Multimedia Streams Retrieval in Distributed Systems Using Learning Automata

Safiye Ghasemi[1], Amir Masoud Rahmani[2]
[1]Department of Computer Engineering
Sepidan Branch, Islamic Azad University
Sepidan, Islamic Republic of Iran
[2]Islamic Azad University, Science and Research
Branch, Tehran
ghasemi.ss@gmail.com, rahmani@srbiau.ac.ir

ABSTRACT: *Current academic Internet environment has enabled fast transfers of huge amounts of data, and has made high quality multimedia and collaborative applications a reality. This article describes a model for distributed multimedia retrieval which performs the retrieval of different multimedia to a variety of clients using learning automata algorithm named LAGridMSS. LAGridMSS allocates a proportion of bandwidth of each node for sending a specified file, and then applies learning automata to allocate packets of files to each node that contains the context. The files' popularity and the remainder bandwidth of nodes are two main factors here for allocating packets to each node.*

Simulation results show improvements in proposed model, LAGridMSS, in some QoS factors such as delay, jitter, and reliability compared to previous multimedia retrieval system such as GridMSS and GridMedia.

Keywords: Grid Computing Environment, Streaming Files, Learning Automata, and Quality of Service

Received: 18 November 2010, Revised 27 December 2010, Accepted 31 December 2010

1. Introduction

The term "multimedia" has been used in many different contexts and means different things to different people [2]. In this issue, multimedia mainly relates the individual or combined use of large-volume, high-quality continuous digital media such as audio and video.

Grid computing has been evolving over recent years towards the use of distributed and parallel computing and also it provides dynamic, secure and coordinated sharing of heterogeneous resources that may be distributed geographically as well as organizationally [11]. One of the major challenges for grid computing is to ensure "non-trivial qualities of service" to the users. A grid computing platform is exploited and content adaptation is performed through appropriate agents that are allocated as jobs and executed on grid systems. These systems can be used for sharing resources of heterogeneous computers. In such a newborn environment, there are so much researching area, one of them is streaming applications which a poor work has been done on it so far.

Resources and information sharing in grid environment should be done with a desired QoS (Quality of Service), especially when sharing is in the field of time dependant applications. Quality of Service of a specific system describes how good that system will function with respect to a set of parameters specified by client. In this paper QoS is considered as a set of quality and quantity specifications of a distributed multimedia that is important for an application to satisfy the clients.

The client specifies the desired quality of media. An application's QoS requirements are conveyed in terms of high-level parameters that specify what the user requires. The specification of requirements of an application occurs according to client's parameters and is known as QoS parameters. After QoS specification, a system has to provide the demanded services according to accessible bandwidth. If it was not available the system accesses the related agents to negotiate the requirements identified by the client and may remove, change or adopt some parameters. As this negotiation successfully finishes then the streaming application will be done. Adding agents to multimedia systems causes the model to be modular and behave more flexible and adoptable in grid environment. Agent's changing role adds the system flexibility in replying different queries.

Many multimedia streaming algorithms in grid environment have been introduced, but none of them support all QoS parameters. Some of them use much more time for coordinating resource nodes. In the GridMSS framework [1] many of important QoS parameters such as reliability, delay, jitter, accessibility and flexibility are supported and the overall framework performance is high.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work about improving QoS of media streaming in grid systems. In Section 3, we analyze the concepts of learning automata. Section 4 presents the specification of existed problem and in Section 5 the proposed model for streaming applications is presented and we discuss in detail our experimental result in Section 6, while in Section 7 we conclude the paper.

2. Related works

GridMedia [18] is a peer-to-peer based multicast architecture designed especially for large scale video streaming with quality requirement in terms of real time, low latency and bandwidth demanding. This architecture mainly contains the overlay protocol and the transmitting algorithm. The overlay protocol MSOMP advocates mesh-based two layer structure, which makes the upper layer much more robust than traditional tree-based scheme and ensures the demanding bandwidth.

