

QoS Management By Mobile Agents in Multimedia Communication

S.S.Manvi and P. Venkataram
Electrical Communication Engineering Department,
Indian Institute of Science, Bangalore-560012 INDIA,
E-mail: {sunil,pallapa}@ece.iisc.ernet.in

Abstract

In this paper we propose a mobile agent based QoS management system to satisfy the five functional principles of QoS architecture, i.e., integration, separation, transparency, asynchronous resource management and performance. Mobile agent paradigm is a unique paradigm in contrast to traditional client/server paradigm in a problem inherently distributed and complex, such as QoS management. This paradigm saves considerable amount of bandwidth and reduces network traffic. We present an analytical and simulation models of a mobile agent based bandwidth negotiation and management system for multimedia communication. Response time of the mobile agent is computed with different number of retransmissions and the intermediate "hops" for both the models. It is observed that the response time of the agent increases with increase in number of retransmissions and the hops. We have also showed that the increase in the arrival of mobile agents would affect the admission of new multimedia applications.

keywords: mobile agent, QoS, bandwidth, multimedia communication.

1 Introduction

Distributed Multimedia applications demand real time multimedia communication because they are of isochronous nature. These applications depend on a certain level of quality of service (QoS). They need a mechanism for QoS adaptation in order to deal with temporary changes in the available QoS parameters. Such applications require a QoS negotiation and management system to provide guaranteed real-time services in multimedia communication. The main functions of QoS management are: QoS negotiation, QoS renegotiation, QoS mapping at different levels, resource reservation, QoS monitoring and QoS adaptation. If negotiation ends with an agreement on the required values, application can be launched and managed in later phases. The types of agreements made could be *best effort*, *stochastic* or *guaranteed*. Most of the networks presently work on "best effort" service without QoS guarantees. Hence, there is a need to design a QoS architecture which provides guaranteed services in multimedia communication.

1.1 QoS requirements in Multimedia Communication

QoS states how valuable the services provided by the multimedia systems are. QoS parameters can be considered at the three levels: *application*, *system(operating system)* and *network level* [1]. The *application or user parameters (up)* consists of media quality descriptions for the specific media characteristics of each device, such as sample size, sample rate and priority. *System parameters (sp)* include CPU power, buffers and secondary storage capacity. The *network parameters (np)* are packet size, packet error rate, end-to-end delay, packet rate, loss rate(reliability), jitter and bandwidth. The QoS of a given system is expressed as a set of (parameter-value) pairs, something called tuple: each parameter is considered as a typed variable whose values can range over a given set for eg., delay:<5sec, 6sec>. Different applications on same distributed systems can have different values required. A QoS mapper is required to map these parameters, i.e., $sp=f(up)$ and $np=f(sp)$.

The generally considered parameters for multimedia communication are: end-to-end delay: the elapsed time between the generation of a service and presentation of service; delay jitter: the variation of end-to-end delay; packet error rate: the percentage of packets discarded due to transmission failure in the path; bandwidth: the transfer bit rate of a service in the path.

1.2 Mobile agents

Agents are the autonomous programs situated within an environment, which senses it and acts upon it using its knowledge base, and learns so as to act in future. Agents are classified as: local or user interface agents, networked agents, distributed AI (Artificial Intelligence) agents and mobile agents based on the attributes they possess [2]. The concept of mobile agents grew out of three earlier technologies: *process migration*, *remote evaluation* and *mobile objects*. All these concepts are an improvement over RPC (remote procedure call) for distributed programming. Mobile agents are the multiagent systems which possess all the attributes specified for an agent. It is an itinerant agent dispatched from source computer which contains program, data, execution state information, migrates from one host to another host in the heterogeneous network

and executes at remote host until they accomplish their task. The mobile code should be platform independent, so that, it can execute at any remote host in the heterogenous network environment. They communicate and cooperate with other agents to achieve its goals. Interagent communication can be achieved by message passing, RPC or common knowledge base(black board)[3].

Mobile agent paradigm significantly reduces bandwidth consumption and network traffic in contrast to client/server paradigm. Mobile agents can be used in network management, telecommunication services management, QoS management and mobile computing. There are certain issues to be resolved in implementation of mobile agents such as agent transfer mechanisms, addressing, exporting agent state information, communication language, secrecy, privacy, agent data transfer, authority and portability and security [3].

