Application of Heuristic MMKP in Admission Control and QoS Adaptation for Distributed Video on Demand Service


Department of Computer Science and Engineering
Rajshahi University of Engineering & Technology
Rajshahi-6204, Bangladesh
{shamsulbd, khrajeeb, sohail_ruet, hsagor, boshir_bd}@yahoo.com

Abstract

Allocation and reservation 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) for multimedia services delivered over the Internet. In this paper, we have proposed a new semi-distributed architecture for admission control and QoS adaptation of multimedia sessions to maximize revenue from multimedia services for Distributed Video on Demand System (DVoDS). We have introduced the mapping of Utility Model - Distributed (UM-D) by semi-distributed controller to the Multidimensional Multiple-choice Knapsack Problem (MMKP), a variant of the classical 0-1 Knapsack Problem. An exact solution of MMKP, an NP-hard problem, is not applicable for the online admission control problem in the VoD System. Therefore we have applied heuristic, I-HEU for solving the MMKP for online real-time admission control and QoS adaptation. We have applied the admission control strategy described in the UM-D to the set of Media Server Farms providing streaming videos to users. The performance of semi-distributed architecture applied in a simulated environment over a set of Media Server Farm has been discussed detail using the experimental outcome.

1. Introduction

Recent years have witnessed the use of digital video and audio in several important applications that affect various aspects of our daily life. These applications include distance learning, digital libraries, Movie-on-Demand (MoD), electronic commerce, etc. Some applications, such as video conferencing or streaming audio, video on demand, produce network traffic, which require a guaranteed level of Quality-of-Service (QoS) to work properly. These QoS requirements may be in terms of a minimum bandwidth, bounded end-to-end delays, or maximum packet loss rate suffered by a flow. To support such flows and to allocate and maintain their finite network resources to uphold their guarantees, some form of admission control is important to accept profitable flows and to reject new traffic flows that would cause the controller to violate its promises. In this paper we have proposed a semi-distributed model of VoD system and semi-distributed controller architecture to do admission control and QoS adaptation for the set of Media Server Farm. We have proposed the mapping of utility model-distributed (UM-D) to the MMKP for the semi-distributed admission controller. Finally, to analyze the performance, we have presented simulation of admission controllers over a set of Media Server Farms.

2. Related Work

Guaranteed transmission of voice and video is a challenge of the researchers in Internet computing field. There has been lot of interesting works in recent years on reservation-based management of resources like CPU cycles and bandwidth carrying the video and voice [1, 2, 3]. Transmission of video requires guaranteed Quality of Service (QoS), which is defined by the amount of bandwidth, latency and jitter bound; otherwise the customers will not pay for the services. Delivery of multimedia streams with absolute guaranteed QoS from a single multimedia server has been proposed by Khan [4]. Hua[1] describes video delivery using multicast, streaming strategies with application layer multicast, and proxy caching techniques. To do admission control and QoS adaptation in a set of Media Server Farms, centralized broker architecture has been proposed by Akbar [5]. The load balancing problems of centralized scheme lead us to the development of a fully distributed scheme for MM servers with more scalability and fault tolerance [7]. But, fully distributed architecture faces message passing complexity and time requirement overhead for specific session. Mundur [8] proposed a global request handling and admission control strategies based on limited redirection of blocked requests to other resources for distributed VoD system. Redirect request handling polices require higher implementation overhead and extra connection setup time. The Utility Model- Distributed...
(UM-D) [4,6] provides a sound strategy for selecting an optimal set of sessions, with specified QoS levels and a set of servers to provide each of MM video streams. The goal is to guaranteeing the agreed levels of QoS, facing resource constraints of the servers and the network. The Utility Model is mapped to the MMKP, whose exact solution is NP hard. The proposed heuristic algorithm [6] is suitable enough for finding a set of QoS levels for the multimedia sessions quickly to allow admission and adaptation decisions in real time.

