Intelligent Mobile Agents for Distributed Multimedia Synchronization

王英宏
淡江大學資訊工程系
台北縣淡水鎮英專路
151 號 E-mail:
Inhon@mail.tku.edu.t

莊淇銘
淡江大學資訊工程系
台北縣淡水鎮英專路
151 號 E-mail:
cmchung@cs.tku.edu.t
w

林宏儒
淡江大學資訊工程系
台北縣淡水鎮英專路
151 號 E-mail:
g7190063@tkgis.tku.ed
u.tw

莊智傑
淡江大學資訊工程系
台北縣淡水鎮英專路
151 號 E-mail:
g6190058@tkgis.tku.ed
u.tw

Abstract

With the developing of the distributed multimedia system, there are many researches about how to reduce the network traffic and how to initialize and gather the distributed multimedia resources. In this paper, we propose a robust system using Intelligent Mobile Agent s to construct an automatic and adaptation mechanism for distributed multimedia synchronization and its adaptive Quality of Service (QoS), which is the Distributed Multimedia Synchronization Agent (DMSAgent) system. The DMSAgent system is proposed to improve the efficiency of distributed multimedia network, and to reduce the frequencies of handshaking between client and server by using Intelligent and Mobile Agent technology.

Keywords: *Distributed Multimedia, Intelligent Mobil Agent, Distributed Multimedia Synchronization*

1 Introduction

There are many synchronization models or systems, which were proposed to deal with the synchronization of the distributed multimedia objects and their Quality of Service [5,6,7,8,9,10]. But, almost those models or systems contain a lot of handshaking messages between client and server to tune the flow rate of the multimedia streams or the qualit of service for synchronizing the distributed multimedia objects. Furthermore, in those models or systems, the adaptive mechanisms are based on the network traffic, the utilization of the client-side buffer or the efficiency of the server. In **DMSAgen** system, an intelligent agent is used t learn about the usage of users and to represent it in his knowledge base for initializing and pre-loading the multimedia objets. A **monitor agent** is used to monitor the efficiency of the client and the utilization of the client buffer to adjust the quality of service automatically by messaging the server agents. A mobile agent who contains the scenario migrates among distributed multimedia servers to handshake with the server agents to gather the multimedia objects.

To represent the knowledge base of intelligent agents, we introduce a special language *Knowledge Interchange Format (KIF* [4] and a *Knowledge Query and Manipulation Language (KQML)* [3] for communicatin among agents in the Intelligent Mobile DMSAgent system. A Prioritized Object Composed Petri Net (P-Nets) will be introduced for synchronizing the distributed multimedia objects and representing the scenario of the application.

This paper is organized as follows. In section 2, we explain some technicality about the intelligent mobile agent, knowledge representation and the distributed synchronization model. In section 3, we introduce the intelligent mobile agents of DMSAgent system amply and respectively. In section 4, a DMSAgent system is proposed for distributed multimedia synchronization and its quality of service. Section 5 gives the conclusion and future work.

2 Related Work

A mobile agent who has the ability to migrate is not onl bounded on a context of a system. It executes in its context and can be dispatched to another one over the wire-line or wireless network. The mobile agents contain two separate and distinct concepts: mobility and agency [1]. The Aglet Application Programming Interface (API) which was proposed by Lange and Osima [2] offers a mobile agent system and their functions to create a mobile agent system. There are many functions within a mobile agent system t support mobility, messaging and its lift cycle

- Create a agent
- Clone a agent
- Dispatching a agent
- Retraction a agent
- Activation and Deactivation a agent
- Disposal of a agent
- Messaging to a agent

An intelligent agent has three major parts: knowledge representation, reasoning system and learning system. The intelligent agents of this paper presented contain two knowledge bases; one is the user’ s behavior knowledge base and the other is the scenario knowledge base. A multimedia synchronization model and a Knowledge Interchange Format (KIF) [3] are proposed to represent those two knowledge bases. In DMSAgent system, an agent must communicate with others to exchange their knowledge and inform the other agents. To communicate among agents, a *Knowledge Query and Manipulation Language* (KQML) [4] is proposed to communicate other agents to share their knowledge.

