A MULTIAGENT SYSTEM ARCHITECTURE FOR QoS MANAGEMENT IN DISTRIBUTED MULTIMEDIA SYSTEMS

Mamadou Tadiou Kone and Tatsuo Nakajima

School of Information Science,

Japan Advanced Institute of Science and Technology

{mamadou,tatsuo}@jaist.ac.jp

Abstract The performance of computer systems relies on several factors. In particular, Quality of Service (QoS) research encompasses all the aspects of the management and delivery of a sound service. As fluctuations and degradations are inherent part of any heterogeneous system, QoS maintenance and adaptation become crucial parts of end-to-end QoS provision.

We propose in this paper a QoS adaptation scheme based on the *mobile agent technology*. We introduce the idea of *macro-adaptation* as the coarse-grained adjustments, and *micro-adaptation* that describes all the fine-grained corrective actions done on resources. Here agents are organized in a hierarchy along the components of the distributed multimedia system. Our idea in this model is to structure in an integrated and hierarchical fashion an adaptation strategy that span all the configurations made of components selected by a QoS manager. Computation is adapted to the available services or moved to other components if necessary. The interactions between the mobile agents and the components interfaces incorporate the adaptation mechanisms.

Keywords: Multiagents, Mobile Agents, Quality of Service (QoS), QoS management, Distributed Multimedia Systems.

1. INTRODUCTION

The improvement of general computing systems performance and services has long been a common goal for many research areas. Specially, modern computing systems need to assure end-users about the quality of the service being provided. Nowadays, a number of applications like Video-on-Demand (VoD), Desktop Video Conferencing, Distributed Col-

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35581-8_35](https://doi.org/10.1007/978-0-387-35581-8_35)

laborative Environments, Distant Learning are taking advantage of the new capabilities offered by performant networks and end-systems. However, these applications will not be able to provide satisfactory level of service if a suitable QoS provision scheme is not designed to sustain their activities.

Originally, the principles of Quality of Service (QoS) were applied to computer network communications. As QoS is gaining wide acceptance, its mechanisms are being applied to end-systems in distributed multimedia systems (DMS) as well. The concept of QoS is best described through the levels of service, a given provider can offer in order to sustain an activity. QoS-driven systems must deal appropriately with failure or changing conditions inherent to their environment in order to keep user confidence. As a result, **QoS adaptation** becomes a crucial part of the whole QoS management process.

We propose the **Mobile Agent** technology to support an efficient QoS adaptation scheme in distributed multimedia environments. Mobile agents are new and attractive alternatives to the traditional client-server computing model. In this scheme, computation and interactions are moved to the location of the resources. This paper is concerned with the adaptation of QoS in distributed multimedia systems (DMS) using the mobile agent technology. A number of works ([4], [7] and [8]) have addressed the issue of adaptation with interesting techniques. Here, we suggest that QoS negotiation, resource reservation and admission control should be planned wisely to prevent significant QoS violation. Our approach based on the flexibility provided by mobile agents aims to honor user requests transparently. Agents, organized in a hierarchy along the components in a distributed multimedia system (DMS), cooperatively maintain the required QoS.

Following this introduction, in section 2, we introduce mobile agents and explain the rationale behind our choice in QoS adaptation. In section 3, we give a description of the mechanisms of QoS provision for distributed multimedia systems. We present in section 4 some noteworthy approaches to QoS adaptation and draw some comparison lines. Our integrated and hierarchical QoS adaptation strategy through the mobile agent technology is explained in section 5. Finally, we state concluding remarks and hints to further directions of research in section 6.

2. AGENTS: WHERE THEY FIT

2.1 Nature of Agents

There are two perspectives in defining the word *agent*. There is the software engineering perspective and the cognitive science perspective.

The first refers to a piece of software that can migrate autonomously inside a network while the second defines an entity or a set of entities called *multiagent system* that has capabilities, provides services and can act on their environment.

2.2 Mobile agents in DMS

A mobile agent is a program that can migrate autonomously across a heterogeneous network and perform tasks on behalf of its owner as shown in figure 1. At anytime it can halt, move together with its state and data to a new machine, and resume execution at the point where it stopped. This ability of agents to migrate from host to host makes mobility support a fundamental requirement for agents systems. In addition, the introduction of mobile code in local or open network like the internet raises several security issues. Agent and host authentication, authorization, portability and access control are some of these issues.