3. Preliminaries

3.1 The Multidimensional Multiple Choice Knapsack Problem (MMKP)

The MMKP is a variant of the KP. Let there be $n$ groups of items. Group $i$ has $l_i$ items. Each item of the group has a value and its required quantities of each of the $m$ resources. The objective of MMKP is to pick up exactly one item from each group, for maximum total value of the collected items, subject to $m$ resource constraints of the knapsack [6].

3.2 Utility Model - Distributed (UM-D)

The Utility Model - Distributed provides a strategy for selecting an optimal set of sessions (with particular QoS levels and selection of servers to provide multimedia stream components) from the submitted requests for multimedia sessions. The goal is to maximize the revenues earned from the selected sessions, subject to the resource constraints imposed by the servers and networks of the DVoDS. The UM-D mainly describes the admission control and QoS adaptation strategy for the multimedia sessions in mathematical notation [4,5].

3.3 Video on demand service

A Video on Demand (VoD) system consists of a video server with a video archive and a number of client machines connected via a local area network. Users use client software to make a request for their desired video. In response to a service request, the server delivers the requested video to the user. The VoD service is replicated in order to achieve high availability and fault tolerance. Each video file is held by a subset of the servers. The service is provided by a number of servers having the capability of dynamic group formation. The service from the servers to carry multimedia video streams for all customers is provided over a local distribution network, such as ATM LAN, HFC, or xDSL. It is assumed that there are sufficient network resources at the local distribution network to deliver videos to the customers and that there is no resource contention on the local distribution network.

4. Distributed admission control architecture

4.1 Broker architecture

Akbar [5] presented centralized broker architecture. This architecture utilizes centralized computing performed by a broker. The broker is interposed between the clients and servers of the system. This architecture faces load-balancing problem with the increase of users.

4.2 Semi-distributed architecture

In this approach, each Media Server Farm (MSF) is connected with a Local Controller (LC) and all the LCs are interconnected through a Central Controller (CC) for negotiation among them. Only the LC of a MSF has the authority to allocate resources of the servers for providing VoD services to the users. Each LC keeps the record of available resources of the corresponding media servers. Users are required to communicate with (one of) the local admission controllers of the DVoDS to submit their requests and to receive admission or rejection decisions made by the controllers. The semi-distributed admission control and QoS adaptation architecture is shown in figure 1.

![Figure 1: The semi-distributed admission control and QoS adaptation architecture](image-url)
5. Mapping of the UM-D by semi-distributed architecture to the MMKP

Admission control and QoS adaptation by the semi-distributed architecture can be mapped to the MMKP. Each server in the DVoDS represents one knapsack with multidimensional resources such as CPU cycles, I/O and Memory. Here, we have ignored the link bandwidths for the simplicity of mapping. Knapsack Problem specifies resource management. Our policy is to do admission control by allocating resources. That is why Knapsack Problem is suitable to represent the admission control problem.

5.1 Mapping of sessions to the groups

Each multimedia session request with a set of QoS levels submitted to the controller, represents a group in the MMKP.

5.2 Mapping of the QoS levels of a session to the items of a group

A particular level of QoS $p_{ij}$ can be provided by one or more servers, because the components of the multimedia stream can be partitioned and replicated in multiple servers. We can find all the combinations of servers that can serve a particular level of QoS of a particular requested multimedia session. Let there be M servers. Now this level of QoS can be provided by $M$ combinations of servers These combinations might have different utilities. These are indistinguishable from the user’s point of view, but are very different from the DVoDS controller’s point of view. Thus one QoS level is transformed to multiple options. Each of these options is defined as QoS with respect to Controller (QoS-C) and represents an item of a group in the MMKP.

