (VoD), video clips, audio, etc. The admission control problem is the selection of an appropriate set of multimedia sessions from many contesting parties so that guaranteed Quality of Service (QoS) of all admitted sessions can be maintained. Maximization of both earned revenue and resource utilization is major concern for the network owner. Guaranteed QoS in terms of both network and server parameters is a vital issue in any multimedia transmission. The delivery of multimedia information with absolute QoS guarantees has presented Internet Service Providers (ISPs) with some challenges and opportunities:

- Users are satisfied only with desired level of QoS of multimedia data stream. So, networks must be able to carry multimedia streams with guaranteed absolute QoS. However, the present Internet is based on the best effort connectionless datagram service, without any QoS guarantee. This service, without any guarantees, is not suitable for paid service.

- Multimedia streams require timely and ordered delivery with very few retransmissions as possible. But existing TCP/IP or UDP/IP creates out of order delivery which requires reassembly. This happens due to the fact that they do not guarantee to follow same path all the time. Moreover, one of them also requires retransmission. Both problems create variable delay, called jitter which is unexpected in multimedia transmission.
• The introduction of some form of connection-oriented service atop the IP datagram service is necessary to solve this problem. The required bandwidth for the multimedia sessions must be reserved on a fixed path from the server to the user with low enough latency and jitter in order to guarantee absolute QoS over the network. All IP datagrams of the session must be routed along the fixed path because best effort datagram service does not guarantee timely delivery of multimedia streams on a particular path.

• One potential problem of the above solution is the congestion. If every transmission reserves network bandwidth on same path, it will be congested very soon. Moreover, simultaneous request of data from the same server by many users will make the server congested.

• So, to solve the above problem, the necessary resources like CPU cycles, I/O bandwidth, and memory in multimedia server(s) must be reserved, to ensure guaranteed delivery of multimedia streams from the multimedia server.

• The resources in the multimedia server (CPU cycles, I/O bandwidth and memory) and in the network (link bandwidths) are finite. If these resources are overbooked then QoS may not be maintained. Therefore, some form of admission control is necessary, to reject some of the applicants’ sessions when insufficient resources are available to serve all of them. Maximization of revenue by admitting profitable sessions is a vital issue for the network owner and an important objective of the admission controller.

To solve all the problems as a whole, a selection criterion is necessary when there are many contesting parties requesting resources from a finite amount and when cost of service or revenue is also a major concern for the service provider. An admission controller implements this selection criterion.

1.2 Problem Definition and Previous Work

Ensuring guaranteed QoS of multimedia stream is essential for users’ satisfaction. Guaranteed absolute QoS for multimedia service requires end-to-end guarantees,
covering the server, network and client. A connection of sufficiently high bandwidth is required from the source of multimedia stream to the destination. A multimedia stream must follow a fixed path from the server to the user during the period of the admitted session, and each link must have enough link bandwidth, with low enough latency and jitter, to carry the multimedia stream with the required absolute level of QoS. The source server must have enough capability to deliver multimedia streams to all admitted users with guaranteed absolute QoS. Sufficient network bandwidth and server resources (CPU cycles, I/O BW and memory) have to be reserved for each admitted multimedia session. Moreover, the user's machine must have enough CPU cycles, I/O bandwidth, memory and hard disk space to play the multimedia stream with guaranteed absolute QoS.

The Admission Controller works as a central resource allocator that allocates the resources of network bandwidth and of the server such as CPU cycles, I/O BW, memory, etc. to the user when her multimedia session starts. It works also as a selector of optimal number of users along with their delivery routes and media source. Each prospective user provides a revenue offer for each level of QoS for the multimedia service. The resource requirement of each quality level is actually set by the owner of the network. Admission control will be done based on the offered revenue, available and required resources.

Finding the near optimal selection of both a data server and delivery routes from server to customer is a very important topic in content routing research [1]. Delivery of multimedia streams with guaranteed QoS from a single multimedia server to maximize the utility (earned revenue) is proposed by Khan et al [2]. But usage of a single server to provide multimedia stream is a bottleneck in data communication. Although high performance server may be used, but the amount of data to be delivered, and delivered with real time constraints, are so huge, and in many cases the anticipated participants or viewers are so many, that it will be clearly impossible to maintain all the data in just one server. The limited capacities of servers, the desire to exploit geographic locality of reference, and the need for fault tolerance, are all reasons to partition or replicate multimedia streams across multiple servers.
Replicating multimedia data among various servers and selecting optimal one based on server parameters is studied by Akbar et al [3].

In [3], they propose an admission controller of a set of Media Server Farm, where multiple servers are organized as a farm and several farms are connected to a network as shown in Figure 1.1. The media servers store a large collection of movies which are replicated in multiple servers. The main purpose of a Media Server Farm is to provide videos with guaranteed QoS to the customers, who are connected to the Media Server Farms through an Enterprise Network (EN). The broker connected to the EN works as an admission controller for requests from the customers connected to the EN. The broker does admission control based solely on resources available in the media servers without considering network bandwidth. Therefore, the selection of optimal routes from the chosen server to the user is not discussed.

Transmission of multimedia streams through the links of an Enterprise Network (EN) by controlling admission based on Service Level Agreement (SLA) between the user and the network owner has been proposed by Watson [4] and further studied by Akbar et al [5]. Figure 1.2 shows their single Enterprise Network based architecture to support real time multimedia service with absolute QoS guarantees. The users (denoted by workstations) and servers are connected to the switches represented by circular nodes of the network.
In Figure 1.2 the Bandwidth Broker (BB), connected to a switch of the Enterprise Network, is a machine running an admission control algorithm for SLA admission control. The users submit their SLAs to the BB, mentioning the server from where they want to get data. The BB finds the suitable paths from the source to destination and determines whether the multimedia streams are allowed to transmit through any of these paths. The broken line in the figure shows the logical connection between a user and the BB. The BB determines the path and QoS level of each multimedia session between the source and the destination.

But one obvious problem in architecture of Figure 1.2 is that the BB determines the path and QoS level based on the network resource only without considering the server parameters, and, thereby, it can not find the optimal server. On the other hand, the architecture presented in Figure 1.1 finds only the appropriate server but does not discuss about the delivery routes. So, simultaneous selection of both servers and routes in an Enterprise Network based on the respective parameters with guaranteed agreement is still undone and an important issue for customer satisfaction. Also optimal utilization of resources of network link and server resources to earn maximum revenue is important for the network owners.
1.3 Scope and Focus of the Thesis

Allocation of resources, such as memory, CPU cycles and I/O bandwidth of multimedia servers and link bandwidth in the network, is essential to ensure Quality of Service (QoS) of multimedia services delivered over the network. Our aim is to address this problem in an Enterprise Network with a set of multimedia servers as follows:

- Definition of an SLA allowing the users to declare
  - multimedia data stream that they want to enjoy
  - level of satisfaction in terms of both server and network parameters
  - revenue to be given to the network owner

- An admission control algorithm to
  - admit users satisfying guaranteed QoS
  - maximize the earned revenue subject to constraints of the total available resources in the servers and in the Enterprise network

- Analyze the performance of the proposed admission control algorithm by simulating the admission controller and varying different server and network parameters such as:
  - system size
  - epoch
  - no. of shortest paths from source to destination
  - no. of replication of multimedia data streams
  - no. of servers
  - no. of expected users

This research presents an admission controller for an Enterprise Network. A universal admission controller for a public network such as the Internet is out of scope of this research. Users' requests are made in the form of SLAs as mentioned earlier and bandwidth requirement for these requests is ignored compared to the bandwidth requirement for actual data transmission. We also assume that a central database will keep all network and server resource information but the actual multimedia data stream will be kept only on the servers with possible replications of each data on different servers. The media servers and the underlying network are assumed to be able to reserve resources for admitted users’ sessions. Another point is that a
simulation is required to measure the profit maximization and to observe the performance measurement of the admission controller, but it will not designed to demonstrate all the communication protocols required for the system.

1.4 Outline of the Thesis

This thesis describes in detail an admission control algorithm for multimedia sessions to a set of multimedia servers connected through an Enterprise Network. The focus of the contribution is on the maximization of revenue earned by the network owner while maintaining Quality of Service of admitted sessions considering resources of both servers and underlying network. This introductory chapter summarizes the contribution by depicting focus and related works.

In Chapter 2, a detail description of Enterprise Network, Utility Model and extended definition of Service Level Agreement has been given. A literature review of knapsack problem and its variants, specially about MMKP, has been done. We have also pointed out literatures related to finding shortest paths in a network. Finally some more detail review of different admission control algorithms has been presented.

Chapter 3 presents our proposed architecture of multimedia server system. Detail description of different components, its workflow, practicality, necessary protocols have been presented. A new algorithm for admission control in the multimedia server system is described with detail explanation and an example. The complexity analysis has been presented.

In Chapter 4 we present the results from extensive simulations. The simulation procedure is described with its necessary parameters. The performance of the proposed system has been analyzed based on the results from the simulations.

We conclude the thesis in Chapter 5 by describing the major contributions of this research. Some future research works have also been suggested in this chapter.
Chapter Two

Preliminaries

2.1 Introduction

In this chapter we will review different admission control algorithms and architectures used in different systems. Service Level Agreement for specifying multimedia requests and Utility Model for doing admission control will be presented. Knapsack problems and its variants will also be reviewed because a special variant of the knapsack problem will be used as a technique to solve the admission control algorithm developed for our architecture.

2.2 Enterprise Network

An Enterprise Network (EN) is a network with limited number of nodes and links administered by a single organization or an autonomous subsidiary of an organization. An EN can be used for video and teleconferencing among distant employees. Similarly, a telecommunication company can set up an EN especially for multimedia stream transmission. A multimedia user will submit her request to the owner of the EN for network bandwidth or server resources to transfer data from a source node to a destination node of the EN. An enterprise network may be one that connects every computer in every location of a company, and runs the company’s mission-critical applications. Figure 2.1 shows a small Enterprise Network with six nodes and three SLAs from node S to node D, represented by the dotted lines, thick lines and thick broken lines respectively. The thin lines in Figure 2.1 represent the links of the network.
2.3 Service Level Agreement

A Service Level Agreement (SLA) in an Enterprise Network is a contract between the network owner of an EN and a customer. It is an agreement to provide a certain level of service, described by a bandwidth and other server resource requirement with a delay bound from a source to a destination at a given price. There are many proposed definitions of SLA [5]. In our research we use our own definition of SLA. \( SLA_i \), the \( i \)th SLA of the set of submitted SLAs to the owner of the system is defined as follows:

**Multimedia Data, \( m_i \):** Name of the movie or other multimedia stream to enjoy. Sources of data will be identified by the admission controller from its central database.