1.3 Proposed Mobile Agent Based QoS Management Technique

Mobile agent based QoS management system is proposed to negotiate and manage the the flows of multimedia communication. It consists of mobile agents and static agents which interact with each other through shared knowledge base. Mobile agents are the piece of code written in shell scripts. Every node in the networked multimedia systems must have an agent execution environment consisting of an agent server and shell interpreter and the agents. Agent sever is responsible for receiving the mobile agent, sending agent to shell interpreter for execution, sending the executed mobile code with data to next node in the network. In the proposed scheme, the service user (multimedia client) host negotiates for bandwidth, with the service provider (multimedia server) by sending a bandwidth negotiator mobile agent across the path before launching multimedia application. The intermediate nodes in the path between two end-systems (service user and provider) are indirectly involved in the process of negotiation and management. The mobile agent interacts with the local agents in the visiting node through shared knowledge base and collects the information about the availability of bandwidth for that flow during its upstream travel from user to service provider (multimedia server).

2 QoS Management by Mobile Agents

Mobile agent based QoS architecture offers an interface for defining desired level of QoS, negotiating resources for attending desired level of QoS, monitoring QoS for adaptation and renegotiating resources when QoS degrades. This architecture provides QoS end-to-end control for each flow. All the intermediate nodes between the source and sink are indirectly involved in the process of negotiation and management.

These systems satisfy all the five functional principles of QoS architecture[4,5]. *Integration* is achieved by defining agents that handle QoS at different abstraction levels (user, system, network). The *separation* principle is due to autonomy of agents. So, they

do not interfere with other functions and flow transmissions of the system. *Transparency* is achieved, because, users and applications can transparently delegate tasks of negotiation and management to the agents. Agents execute concurrently being asynchronous entities, hence *asynchronous resource management* principle is satisfied. Since agents are solely used for management functions, it does not affect other network protocols, thus it achieves *performance* principle.

The proposed architecture consists of a QoS agency installed in all the networked multimedia systems. It comprises of agents (fixed and mobile), agent facilitator and a multimedia application server (MAS) as depicted in fig.1. Flow application parameters defines QoS requirements for each flow. A flow is referred as a path from service generation point to service presentation point. These are specified as a set of parameter-value pairs. MAS manages these flow application parameters. Agent facilitator instantiates agents when it receives a message containing the type of agent to instantiate and the parameters to be passed to that agent. This new agent starts executing, during execution it may communicate with other agents or resources to complete its task. Interaction among the agents takes place through shared knowledge base. The negotiation takes place in all the nodes of flow. Flow application parameters can be renegotiated, if resources are not available as per the requirements of the user.

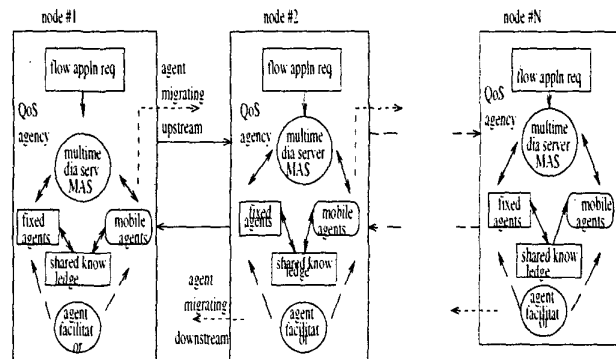


Fig 1. Mobile agent based QoS negotiation and management

The fixed or local agents used in QoS agency are:

- user interface agent*: defines an application or user level parameters;
- QoS mapper agent*: maps parameters from one level to other level;
- QoS monitor agent*: monitors the multimedia flows;
- resource manager agent*: manages local resources at hosting node;
- local negotiator agent*: negotiates locally by interacting with the resource manager agent;
- QoS adaptor agent*: corrects locally if there is a minor degradation QoS in the flow.

The mobile agents used in the QoS agency are:

- flow negotiator agent*: performs global negotiation by interacting with local negotiator agent at the migrated node;
- flow renegotiator agent*: acts on behalf of adaptor agent when some parameters are changed for the flow;
- flow monitor agent*: travels periodically to nodes in the flow, interacts with monitor agent and reports degradation of QoS to the host node.

QoS negotiation Scheme

The negotiation scenario in this QoS management framework can be explained in sequence of steps as follows:

1. The user interface agent reads the user/application level flow parameters and passes flow application parameters to QoS mapper agent;
2. The mapper agent maps user/application parameters to system level parameters and passes to local negotiator agent;
3. The local negotiator agent tries to reserve resources locally as per the flow requirements by interacting with resource manager agent;
4. If local negotiator agent is successful in local negotiation, flow negotiator mobile agent is instantiated to negotiate resources outside node;
5. When arriving at remote nodes in the flow route, this mobile agent interacts with local negotiator and reserve resources temporarily;
6. The flow negotiator agent returns to its hosting node in the reverse route by fixing up the reserved resources permanently at each node for the flow until the flow is processed.