A framework for QoS-based service discovery in manufacturing grid is described in [15]. The main focus of this framework is to provide a means for the service requesters to search for services based on QoS criteria in MG, to provide QoS guarantees for service execution and to enforce these guarantees by establishing SLAs.

Adrizona in [2] has presented an approach to multimedia content retrieval in a distributed digital library implemented on a P2P network. The proposed system employs an adaptive technique for routing queries and addresses the issues arising from the presence huge amounts of data, their peculiar nature and, finally, the lack of a centralized index.

Our approach employs a decentralized architecture which fully exploits the storage and computation capability of computers in the Internet and forwards queries throughout the network using an adaptive routing strategy that dynamically performs local topology adaptations.

In [5], the heterogeneous asynchronous multi-source streaming (HAMS) model for transmitting continuous multimedia files from multiple peers to a leaf peer is discussed. Goudarzi[6] discusses how to support a receiver peer with enough QoS of the multimedia steaming service by multiple source peers in [6]. Not only a receiver peer but also source peer is moving in a network. Here, QoS supported by source peer is changing according to the movement of the receiver peer and source peer.

GridMSS [1] is a streaming contents retrieval framework which supports the QoS. The main characteristics of this framework are highlighted by its promising QoS supporting such as accessibility, low latency and reliability, its flexibility and packet allocation mechanisms of nodes. The approach is that the allocation is done according to available bandwidth of each node. GridMSS provides a high performance files streaming framework for a large population of users and ensures the QoS in terms of accessibility, low latency and reliability. To tide over the bandwidth bottleneck and the rate of losing packets some new relations between bitrates of nodes and services they can provide are determined and in the extended GridMSS a genetic algorithm is used to find the best nodes with suitable bit rate for sending demanded content to receiver.

CoopNet (Distributing Streaming Media Content Using Cooperative Networking) [20] divides streaming media content into multiple sub-streams using MDC and each sub-stream is delivered to the requesting client via a different peer. This improves robustness and also helps balance load amongst peers. The system has two cases. First is live streaming refers to the synchronized distribution of streaming media content to one or more clients, and second one is on-demand streaming, which refers to the distribution of pre-recorded streaming media content on demand (e.g., when a user clicks on the corresponding link).

Another streaming system is a QoS-Aware P2P streaming framework called Whirlpool which organizes peers into different levels based on their end-to-end latencies from the streaming source. The key design issues of Whirlpool can be mentioned as presenting a membership service, which selects potential good neighbors with higher probabilities, and analyzing the convergence property of the level calculation the key design issues of Whirlpool, presented a membership service, which selects potential good neighbors with higher probabilities, and analyzed the convergence property of the level calculation algorithm.

3. Learning Automata

Learning Automata (LA) are adaptive decision making units that can learn to choose the optimal action from a set of actions by interaction with an unknown random environment. At each instant, *n*, the LA chooses an action a_n from its action probability distribution and applies it to the random environment. The random environment provides a stochastic response, which is called a reinforcement signal to the LA. Then the LA uses the reinforcement signal and learning algorithm to update the action probability distribution.

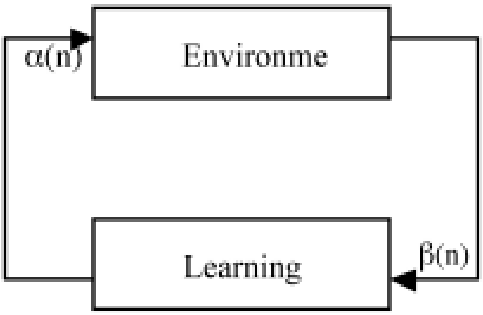


Figure 1. Learning automata connection with environment

A learning automaton is an automaton that improves its performance by interacting with the random environment in which it operates. Its goal is to find among a set of μ actions the optimal one, so that the average penalty received by the environment is minimized. This means that there exists a feedback mechanism that notifies the automaton about the environment's response to a specific action.