**Destination, \( D_i \):** User’s location, ie, node of the network.

**Delay Bound, \( d_{ij} \):** The network link propagation delays plus the switching delays between Source and Destination nodes must be bounded above by \( d_{ij} \) for the \( j \)th level of QoS for \( SLA_i \).

**QoS levels:** Users are allowed to submit different options for desired QoS levels and bid prices. So, an SLA specifies a set of QoS levels determined by the required bandwidth and delay constraints, plus offered price rates. The \( j \)th QoS level of the \( i \)th SLA is defined as follows:

**Bandwidth requirement, \( B_{ij} \):** QoS level \( j \) of \( SLA_i \) requires bandwidth \( B_{ij} \) between a particular source of \( m_i \) and \( D_i \). Thus, all the links on the chosen path between a source and \( D_i \) must reserve bandwidth \( B_{ij} \) for \( SLA_i \), when operating at QoS level \( j \).
Server Resource Requirement: This defines how much portion of following server parameters is needed to guarantee respective QoS:

i) CPU Cycles,
ii) Memory, and
iii) I/O bandwidth

Revenue: For each QoS level \( j \), there must be a rate of revenue or utility paid by the Customer \( i \), and earned by the network owner.

Flow Attributes: An SLA also specifies several attributes of the data flow from the source to the destination. These are as follows:

Path Restriction Flag: This flag specifies whether or not the path can be changed after the session starts. If the flag is set we cannot re-route this SLA in order to increase the QoS level of this SLA or other SLAs, or to admit a new SLA.

Server Restriction Flag: In our architecture a movie is replicated in multiple servers. This flag specifies whether or not the server can be changed after the session starts. The effect of this flag is similar as path restriction flag. That is, if this flag is set we cannot change the server for this SLA.

Down Flag: If this flag is set then the level of QoS of an admitted SLA can be downgraded at times of quality adaptation. (This is not always advisable for psychological or marketing reasons, just as it is not always advisable to downgrade a business class airline passenger to the economy class.). The users who do not care about quality downgrading during the session might set this flag in order to get a cheaper rate.

Up Flag: If this flag is set, then upgrading of the QoS level is permitted after the admission of an SLA. A user who does not expect change of QoS level when she gets used to it after getting admission might not set this flag.

These four flags are set by the users and changes will be considered by the Admission Controller during the next computation of QoS levels and paths.
2.4 The Utility Model

In this thesis we define utility as the revenue earned by the network owner from a user enjoying a service. The term utility can also be defined by different kinds of satisfaction of the user (response time, quality of the service etc.), but we are not considering those in this thesis. Khan et al [2] proposed the Utility Model for the admission and adaptation of sessions in a multi-session Multimedia Service Provider.

Let there be \( n \) session requests \( s_1, s_2, \ldots, s_n \) from \( n \) users in a multimedia service provider. Each session \( s_i \) has \( l_i \) QoS levels, e.g., Gold, Silver, Bronze, etc, mathematically denoted as \( \vec{q}_{i1}, \vec{q}_{i2}, \vec{q}_{i3}, \ldots, \vec{q}_{in} \), together with the corresponding revenue offered by the user. Each QoS level \( \vec{q}_y \) requires \( m \) resources from the Enterprise network, which can be denoted by the vector \( r(\vec{q}_y) = (r_{y1}, r_{y2}, \ldots, r_{ym}) \).

Each session \( s_i \) is admitted at one of the offered QoS levels \( \vec{q}_y \). (QoS level \( \vec{q}_{00} \), with zero resource and zero revenue, is by convention used to represent rejection of the SLA.) The chosen QoS level then implies the session utility \( u_{iy} \), and the session resource usage, \( r(\vec{q}_y) = (r_{y1}, r_{y2}, \ldots, r_{ym}) \). The objective of admission control of sessions is to maximize the revenue earned from the admitted sessions, subject to honoring all QoS guarantees to these admitted sessions. This implies observing the resource constraints of the system. Mathematically we can say,

**Objective:**

\[
U = \max \sum_{i=1}^{n} \sum_{j=1}^{l_i} x_{ij} u_{iy}
\]

Where \( x_{iy} \in \{0,1\} \) is the selection variable.

**Constraints:**

\[
\sum_{j=1}^{l_i} x_{ij} = 1 \text{ (i.e., at most one QoS level is selected from a session including null QoS)}
\]

and \( \sum_{i=1}^{n} \sum_{j=1}^{l_i} x_{ij} r_{jk} \leq R_k \) (i.e., the sessions must consume less resources than the total capacity)

Where, \( R_k \) is the total amount of the \( kth \) resource in the system.
Figure 2.2 shows the relations between the system and session utilities (revenues), and between resource mappings and constraints, all established via the choice of session QoS.

The Utility Model ensures the optimal revenue (utility) for the multimedia service provider, as well as the satisfaction of users. It gives us a strategy to select a revenue-optimal set of sessions at QoS levels to admit. The multimedia servers and the underlying network together provide this agreed, well-defined level of QoS for each session, in order to respect the commitments made in all SLAs and thus keep users satisfied.

Now, the Utility Model can be mapped to a variant of the classical Knapsack Problem called MMKP (Multiple Choice Multi-Constraint Knapsack Problem). Although some algorithm like the Branch and Bound Linear Programming (BBLP) method finds the optimal solution of the MMKP, but NP-hardness and thus impracticality of BBLP is the obstacle for its use in real time decision-making. So, different heuristic solutions are proposed to solve the MMKP.
2.5 Finding the Shortest Paths

A shortest path between two nodes in a graph or network is a directed simple path from the source to the destination with the property that no other such path has a lower weight. The weight may be distance, cost or network delay, etc. The applications of shortest path computations are too numerous to cite in detail. They include situations in which an actual path is the desired output, for example, network connection routing and many optimization problems solved by dynamic programming.

Numerous works have been done to find shortest paths between nodes of a graph. Dijkstra [6] solves the single source shortest problem if all edge weights are greater than or equal to zero. Without worsening the run time, this algorithm can in fact compute the shortest paths from a given start point to all other nodes. The Bellman-Ford algorithm which is based on separate algorithms by Bellman [7] and Ford and Fulkerson [8] can handle negative weight edge but with longer running time than Dijkstra’s algorithm [6]. These algorithms find shortest paths from a single source to all other nodes. Although we can use them to find shortest paths for every pair of vertices, but better algorithm exists. For example, the Floyd-Warshall algorithm is used to find all pairs of shortest paths for general graphs. This algorithm is due to Floyd [9] who based it on a theorem of Warshall [10]. Johnson’s algorithm [11] is used for sparse graph.

Figure 2.3: Multiple routes between two nodes in a small network
All algorithms cited above find only a single path between two nodes. But sometimes we need alternate paths between two nodes. For example, in Figure 2.3, where circles represent nodes, lines represent edges and numbers in edges represent distances between nodes, there are multiple paths between nodes S₁ and S₄. If we consider three shortest paths between them, they will be S₁ → S₂ → S₄, S₁ → S₆ → S₅ → S₄ and S₁ → S₅ → S₃ → S₄.

Considerable works have been done on finding \( k \) shortest paths between two nodes of a graph or network. Eppstein [12] presents algorithm for finding the \( k \) shortest paths connecting a pair of vertices in a digraph. His algorithm calculates these shortest paths in a digraph with \( n \) vertices and \( m \) edges, in \( O(m + n \log n + k) \) time and also finds the \( k \) shortest paths from a given source \( s \) to each vertex in the graph, in total \( O(m + n \log n + kn) \) time.

Graph algorithms including Eppstein’s algorithm can handle different types of weights on the edges of a graph. The example in Figure 2.3 uses distance between nodes as weights. It may be cost or delay to convey goods or to transmit data, or it may be amount traffic congestion in the path or any other measurable properties that a typical path may have. For example in our simulations, we need to find multiple shortest paths from a source to a destination based on delays in different paths. So, in our simulation the shortest path means the fastest path.

### 2.6 Knapsack Problem and its variants

Knapsack problem and its variants are widely used in formulating different practical problems such as menu planning, resource scheduling, admission control and profit maximization. So, researchers rendered their effort to develop various algorithms for solving those problems.

The classical 0-1 Knapsack Problem (KP) is to pick up items for a knapsack for maximum total value, so that the total resource required does not exceed the resource constraint \( R \) of the knapsack. Let there be \( n \) items with values \( v_1, v_2, \ldots, v_n \) and let the corresponding resources required to pick the items be \( r_1, r_2, \ldots, r_n \) respectively. The
items can represent services and their associated values can be values of revenue earned from that service. In mathematical notation, the 0-1 Knapsack Problem is to find $V = \sum_{i=1}^{n} x_i v_i$, subject to the constraint $\sum_{i=1}^{n} x_i r_i \leq R$ and $x_i \in \{0,1\}$. Knapsack Problem is an NP-Hard problem [13]. There is a pseudo polynomial algorithm with $O(nR)$ computational complexity by using the concept of dynamic programming [14].