A mobile agent can ping the neighbouring node before migration so as to ensure that node is working. If a node is not working, it will communicate to the user or process which has sent it, so that another route can be chosen. Thus the agent can learn about bad nodes in the network and communicate with other agents.

Management of negotiated flow is achieved by flow monitor and local QoS monitor agents. The mobile flow monitor agent acquires the values of the parameters of the flow exhibited at the visiting node. If a degradation in QoS is detected, QoS adaptor agent at the agent hosting node is informed. The adaptor agent tries to make minor adjustments locally to attain original level of QoS. If it is not successful, flow renegotiation mobile agent is instantiated to renegotiate the resources.

3 Analytical Model

A discrete state, discrete time markovian model is considered to show the negotiation for one QoS parameter (bandwidth) using mobile agents as depicted in fig.2. This model uses "N" number of nodes in the flow from source to destination. Node 1 is considered as agent hosting (source) node.

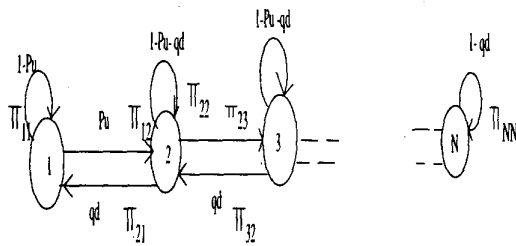


Fig. 2. Analytical model for negotiation agent.

Agent migrates to upstream node starting from hosting node 1, executes bandwidth negotiation at each nodes until "N" th node.

Now agent with negotiation results migrates downstream confirming or fixing negotiated values from "N" th node to hosting (source) node 1.

p_u , probability of agent migrating to upstream node from state i to state j is given as:

$P_u = 1 - P_{icd}$, P_{icd} = probability of agent corrupted or delayed or lost.

q_d = probability of agent migrating downstream from state i to state j is given as:

$q_d = 1 - P_{icd}$, Transition probabilities are p_u , q_d , $1 - p_u$, $1 - q_d$, $1 - p_u - q_d$.

These transition probabilities are represented as π_{ij} in the state diagram where: $i, j = 1, 2, \dots, N$.

p_1, p_2, \dots, p_N are state probabilities of state space $\{1, \dots, N\}$ and probability of being in state i is given by: $p_i[N] = p\{X_N = i\}$. Because of stationary probability property of markov chain, transition probabilities are defined by one step transition probabilities:

$$\pi_{ij}[n, n+1] = p\{X_{n+1} = j | X_n = i\}$$

$$\sum_j \pi_{ij} = 1 \quad j = 1, \dots, N.$$

If initial transition probabilities are considered steady state probabilities can be obtained from steady state equation

$$PQ = P \quad \text{where: } p = p_i[N] \quad \text{and} \quad \sum_{i=1}^N p_i = 1.$$

3.1 Response Time of Mobile Agent

After having computed the steady state probabilities, response time of agent " t_r " is given by

$$t_r = \sum_{i=1}^N p_i t_{ser} + 2 * (n-1) * t_{avt} + T * R, \text{ where:}$$

N = no. of states, n = no. of hops, T = timeout interval(sec), C = capacity of link(bps), p_i = steady state probability of being in state, x = average agent size(kilobits), t_{avt} = average transmission time of agent on link = x/c , t_{ser} = total service time of agent at all the nodes, R = number of retransmissions = $(1/p_1 - 1)$, if $p_1 = 1$, then $R = 0$.

4 Simulation Model

The QoS parameter we have considered for implementation is bandwidth. This parameter is very important in multimedia communication to provide real-time services. Layered functions of mobile agent framework used for the implementation of mobile agent based bandwidth negotiation comprises of agents (top layer), shell interpreter, agent server and TCP/IP (transport mechanism is the bottom layer). Agent server is responsible for receiving the agent, executing the agent with the aid of interpreter and sending the mobile agent to outside world. The shell interpreter executes the mobile shell script (shell script for negotiating the bandwidth). Transport mechanism used for mobile agents is TCP. A mobile agent is hosted from the host in need of running an application from multimedia server. It consists of shell script containing the lower and upper range of bandwidth, path of travel, statements for negotiation and migration. Pseudocode of the mobile agent negotiation is as follows.

Algorithm 1: Mobile agent negotiation procedure

```

Low_range_bwidth = x; High_range_bwidth = y;
path=jala:protocol:pet:proptocol:jala;
check_QoS_profile( );
If (bandwidth_at_node > x && bandwidth_at_node < y)
    then collect data and set_bwidth_from_host = band-
width_at_node
else if (bandwidth_at_node > y)
    then collect data and set_bwidth_from_host = y
else collect data and set_bwidth_from_host = 0
    /* 0 indicate unsuccessful*/
migrate(path)
stop.