2.1 Knowledge Interchange Format (KIF)

Knowledge Interchange Format (KIF) [3] is a language designed for the interchange of knowledge among heterogeneous computer environment. KIF is not a language designed for interacting between human and computer. KIF is designed to define a common language when the computer system wants to interact with another. Different computer systems and their applications have their own appropriate languages to represent their knowledge bases. But, if they want to interchange their knowledge to another one, they must write out their knowledge base to KIF and other computer system, or application may read the knowledge represented in KIF and convert the language to its own internal form. For communicating among agents, the KIF is not enough to specify for carrying their knowledge to another agent. So, a *Knowledge Query and Manipulation Language* (KQM) [4] is proposed to communicate other agents to share their knowledg .

2.2 Knowledge Query and Manipulation Language (KQML)

As the KIF, the Knowledge Query and Manipulation Language (KQML) [4] is a formal language for specifying the communication format in DMSAg ent system between agents. KQML is a language for programs to use to communicate about the information, such as messaging, requesting and the knowledge base. In intelligent multiple agent system, the KQML often contains a sub-expression, which is scripted by another language to represent the knowledge base and to share it to another agent. But, in DMSAgent system, if an agent just tries to send a message to another agent, a simple message passing mechanism is used to replace the KQML for lowering the syste overhead.

2.3 Prioritized Object Composition Petri-Net (P-Nets)

In DMSAgent system, the distributed Prioritized Object Composition Petri-Net (P-Nets) [8] is used to represent the knowledge of scenario in a formal method. The P -Nets is a synchronization model extended from Object Composed Petri Net (OCPN) [5,6] (shown as Figure 1) and the extended OCPN (XOCPN) [5,7] (see Figure 2). Furthermore, the P-Nets can be extended from the Distributed OCPN (DOCPN) [8] for synchronizing distributed objects in distributed multimedia systems.

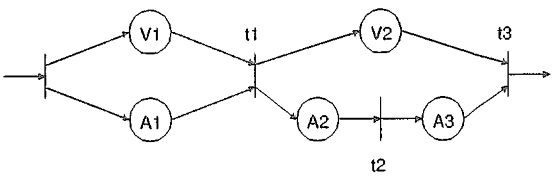

Figure 1: The example of OCPN

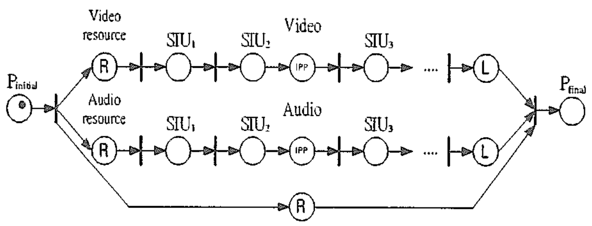

Figure 2: The example of XOCPN

Definition (P-Nets): A Prioritized Object Composed Petri Net (P-Nets) is defined by the quintuple P-Nets = (P, T, I, I_p, O), where

$P = \{P_1, P_2, \ldots, P_n\}$ is a finite set of places, $n \geq 0$.

$T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transactions $m \geq 0$. The set of place and the set of transactions are disjoint, $P \cap T = \emptyset$

I: $T \rightarrow P^{\infty}$ is the input function, a mapping from transactions to bags of places.

I_p: $T \rightarrow P^{\infty}$ is the prioritized input function, a mapping from transactions to bags of places.

O: $T \rightarrow P^{\infty}$ is the output function, a mapping from transactions to bags of places.