There are some variants of the classical knapsack problem. The Multidimensional Knapsack Problem (MDKP) is one kind of KP where the resources are multidimensional, i.e. there are multiple resource constraints for the knapsack. The Multiple Choice Knapsack Problem (MCKP) is another KP where the picking criteria for items are restricted. In this variant of KP there are one or more groups of items. Exactly one item will be picked from each group. The Multidimensional Multiple-choice Knapsack Problem (MMKP) is actually a combination of the MDKP and the MCKP. Let there be $n$ groups of items. Group $i$ has $l_i$ items. Each item of the group has a particular value and it requires $m$ resources. The objective of the MMKP is to pick exactly one item from each group for maximum total value of the collected items, subject to $m$ resource constraints of the knapsack. In mathematical notation, let $v_{ij}$ be the value of the $j$th item of the $i$th group, $\vec{r}_{ij} = (r_{ij1}, r_{ij2}, \ldots, r_{ijm})$ be the required resource vector for the $j$th item of the $i$th group and $\vec{R} = (R_1, R_2, \ldots, R_m)$ be the resource bound of the knapsack. Now, the problem is to find

$$V = \text{maximize} \sum_{i=1}^{n} \sum_{j=1}^{l_i} x_{ij} v_{ij} \quad \text{(objective function)}$$

with constraints, $\sum_{i=1}^{n} \sum_{j=1}^{l_i} x_{ij} r_{ijk} \leq R_k$, $x_{ij} \in \{0,1\}$ and $\sum_{j=1}^{l_i} x_{ij} = 1$.

There are two methods of finding solutions for an MMKP: one is the method for finding exact solutions and the other is using heuristics. Finding exact solutions to the MMKP is an NP hard problem [15]. Khan et al [2] presented an exact algorithm for the MMKP using the Branch and Bound Linear Programming (BBLP) technique. This method of finding exact solution to MMKP is not applicable to real-time admission control problem and non real-time large problem sets due to its exponential time requirement.
Since method of finding exact solution to the MMKP is not applicable to the real-time admission control problem, heuristic solutions have been developed. HEU, a heuristic developed by Khan et al [15], finds the solution of the MMKP using the concept of aggregate resource consumption. Later Akbar et al [16] presented a modification of HEU, I-HEU, which achieves a better of optimality than HEU. C-HEU, another heuristic developed by Akbar, provides solution to the MMKP in logarithmic worst-case time complexity [17]. But the optimality achieved by this heuristic is much inferior to the other two heuristics.

2.7 Literature Review of Admission Control Algorithms

As the current Internet is based on a best-effort datagram service model: this model does not require (and generally does not permit) resource reservation prior to data transmission. So the service quality of today’s Internet is quite unpredictable and not reliable enough to provide satisfactory services for the emerging applications, such as real-time audio and video that require strict Quality of Service (QoS) parameters. When a packet arrives at a router, and sufficient resources (such as time and buffer-space on the outgoing link) are available, the packet is forwarded to the next router. However, if the necessary resources are not available, the incoming packet may be delayed, or even dropped. It is therefore difficult to predict, let alone guarantee, the bandwidth or latency experienced by a stream of packets under best-effort datagram services. And since each packet of a session is forwarded through the network independently, packets may experience variable and unpredictable delays, and may arrive at the destination out of order. This service has many advantages, but it is unworkable for real-time multimedia applications requiring absolute standards of performance such as continuous bandwidth during a Video on Demand session with maximum delay and jitter constraints. Hence a best-effort datagram service model is not considered suitable for the Internet2 [18], which has been proposed to offer the end-to-end quality-of-service guarantees similar to that of the telephone network.

Blake et al [19] introduces architecture that uses class based forwarding proposals. In this architecture packets requiring guaranteed QoS are assigned higher priority classes than the best-effort traffic. Higher priority classes provide superior service to the
QoS-sensitive application relative to the best-effort traffic. But they cannot guarantee absolute standards of QoS to applications, including telephony and interactive video communication, which require such standards.

RSVP [20] is a protocol used to reserve resources i.e., link bandwidth over the Internet. It requires reservation in each switch from the source to the destination, which clearly requires the determination of a fixed path on which all datagram of the flow are carried. This protocol works for real-time audio and video transmission, and in that sense could provide a basis for guaranteed QoS, but scalability is a problem.

MPLS [21] provides a mechanism for sending data independent of the IP routing tables in the routers. In this mechanism each packet is routed through a predefined path, which is determined before data transmission. A label is added to the packet, and this label is used for table look up in the router for forwarding packets to the next router with another label. The label-forwarding table is created at the time of fixing the path and it contains additional information such as Class-of-Service (CoS) values that can be used to prioritize packet forwarding. As MPLS is a lower layer protocol than IP and UDP, real-time multimedia transmission using IP or UDP over MPLS is considered plausible.

Ali [22] proposed a new flow-based admission control algorithm for routing multimedia traffics with a predefined QoS through a DiffServ-Aware ATM based MPLS network. The routers in the networks work as distributed admission controllers to find a particular path satisfying a particular QoS. A policy of rerouting some of the flows during the congestion in a particular part of the network is also presented. Admission Controlling is done by the router connected to the source of the multimedia traffic by observing the current delay and jitter of the calculated path for the multimedia request. Virtual Circuits (VC) in an ATM-based MPLS network are used to reduce end-to-end delay and router load and to increase throughput.

Schreier and Davis [23] proposed the Benefit Model for adaptation of quality attributes of a single user multimedia application. The quality of the service is expressed by the video frame rate, audio/video quality and audio/video
synchronization. Each of these quality parameters has an associated benefit function. The objective is to maximize the benefit of the service by adjusting the quality parameters. Moser [24] presented an Optimally Graceful QoS Degradation Model (OGQD) where a single session’s quality is gracefully degraded to meet resource constraints. For a multimedia session the system calculates the set of services for maximum utility subject to resource constraints using a heuristic for solving the MMKP [25].

However, the Benefit Model and OGQD discuss the adaptation of QoS for a single multimedia session. They do not address the problem of adaptation in a multi session environment with a predefined objective like revenue or utility maximization. Venkatasubramanian and Nahrstedt [26] proposed an economic framework for a multi-user multimedia service provider with different objectives for the users and for the service provider. In this framework a user’s objective is to maximize QoS with respect to paid price but the service provider’s objective is to maximize revenue with respect to resource usage of the system. This principle does not ensure the maximum utilization of resources of the service provider.

Ahmed [27] designed a single server based architecture and an admission control algorithm to accept enough traffic to efficiently utilize server resources, while not accepting clients whose admission may lead to the violations of the service requirements of clients. In his work, he proposed a new admission control algorithm that can handle a good number of clients simultaneously. One interesting feature of his technique is different admission policies for different clients based on their service requirements.

Elnikety et al [28] presents an admission control algorithm for e-commerce based websites to achieve both stable behavior during overload and improved response times. This algorithm distinguishes different types of requests based on execution costs. It performs overload protection and schedules the requests using relatively simple measurements and a straightforward control mechanism.
Admission control algorithm is required not only for requesting media stream from multimedia servers, but sometimes multiple users may try to store or record media data to the servers. Zimmermann and Fu [29] presents a statistical admission control algorithm based on a comprehensive random variable model to support both reading and writing of multiple variable bit rate media streams on multimedia servers.

Cheng et al [30] proposed a quota-based admission control algorithm for multimedia servers where server capacity is divided into three partitions based on the quota values. Two partitions are for each of two different priority classes of requests and one common pool shared by these two classes of requests. High-priority requests are associated with higher values of reward as well as penalty than low-priority ones. Given the characteristics of the system workload, this algorithm finds the best partitions, optimizing the system performance based on the objective function of the total reward minus the total penalty. A similar admission control algorithm was proposed by Chen and Chen [31] based on the idea that admission control can be driven not only by hardware requirements of the servers, but also by knowledge regarding the workload characteristics of client requests.

Yum et al [32] developed admission and congestion control algorithms for cluster servers in a wormhole-switched network that uses QoS-capable wormhole routers and QoS-capable network interface cards (NICs), referred to as Host Channel Adapters (HCAs). The admission control is applied at the HCAs and the routers, while the congestion control is deployed only at the HCAs.

2.8 Literature Review of Admission Control Architectures

DiffServ [19] is one of the prominent architectures for support QoS in the Internet. The architecture allows the network owner or service provider to offer different kinds of services to different users. The key point in this architecture is that routers are capable to forward packets of different traffic flow in different per hop behaviors, which are indicated by DiffServ code point in IP headers of respective packets.
Cetinkaya et al [33] developed a scalable architecture and an endpoint admission control algorithm for quality-of-service management by resource management and admission control only at egress routers, without any coordination among backbone nodes or per-flow management. By monitoring and controlling egress routers’ class-based arrival and service envelopes, the algorithm shows how network services can be provisioned via scalable control at the network edge.

Bhatnagar and Nath [34] propose a distributed admission control architecture to support core-stateless guaranteed services. This architecture maintains high network utilization while ensuring that resources are not over allocated. Admission control is performed at the ingress edge routers of a request on an edge-to-edge path basis. A token-passing mechanism is used as the resource management framework. The edge routers co-operate to provide fault tolerance effectively. This admission control framework can support statistical guarantees and DiffServ architecture’s premium service as well. The resource management part of the architecture is well-suited to aid QoS routing algorithms.

Bouras and Stamos [35] proposes an architecture called Bandwidth Broker that is responsible for managing the bandwidth within a network domain and for the communication with Bandwidth Brokers of neighboring domains. The admission control module of this architecture aims at achieving a satisfactory balance between maximizing the resource utilization for the network provider and minimizing the overhead of the module. To achieve this, the architecture gathers and examines sets of book-ahead requests and adapts the size of the set so that the network utilization and the computation overhead are appropriately balanced.

Bhatnagar et al [36] propose an architecture to implement distributed admission control for multicast flows with heterogeneous user requirements. The architecture does not require core routers to perform any admission control. The framework guarantees that a request is only admitted if there is sufficient bandwidth available and only requires the edge routers of a domain to take admission decisions. An intra-domain signaling mechanism is used in conjunction with the admission control framework to install the forwarding state inside the network core.
2.9 Chapter Summary

In this chapter we have described the necessary key components for designing an admission controller. Recent works on Knapsack Problem algorithms have also been described. Applications of the Utility Model for admission control in multimedia server system connected through an Enterprise Networks have been introduced. The chapter concludes with the detailed review of different types of existing admission control algorithms and architectures used in different systems. The next chapter describes details of the architecture of a Multimedia Server System and a new algorithm for admission control of multimedia sessions in this architecture.
Chapter Three

Admission Control Methodology

3.1 Introduction

In this chapter we will describe details of our proposed architecture of a Multimedia Server System along with its various aspects. A new admission control algorithm for the architecture with its complexity will also be discussed.