5.3 Mapping of server resources to the resources of the knapsack

If there are $m_e$ resource dimensions in each of $M$ multimedia servers, then we can think of a merged server with $M \times m_e$ resources; it is in effect the union of the physical servers, and its resources belong to the cross-product space (server, resource). In this way we can map multiple servers to one merged server and hence to one knapsack. Now, our problem is to find a QoS-C level for each session for maximum utility of the DVoDS, subject to resource constraints of the merged server, which is treated as an MMKP.

6. Admission control and QoS adaptation methodology

New sessions are collected into a batch over a time interval. We have used I-HEU, the incremental heuristic solution of the MMKP proposed in [6] for admission control and QoS adaptation. A batch of multimedia sessions is submitted to the controller which finds out the QoS-C levels. For the new sessions, a dummy null level is added to the QoS-C profile of each session. This QoS level gives null revenue with null resource consumption. I-HEU is applied to the new sessions as well as to the already admitted one, once in every batch. I-HEU finds the new set of QoS-C levels for each session by upgrading, by downgrading, or without changing the QoS-C levels of the sessions in order to maximize the earned revenue from the sessions. If the QoS-C level of any new session in the new batch remains null after applying I-HEU, then it is rejected. The other sessions are admitted at their non-null QoS-C levels. The admitted sessions enjoy the assigned QoS-C level at least for the next epoch.

When any session leaves, some resources are released from the multimedia server system. The computation of I-HEU for the next batch will get the opportunity of these released resources to upgrade some sessions or to admit more sessions.

When some resources of the system decreases, for example if the communication bandwidth decreases due to the failure of any communication link or a server goes down, then QoS adaptation is done as soon as the fault is detected. It is done in three steps:

**Step 1:** All the admitted sessions are downgraded to the lowest QoS level (not null). If I-HEU gives a feasible solution then adaptation is complete. Otherwise execute step 2.

**Step 2:** All the admitted sessions are downgraded to the null QoS level. Now I-HEU will admit some sessions and some may be rejected. If any session is rejected then execute step 3.

**Step 3:** All the sessions admitted in step 2 are downgraded to the lowest level of QoS. Now some resource will be released and I-HEU is applied to the rejected sessions from step 2. This will allow some sessions to be admitted and the rest will be rejected. These rejected sessions will not be considered further. The admitted sessions in step 2 and step 3 will be upgraded again by I-HEU to do final adaptation.

7. Computational complexity of the controller

The complexity of the controller depends on the size of the system, the number of users in the system and the algorithm used to solve the MMKP. We have presented the complexity analysis of the broker initialized here to run the simulation over a set of Media Server Farms. The total number of users in this set of Media Server Farms with $M$ servers is approximately $M \times n_k$, where $n_k =$
estimated number of users enjoying Silver QoS from a server at full load. There are \( M \) video servers. Each server has three resources, CPU cycles, Memory and I/O BW. So, the total number of resources is \( 3M \).

Each video stream is replicated in more than one server. The number of replications can be expressed by \( M \times 2/10 \). On the average, there are 2 QoS levels in each session. So, the number of QoS-B levels for a session is approximately \( M/5 \). The MMKP solved by the broker has \( n = M \times n_k \) groups, \( t = M/5 \) items in each group and \( m = 3M \) resource dimensions. Using the complexity analysis of the heuristics of the MMKP, we get the following worst-case complexity of the controller using I-HEU,

\[
O\left(\frac{M^2n_k^2 M^3}{25}\right) \cong O(M^3).
\]

8. Simulation of the admission controller

We have simulated the admission controllers of the set of Media Server Farms for different distributed control architectures using greedy and heuristic approach. The performance data has been collected by varying the number of users and the number of video servers. The following tables present the parameters of the set of Media Server Farms required for the simulation.