Fire rules: A transaction can fire until all of the non-prioritized input events are complete. If a transaction contains a prioritized input event and other non-prioritized input events, the transaction could fire immediately when the prioritized input event is complete without waiting for the other non -prioritized input events. If a place contains prioritized event and non-prioritized output events currently, then the place would fire the transaction with the prioritized output event only and the transaction with non-prioritized input arc would not be fired [8] (shown as Figure 3 for example).

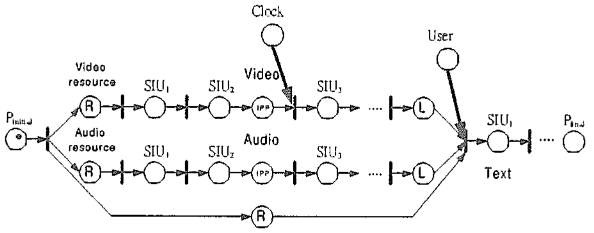

Figure 3: The example of P-Nets

The P-Nets could model the high -interactive distributed multimedia system. In a high-interactive multimedia system, the multimedia scenario may be paused or stopped by users at any time. A higher prioritized event can be used to model this situation. Furthermore, the priority of agent' event is higher than the multimedia objects for dealing with the user interaction and the quality of service.

3 Intelligent Mobile DMSAgents

In this section, we describe all kinds of agents in Intelligent Mobile DMSAgent system particularly. There are four different agents in DMSAgent system: Client-side Intelligent Monitor Agent (CIMA), Server -side Monitor Agent (SMA), Multimedia Resource Management Agent (MRMA) and Intelligent Mobile Multimedia Synchronization Agent (IMMSA).

3.1 Client-side Intelligent Monitor Agent (CIMA)

A Client-side Intelligent Monitoring Agent (CIMA) contains a software sensor module to sense the performance and efficiency of the client-side environment, and monitors the buffer utilizati on of Multimedia presentation. Furthermore, The CIMA will send the information to the Intelligent Mobile Multimedia Synchronization Agent (IMMSA) and the information will be used to tune the Quality of Service (QoS) of the multimedia objects transfer to perform spatial and temporal adaptation.

In addition, the CIMA is also the manager of the multimedia object buffers. To smooth the presentation of temporal multimedia objects, a local memory space has been used to "buffer" the packets of the multimedia objects. According to the flow rate of the network environment, the "pre-loading" buffer size of the multimedia packets is a issue. A low pre-loading buffer size might lead the break of the multimedia presentation, but a high pre-loading buffer size might grow the waiting time of the user and the multimedia presentation might be canceled by user in high interactive multimedia system (See Figure 4).

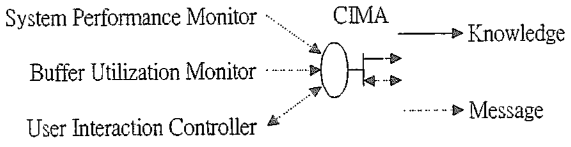
System Performance Monitor
CIMA
Knowledge
Buffer Utilization Monitor
Message
User Interaction Controller

Figure 4: Client-side Monitor Agent

3.2 Server-side Monitor Agent (SMA)

As the CIMA, Server-side Monitor Agent (SMA) (see Figure 5) contains a software sensor module to sense the performance and efficiency of the server-side environment, and schedules the request of the Multimedia Resource Management Agent (MRMA). Specificaly, the SM manages the hardware and multimedia objects resource of the multimedia server where it executes. To schedule the multimedia objects, the SMA must perform a critical section and two -phase lock mechanism to prevent the deadlock situation and the multimedia resource consistency. A *distributed operating system* or *distributed multimedia database* could be used to manage the multimedia resources and the SMA may only pay attention to manage the scheduling of multimedia objects.

In DMSAgent system, a multimedia server could serve the client application by constructing a SMA to register t MRMA, and if the multimedia server crashed, it must re-register to the MRMA when it recuperated from crash.