3.2 Proposed Architecture for a set of Multimedia Servers of an Enterprise Network

In this section, we will describe our proposed architecture for a set of multimedia servers connected through an Enterprise Network. The work flow of the admission controller, necessary protocols for data transmission and a practical feasibility of the architecture will also be described.

3.2.1 Motivation for a New Architecture

Every admission control algorithm is applied to an architecture. The choice of architectures depends on many things like the algorithm used, network size, fault tolerance etc. Some architectures [17] use multiple admission controllers which communicate with each other to do admission control. This type of architecture is suitable for multiple connected enterprise networks or the Internet where admission control from a single machine is clearly a bottle neck. But complex algorithm is necessary for admission control and thereby more admission time is required because an individual admission controller cannot admit users alone without consulting others. So a single admission controller is suitable and sufficient for a system with a single enterprise network.

Other architectures like [33], admission control is done on egress routers without using any separate admission controller. In this architecture, a user's request for
reservation on a path is sent to an egress router located on that path. The router then
continuously monitors available service on the path to admit a new flow for the user.
So continuous monitoring is a bottle neck. Moreover the egress router must be able to
predict how much resource of some links on the path is used by other flows. So the
performance of the architecture depends on a good prediction of services used by
other flows. Besides, handling and monitoring multiple source servers is another
problem for the architecture because routers can monitors the load on paths only. So
the admission control decision based only on path monitoring will not reflect the
proper server utilization.

From the above discussion we can say that a new architecture with a single admission
controller is required for a small scale system where users will be completely unaware
of media sources. The admission control decision must be based on resources
available both on the servers and the paths. The next subsection gives detail
description of the new architecture.

3.2.2 The Architecture

As we mentioned previously, our proposed system is for an Enterprise Network not
for the Internet. Figure 3.1 shows the proposed architecture of the Multimedia Server
System to support real time multimedia services with absolute QoS guarantees. The
system contains following components.

- **The Enterprise Network**: The network is a privately owned small Enterprise
  Network. It contains network switches as nodes and links between them. In
  Figure 3.1, S_1 through S_6 are switches. Connections between them are shown
  by different thin lines.

- **Workstations**: Workstations denote users. They are personal computers with
  sufficient resources to play multimedia stream and connected to the network.

- **Media Servers**: Media servers are connected to the network. The servers
  contain multimedia data like movies with possible multiple replication of them
  but no replication of the same data on the same server.
- **Admission Controller**: The admission controller is a high performance machine and connected to a switch. The admission controller contains a centralized database that contains all information of the system, i.e., available memory, CPU cycles and I/O bandwidth of different servers and bandwidth of connecting links. Multimedia data information, that is, which server contains which data is also stored in the database. Note that the centralized database does not contain actual multimedia data; actually it contains information of media stream.

![Figure 3.1: Architecture of the Multimedia Server System](image)

3.2.3 Work Flow of the Admission Controller

In our proposed architecture different control and multimedia data transmission occurs in different stages. Different components of the system communicate with each other to do admission control of the submitted requests. The work flows of data transmissions between different components are described as follows:
1. The servers advertise about their available multimedia data and other resources to the admission controller. Thick squared dotted lines in Figure 3.1, for example, \(S_5 \rightarrow S_5\), \(S_6 \rightarrow S_5\) and \(S_3 \rightarrow S_4 \rightarrow S_5\) show advertisements of the media servers to the admission controller.

2. The users will put their multimedia session requests in terms of SLAs to the admission controller. These SLAs are called proposed SLAs. Thin dotted lines in Fig 3.1, for example, \(S_5 \rightarrow S_1 \rightarrow S_5\) and \(S_4 \rightarrow S_5\), show submissions of SLAs to the admission controller by the users in workstation positions.

3. An SLA contains different QoS levels, offered revenue and resource requirement of each QoS level, media stream to be enjoyed, and other quality parameters. The admission controller determines whether the SLAs will be admitted or rejected satisfying the constraints about the available resources and requested resources by the users according to its admission control principle described in the next section. The accepted SLAs by the admission controller are called active SLAs and eventually respective users will be notified. Thin dotted lines in Figure 3.1, \(S_5 \rightarrow S_1 \rightarrow S_2\) and \(S_5 \rightarrow S_4\) represent notifications to the users.

4. There may be multiple sources for a particular data requested by the user and multiple paths may exist from any of them to the user. For example, Fig 3.1 shows two sources of Movie1. These are Server1 and Server2. The figure shows multiple paths from Server1 to Workstation2 shown by thick broken lines, \(S_1 \rightarrow S_5 \rightarrow S_4\) and \(S_1 \rightarrow S_6 \rightarrow S_4\). The figure also shows two paths from Server2 to the same workstation shown by thick lines, \(S_3 \rightarrow S_4\) and \(S_3 \rightarrow S_6 \rightarrow S_4\). An active SLA contains accepted QoS level, offered revenue and resource requirement of that QoS level, optimal source of the media stream, and optimal routing path from the source server to the user. The thick circled dotted lines, \(S_1 \rightarrow S_2\) and \(S_3 \rightarrow S_4\), in Fig 3.1 show corresponding routes and thereby respective serves of accepted SLAs for users requesting from Workstation1 and Workstation3, respectively. The underlying network and
servers reserve resources for the admitted SLAs. Then the actual data transmission begins.

3.2.4 A Practical Description of the Architecture

In a previous section we described the proposed architecture and its various components. This section describes how it will be implemented practically. Following points present some practical details of different components.

- **Connectivity**: The architecture has connectivity in different stages. The nodes or switches are interconnected through fiber optics backbone. Links with Optical Carrier level 1 (OC1), can be used when the number of users in the system is small. The connectivity can be updated later with links of higher level carrier when the number of servers and users increases in the system. Gigabit Ethernet connectivity like 1000baseF fiber optics or 1000baseT copper wires can be used to connect the servers to the switches. The workstations are not directly connected to the switches. They might be connected through an Ethernet Local Area Network (LAN) or an Asymmetric Digital Subscriber Line (ADSL).

- **Nodes**: Nodes in the network are actually high speed switches. These switches have appropriate number of fiber based or copper based Gigabit ports to be connected with servers and with other switches. They might have necessary ports to connect workstations.

- **Servers**: The servers are special Network Attached Storage (NAS) devices, special servers that are dedicated for file sharing. These machines are connected to the switches directly through Gigabit network cards or indirectly through Gigabit LANs.

- **Workstations**: The workstations are standalone personal computers connected to the network through LAN or ADSL. They have sufficient resources like memory, disk space, graphics card, and processing power to play multimedia stream.
• **Admission Controller:** The admission controller is a high performance computing server. It is directly connected to the switch or indirectly through a high speed Gigabit Ethernet LAN. A NAS can be used for the centralized database attached to it or we can use a cluster solution replacing both the high speed server and the NAS, as cluster solution has the capability to act as both computing server and storage server. Whether we use cluster solution or separate computing and storage server, they require fiber or copper based Gigabit Ethernet network card to be connected to the network.

Besides the components described above, another practical issue related to the connectivity and the switches is delay. These are the two sources that may cause delay in the proposed architecture. As we reserve resources both in the servers and in the network we assume that the switches will be able to transmit data from their incoming ports to the corresponding outgoing ports without storing the packets in the internal buffers for a long time. So there will be no congestion and thereby no queuing delays in the switches. They will contribute only switching delay which is assumed constant for all switches. The effective delay comes actually from the links of the network. They contribute the propagation delay which is the time required to transmit one bit of data from one end of a link to the other end. This propagation delay is assumed same for all links in the network if the same technology is used for all links of the network. The delay of a path is calculated by summing up all switching delays and propagation delays of respective switches and links that comprise the path. So the delay of a path is proportional to the number of links in the path. Because resources for an admitted SLA are reserved and all packets of a particular admitted SLA will follow same path, so packets will arrive to the destination with same regular interval, i.e., without any variable delay. So we can say that our system will be jitter free.

### 3.2.5 Protocols Used in Different Stages of the Admission Controller

Although we mentioned previously that our work will not concentrate on network protocol, but we need to say something about the protocols to be used in different stages of the system while transmitting session requests, notification and multimedia data.
We know that TCP is a guaranteed protocol and it needs retransmission. So, we can use it to transmit control data. When different servers advertise about their available resources and media data and users make requests to the admission controller, both require guaranteed transmission. Notification from the admission controller to the users also requires guaranteed transmission. Data transmission for these purposes is non-multimedia type. So, we can use TCP in these cases. Resource requirement for these transmissions is considered negligible compared to the actual media data transmission.

On the other hand, by the nature of multimedia data transmission, it requires ordered and timely arrival of packets with no retransmission. It is possible only when a particular path and a server will be used to transmit multimedia stream for an admitted SLA. As the admission controller selects SLAs satisfying resource constraint of the system, so it is possible to fix up a server and a delivery route for an SLA. We assume that an end to end virtual path is created in the packet switching network using Multi Protocol Level Switching (MPLS). The admission controller sets the MPLS labels in the switches of the Enterprise Network for the path with a particular Class of Service (CoS). The CoS in the MPLS table represents the QoS levels of the SLAs. Then a multimedia session starts transmitting the multimedia stream using UDP over MPLS. As UDP does not create retransmissions, so it will be best protocol in this case. Moreover as we fix a path and a server with required resources, we can think that packets will arrive to the destination timely and orderly.

3.3 Admission Control Methodology

In this section, we will give detail description of the admission control algorithm. The complexity of the algorithm will be analyzed in the next section.

3.3.1 The Algorithm

Fig 3.2 shows the pseudo code of the algorithm used by the admission controller to admit SLAs in the system. Next subsection presents detailed description of different steps of the algorithm. Necessary data structures required for the algorithm are as follows:
QoS_list[ ]: The list of QoS levels for an SLA.
QoS_path_list[ ]: The list of candidate paths satisfying different QoS levels of an SLA. Each element of the list is represented by the (candidate path, QoS_level) tuple.
active_sla_list[ ]: The list of SLAs that are already admitted.
new_batched_sla_list[ ]: The SLAs batched in the last epoch that are queued for admission.