**Table 1:** Different simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value(s) for the Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum length of movie</td>
<td>3 hours</td>
</tr>
<tr>
<td>Total Number of movies in MSF</td>
<td>10</td>
</tr>
<tr>
<td>Number of video servers in MSF</td>
<td>5, 10, 15, 20, 25, 30, 35, 40, 45</td>
</tr>
<tr>
<td>Number of replications of video</td>
<td>3, 3, 3, 3, 4, 5, 6, 7, 8, 9</td>
</tr>
<tr>
<td>Number of movie copies per server</td>
<td>2, 6, 3, 3, 2, 2, 2, 2, 2, 2</td>
</tr>
<tr>
<td>Repetitions of the simulation experiment</td>
<td>5</td>
</tr>
</tbody>
</table>

The servers are initialized with CPU cycles, memory and I/O bandwidth capacity, where all the parameters are randomized by uniform distribution \( U(0.95, 1.05) \), to simulate the fluctuation of the available resources in different servers. The users can enjoy the movie in three QoS levels (Gold, Silver, Bronze), where each QoS level resource requirement has been defined with randomization using \( U(0.75, 1.25) \).

**Table 2:** Initialization of server resources in the media server farm

<table>
<thead>
<tr>
<th>Type of resource</th>
<th>Server resource value</th>
<th>Initialization function for a video server</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>460 MHz dual processor</td>
<td>100 X U(0.95, 1.05)</td>
<td>Total 800 MHz is equivalent to 100 cycles.</td>
</tr>
<tr>
<td>RAM</td>
<td>256 MB</td>
<td>120 X U(0.95, 1.05)</td>
<td>136 MB is used by the O/S.</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>640 Mbps</td>
<td>600 X U(0.95, 1.05)</td>
<td>40 Mbps is reserved for the system.</td>
</tr>
</tbody>
</table>

**Table 3:** Different QoS levels supported by the DVoDS

<table>
<thead>
<tr>
<th>QoS levels</th>
<th>Average I/O BW req.</th>
<th>Average CPU cycles req.</th>
<th>Average Memory req.</th>
<th>Average offered price by the user</th>
<th>Average cost of providing media</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS 1 (Bronze)</td>
<td>1.5 Mbps</td>
<td>0.25%</td>
<td>0.3 MB</td>
<td>$1.0</td>
<td>$0.75</td>
</tr>
<tr>
<td>QoS 2 (Silver)</td>
<td>3.0 Mbps</td>
<td>0.50%</td>
<td>0.6 MB</td>
<td>$3.0</td>
<td>$2.25</td>
</tr>
<tr>
<td>QoS 3 (Gold)</td>
<td>4.5 Mbps</td>
<td>0.75%</td>
<td>0.9 MB</td>
<td>$5.0</td>
<td>$3.75</td>
</tr>
</tbody>
</table>

**Table 4:** Initialization of resource requirements for different QoS levels?

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initialization function</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O bandwidth requirement for the kth QoS level</td>
<td>( 1.5 \times k \times U(0.75, 1.25) )</td>
</tr>
<tr>
<td>Memory requirement for the kth QoS level</td>
<td>( 0.3 \times k \times U(0.75, 1.25) )</td>
</tr>
<tr>
<td>CPU cycles for the kth QoS level</td>
<td>( 0.25 \times k \times U(0.75, 1.25) )</td>
</tr>
<tr>
<td>Offered price by user for the kth QoS level</td>
<td>( (2k+1) \times U(0.75, 1.25) )</td>
</tr>
<tr>
<td>Cost of providing kth QoS level of a movie</td>
<td>( (2k+1) \times U(0.55, 0.95) )</td>
</tr>
</tbody>
</table>

We have coded the simulation of the LC & CC using the Java programming language (JDK ver 1.5). LC generates a big batch of user session request initially, having double size of Media Server Farm’s server resource capacity, to get admission for creating contention of resources in the resided servers. We have discarded 10 percent of accepted sessions to create space for future sessions, then generate 15 percent new sessions as a small batch. The total simulation ends when all LCs have generated 100 batch. We have calculated the Revenue earned from the users and the time required by LC to do admission control and QoS adaptation. The summation of the revenues of the sessions and average time requirement by each session is the measurement criteria of the distributed admission controller’s performance in our simulation. The simulation was implemented on a 1.7 GHz Pentium IV machine plugged in a LAN and was running under Windows XP having a physical memory of 128MB RAM.