The operation of a learning automaton constitutes a sequence of cycles that eventually lead to minimization of average penalty. The learning automaton uses a vector $p(n) = \{p_1(n), p_2(n), \ldots, p_\mu(n)\}$, which represents the probability distribution for choosing one of the actions $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. The core of the operation of the learning automaton is the probability updating algorithm, also known as the reinforcement scheme, which uses the environmental response $\beta(n)$ triggered by the action α_i, selected at cycle to update the probability distribution vector . After the updating is finished, the automaton selects the action to perform at cycle $n+1$, according to the updated probability distribution vector $p(n+1)$. A general reinforcement scheme has the form of equations (1, 2).

$$\begin{aligned} p_i(n+1) &= p_i(n) + \alpha(1\text{-}p_i(n)) \\ p_j(n+1) &= p_j(n) - \alpha p_j(n)) \; \forall j \neq i \end{aligned} \tag{1}$$

$$\begin{aligned} p_i(n+1) &= (1-b)\, p_i(n) \\ p_i(n+1) &= \frac{b}{r\text{-}1}(1-b)\, p_j(n) \; \forall j \neq i \end{aligned} \tag{2}$$

In these two equations, *a* and *b* are reward and penalty parameters respectively. For a=b, learning algorithm is called $L_{R\_P}^{1}$, for a<<b, it is called $L_{R\varepsilon P}$, and for b=0, it is called $L_{R\text{-}I}^{3}$.

4. Problem Specification

There are a variety of nodes with different files in grid computing. Care is taken that some of the files may exist in several nodes. It means that the number of content all over the grid system is greater than one. In this research we suppose that the frequency of files in grid computing have been defined. Furthermore the nodes supposed to be on line during the process of sending. The considered system has two parts. First is the user part which gets requests of users, second one is information part which has the information of current active nodes (e.g. list of contained files and remainder bandwidth).

The process starts by getting requests of clients, and then these requests are sent to information part. After receiving requests, the information part allocates a quote of bandwidth to each file in each node. The allocation procedure is done according to remain bandwidth of nodes and the popularity of files which nodes contain.

As mentioned before, each node should send some parts of files to the buffer of receiver nodes. Actually nodes send some part of each file according to the quotas assigned to them. In the buffers after arriving enough parts user can use them.

Files are comprised of numbers of packets. Size of packets can be the same or not. In this paper we assumed different size of packets. But all copies of files have the same number and size of packets, therefore allocation can be done centrally, and then the results are sent to nodes to transfer the packets which they have to send.

5. Proposed Model: LAGridMSS

Consider a scenario which clients send their requests to the user part of system. Each request contains the name and some special specification of file such as bit rate. After gathering requests, the user part sends them to the information part. This part calculates the popularity of each file according to requests. *Req* is an array which consists of requests of the clients. It contains name of files and the bit rate which the user demanded before. Clients prefer to get the file with specified bit rate.

To find the popularity of a file such as *i* we used equation (3).

$$P_i = \frac{NReq_i}{NoF_i} \qquad (3)$$

Where $NReq_i$ shows the number of all requests for receiving content with the label of *i* in *Req*, and NoF_i is related to frequency of the content with label *i* in the system. To make it more clear this parameter is the number of content *i* in all over the considered grid system.

The user part which receives clients' requests can wait for a specified duration of time and then sends requests to information part or it can wait until a specified amount of requests. Selecting one of these situations depends on to the distribution of requests and to the number of requests that are gotten in a specified duration of time.

The bandwidth for sending files to receivers' buffer can be determined according to the popularity of files in each node. We used the relation shown in equation (4) for specifying a quote of remainder bandwidth of nodes to each of files it belongs.

$$BW_j^i = \frac{P_i}{\sum_{k-1}^{N_j} P_k} * \beta_j \qquad (4)$$

Where BW_j^i shows the bandwidth which is assigned to file *i*, in node *j*, N_j is the number of files in node *j* and β_j is remainder bandwidth of node *j*.