I_HEU (sla_list, option) : Apply I-HEU to determine the current path and QoS level of the SLAs in the sla_list. The parameter option indicates whether heuristic is applied with RESPECTING_RESTRICTIONS or IGNORING_RESTRICTIONS as specified by the flags of an SLA. If IGNORING_RESTRICTIONS is set to option then servers, paths or QoS levels of previously admitted SLAs may be changed whenever the change is necessary to maximize the earned revenue. On the other hand if RESPECTING_RESTRICTIONS is set, respective flags, which are set in the SLAs, will be checked whether they allow any change.

accepted_slas( sla_list ) : This function returns the SLAs which have non null current QoS levels.
rejected_slas ( sla_list ) : This function returns the SLAs which have null current QoS levels.

3.3.2 Detailed Description of the Algorithm
Admission control and QoS adaptation is done on batches of SLAs. The heuristic for solving the MMKP is applied to a batch of sessions once in a regular time interval which is called an epoch. In each epoch some of the old sessions leave and some new session requests are batched for admission. All SLAs, whether new or old, are batched (Step 1).

In Step 2 we actually map SLAs to an MMKP, a variant of classical knapsack problem, described previously. Users mention their requested multimedia data; they do not need to mention the source of the data. In Step 2.1, we find the sources of the data from the centralized database contained in the admission controller.

There might be more than one path from source to destination in a network. Because of the delay constraint, some of the paths will be considered infeasible to carry the
multimedia traffic of an SLA. In Step 2.2, we apply a $k$ shortest paths discovery algorithm. We use Eppstein's $k$ shortest paths algorithm [12] to find first $k$ shortest paths from all sources of the data to the destination. The paths whose latencies fit the requirements for the lowest nontrivial QoS level of the flow are the candidate paths for routing the multimedia session. The network topology will also be stored in the database and the admission controller makes query for information about the network.

```
Procedure admission()
Begin
  hatched_sla_list ← new_batched_sla_list // new SLAs are batched
  for i←1 to size(hatched_sla_list[ ]) do // for each SLA
    //determine sources
    sources← hatched_sla_list[i].get_sources(hatched_sla_list[i].data)
    path_list ← null //determine paths for all pair of
    for s←1 to size(sources) do // source and destination
      path_list ← path_list + determine_K_candidate_paths(sources[s],
                                                  hatched_sla_list[i].destination)
    endfor
    //for each pair of satisfied path and QoS Level item of an MMKP is generated
    for j←1 to size(hatched_sla_list[i].QoS_list) do
      for k←1 to size(path_list[ ]) do
        if delay(path_list[k]) < delay(hatched_sla_list[i].QoS_list[j]) then
          add (path_list[k], bid (hatched_sla_list[i].QoS_list[j])) in
          hatched_sla_list[i].QoS_path_list
        endif
      endfor
    endfor
    // A null QoS level with null resource usage is added
    add (null, null) in the hatched_sla_list[i].QoS_path_list
  endfor
  temp_sla_list ← active_sla_list + hatched_sla_list //heuristics solves the MMKP
  I-HEU (temp_sla_list, IGNORING_RESTRICTIONS)
  active_sla_list ← active_sla_list + accepted_slas(temp_sla_list)
  rejected_sla_list ← rejected_slas(temp_sla_list)
endProcedure
```

Figure 3.2: The admission control algorithm
Each SLA is actually a group of MMKP; and Step 2.3 creates items for the group from every pair of paths and QoS levels. Each path is checked for delay constraint and each pair of QoS levels and satisfied paths represents the items of MMKP and they are inserted into an array named $QoS_{path\_list}$.

For each new SLA, we add a null QoS level to the QoS profile (Step 2.4). This is actually a dummy QoS level with no offered price and no resource utilization. If the final result of the MMKP assigns the null QoS level to an SLA, that SLA is rejected, i.e., admission at QoS level 0 is equivalent to rejection. The null QoS level indicates whether the SLA is active or inactive. An SLA becomes active when it is given a non-null QoS level.

Finally in Step 3, we apply a heuristic algorithm I-HEU [16], to solve the MMKP which actually admits some new SLAs from current batch. As we ignore the restrictions applied as specified by the flags in the SLAs, the solution of MMKP may change QoS levels, paths or servers for some previously accepted SLAs to find optimal revenue. Underlying network and servers use information about these admitted SLAs to reserve resources for transmission of multimedia data for respective users. The new accepted QoS levels can be interpreted as follows:

- If a new SLA request has a non-null QoS level then the user starts enjoying multimedia service. Thus the user gets admission with that QoS level.
- If the QoS level of a new SLA remains null then the request is rejected.
- The QoS level of an existing SLA may be upgraded or downgraded to a higher or lower QoS level or its path or server may be changed.

3.3.3 An Example Explaining the Algorithm

To explain the algorithm we use information from our proposed architecture. Figure 3.3 shows different paths from different servers to users. These paths are found from $k$ shortest path algorithm. Let the algorithm derives the paths as follows:

Path$_1$: $S_1 \rightarrow S_2$
Path$_2$: $S_1 \rightarrow S_6 \rightarrow S_3 \rightarrow S_2$
Path 3: $S_1 \rightarrow S_6 \rightarrow S_4$
Path 4: $S_1 \rightarrow S_5 \rightarrow S_4$
Path 5: $S_3 \rightarrow S_4$
Path 6: $S_3 \rightarrow S_6 \rightarrow S_4$
Path 7: $S_6 \rightarrow S_3 \rightarrow S_2$
Path 8: $S_6 \rightarrow S_1 \rightarrow S_2$

Suppose that for delay constraint Path 2, Path 3, Path 6 and Path 8 are not considered. Let, QoS levels of SLA 1 be QoS 11 and QoS 12 and QoS levels of SLA 2 be QoS 21 and QoS 22.

Now entries, in Figure 3.4, are items of an MMKP and inserted into the $i$th QoS_path_list which is a group of MMKP, where $i$ is 1 or 2.

A centralized database contains information about available resources in the paths and servers. Information related to the offered revenue and requirement of server resources and link bandwidth is specified in different QoS levels of SLAs. From these information the heuristic solution pick an item from each group so that earned revenue is maximized and QoS is maintained for each admitted SLA.
3.4 Complexity Analysis

Suppose,

The number of SLA submitted to the admission controller = \( n \)

Each SLA contains \( I \) QoS levels.

The number of nodes in the Enterprise Network = \( N \)

The number of edge, \( m = O(N) \)

Each multimedia data is replicated on \( s \) servers

The number of alternate paths from each source to user destination = \( K \)

For SLA admission we have two types of computation. First, we must find the \( K \) shortest paths for a particular SLA, and second, we execute I-HEU for SLA admission control.
If we use Epstein algorithm [12] to find the shortest path in an EN, we need \( O(m + N \log N + KN) \) or \( O(N \log N + KN) \) computation, as \( m \) is \( O(N) \).

But for each SLA media data may come from one of \( s \) servers. So total computation to find all paths is \( O(sN \log N + sKN) \).

After \( K \) shortest path computation, we map the problem to an MMKP with \( O(N) \) resources or volume dimensions, \( n \) piles or groups, and a maximum of \( IsK \) items in each group, where \( I \) is the maximum number of QoS levels in an SLA and \( K \) is the maximum number of alternative paths considered. The worst-case complexity of 1-HEU [16] to find rejection or admission with QoS levels and routing paths to achieve maximum revenue is \( O(Nn^2(IsK - 1)^2) \).

So, the total complexity is \( O(sN \log N + sKN) + O(Nn^2(IsK - 1)^2) \).

Therefore, the response time in SLA admission will increase almost linearly \( (O(N \log N)) \) with the expansion of the network. But it will increase in a quadratic manner with the increase in the number of SLAs \( (n) \), the number of QoS levels \( (I) \), the number of shortest paths \( (K) \) or the number of servers or replication \( (s) \).

### 3.5 Chapter Summary

We have presented our new architecture for an admission controller. The work flow and necessary protocol of different stages have also been described. We have also given detailed description of admission control methodology and its complexity has also been computed. The next chapter describes extensive simulations of the admission controller from different aspects.
Chapter Four

Simulation Results

4.1 Introduction

Allocation of resources, such as CPU cycles and I/O bandwidth of multimedia servers and link bandwidth in the network, is essential to ensure Quality of Service (QoS) of multimedia services delivered over the network. In the previous chapters we have described a new definition of SLA, the architecture and algorithm of the admission controller along with its complexity analysis. In this chapter, we demonstrate and measure the performance of the admission controller varying different parameters by simulating the admission controller.

4.2 Simulation Steps of the Admission Controller

Figure 4.1 shows the main simulation steps of our admission controller. We have coded the simulation of the admission controller using the Java programming language. We ran the simulation on a single processor Pentium-IV PC with 512 MB RAM. After Initialization of simulation parameters, a big batch of SLAs is submitted to the admission controller to make the system congested. SLAs are mapped to an MMKP as described in Chapter 3, and it is solved for the initial batch. Then the main loop of the simulation starts. In each loop new session requests are collected into a batch over a time interval called an epoch along with previously admitted SLAs. Timed-out SLAs of previous iteration are purged. The remaining and new comers are again mapped to an MMKP, and it is solved. Necessary simulation data such as, the revenue earned from the users, the time required by the admission controller, resource usage by users and the number of accepted users are gathered before the new epoch starts.
4.2.1 Imitating the Real Environment

In practice there are different kinds of video players with different qualities. So we provide different QoS levels to imitate this. The network bandwidth and server resource requirements for these video qualities are not the same. So, the cost of providing different QoS levels must be different. Besides this, new movies (first round movies) are generally costlier than old movies (second round movies). Users would likely be willing to pay more to see a first round movie. In the simulation we simulate these variations in the resource requirements and revenue for a given quality level of a movie by using random numbers.

Movies are replicated in the servers of the multimedia server system. It is impractical to replicate a particular movie in all the video servers of the system, because each copy would require a substantial licensing fee. In the simulation, we have arbitrarily
considered at least one replication of each movie in the system and exactly five different movie copies on each video server. Number of replication of movies varies with the variation of the number of servers.