Figure 1 A new approach to client/server communications

Mobile agents are a new kind of abstraction in the client/server communications world. The mobile agent technology has found many interesting applications in areas where QoS control and management is an important issue like in [10].

In distributed multimedia environments, a component is defined as a physical (peripheral, network node, CPU ...) or processing entity with the ability to produce, transform or consume a multimedia data. A component as shown in figure 2, is characterized by its interface and services it provides. In addition to describing interaction requirements, behaviors and capabilities, interfaces present the set of services a component has to offer.

In place of the previous concepts of object oriented programming, we

use multiagents and mobile agents. With mobile agents come the possibility to move the interaction point, the service access point and computation to the target component side. An agent uses a service by invoking the component interface.

Figure 2 Anatomy of a component in a DMS

2.3 Multiagents systems

Multiagent systems - usually studied in AI - are distributed computing systems composed of several interacting computational entities called agents. These constituent agents have capabilities, provide services, can perceive and act on their environment. In our model, distributed multimedia components are treated as this type of agent.

2.4 Mobile agents and Fault Tolerance

As mobile agents are bound to perform many operations and use services provided at multiple hosts. Some concerns naturally arise about their reliability, their execution environment, the host machine and the network that support them. To deal with these problems we propose a simple scheme for failure recovery. At the time it leaves a host, an agent creates a checkpoint on a persistent store. On regular intervals, the replicated agent sends a message (ping) to check if the remote agent is alive. This way, when the current host crashes together with an agent or a communication failure occurs, the replicated agent at the last host visited recreates the previous state and carry on the task.

3. QUALITY OF SERVICE (QOS)

3.1 QoS Concepts and Management

In a distributed multimedia environment, any activity that contributes to the production, delivery and consumption of some multimedia data can be characterized by the quality of its service. The followings are some well accepted phases of Quality of Service provision borrowed from the network communications area that are used in distributed multimedia environments:

QoS Specification: A service user must specify to the provider with objective and subjective QoS parameters the level of service that it expects.

QoS Negotiation and Mapping: In this process, the system must identify the set of components that are able to provide and sustain completely the required level of service. Then, with adequate mapping, the required QoS are translated from one system layer to another.

Resource Allocation and Admission Control: In conjunction with the negotiation phase, necessary resources are reserved for an activity. The required resources availability could be an issue unless the admission control functions compares beforehand the activity needs with the existing resources. Admission is then granted when enough resources exist.

Monitoring: Once an activity has started, a given system layer constantly monitors the lower layer to check for alteration in the value of the performance parameters.

Maintenance and Adaptation: These processes aim to counteract the perceived degradation or violation of the contracted QoS. Adaptation is achieved through change in the topology of the components selected by the negotiation process, a coarsened or fine grained resource adjustment or an explicit change of service by the user.

Re-negotiation: When a degradation of the QoS level has reached a critical level and nothing further can be done to recover or adapt to a lower level of service, a re-negotiation is initiated by the system. This scheme proposes an entirely new QoS provision process. A change of service preferences by a user also can trigger a re-negotiation. The whole management process is depicted in figure 3.

In addition, the *QOS contract* binds the entities (network provider, network and end-system resources ...) involved in the QoS provision. We define it in generic terms as follows:

```
typedef struct{
    service-spec_t : service-spec ;
    committment_t : committment ;
```

```

monitoring_t : monitoring ;
adaptation_t : adaptability ;
dependability_t : dependability ;    }service-contract_t

```

The contract is composed of the *service specification*, the class of *commitment* the resources are able to provide to sustain a given activity, the *monitoring* process, the *adaptation* process, and a subjective parameter we call *dependability*.

A user states with objective and subjective parameters in the service specification the performance expected. For instance, in dealing with timeliness parameters like loss, delay, jitter and throughput could be considered.

Figure 3 A simple view of QoS Management

The commitment clauses are known as **Deterministic**, **Statistical** and **Best effort**. The first guarantees that resources are exclusively dedicated to an activity. The second is a loose version of the first where a service tolerates some fluctuations because resources are shared and eventually pre-empted. The best effort commitment does not guarantee any level of service nor does it permit control over any resource. These

clauses are qualitative expressions that helps precise the nature of the service required.