After this phase, the bandwidth of each node for sending any of included files will be determined. In other words the bit rate for sending requested files in each of nodes is calculated. Take care that popularity of files which no requests have received to get them is zero according to the equation (3). Therefore no bandwidth will be assigned to these files.

[1] *Linear Reward-Penalty*
[2] *Linear Reward epsilon Penalty*
[3] *Linear Reward Inaction*

The files are comprised of some packets with different sizes. As mentioned before the files are distributed in the grid computing. It means that content can be in a variety of nodes in the system. According to our proposed approach, some simplification assumptions have been considered. We have assumed that each content has the same number and size of packets all over the system. This assumption eases the allocation of packets to nodes of grid system.

The packets in each content are shown by *pkt*. Therefore a streaming content can be considered as streaming content = <pkt_1, pkt_2, …, pkt_l>. *l* is the number of packets in the content. Care is taken so that *l* has the same value for a specified content in all of the nodes which include that content.

Suppose that *m* requests have been received for getting *n* files. These requests are arranged in an array by user part of the considered system.

When the bandwidth allocated to demanded files in each node is determined, the approach should assign packets of files to each node to be sent. This phase can be performed randomly. Therefore the assignment will not be the optimal, but if a learning method is used the results may be optimal. Here we have used learning automata for selecting a node to send each packet of files.

As mentioned before, size of packets are not the same and there are different files in each node which may be requested. In previous phase the bandwidth for sending of each file in nodes was calculated. The allocated bandwidth for sending a file has different values in nodes which contains the file. According to this fact some packets are better to be sent by some special nodes as the assigned bandwidth of nodes for sending each file is different. We used learning automata to select the best nodes for sending each packet of content. The number of automata used for each file is related to the size of buffer in receiver node. We used automata as much as number of packets which fill the buffer.

According to previous parts of the paper, each automaton has a set of actions. Here actions of them are the nodes which have the requested content. In fact, a LA selects one of the nodes to send the associated packet to the buffer. As mentioned before the selected action should be evaluated according to the input from environment. For evaluating each action the equation (5) is used.

$$B_j^i(n) = \frac{T_{ideal}^i}{T_j^i} \qquad (5)$$

$$T_j^i = \frac{sizeof(pkt_i)}{BW_j^i} + delay(node_j)$$

$$T_{ideal}^i = \frac{sizeof(pkt_i)}{Req\_BR(File_c)}$$

Where $B_j^i(n)$ related to evaluation factor of the LA assigned to packet *i*, in its *n*'th repetition when node *j* is selected as the sender of pkt_i. This factor is related to T^i_{Idael} and T_j^i. The first is the time needed for sending pkt_i by the requested bit rate which is known as ideal transfer time, and the former one is the time which is required for sending pkt_i when is transferred by node *j*. As can be concluded from these relations, $B_j^i(n)$ has a value less or equal to one.

Each LA selects a node for sending its packet. In the first run of LA probability of all nodes are equal. Then the selected action is evaluated and according to the evaluation results, probability of selected node will be changed. The probability may be decreased or increased, and so the probabilities of other nodes will be increased or decreased.

The automata model will be iterated until the probability of a chosen action in each automaton exceeds 95% [2], or transfer time does not change for much iteration, or the number of iterations reaches a maximum limit. If first or second condition stop the automata, then we say that the model coverages.

6. Simulation

For simulating the mentioned approach, a set of predefined nodes with their specifications as the included files and properties, the remainder of their bandwidth and some requests to the system were used. The following section is about these simulations in detail.

6.1 Simulation Environment Setup

We considered a grid system with predefined number of nodes and we used some assumptions in the model, such as no variability of nodes while sending process. Every node was assumed to have some files and each stream is initially set to have some different sized packets in the interval [20, …, Maxpacket], with Maxpacket less or equal to 10000.