```

Mobile agent visits all the nodes in the path, and negotiates with local agents at that node through common knowledge base (QoS_profile). After reaching the service provider or multimedia server (end host), the agent travels back in the same path, fixes the negotiated bandwidth, returns to hosted node. The user or a process at the hosted node decides to run the application based on the information gathered by the agent during its travel.

A linear system configuration is considered for simulation. Simulation has been carried out for two node and three node linear system configuration as shown in fig.3. The machine named “jala-system” is the host of agent and destination is “protocol” for two node systems. The destination (multimedia server) is “pet-system” for three node system. The specifications of these machines are pentium-100mhz, and they are connected by linux network which has 8 systems. We have partially simulated the mobile agent based QoS management framework. The model simulated considers only two agents, local negotiator and the flow negotiator mobile agent on it.

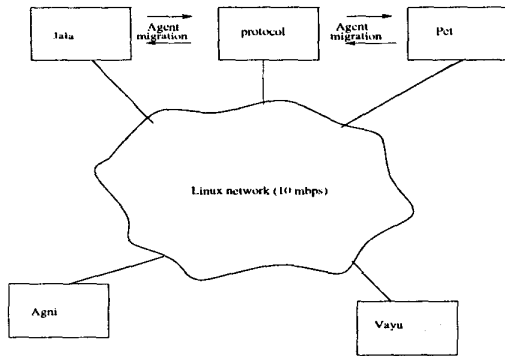


Fig. 3. Simulation model for QoS negotiation

The local negotiator agent at all the nodes generate random values for available bandwidth and stores in shared file (QoS_profile). The QoS negotiation mobile agent is hosted from jala which migrates to next node in the path, negotiates with local agent through a shared file consisting of available bandwidth at that node. The mobile agent after negotiation returns back to its host in the reverse

route and confirms bandwidth, reports about bandwidth availability at the nodes in the flow. In case, the agent is lost, agent host will time out for “T” seconds and retransmit the agent. Even then, if the agent does not respond, this process is carried out for ten attempts. If it fails in all the attempts, agent host has to try for another route. This failure may be due to broken link or node failure in the path. The pseudocode for this is as follows.

Algorithm 2: Fault finding

1. send mobile agent, initialize attempt=1.
2. wait for results until time_elapsed=time_out(T)
3. If result_received within timeout
 - then display results(succ or unsucc)
 - goto step 5
 - else attempt=attempt+1.
4. If attempt <= 10 then goto step 1
 - else display “link failed try another”
5. stop.

The response time t_{rs} for the agent is given as

$$t_{rs} = \sum_{i=1}^n t_c + 2 * \sum_{i=1}^N t_{ser_i}$$

where N =number of nodes,
 $n=2*(N-1)$ =no of hops

t_c = communication time or transmission time

t_{ser} =service time of agent at node(waiting+executing)

A partial implementation of this kind of architecture is simulated using AWB and CORBA (common object request broker) for teleconferencing test bed[5]. AWB is aglet work bench for implementation of mobile agents. Aglets are piece of code in java which can migrate from node to node and execute. Interagent communication is achieved through two mechanisms:message passing of AWB; ORB(object request broker) invocations. Several overheads are associated with this agent based approach for QoS negotiation. They are: communication time; mobile agent has to be transmitted from one node to another (for ex., a 2mb mobile agent needs 0.2sec time for migrating on a 10 mbps network); memory space is required to store the mobile agent and create mobile agent platform; CPU time for computation of mobile agents.

5 Results and Discussion

We have carried out several analytical and simulation experiments using different types of mobile agents for different QoS parameters allocation. One of results of several experiments is presented here for discussion. One of the requirements in mobile agent based systems is to incur low delay in migrating agents. Thus we measured the time taken for the agent to complete its task of bandwidth negotiation. This time is referred to as response time of the agent. The constant parameters which have been considered for computation of response time of agent are $C=10$ mbps, $x=4$ kilobits, $t_{ser}=1.5$ sec and 0.8 sec for three node and two node system, timeout interval $T=4$ sec and the steady state probabilities are obtained as given in section 3. The response time is computed for both analytical and simulation models using the above equations given in section 5 and

6. The results are tabulated in Table I for N=3 and N=2 (N represents the number of nodes in the network). Fig.4 shows response time of the agent with increasing number of retransmissions for three nodes (N=3) and two nodes (N=2).