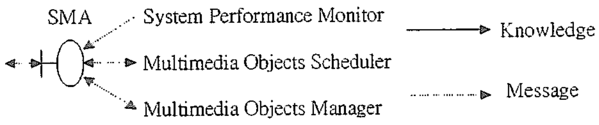
SMA
System Performance Monitor
Knowledge
Multimedia Objects Scheduler
Message
Multimedia Objects Manager

Figure 5: Server-side Monitor Agent

3.3 Multimedia Resource Management Agent (MRMA)

As the DMSAgent system, the multimedia servers are full distributed and the same multimedia object may have man redundant copies in many distributed multimedia servers. The IMMSA may just know the only one server of the whole system, which is called "virtual multimedia server" and may not need to know where it could gather the multimedia objects completely from distributed multimedia server groups. In addition, the IMMSA also does not need to know what the multimedia server, which contains the multimedia objects of the IMMSA requested, is the most efficient. The Multimedia Resource Management Agent (MRMA) (see Figure 6) is a high-level distributed server resources manager. It hides the information of the location of multimedia resources and leads the IMMSA to where it can "gather" the multimedia objects that it needed from the distributed multimedia servers. The property of "location transparency" in DMSAgent system is provided b cooperation of MRMAs. Furthermore, for constructing a robust and fault-tolerance system, the MRMA provides a service-redirecting and best-effort mechanism when one of the multimedia servers crashed.

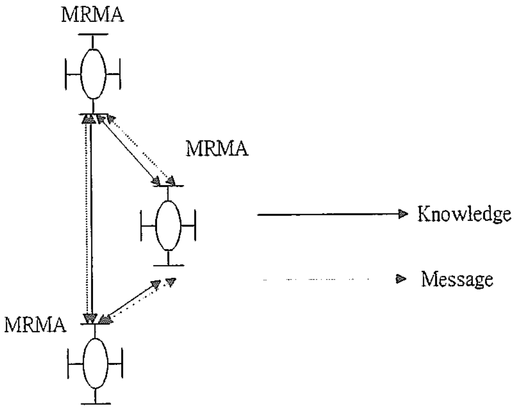

Figure 6: Communication of the MRMAs

3.4 Intelligent Mobile Multimedia Synchronization Agent (IMMSA)

The Intelligent Mobile Multimedia Synchronization Agent (IMMSA) (see Figure 7) contains the knowledge base about the user's behavior and the multimedia scenario. The knowledge base that the IMMSA contained is provided b CIMA. And the CIMA dispatches the IMMSA to "gather" the multimedia objects, which are demanded by the client application. The IMMSA was dispatched to one of the multimedia servers that are most closed to the client syste and the IMMS tells the MRMA the information about what multimedia objects it needs, when they must be dispatched, and what quality of service of the multimedia object is. The distributed Prioritized Object Composition Petri Net (P-Nets) [8] is the distributed multimedia synchronization model of the DMSAgent system . The user's behavior is represented b a KIF-based knowledge base and the agent communicates with other agents b KQML and default messaging mechanism.

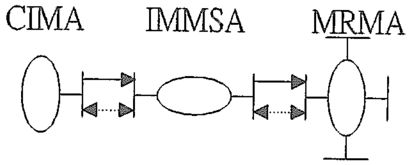

Figure 7: Communication between CIMA, IMMSA and MRMA

4 DMSAgent System

In DMSAgent system, there are three knowledge bases: the user usage base, the distributed synchronization base, and the distributed multimedia objects base. The user usage base is for representing the usage of user, and it also records the situation of the distributed multimedia servers group by communicating with MRMA. The distributed multimedia synchronization base is for representing the distributed multimedia synchronization by using P-Nets. And the distributed multimedia objects base is for recording the information of the multimedia objects. Furthermore, two formal languages: KIF and KQML have been used in DMSAgent system for representin knowledge and interchanging the knowledge among agents. Cooperation among agents provides an automatic and adaptive mechanism for distributed multimedia synchronization and qualit y of service without resource misspending.