Number of nodes, the files in nodes, all requests of the system, frequency of files in grid system and the properties of demanded files are produced randomly. Transfer rate of each node is produced by uniform distribution with average of [10^2 kb/s,10^6 kb/s].

To accurate the results of our simulation, we test it by using a variety of parameters of learning automata algorithm. These parameters are reward parameter and penalty parameter. To illustrate some results of the simulation, we have given maxpackets = 10000, reward parameter is randomly generated between 0.2 and 0.7 and penalty parameter is varied from 0.05 to 0.3. According to the results of simulation the best observed value of reward parameter can be 0.6 and for the penalty one is 0.1 (figure 2).

In the following depicted figures the horizontal ax is the number of LA algorithm iterations before convergence. Care is taken that each repetition in ax can be a composition of several repetitions. As mentioned before the termination condition of automata as no change in the value of transfer time can be claimed as convergence of the algorithm. The vertical ax shows the average time needed for transferring demanded files to buffers.

In figure 3, we suppose reward parameter with the value of 0.2 and the penalty as 0.3. The automata need more iteration to be converged in comparison with the previous run in figure 2. In figure 2 the convergence occurred much earlier than figure 3.

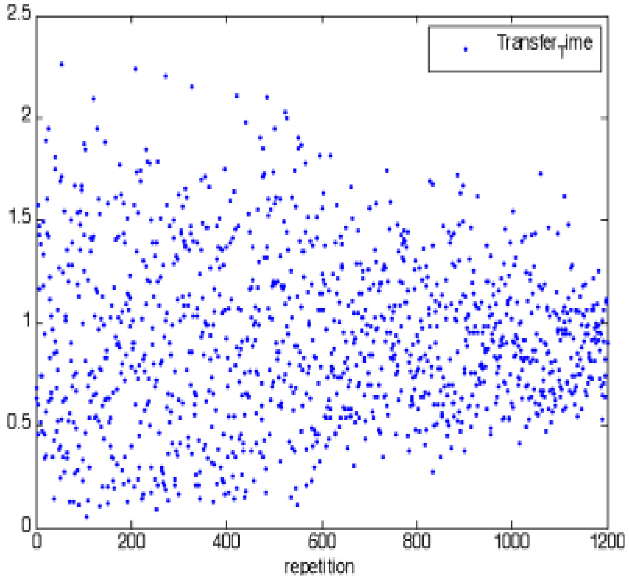


Figure 2. Conergence of Learning Automata Algorithm with Reward Parameter of 0.6 and Reward Parameter of 0.1

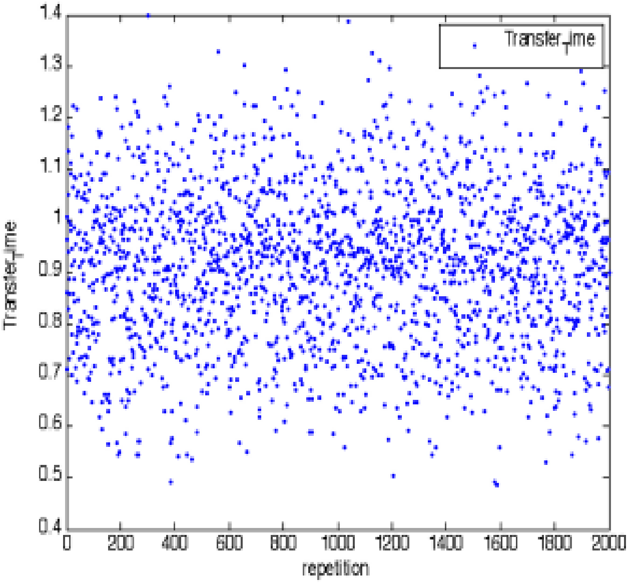
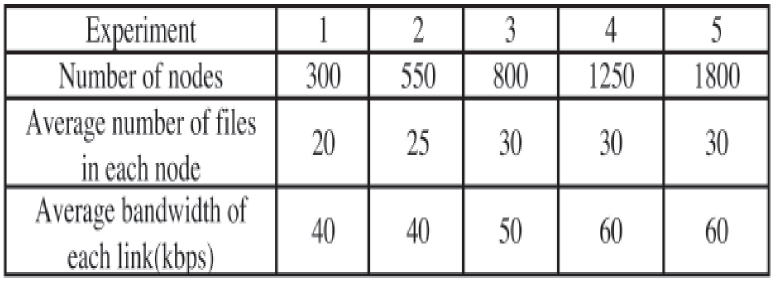