4.3 General Simulation Parameters

The results have been gathered from the extensive set of simulations. Performance of the admission controller is analyzed from six different aspects. This section describes different simulation parameters with their possible values for different sets of experiments. Table 4.1 presents some of these parameters.

Table 4.1: Different Simulation Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values when running simulation for measuring effect of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>system size</td>
</tr>
<tr>
<td>Service duration (Movie length)</td>
<td>3 Hours</td>
</tr>
<tr>
<td>Total no. of movies</td>
<td>10 movies</td>
</tr>
<tr>
<td>Number of servers</td>
<td>varies as 2, 4, 6, 8 and 10</td>
</tr>
<tr>
<td></td>
<td>2, 6, and 10</td>
</tr>
<tr>
<td>Replication of each movies</td>
<td>varies as 1, 2, 3, 4 and 5</td>
</tr>
<tr>
<td></td>
<td>1, 3 and 5</td>
</tr>
<tr>
<td>Number of movies in each server</td>
<td>5 movies</td>
</tr>
<tr>
<td>Epoch (simulation second)</td>
<td>15 sec varies as 15, 30,45,...,120</td>
</tr>
<tr>
<td></td>
<td>15 sec</td>
</tr>
<tr>
<td>Number of shortest paths between source and destination</td>
<td>3 paths varies as 1, 2, 3, 4 and 5 paths</td>
</tr>
<tr>
<td>Percentage of movie replication</td>
<td>100% varies as 10%, 20%, ...100%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>
### Values when running simulation for measuring effect of

<table>
<thead>
<tr>
<th>Parameters</th>
<th>system size</th>
<th>epoch</th>
<th>number of shortest paths</th>
<th>percentage of replica considered</th>
<th>number of servers varying per server resource constant</th>
<th>number of servers keeping per server resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth per link</td>
<td>varies as 100, 200, 300, 400 and 500 units</td>
<td>100, 300 and 500 units</td>
<td>300 units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of nodes in the network</td>
<td>31 nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of edges in the network</td>
<td>62 edges</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph structure of the network</td>
<td>Harary graph [37]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Following points are assumed for the simplicity of the simulation:

1. Each server is connected to a particular node of the network and at most one server is connected to each node.
2. Replication of movies is a function of the number of servers used and can be represented as follows: replication = no_of_servers / 2.
3. A data structure keeps tracks of the movie content of the servers.
4. We think that whenever we want to serve more users, both the link bandwidth and server resources or the number of servers must be increased. So we keep the network link bandwidth proportional to the number of servers used in the system in most of the simulations. In our simulation, we use 50 units of link bandwidth in each link for each server.
5. The time required by the heuristics depends on the problem set. That is why we generate multiple small batches and calculate the average response time from 200 iterations.

The users enjoy three QoS levels defined by three different resolutions of video and image quality. Table 4.2 shows average resource requirements in video server and link bandwidth and offered revenues for different QoS levels. We assume that each QoS level requires three types of server resources namely memory, CPU cycles and
I/O bandwidth. Total resource of each type in a server is considered 100% and so, the server resource requirements are presented in terms of the percentage of the total amount. Although Table 4.2 shows only one column for server resource requirement, it actually represents three types of resources because in simulation three random numbers are used to generate three different requirements. Server resource requirement is kept proportional to the network bandwidth requirement to resemble real systems where I/O bandwidth is proportional to the link bandwidth. Yet, sufficient randomness is present in the system.

Table 4.2: Average QoS requirements and revenues

<table>
<thead>
<tr>
<th>QoS Levels</th>
<th>Average network bandwidth requirement</th>
<th>Average server resource requirement</th>
<th>Average offered revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS1</td>
<td>2.5 units</td>
<td>0.5%</td>
<td>3</td>
</tr>
<tr>
<td>QoS2</td>
<td>5.0 units</td>
<td>1.0%</td>
<td>5</td>
</tr>
<tr>
<td>QoS3</td>
<td>7.5 units</td>
<td>1.5%</td>
<td>6</td>
</tr>
</tbody>
</table>

The resource requirements of media streams for different movies might vary drastically from the average resource requirements listed in Table 4.2. Offered revenue would vary from one user to another. In order to simulate these variations we initialize the resource requirements and revenue using formula as shown in Table 4.3. These initializations resemble the fact that sometimes users offer relatively lower revenue for higher QoS, and vice versa.

Table 4.3: Initialization of different QoS requirements and revenues

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network bandwidth requirement of the kth level</td>
<td>$2.5kU(0.75, 1.25)$</td>
</tr>
<tr>
<td>Server resource requirement of the kth level</td>
<td>$0.5kU(0.75, 1.25)$</td>
</tr>
<tr>
<td>Offered revenue of the kth level</td>
<td>$(2k+1)U(0.75, 1.25)$</td>
</tr>
</tbody>
</table>

$\text{U}(\text{low, high})$ is uniform random number generator between low and high.

The movie enjoyed by the user is $\text{U}(10) + 1$, where $\text{U}(i)$ is a random integer from 0 to $(i - 1)$ and duration of the movie session is $\text{U}(0.00, 10800)$. 

4.3.1 Network Used in the simulation

As mentioned in Table 4.1, we use the Harary graph, \( H_{4,31} \) as our Enterprise Network for simulation purpose, where 4 is the degree of each node and 31 is the total number of nodes in the graph. Although practical networks are not usually regular graph, but for simplicity of some analysis we use \( H_{4,31} \) as our simulation network. We present a simple picture of \( H_{4,9} \) in Figure 4.1. \( H_{4,31} \) can be drawn in a similar fashion.

![Figure 4.2: A sample Harary graph, \( H_{4,9} \)](image)

A Harary graph, \( H_{2r,N} \) is drawn by placing \( N \) circles in order. Then we form \( H_{2r,N} \) by making each node adjacent to the nearest \( r \) nodes in each direction around the circle. The graph has following properties:

- regular graph
- total \( N \) nodes
- \( r \) degree of each node
- total \( nr \) edges [38]

As mentioned previously, practical networks are not regular. But it is natural to connect a switch of a network to its nearby switches to design a network, when long distance between distant nodes creates problems due to maximum segment limit and cost constraints. So consideration of a Harary graph as a network in the simulation is not impractical.
4.3.2 Simulation Events

User arrival: The inter-arrival times of users in service oriented systems such as telephony, post office generally follow the exponential random distribution, i.e., the number of new users requesting for service in each batch will not vary substantially. Here we assume the exponential inter-arrival rate of the movie customers with a mean of $\frac{\text{number admissible users}}{\text{service duration}}$.

Departure of the user: We assume that our proposed system is not like a television system where viewers do not have any option to tune off the movie temporarily and to play from the same position where they leave off. Rather we think our system is like a video player system where viewers can stop viewing movie at any time and can view it later. Thus the chance of giving up multimedia session at any time during the total duration of a movie is equally probable. So we assume that the service time is uniformly distributed throughout the duration. There will be no reservation of the resources after the user leave the system.

4.4 Simulation Results

This section describes our simulation results. We have measured the revenue earned from users, the number of users by the admission controller, resource utilization by the users and time required by the admission controller varying the system size, the number of shortest paths, epoch, and the number of servers. Following subsections describe the effect of each variation.

4.4.1 Effect of the System Size

In this section we describe effects of system size on different performance parameters. The system size is defined by the number of servers in the system and the capacities of links in the Enterprise Network. We set the bandwidth of each link proportional to the number of servers in the system as mentioned in Section 4.3. Findings of the simulation are presented with graphs as follows.

The total available resource is almost proportional to the number of servers in the system. Thus the number of admissible SLAs increases proportionally with the
increase in number of servers, as shown in Figure 4.3 (b). More SLAs contribute more revenue to the system resulting proportional increase in total revenue by the system, as indicated in Figure 4.3 (a).

![Revenue Vs. No. of Servers](image1)

![No. of Accepted SLAs Vs. No. of Servers](image2)

Figure 4.3: Effect of system size on revenue and the number of accepted SLAs

![AverageTime Vs. No. of Servers](image3)

Figure 4.4: Effect of system size on average admission time

The complexity of the admission controller is approximately $O(Nn^2l^2s^2K^2)$, where $K$ is the number of shortest paths, $l$ is the number of different QoSs and others are
defined previously. As $N$, $K$, and $I$ are fixed, so the time is dependant on $n$ and $s$. Moreover, $n$ is linearly proportional to $s$, so the time is a fourth degree polynomial of $s$, which is reflected in the graph of Figure 4.4.

Figure 4.5: Effect of system size on the average utilization

Figure 4.5 shows the effect of system size on the average resource utilization. In this simulation, $utilization = 100 \times \frac{total\_used\_resource}{total\_available\_resource}$. As we consider different types of resources in the system, so utilization of each resource is calculated first. Then these individual utilizations are averaged to find the overall average utilization.

In a system with a few servers, the links near the servers become exhausted although most of the links remain underutilized. So the number of accepted SLAs is low in this case and average utilization may be relatively lower. That is why Figure 4.5 shows the lowest average utilization when the number of servers is 2.

When the number of servers increases in a system, they are sparsed in different parts of the network and links are utilized more efficiently, which increases the chance of getting admission of an SLA. For this reason, Figure 4.5 shows some increase in average utilization when the number of servers is 4.
There are some SLAs where the destination workstation and the movie server are connected to the same node of the network. We call these SLAs as SLAs without link bandwidth. An admitted SLA must consume server resource but it may not consume link resource if it is an SLA without link bandwidth. There will be more SLAs like this if we have more movie servers in the system because admission of these SLAs is not affected by the heavy utilization of nearby links of their respective servers. They consume only server resources but no link resources. This results a decreased overall utilization although the number of accepted SLAs may be high. That is why Figure 4.5 shows fall in the average utilization when the number of servers is increased from 6 to 10.

4.4.2 Effect of Epoch

This section describes the effect of epoch on different performance measures. Epoch is varied from 15 seconds to 120 seconds. In each epoch, new SLAs come with an exponential rate as mentioned previously. New SLAs and previously admitted SLAs are considered for admission in the subsequent iteration. Results are gathered from simulation with three different system sizes. The other parameters have values as given in Table 4.1. Following figures present our results in graphical format.