4.1 The knowledge base of the user's usage

A user's usage knowledge base is built for recording the quality of service and the pre-loading time of multimedia scenario. The CIMA could use the knowledge to specify the quality of service to the IMMSA automatically when the user requests a multimedia objects service. A user's usage knowledge base is built using KIF language for interchanging the knowledge to another agents.

4.2 The distributed multimedia synchronization knowledge base

The distributed multimedia synchronization knowledge base is built for the multimedia scenario and the quality of service specification. The synchronization knowledge base is built from authors or programmer s by using P-Nets [8] model (shown as Figure 8), and it would be transferred t the IMMSA and MRMA for requesting the multimedia service and specifying its quality of service by cooperatin among IMMSA and MRMA.

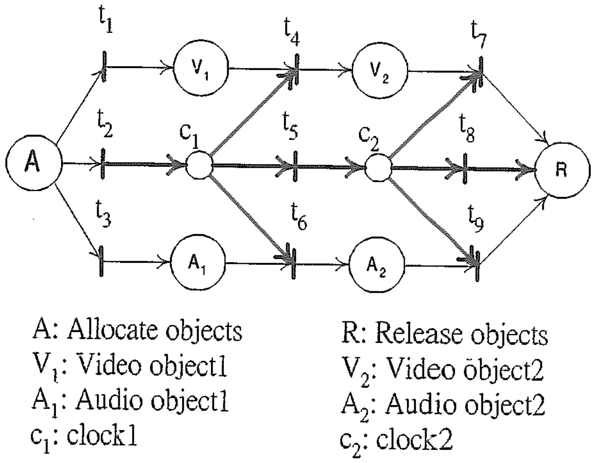

Figure 8: P-Nets Multimedia Objects Scenario

4.3 Communication and Knowledge Interchange

There are many agent-communication systems have been proposed for multimedia or agent communication service [11,12,13,14,15]. In this paper, we focus on the specification of the communication among agents. The specification of communication and knowledge interchange in DMSAgent system follows the KQML [4] and KIF [3]. An agent who wants to transfer its knowledge to another could specify the KQML that contains its knowledge. For example, the sentence shown below specifies the IMMS sending its P-Nets to MRSA by negotiating QoS. The strin IMMSA-01 is generated by system automatically for a unique reply ID.

(ask-one

:language P-Nets

:contain (

$P=\{A, R, c_1, c_2, V_1, V_2, A_1, A_2\}$,

$T=\{t_1, t_2, t3, t_4, t_5, t_6, t_7, t_8, t_9\}$,

$I(t_1)=\{A\}, O(t_1)=\{V_1\}$,

$I(t_2)=\{A\}, O(t_2)=\{c_1\}$,

$I(t_3)=\{A\}, O(t_3)=\{A_1\}$,

$I(t_4)=\{V_1\}, I_p(t_4)=\{c_1\}, O(t_4)=\{V_2\}$,

$I_p(t_5)=\{c_1\}, O(t_5)=\{c_2\}$,

$I(t_6)=\{A_1\}, I_p(t_6)=\{c_1\}, O(t_6)=\{A_2\}$,

$I(t_7)=\{V_2\}, I_p(t_7)=\{c_2\}, O(t_7)=\{R\}$,

$I_p(t_8)=\{c_2\}, O(t_8)=\{R\}$,

$I(t_9)=\{A_2\}, I_p(t_9)=\{c_2\}, O(t_9)=\{R\}$)

sender IMMS

:receiver MRMA

reply-with IMMS -01).

The MRMA may reply that it agrees the request of the specified multimedia synchronization and QoS by IMMSA:

(repl

:language P-Nets

:contain . (Agreement)

:sender MRMA

:receiver IMMS

in-reply-to IMMS -01).