Figure 3. Automata Convergence with Reward Parameter of 0.2 and Reward Parameter of 0.3

6.2 Detection Simulation Result

We performed our experiments with an event-driven simulator coded in C. The simulator models the grid environment, the nodes, communication channels and files in the system. After determining the accurate values of required parameters of proposed approach, we compare it with some other previous methods of streaming files. One of these algorithms is MSS which is introduced in previous sections. Another one is GridMSS which is the authors' previous investigation results on streams retrieval in grid system.

In the obtained results of our simulations, we use LAGridMSS for the proposed approach in this research. Table 1 shows the parameters of simulated experiments.



Experiment	1	2	3	4	5
Number of nodes	300	550	800	1250	1800
Average number of files in each node	20	25	30	30	30
Average bandwidth of each link(kbps)	40	40	50	60	60

Table 1. Main parameters used in some simulated experiments

In figure 4 the comparison results of these methods is depicted. The vertical ax is the duration of time needed for transferring the demanded files by entered requests to the system and the horizontal one shows the number of runs with some variations in specification of nodes or requested files.

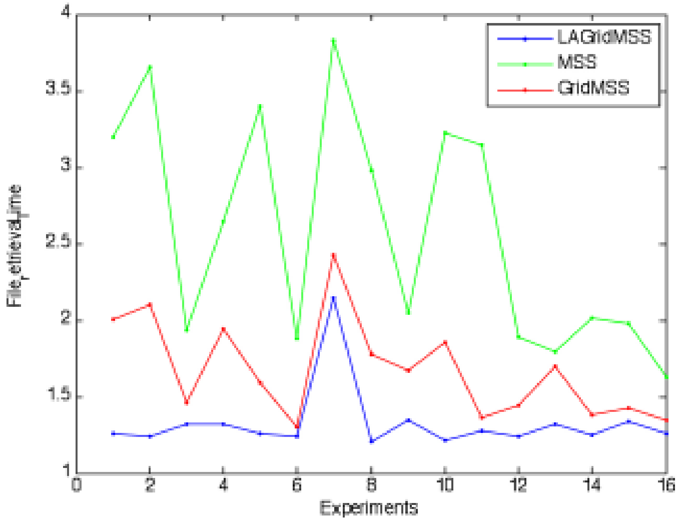


Figure 4. Retrieval time needed in MSS, GridMSS and LAGridMSS

In this comparison depicted in figure 4, we run the simulation 16 times for different values of number of nodes, size and distribution of files, number of requests and bit rates. The performance of LA based approach was much better than others.

7. Conclusion

This research proposed an intelligent streams retrieval method in grid environments. In first phase the appropriate bandwidth for each file, in nodes is considered using the popularity of files in grid system. There is a LA for each packet of files. These LA selects a node for transmitting the packet and then tries to improve its next selection. Transfer time of packets in comparison to ideal transfer time is the evaluation factor of selections. According to the results of evaluations the probability for selecting each node is changed. Finally, when the automata converge, the probabilities of nodes are calculated. Simulations of the model showed the decreased delay in retrieving requested files in comparison to existing models such as MSS and GridMSS.