4. A NEW ADAPTATION SCHEME

In a distributed multimedia environment, unstability, fluctuations in the level of service or temporary failures are inherent parts of the activity of any system. As a result, service maintenance or adaptation are required if user confidence is important. In our context some particular motivations are the following:

1. In distributed systems environments, shared resources due to system policies (resources management, priorities, ...) may not be available on request.
2. Congestions in the network or end-systems often occur.
3. Different clients QoS requirements, depending on preferences, may create conflicts.
4. Granted resources may be pre-empted and not be guaranteed over time.

In our approach to adaptation, we define **macro-adaptation** as all the coarse-grained adjustments relying on agents movement with change in components topology; whereas **micro-adaptation** describes all the fine-grained corrective actions done at the lower level resources along agents hierarchy. In addition, in this agent-centric design, we assume that QoS monitoring and resource reservation are done at the component level. Our idea is to organize an adaptation strategy where mobile agents move computation - like in figure 6 to components where services are available.

4.1 Agent-based adaptation strategy

How do mobile agent identify their destination before migration? Our scheme models components as a multiagent systems whose constituents can communicate in an appropriate agent communication language like the Knowledge Query and Manipulation Language (KQML). Before the start of a session, all agents (components) in the multiagent systems register their capabilities to a facilitator agent. Then, when a component agent fails to meet its requirements, it sends a message to the facilitator and ask for another agent component that is able to satisfy these requirements. When a suitable component is identified this way, a mobile agent can carry on an appropriate adaptation scheme.

An integrated approach:

The system we propose is integrated in two aspects:

First, all components spanned by related mobile agents, are integrated

in a single end-to-end QoS provision process.

Traditionally, the negotiation process identify a finite set of components that are able to sustain an activity and assign to each, in a rigid configuration, its share in the overall end-to-end QoS provision. In contrast, with the introduction of mobile agents, several components assembled in a loose configuration, are connected dynamically to provide the required QoS. Such an approach is described in [1]. In our scheme, system agents coordinate the operations of system components and communicate with network agents responsible of network transactions at the transport layer of the network as illustrated by figure 4.

Second, all adaptation mechanisms in research related to QoS provision have been implemented as isolated processes activated when a degradation or violation actually occurs. However, unstability, degradation are unavoidable events in heterogeneous environments. The QoS negotiation process, the resource allocation and admission control should incorporate strategies that prevent as much as possible violations of QoS requirements.

Figure 4 Agents structure in end-to-end QoS provision

A hierarchical approach:

When a degradation or violation is detected, maintenance or adaptation is carried out either locally or with the support of a remote component along the following steps:

1. The system identifies the component that fails to satisfy the local QoS.
2. The related agent (ex. end-system agent) attempts to solve the problem locally by either:
 - ◊ moving the computation to a peer component (ex. end-system component);
 - ◊ sending a message to a sub-component agent to make the necessary corrections down into the hierarchy or interact with the resource

manager.

◇ sending a message to another agent located elsewhere and the previous steps are performed there.

3. If local adaptation fails, the adaptation process is moved to another part of the system (ex. network) and step 2 is re-enacted.

These steps may be conducted in two ways:

Horizontally, computation is moved with the QoS access point from the current component to a peer component.

Vertically, a message is sent to an agent at a sub-component down in the hierarchy. The target agent can further move the computation to a peer sub-component if necessary. Should the QoS maintenance or adaptation to a lower QoS fail, the end-system agent sends a message to the network agent to carry out a similar strategy.

Figure 5 Agent-based hierarchical adaptation

4.2 Agent-based adaptation mechanisms

Agents interacting with components

The main purpose of our system is to make existing services at a component available on request to agents. In normal situation, the level of the service offered matches the request. Otherwise, a maintenance or adaptation to a lower level is proposed. Upon receiving from the user agent the QoS contract with the specified QoS parameters, an end-

system agent moves to the *agent execution environment* of a component as depicted in figure 5. In this environment, the agent makes the necessary computations, calls the *Application Programmer Interface (API)* in order to use the services.

An example of interface is the Java Database Connectivity Layer and the Java Advanced Windowing Toolkit. Here we assume that during the previous Resource Allocation and Admission Control, resources necessary to sustain the computation have been reserved. When there is a degradation and the level of service is not satisfactory and all attempts to maintain it has failed, the agent has the option to adapt its computation to the available resources. That is, the mobile agent alters a QoS level to a lower level.