4.4 QoS negotiation

There are many researches about the QoS negotiation and they were proposed by using agent-based mechanism [14,15,16]. In this paper, we propose an automatic and intelligent QoS negotiation by cooperating among agents (see Figure 9). The IMMSA carries the synchronization knowledge and sends it to the MRMA; the MRM schedules the request by reasoning the synchronization knowledge with others that requested from the other IMMSAs. The MRMA would reference the syste performance, which is monitored by SMA, and negotiate with the requesting IMMSA. MRMA is a long-term scheduler and coordinator who manages more than one multimedia server by communicating with SMAs and other MRMAs in DMSAgent system. Unlike MRMA, the SMA is a short-term scheduler and monitor of a single multimedia server in DMSAgent system. In addition, the SMA is a system-dependent agent in a heterogeneous environment.

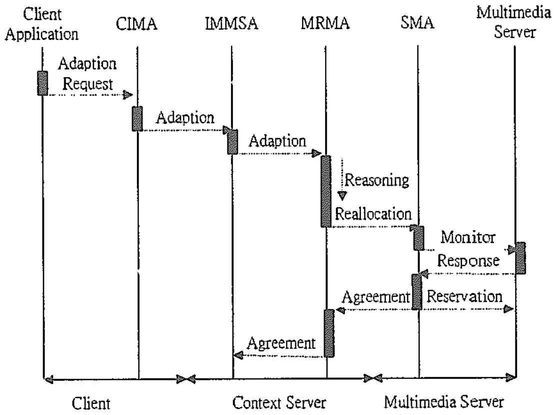

Figure 9: QoS negotiation by cooperating among agents

4.5 Robustness and Fault-Tolerance

To construct a robust and fault-tolerance infrastructure is a key issue in a distributed multimedia system [17,18,19]. In DMSAgent system, we propose a best-effort mechanism for constructing a robust and fault-tolerance system by cooperating among agents.

The MRMA will trigger the redirection and the adaptation when it greets the SMA or the multimedia server crashed. In addition, MRMA will inform the IMMSA to clone and dispatch it to another MRMA that contains the copy of the object in the crashed multimedia server, and the clone-IMMSA will request a continuation of the multimedia object for negotiating with the new MRMA. The IMMSA would know about the new efficient MRM by informed from the original MRMA. For constructing the robust mechanism, the communication among MRMAs would also interchange the knowledge about the multimedia resources and system efficiency (shown Figure 10).

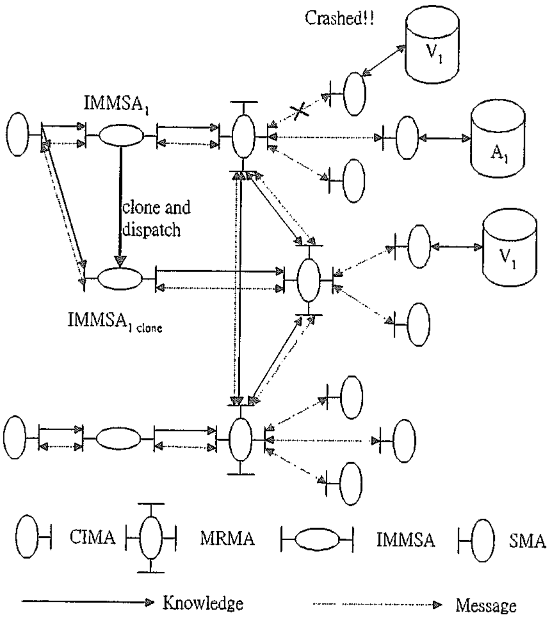

Figure 10: Robustness and Fault-tolerance

5 Conclusions and Future work

This paper proposes an agent -oriented distributed multimedia system and its distributed multimedia synchronization mechanism. With the communication and handshaking among intelligent mobile agents, we improve the system efficiency and reduce the network traffic by lowering the resource misspent and automatic agenc y.

A best-effort automatic agency in DMSAgent system that we proposed does not guarantee the Quality of Service. A resource ReSerVation Protocol (RSVP) was proposed for guaranteeing the QoS and the real -time service in software layer. Unfortunately, the so ftware-layer protocol is not full supported by router s or other network resources. An agent-based automatic and adaptive " *guaranteed*' QoS mechanism will be the key issue of the research in DMSAgent system.