It is clear from the graphs that response time increases with number of transmissions, so, we should reduce the number of retransmissions to get better performance for real time applications. It can be observed from both Fig.4(a) and (b), that response time decreases with increasing number of retransmissions for two node network as compared to three node network. This indicates that, source hosting agent should try to choose the shortest route to the destination.

One of the important aspect is to predict the agent failures in allocating the required QoS parameter (bandwidth) to the application. We considered a fully connected five node network with link bandwidth= 10mbps and total network capacity = 10links X 10mbps = 100mbps. Maximum number of agents that can be generated in the network are assumed to be 25. Each agent can request for bandwidth between 1 to 10mbps. It is observed in the simulation, that, the agent rejection percentage increases with the increase in the number of mobile agents, thus affecting the admission of new multimedia applications. The rejection starts at $\lambda = 0.4$ (agents generated are = $0.4 \times 25 = 12$). The rejection percentage is 0.135 for $\lambda = 0.4$ and rises to 0.420 for $\lambda = 1.0$.

TABLE I (N=3 and N=2)

Retransmission	trs (analytical)	trs (simulation)	trs(analytical)	trs(simulation)
0	1.5sec	1.7 sec	0.8 sec	0.9 sec
1	5.5sec	5.8 sec	4.8 sec	4.9 sec
2	9.5sec	9.6 sec	8.8 sec	9.1 sec
3	13.5sec	13.9 sec	12.8 sec	12.9 sec
4	17.5sec	17.6 sec	16.8 sec	17.0 sec
5	21.5 sec	21.8 sec	20.8 sec	20.9 sec

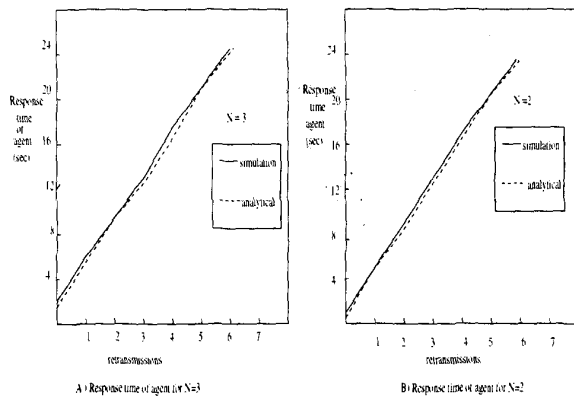


Fig 4. Response time of agent Vs Retransmissions for N=3 and N=2

Counterproposal: For an unsuccessful flow negotiation at any intermediate node, a counterproposal can be made by local negotiator agent at that node. This counterproposal will be carried to agent hosting node by flow negotiator mobile agent, which may tune its application parameters to suit the proposal or reject it. For example, it is required to negotiate for a bandwidth of 4mbps, consider that negotiation failed at node 3 in a four node network path. The local negotiator at node 3 makes a counterproposal for 3mbps, then the flow negotiator agent will travel with this proposal to hosting node 1 and communicates with local negotiator agent at node 1. If it is acceptable, then counterproposal is successful. The time required to make this counterproposal is twice the time needed for the first proposal (for ex., response time for first proposal = 1.7 sec, response time for counterproposal = 3.1 sec).

6 Conclusions

The paper presented different levels of QoS parameters, QoS requirements and its negotiation and management techniques in multimedia communication. The partial implementation of mobile agent based bandwidth negotiation has been shown. It reduces bandwidth and network traffic required for QoS management. The results showed that response time of agent depends on the various factors such as delayed execution, lost agent, corrupted agent, number of nodes in the path. So, always agent host should try to choose a shortest path for mobile agent to respond immediately. We also observed through simulation, that the increase in arrival of mobile agents would affect the admission of new multimedia application.

References

- [1] C. Aurecochea, Andrew T. Campbell, Linda Hauw, " A survey of QoS architectures", *Multimedia systems*, Vol .6, 1998, PP. 138-151.
- [2] D. Wong, N. Paciorek, D. Moore, " Java based mobile agents", *Communications of ACM*, Vol 42, No.3, March 1999.
- [3] V.Oham, A. Karmouch, " Mobile software agents: An Overview", *IEEE Communication magazine*, July 1998, PP. 25-37.
- [4] A. Puliafito, O. Tomarchio, H. Meer, " Agent based framework for QoS management", *4th int.confrence on Analytical and numerical modeling tech.-QoS modelling, singapore*, sep. 1997.
- [5] L.A.Guedes, P.C.Olivera, L.F.Paina, E.Cardozo, " An agent based approach for supporting quality of services in distributed multimedia systems", *Computer Communications*, Vol.21, 1998, PP. 1269-1278.