Figure 6 Agent-based Adaptation Model

Agents migration:

Agents mobility is supported by the object serialization functions included in the Java language. For the three types of agents (user agent, end-system agent and network agent) migration happens inside their respective domain (end system or network). An agent migrates between peer components of the same domain. A network agent assigned to nodes can migrate only between nodes and so do end-system agents. Often, the operation of a component depends on other sub-components. These sub-components are visited by related agents down in the hierarchy. All computations are identical at this level as at the upper level. At the bottom of the hierarchy, in connection with the resource manager, resource agents participate in fine grained adjustments called micro-adaptation by travelling between locations. An adaptation scenario is explained in table 1.1.

Inter-agents interactions:

Along the hierarchy, mobile agents send messages in a serialized form horizontally to peer agents located in another domain or vertically to agents related to sub-components or resources. The two types of communication are at the heart of agents cooperation for maintenance or adaptation.

QoS Adaptation Strategy	Context	Agent Migration	Agent Interaction
Macro-adaptation	End-system	WS ----> WS NSAP ----> NSAP	Component API Message
	Network	Node ----> Node Router ----> Router	Component API Message
	Peripheral (Video camera, Mic.)	Video Cards ----> Video Cards	Component API Message
Micro-adaptation	End-system	CPU/Memory ----> CPU/ Memory	Resource Manager Message
	Network	Buffer ----> Buffer	Resource Manager Message
	Peripheral	Codec ----> Codec	Resource Manager Message

Table 1.1 Agent-based adaptation scenario

4.3 An example

Let's consider a system that captures images and stores them in a remote multimedia database storage (MMDBS) as illustrated in figure 7. In addition, this data in the remote server is accessible to other users ready to be displayed locally. That is, the whole system can be used in two modes. When a user captures data and stores it remotely. The source is a camera together with a set of workstations linked in a sub-network and the sink is the multimedia database server. Conversely, a user in need of some multimedia data could fetch it from the database server and display it locally.

In the first mode, a QoS manager in its negotiation phase would simply select a configuration made of the following components: the camera, a workstation with a video capture card, a path of nodes and routers in the network and the database storage.

When all the resources allocation and admission tests complete, the service starts. We assume all the workstations in the subnetwork have the same capabilities and resources. The camera physically linked to a given workstation capture images which are compressed, sent accross the network and stored in the database.

In the second mode depicted in figure 8, data is fetched from the remote server. The QoS manager selects a configuration made of the following components: the database server (the source), a path of nodes and routers in the network and a workstation (display sink). In fact, during the QoS negotiation phase, several potential configurations of components and resources were identified and recorded by the QoS manager.

Figure 7 Example system: storing data in a multimedia database

Figure 8 Example system: fetching data from a multimedia database

Here the user agent, system agents and network agents are located respectively in the user application, the end-system and the network. When a degradation or violation occurs, the QoS manager tries to locate and identifies the failing component (source workstation, network node, router or database server). The mobile agent located at the failing component has two alternatives. On one hand, it can send a message to other agents located at sub-components like the operating system, the hard disk, the memory or video capture card. On the other hand, the mobile agent can move the necessary data to another workstation (macro-adaptation) and restart computation where it stopped. As a result, the initial QoS level is maintained or adapted to a lower level. In case violation is detection at network node or router, the corresponding agent migrates with all the data to another node or router. Although the service started with a given configuration of system components, the mobile agents give us the possibility to move computation from one component to another. The configuration changes dynamically to cope with violations.

5. RELATED WORK

A number of works have addressed the problem of QoS adaptation with various techniques in different environments. Few of these ap-

proaches use the agent technology to improve QoS provision.

In [8], a QoS manager conducts QoS adaptation by interacting with static QoS agents bound to components. The QoS adaptation strategy relies on three schemes are. The first called Component Reconfiguration Scheme (CRS) performs adaptation by altering the components topology and initiating a smooth and transparent transition. The second named Resource Reconfiguratio Scheme (RRS), redistributes the level of QoS supported by each component in a given configuration to cope with violation and keep the overall end-to-end QoS.

In [6], the concept of binding as abstraction of communications between components is used to sustain the required QoS through monitoring, adaptation and reconfiguration. Here binding objects, organized in a hierarchy, apply adaptation policies dynamically.