9. Experimental Results

Figure2 and figure3 depict that our new semi-distributed architecture requires much less acceptance and rejection time than centralized broker architecture in both heuristic I-HEU and greedy approach. This is because the total workload is distributed among the LCs in semi-distributed architecture and each LC fetches less traffic than one centralized server with one centralized controller in the broker architecture.

In both architectures, the greedy approach seems to be good than I-HEU with respect to response time as it takes less acceptance and rejection time than I-HEU. This is because, I-HEU is applied to the new sessions as well as to
the already admitted ones, once in every batch. I-HEU finds the new set of QoS-C levels for each session by upgrading, by downgrading, or without changing the QoS-C levels of the sessions, in order to maximize the earned revenue from the sessions. For this reason I-HEU requires higher time than greedy.

The rejection rate of semi-distributed architecture is much less than broker architecture. In both architectures, I-HEU gives better result than greedy approach. The rejection rate of semi-distributed architecture using I-HEU is initially high, but decreases quickly with some fluctuations as the number of server increases and the rejection rate is lowest after 40 servers shown in figure 4. This is because as the number of server increases, there is less resources contention in the system.

Figure 5 illustrates the over all response time of different admission control architectures using different admission control algorithms. Semi-distributed architecture requires fewer overall response time than broker architecture in both approaches. For all admission control architectures, the response time increases if the number of server increases, due to extra search among larger number of servers. So, for a large VoD system our proposed semi-distributed architecture is better and efficient as it provides better fault tolerant.

On the contrary, in figure 4, we see that greedy approach has much higher rejection rate than I-HEU and in figure 5, greedy approach has much less revenue than I-HEU. Our heuristic determines the operating QoS of the admitted session after analyzing all the sessions (admitted and requested) in the system. That is why it gives better results than greedy method, which is based on the FCFS rule and does not search extensively to achieve better utilization of resources. Our main goal is to maximize revenue by admitting more profitable sessions. So, I-HEU gives better result by achieving our objective.
From figure 6, we observed that the earned revenue for all the architectures increases as the number of server increases. The revenue increases as we add more servers in the system and if there are more users coming proportionally in the system as it gets more users in a batch to select a profitable one. For small number of servers, the controller may select a non-profitable user for admission and may be unable to take advantage of a better session later as resources are reserved by the non-profitable user. On the other hand the larger batch size defers the admission of sessions and the controller may be deprived of earning more money. Sometimes a controller with smaller batch size helps to admit sessions early as some sessions have already been left. Revenue earned is greater when we used I-HEU for both architectures. The semi-distributed architecture has slightly less revenue than broker. But the semi-distributed architecture is inherently more scalable and better fault tolerant which are the key issues for multimedia services.

10. Conclusion

In this paper, we have analyzed the performance of different distributed admission controller architecture using different admission control algorithm. We have presented a new model for admission control and QoS adaptation architecture which addresses the issues of optimal admission of user requested sessions, server failures and scalability that are the key features of video on demand service. The server architecture of the semi-distributed VoD system is analogous to that of a web server in the Internet. The users get access to an intermediate web server and then the page referred to another web server. Our model approximately fits this web server architecture. Experimental results show that the distributed VoD system requires a heuristic admission control algorithm like I-HEU. The semi-distributed VoD system also demonstrates server fault tolerance. The completely distributed computation of I-HEU will become necessary as the load and hence the computational time requirement increases with the increase of users and number of servers. Distribution of controller functionality (admission control and QoS adaptation) leads the system to be more fault tolerant, as the system can run even if one of the admission controllers fails. The concepts demonstrated in this paper may be exploited to construct a highly available distributed VoD service.

11. References