![No. of Accepted SLAs Vs. Epoch](a)

![Revenue Vs. Epoch](b)

Figure 4.6: Effect of epoch on revenue and the number of SLAs
Figure 4.6 shows the effect of epoch on the number of accepted SLAs and revenue earned by the network owner from the users. The admission controller gets longer batch of new SLAs when the epoch becomes larger. Thus the admission controller would get more alternatives to maximize the total earned revenue by selecting more or less SLAs. On the other hand the space for leaving SLAs will be replenished very quickly if the epoch becomes smaller. Thus there is a chance of loosing revenue for the system with longer epoch.

```
Figure 4.6: Effect of epoch on number of accepted SLAs and revenue earned by the network owner from the users
```

Figure 4.7 shows the effect of epoch on average admission time. From our previous complexity knowledge of the admission controller, we find that as more SLAs are present in the system, it will take longer time to do admission control although the number of admissible SLAs does not change significantly. Increase in epoch means more incoming SLAs, which means longer admission time although all of them will not get the chance to be admitted. Figure 4.7 clearly justifies our views.

```
Figure 4.7: Effect of epoch on average admission time
```

The graphs in Figure 4.8 show the effects of epoch on the average utilization. As mentioned previously SLAs come with an exponential arrival rate. As epoch
increases, more SLAs come within a period. But resource constraints do not allow them all to be admitted and the admission controller gets more chance to admit SLAs so that resources are effectively used. This increases resource utilization, which the graphs in Figure 4.8 justify.

![Avg. Utilization Vs. Epoch](image)

Figure 4.8: Effect of epoch on average utilization

Figure 4.8 shows also that the lowest average utilization is found when there are 2 servers in the system and average utilization increases with the increase in the number of servers but with an exception. The maximum average utilization is observed when there are 6 servers in the system because sparse distribution of servers in the network results effective utilization of all links. But further increase in the number of servers, for example, 10 servers in the system, does not increase the utilization because more SLAs without link bandwidth get chances to be admitted. The admission of these SLAs decreases the average utilization in this case.

### 4.4.3 Effect of the Number of Shortest Paths

In this section we describe the effect of the number of shortest paths on the performance of admission controller. In a network there might be several alternate paths from a source to a destination. In this simulation the number of shortest paths
from each movie server of a movie to the workstation is varied from 1 to 5. Shortest paths are found by applying Eppstein's $k$ shortest path algorithm based on data transmission delay in paths. Finally we gather different results from the simulations and demonstrate our findings in the following figures.

![Graphs showing the effect of the number of shortest paths on the number of accepted SLAs and revenue.](image)

Figure 4.9: Effect of the number of shortest paths on the number of accepted SLAs and revenue

The graphs in Figure 4.9 show the effect of the number of shortest paths on the number of accepted SLAs and on the earned revenue. Figure 4.9(b) shows very little effect on the earned revenue although Figure 4.9(a) shows slightly decreasing number of accepted SLAs. Increase in the number of shortest paths between a movie server and a workstation of an SLA creates many alternate proposals to the admission controller. Generally we would expect better revenue and longer number of accepted SLAs if more shortest paths are considered by the admission controller. But here we use a heuristic algorithm in the simulation of the admission controller, it may lead the solution state to a point of local optima. So sometimes less number of SLAs may be admitted. That is the reason why some points in Figure 4.9(a) shows decreased number of admitted SLAs. On the other hand, the accepted QoS levels of admitted SLAs are not always the same. Moreover there is sufficient randomness in the offered revenue in different QoS levels. So, revenue curve might not always follow the
acceptance curve exactly. That's why revenue curve in Figure 4.9(b) does not show any significant change in revenue.

![Avg. Time Vs. No. of Shortest Paths](image)

Figure 4.10: Effect of the number of shortest paths on average admission time

Figure 4.10 demonstrates the effect of the number of shortest paths on average admission time. In this simulation only the number of shortest paths is changed. Complexity analysis shows that the admission time is a quadratic function of the number of shortest paths, if other terms remain fixed. So our justification is clearly verified from the graph in Figure 4.10.

Figure 4.11 shows the effect of the number of shortest paths on the average utilization. It is natural that with the increase in the number of shortest paths, SLAs with different source and destination get more chance to be admitted. So increased link utilization increases overall resource utilization. So the curves in Figure 4.11 show increasing tendency of utilization. But with only 2 servers in the system, many alternate shortest paths cannot help that much increase utilization because of heavy utilization of nearby links of the servers. So, the curve for 2 servers in Figure 4.11 shows less increase in utilization than the other two curves in the figure.
4.4.4 Effect of the Percentage of Replica Considered

This section describes the performance of admission controller affected by the percentage of movie replication considered. As we mentioned in Section 4.3, each movie is replicated in different servers of the system. For a fixed number of servers in the system, the number of replication is also fixed. For example, if there are 8 servers in the system, each of the 10 movies has 4 replicas in 4 different servers. In this simulation we vary the consideration of percentage of replication of movies. 50% of replication for a particular movie means that we will consider most lightly utilized 2 servers out of 4 available servers to get that particular movie data. While selecting the servers to be considered we sort the servers according to their current resource utilization. First $x\%$ of least utilized servers will be considered as the source of the movie, where $x$ is the percentage of movie replica considered. Results of the simulation are presented in following figures.

Figure 4.12 shows the effect of the percentage of movie replica considered on the number of accepted SLAs and revenue. Increment of percentage of replication means increase in the number of effective servers. So, more SLAs get chances to be admitted and thereby it increases revenue. Some points in the graphs show no admission. This
happens because there remain no effective servers in these cases. For example, when total number of server is 6, each data has 3 replications. So, 10% of replication means data is supposed to come from any of 0.3 server(s). As we round the number of effective servers, so there remain no servers in this case. So, no admission occurred and no revenue is earned. That is why, some points in Figure 4.12 show no admission or revenue.

![Graphs showing the effect of percentage of movie replica considered on the number of accepted SLAs and revenue.](image)

**Figure 4.12:** Effect of the percentage of movie replica considered on the number of accepted SLAs and revenue

Figure 4.13 shows the effect of the percentage of movie replica considered on average admission time. We know that the complexity of the admission controller is quadratic function of the number of serves. Increase in the percentage of replication means increase in the number of effective servers and thereby increase in admission time. Although, there may be no admission in some points, the overhead of the admission controller takes up some time.
Figure 4.13: Effect of the percentage of movie replica considered on average admission time

Figure 4.14: Effect of the percentage of movie replica considered on the average utilization

Figure 4.14 shows the effect of the percentage of movie replica considered on the average utilization. As we observe no significance admission of SLAs up to the 40%
consideration of movie replication, so we plot the graphs in Figure 4.13 from the 50% of the movie replication to the 100% of replica considered. Increasing tendency in average utilization in Figure 4.14 results mainly for two reasons. One is the increase in the number of accepted SLAs for 6 or 10 servers. The other reason needs some explanation to be stated. Used resource in the system comes only from considered effective servers and network resources. And the number of effective servers increases with the increase in the percentage of replication although the total number of servers remains unchanged. From the definition of the average utilization given in Section 4.4.1, we can say that average utilization will increase with the increase in the percentage of replication.

4.4.5 Effect of the Number of Servers
In this section we describe the performance of the admission controller affected by the number of servers. We observe the effect by running the experiment keeping network link capacities fixed and varying the number of servers in the system. The bandwidth of links is kept proportional to the amount necessary for a system with 6 servers. We do two experiments to determine this effect. In the first case we keep the available resource of a server fixed so that the total server resource increases proportionally with the number of servers. In the second case, we keep the total servers resource fixed to determine the effect of distribution of resources among the servers.

4.4.5.1 Keeping per Server Resource Fixed
This section describes what will happen if we add more servers with same configurations in the system without changing the underlying network architecture, i.e. without changing the number of connections between switches or bandwidth of links of the network. As configurations of the servers are the same, so total server resource of each type increases with the increase in the number of servers in the system. Our findings from simulation results are presented in following figures.

Figure 4.15 shows that both the number of accepted SLAs and the earned revenue increase with the increase in the number of servers. Although the underlying network is not changed and it may be exhausted, but the admission controller can admit more SLAs without link bandwidth, because server resources become available with the
increase in the number of servers. So Figure 4.15(a) shows increase in the number of SLAs with the increase in the number of servers. The revenue, becoming a function of accepted SLAs, also increases and follows the acceptance curve, as shown in Figure 4.15(b).

Figure 4.15: Effect of the number of servers on the number of accepted SLAs and revenue

Figure 4.16: Effect of the number of servers on average admission time
Figure 4.16 shows the effect of the number of servers on average admission time. Both the number of servers and the number of admitted SLAs increase. The admission time is a quadratic function of both terms. So it increases accordingly, as shown in the Figure 4.16.

Figure 4.17 shows the effect of the number of servers on the average utilization. In this simulation the bandwidth of network links remains unchanged, as mentioned previously. For the system with small number of servers we have very few SLAs to consume the network link bandwidths. This utilization increases linearly until all the resources in the network links become exhausted, as shown in Figure 4.17 for the range of the number of servers from 2 to 6. For the range of the number of servers from 8 to 10 in Figure 4.17, the utilization curve starts declining as the server resources remain unutilized. This happens because there remains no available link bandwidth in certain links for that range. It is worth mentioning that SLAs without link bandwidth will also contribute to increase the resource utilization in this case but the contribution is not sufficient to make it increasing.
4.4.5.2 Keeping Total Server Resource Fixed

In this section we will describe what will happen if we add more servers with weaker configurations in the system without changing the underlying network architecture, i.e. without changing the number of connections between switches or bandwidth of the links of the network. Sometimes it may happen that we have limited amount of money. So we cannot upgrade the network and we are bound to choose more servers with low amount of resources or few servers with high amount of resources. That is, we cannot increase total amount of server resources. This section projects what will happen in such a situation. Keeping the total amount of server resources constant means decrease in the amount of individual server resource if we increase the number of servers. Our results are presented in the following figures.