In [4], a hierarchical QoS adaptation based on the organization of binding objects is proposed. In this paper, adaptation is carried out along a hierarchy of bindings. From the top with coarse-grained actions to the bottom with fine-grained atomic resource control, the adaptation mechanisms are embedded in the binding components which are conceived as communications abstractions. In addition, these bindings are responsible for the end-to-end QoS maintenance.

In [10], an agent-based approach is introduced to carry out corrective QoS adaptation in a cloud of non-RSVP-capable network routers. Static agents monitor tunnel properties and interact with RSVP routers to deliver the required QoS.

In the above systems, when agents are used, their functions are limited to monitoring the level of QoS and eventually notifying the system of QoS violations at the host component. Our approach that consists of moving the QoS access point from one component to a peer component with mobile agents is new. The initial configuration need not be altered unless no component is able to compensate the degradation. Adaptation is performed smoothly by mobile agents horizontally or vertically in an integrated fashion. System agents - at the source or sink - and network agents cooperatively manage the end-to-end QoS transparently.

6. CONCLUSION

We have presented a new approach to QoS adaptation that relies on the flexibility of mobile agents. In general QoS provision, a static configuration of system components is provided. To this static configuration, the mobile agent technology allowed us to substitute a dynamic one. The adaptation strategy evolves around the interactions of agents and components interfaces on one hand, and the cooperation between agents

themselves on the other hand. To achieve this, we have introduced macro-adaptation and micro-adaptation to describe the operations of agents that cooperate along a hierarchy. Although designing an adaptation strategy that responds to QoS violation might be desirable, planning wisely a negotiation, resources allocation and admission processes still remain the best way to prevent QoS degradation.

Our model - due to its integrated and hierarchical aspect - could improve the overall QoS provision significantly provided the entire system is reliable. We are planning to include the strategy described in this paper in a complete QoS provision scheme based on mobile agent in a near future.

References

- [1] Mamadou Tadiou Kone and Tatsuo Nakajima, "An Architecture for a QoS-based Mobile Agent System." In *Proceedings of 5th Int. Conf. on Real-Time Computing Systems and Applications*, Hiroshima, Japan, Sept. 1998.
- [2] Hutchison D., Coulson G. , Campbell A. , Blair G., "Quality of Service Management in Distributed Systems." In Sloman M.(ed), *Network and Distributed Systems Management*, Chapter 11, Addison Wesley. Reading, Mass.
- [3] C. Aurrecochea, A. Campbell, and L.Hauw, "A Survey of QoS Architectures." In *ACM Multimedia Systems Journal, Special Issue on QoS Architectures*, 1997.
- [4] Daniel G. Waddington and David Hutchison, "End-to-end QoS Provisioning through Resource Adaptation." In *Proceedings of the 8th IFIP Conference on High Performance Networking - HPN'98*, Vienna University of Technology, Vienna, Austria, September 1998.
- [5] D. Waddington, Christopher Edwards and D. Hutchison "Resource Management for Distributed Multimedia Applications." In *Proceedings of the Second European Conference on Multimedia Applications, Services and Techniques - ECMAST*, Milan, Italy, May 1997; LNCS 1242.
- [6] D.G. Waddington and G. Coulson, "A Distributed Multimedia Component Architecture." In *Proceeding of the first International on EnterpriseDistributed Computing*, Gold Coast, Australia, October 1997.
- [7] S. Fischer, A. Hafid,G., Bochmann,and H. de Meer, "Cooperative QoS Management for Multimedia Applications." In "4th IEEE Int. Conf. on Multimedia Computing and Syst.,"Ottawa, Canada, June 1997, pp. 303-310.
- [8] H. Hafid and G. V. Bochmann, "Quality of Service Adaptation in Distributed Multimedia Applications." In *ACM Multimedia Systems Journal, Vol.6*, 1997.
- [9] Luiz A.G. Oliveira, Paulo C. Oliveira and Eleri Cardozo. "An Agent-Based Approach for Quality of Service Negotiation and Management in distributed systems." In *Mobile Agents, Proceedings of the First International Workshop,MA '97*, Berlin, Germany, April 1997, pp 1-12.
- [10] Hermann de Meer, J-P. Richer, A. Puliafito and O. Tomarchio, "Tunnel Agents for Enhanced Internet QoS." In *IEEE Concurrency*, April-June,1998. pp. 30-39.
- [11] The Object Management Group (OMG), "The Common Object Request Broker: Architecture and Specification." OMG Doc.,<http://www.omg.org/>, Dec. 1995.