References

[1] Vu Anh Pham and Ahmed Karmouch, "Mobil Software Agents: An Overview ", IEEE Communication Magazine, July 1998, pp26-37.

[2] Lange, D. and Oshima, M., "Mobile Agents with Java: The Agent API", World Wide Web, 1(3), September 1998.

[3] M.R. Genesereth, "Knowledge Interchange Format draft proposed American National Standard (dpANS)", NCITS.T2/98-004, Comp. Sci. Dept., Stanford Univ.

[4] Tim Finin and Jay Weber, "DRAFT Specification of the KQML Agent-Communication Language", Th DARPA Knowledge Sharing Initiative External Interfaces Working Group.

[5] Y.Y. Al-Salqan and C.K. Chang, "Temporal Relations and Synchronization Agents", IEEE Multimedi Summer 1996Vol. 3, No. 2, pp30 -39.

[6] T.D.C. Little and A. Ghaffoor, "Synchronization and Storage Models for Multimedia Objects", IEEE J Selected Areas Comm., pp. 413 -427, Apr. 1990.

[7] M. Woo, N.U. Qazi, and A. Ghafoor, " Synchronization Framework for Communication of Pre-orchestrated Multimedia Information", IEEE Network, pp.52-61, Jan./Feb. 1994.

[8] Sheng-Uei. Guan, Hsiao-Yeh Yu, and Jen-Shun Yang, "A Prioritized Petri Net m odel and Its Application i Distributed Multimedia System", IEEE Transaction on Computers, Vol. 47, No. 4, April 1998.

[9] Y.Y. Al-Salqan, C.K. Chang, and Y.V. Reddy "MediaWave: On Distributed Multimedia Synchronization", Proc. 2nd IEEE Int'l Conf. on Multimedia Computing and Systems, May 1995, p 150-158

[10] R. Steinmetz, "Multimedia Synchronization Techniques: Experience Based on Different System Structures", Proc. IEEEE Multimedia, 1992, pp 306-314.

[11] W. Kwan and A. Karmouch, "Multimedia agents in a distributed broadband environment", ICC '96, Vol. 2, pp 1123-1127, 1996

[12] R. Kishimoto and N. Sakurai, "Access transport

network system for multimedia communication services", ICC '93,Vol. 2 , pp 656 -660, 1993

[13] Kishimoto R. and Sakurai N., "Agent co mmunication system for multimedia communication services", INFOCOM '96.,Vol. 1, pp 10 -17, 1996

[14] L.A. Guedes, P.C. Oliveira, L.F. Faina and E. Cardozo, "QoS agency: an agent-based architecture for supporting quality of service in distributed multimedia systems", Protocols for Multimedia Systems - Multimedia Networking, pp204-212, 1997

[15] H.de Meer, A. Puliafito, J.P Richter, O. Tomarchio, "QoS Adaptation by Software Agents in the Presence of Defective Reservation Mechanisms in the Internet", Third IEEE Symposium on ISCC '98, pp488-492, 1998

[16] M.T. Kone and T. Nakajima, "An architecture for a QOS-based mobile agent system", Fifth International Conference on Real-Time Computing Systems and Applications, pp145 –148, 1998.

[17] S.M. Deen, "A fault-tolerant cooperative distributed system", The Ninth International Workshop on Database and Expert Systems Applications, pp508-513, 1998.

[18] N.M. Karnik and A.R. Tripathi, "Design issues in mobile agent programming systems", IEEE Concurrency, vol. 6 3, pp52 –61, July-Sept 1998.

[19] R.K. Iyer, Z. Kalbarczyk and S. Bagchi, "Chameleon: adaptive fault tolerance using reliable, mobile agents", The Sixteenth Symposium on Reliable Distributed Systems, pp61 –62, 1997.