![Graph (a)](image1.png)

![Graph (b)](image2.png)

Figure 4.18: Effect of the number of servers on the number of accepted SLAs and revenue

Figure 4.18 shows the effect of the number of servers on the number of accepted SLAs and revenue. From a simplified graph theoretic analysis to be presented in Section 4.5, we know that the number of admissible SLAs depends actually on resources of the system. Although the number of servers increases in the system, but total amount of server resources and link capacity remains fixed. So it is natural that the number of admitted SLAs and total revenue will be more or less same, which clearly verified from the graphs in Figure 4.18. Some increase in the values mainly
because of sparsed distribution of servers in different parts of the network and SLAs without link bandwidth, as explained in Section 4.4.1.

**Figure 4.19: Effect of the number of servers on average admission time**

Figure 4.19 shows the effect of the number of servers on the average admission time. The complexity is a quadratic function of both the number of servers and the number of SLAs. Generally, the number of admitted SLAs is much greater than the number of servers. From Figure 4.18(a), we know that the number of admitted SLAs does not change significantly. So the effect of the constant number of SLAs on the admission time makes the quadratic effect of the number of servers lighter on the admission time. That is why the curve in Figure 4.19 shows some linear increase in average admission time instead of quadratic behavior.

Figure 4.20 shows the effect of the number of servers on the average utilization. The increase in the number of servers affects the average utilization in two ways. It can increase the average utilization as SLAs without link bandwidth will increase the utilization of server resources. On the other hand, it can decrease the average utilization as admitted SLAs can select those servers that are nearer to them, ie, uses
less number of links, which reduce the usage of link capacity. Due to these two effects we find increase in utilization up to a certain point, for example for the range of the number of servers from 2 to 4 in Figure 4.20, and decrease in average utilization for further increase in the number of servers.

![Graph: Avg. Utilization Vs. No. of Servers](image)

Figure 4.20: Effect of the number of servers on the average utilization

4.4.6 Effect of the Expected Number of Users

In this section we will describe what will happen if more users come to the system without changing the underlying network or the number or configuration of servers. In this simulation we increase the incoming rate so that more users can arrive in the system. Our objective to see the effect on the number of admitted users, earned revenue, response time and resource utilization of the system. Our findings are presented in following figures.

Figure 4.21 shows the effect of the expected number of users on the number of admitted users and revenue. As total resource in the system is fixed, so total number of admission and thereby earned revenue will be bounded by some upper limits whatever large the number of arrival users in the system is. The graphs in Figure 4.21 clearly justify our views. Figure 4.21(a) shows around 300 admitted users, when the expected number of users is only 200. This happens because we have sufficient resources in this case and so most of the incoming users get admission. On the other
hand the number of leaving users may be small. So this point shows more admitted users than the expectation.

Figure 4.21: Effect of the expected number of users on the number of admitted users and revenue

Figure 4.22: Effect of the expected number of users on average admission time
Figure 4.22 shows the effect of the expected number of users on the average admission time. It is natural that admission time will increase when there are more arrival of users in the system. But the graph shows low admission time in some points. In this simulation we fix the system resources so that nearly 600 users can be admitted in the system although the upper limit of actual number of admission is nearly below 500. So when the number of arrival users in the system is much larger than 500, the admission controller gets more chance to pick suitable number of users so that its earned revenue and resource utilization becomes optimized. In this case it becomes easier for the admission controller to select servers and paths and to reserve resources on the system for the admitted users. So lower admission time is required in this case compared to the time requirement when there are nearly 400 to 600 users in the system. In the later range the admission controller has limited freedom to select the optimal servers and paths to reserve resources for the admitted users. So it takes comparatively long time to do admission. That is why Figure 4.22 shows long admission time with 400 to 600 arrival users and comparatively short admission time for 800 to 1000 arrivals.

**Figure 4.23: Effect of the expected number of users on average utilization**
Figure 4.23 shows the effect of the expected number of users on the average utilization. Average utilization may be affected by some factors. The increase in the number of admitted users increases resource utilization at least from the servers. So Figure 4.23 shows increasing tendency of average utilization for the range of 200 to 400 expected users. But when there are more users in the system, it is possible that the admission controller is in better position to select more users that are very near to their source servers. So utilization may decrease in this case. That is why Figure 4.23 shows relatively low utilization for the range of 600 to 1000 arrivals of users.

4.5 Validation of Simulation Results

We use the following simple theoretical calculation to show that our simulation results, specially the number of admitted SLAs and the earned revenue from the users, are correct. We need further study to prove the correctness of resource utilization.

The network used in our simulation is nearly a Harary graph [37]. Let, there be $N$ nodes and $N \times r$ links in the network, where $r$ is the degree of nodes.

Maximum distance (edges) between two nodes $= \frac{N}{2r}$

Minimum distance between two nodes $= 1$

Average distance between two nodes, $L = \left\lfloor \frac{N}{2r} + \frac{1}{2} \right\rfloor$

Each SLA, $i.e.$, user requires on the average $L$ links to traverse its data.

Let, each link has bandwidth $= B_L$

Average QoS of an SLA requires $B_Q$ bandwidth.

So, each link allows $\left\lfloor \frac{B_L}{B_Q} \right\rfloor$ SLAs, $i.e.$, users to pass their data.

On average $L$ links will be exhausted by $\left\lfloor \frac{B_L}{B_Q} \right\rfloor$ admitted SLAs.

So, total number of admissible SLAs on the average $= \left\lfloor \frac{B_L}{B_Q} \right\rfloor \times \left\lfloor \frac{N \times r}{L} \right\rfloor$

Now we will find the average number of admissible SLAs with respect to server parameters.
Let the total number of servers be $s$.

Average QoS level of an SLA requires $q$ fraction of server resource.

Resource of each server is assumed to be unity.

So, the average number of total admissible SLAs (according to server resource only) is $\frac{s}{q}$.

If both server and link parameters are considered, then average number of total admissible SLAs will be $\min\left(\frac{B_L}{B_Q} \times \left\lfloor \frac{N \times r}{L} \right\rfloor,\frac{s}{q}\right)$.

Now, in our simulation, $N=31$ and $r=2$. So, $L = \left\lfloor \frac{n + 1}{2} \right\rfloor = 5$

Table 4.4 shows average expected number of accepted SLAs and revenue earned for different QoS levels, for $s=2$ (2 servers in the system) and $B_L$ is 100 (100 units of bandwidth in each link).

Table 4.4: Expected number of accepted SLAs and revenue for different QoS levels

<table>
<thead>
<tr>
<th>QoS Levels</th>
<th>Expected number of accepted SLAs</th>
<th>Maximum expected revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS1</td>
<td>400</td>
<td>1200</td>
</tr>
<tr>
<td>QoS2</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td>QoS3</td>
<td>133</td>
<td>798</td>
</tr>
</tbody>
</table>

Figure 4.3 shows that the accepted number of SLAs is nearly 165 and revenue is nearly 900 which are close to our theoretical expected results. It is worth mentioning that there are SLAs with three different QoS in these 165 SLAs.

4.6 Chapter Summary

This chapter presented the simulation of the admission controller. Simulation parameters, its environment and results have been described and analyzed. We have also verified the validation of our results using a graph theoretical calculation. The
next chapter concludes this thesis with a brief description of our contribution and some points related to the future study.
Chapter Five

Conclusion

In this chapter we summarize the major contributions from this thesis and present suggestions for the future research work.

5.1 Major contributions

A central challenge facing network operators is maintaining customer guarantees whilst sustaining profitability. In this thesis we derive a new admission control algorithm for a multimedia system with multiple servers underlying an Enterprise Network to admit enough users to efficiently utilize the system resources, while maximizing the earned revenue from the users.

To summarize, the main contributions presented in our research are as follows:

1. We presented a new problem and gave a new definition of Service Level Agreements for an Enterprise Network, controlled and owned by an organization. As our admission controller considers both server and network resources, so we had to redefine existing simple SLA definitions.

2. We gave the architecture of a new admission controller along with proposals for data transmission protocols in different stages of the admission controller. We also designed an admission control methodology for the architecture and its worst-case complexity analysis has also been done.

3. We have successfully mapped the problem to a variant of knapsack problem and applied heuristic solution to design the admission controller.

4. The performance of the admission controller has been shown through extensive simulation. A java simulation has been developed. In this simulation we assumed that media servers had limited resources to serve all admitted users and the Enterprise Network had limited bandwidth to carry multimedia
streams for all users from the respective servers. The admission controller did
admission control based solely on available and required resources in the
media servers and the Enterprise Network using heuristic for solving the
MMKP. We used this simulated system to generate performance data. Results
have been analyzed and validated through detailed theoretical analysis.

5.2 Future Research Work

We suggest following further research plans for our present admission controller.

1. We designed the admission controller in a small scale, for an Enterprise
Network, a network with limited number of nodes and edges. Further study is
necessary to extend the architecture for Internet or interconnected multiple
Enterprise Networks.

2. Our admission controller is a centralize one. A distributed system can be
established where multiple admission controllers will try to optimize their
respective revenues from users' proposals which may require data
transmissions among different networks. A new redefined SLA is necessary.

3. We assumed that users' machines are capable of enjoying multimedia data.
We do not consider the resources of users' machines. The change of resources
of servers and network bandwidth in the middle of transmission is not
considered. A new admission control methodology can be developed
considering these issues.

4. We considered that a movie or its replication is entirely kept on a single
server. We do not consider keeping different portion of a multimedia data on
different servers. Collecting and combining data from different sources and
transmitting it to destination is complex. A different methodology has to be
developed to handle segmented storage of multimedia data.

5. All users requesting for data cannot be admitted. Only portion of them is
admitted by the admission controller to maximize the earned revenue.
Rejected users leave the system. But users sometimes prefer to make future reservations. Implementing future reservations require redefinition of SLAs. A new utility model and algorithm have to be designed.

6. We gave a worst-case time complexity of the admission control algorithm. So, some graphs derived from simulation show lower time as expected. A detailed average case analysis of time complexity is required to exactly compare the simulation and the theory. Similarly, a more graph theoretical analysis is required to predict the average utilization.
References