References

[1] Ghasemi, S., Rahmani, A.M., Mohsenzade, M. (2008). An agent-based framework for supporting QoS in grid computing environment, ICCEE, Thialand, IEEE.

[2] Ardizzone, E., Gatani, L., Cascia, M.L., Re M., Ortolani M. (2007). Distributed Multimedia Digital Libraries on Peer-to-Peer Networks, 14th International Conference of Image Analysis and Processing, IEEE.

[3] Itaya, S., Hayashibara, N., Enokido, T., Takizawa, M. (2005). Scalable Peer-to-Peer Multimedia Streaming Model in Heterogeneous Networks, Proceedings of the Seventh IEEE International Symposium on Multimedia, IEEE.

[4] Itaya, S., Hayashibara, N., Enokido, T., Takizawa, M. (2006). Distributed Multimedia Streaming Systems in Peer-to-Peer Overlay Networks, Proceedings of the Eighth IEEE International Symposium on Multimedia, IEEE.

[5] Itaya, S., Enokido, T., Takizawa M., Yamada, A. (2005). A Scalable Multimedia Streaming Model Based-on Multi-source Streaming Concept, Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems, IEEE.

[6] Goudarzi Nemati, A., Takizawa, M. (2008). Application Level QoS in Multimedia Peer-to-Peer (P2P) Networks, IEEE, WAINA, p319-324.

[7] Itaya, S., Hayashibara, N., Enokido, T., Takizawa, M. (2006). Distributed Coordination Protocols to Realize Scalable Multimedia Streaming in Peer-to-Peer Overlay Networks, Proceedings of the 2006 International Conference on Parallel Processing, IEEE.

[8] Itaya, S., Hayashibara, N., Enokido, T., Takizawa, M. (2006). Distributed Coordination for Scalable Multimedia Streaming Model, Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops, IEEE.

[9] Bertino, E., Catarci, T., Elmagarmid, T. K. (2003). Quality of Service Specification in Video Databases, Published by the IEEE Computer Society.

[10] Tang, J. (2005). An Agent-based Peer-to-Peer Grid Computing Architecture, A thesis submitted in fulfillment of the master, university of Wollongong.

[11] Li, S. (1997). Quality of service control for distributed multimedia systems, a thesis submitted to the faculty of Purdue University.

[12] Lu, S., Lyu, M.R. (2006). Constructing Robust and Resilient Framework for Cooperative Video Streaming, ICME confrance, IEEE.

[13] Amoretti, M., Conte, G., Reggiani, M., Zanichelli, F. (2004). Designing Grid Services for Multimedia Streaming in an E-learning Environment, Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE.

[14] Shi, Z., Yu, T., Liu, L. (2004). MG-QoS: QoS-Based Resource Discovery in Manufacturing Grid, Springer-Verlag Berlin Heidelberg.

[15] Guedes, L.A., Oliveira, P.C. (1997). QoS Agency: An Agent-based Architecture for Supporting Quality of Service in Distributed Multimedia Systems, Proceedings of the IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking , IEEE.

[16] Kerherve, B. (1994). On Distributed Multimedia Presentational Applications: Functional and Computational Architecture and QOS Negotiation, Proc. 4th lnt'l Workshop on Protocols for High-speed Networks, Chapman & Hall, London, pp. 1-1 7.

[17] Zhang, M. (2005). Gridmedia: A Multi-Sender Based Peer-to-Peer Multicast System for Video Streaming, IEEE.

[18] Mine, T., Matsuno, D., Takaki, K., Amamiya, M. (2004). Agent Community based Peer-to-Peer Information Retrieval, AAMAS'04, July 19-23, 2004, New York, New York, USA. ACM.

[19] Padmanabhan ,V.N., Wang, H. J., Chou, P. A. (2002). Distributing Streaming Media Content Using Cooperative Networking, NOSSDAV'02, May 12-14, Miami, Florida, USA.