Paradigms for Reliable Communication Protocols in Mobile Agents based Systems

Anirban Banerjee and Shibdas Bandyopadhyay
Indian Institute of Information technology-Calcutta
Sec-5, Block-AQ/13, Kolkata – 700108, INDIA.
Anir_iiit@yahoo.co.uk , shibdas@rediffmail.com

Abstract

Mobile Agents are regarded as a promising paradigm for future distributed computing. Among essential features of mobile agents, communication is a fundamental ability that enables mobile agents to cooperate with each other. But the presence of mobility raises a number of new challenges in designing message delivery protocols for effective and efficient communications between mobile agents, especially the Communication Failure and Message Chasing problem for highly mobile agents. In this paper, we firstly discuss ways to solve Communication Failure, and then describe a two-dimensional solution space for designing inter-agent communication protocols. We also propose a reliable and efficient mechanism for mobile agent communication named MEFS, which is implemented flexibly and adaptively. The MEFS provides a location-transparent tracking mechanism and, in particular, guarantees message delivery under any condition in a fault-free network. The experiments demonstrate the performance of MEFS too.

Keywords **–** Mobile agents, Intelligent systems, Simulation, Chasing problem.

1. Introduction

As a kind of well-encapsulated entity computing in the open and dynamic network environment, Mobile Agent is thought to be the main distributed computing model based on the next generation of Internet. It is widely accepted that the mobile agent has the following key features: autonomy, collaboration, activity and especially mobility. To exhibit these features, mobile agent must interact and cooperate with others. Although the typical use of a mobile agent paradigm is for bypassing a communication link and exploiting local access to resource on a remote server, remote communication, as [1] emphasizes, is still a fundamental facility in Multi-Agent Systems. Because message passing, on which more complex mechanisms such as RPC &RMI can be easily built, is a basic and well-understood form of communication in distributed systems, we adopt it and focus on the underlying transportation layer in our discussion instead of concerning about the common semantic layer for knowledge sharing which is done in KQML [2] and FIPA ACL [3]. But the agent mobility presents distinct challenges to communication frameworks of the agent system that distributed systems cannot currently address. If the agents keep stationary, we can ensure that messages will be received within a limited time in a fault-free network. However, due to the autonomy and mobility of the agents, the communication object may move from one host to another at any time in the mobile agent framework. The physical location change of agents will result in a problem of

Communication Failure, that is, before a message gets to one host, the target agent has left away, making itself unable to receive this message. Even if we use connection oriented protocols such as TCP, we still cannot guarantee the message delivery to mobile agents all the same. Because message cannot be instantly delivered, the problem widely exists in mobile agent systems. In some extreme situations, like the agent moves frequently, whenever a message reaches a host where the target agent used to reside, the agent has just left away so it can not get the massage all the time and the message keeps chasing the recipient around the system but never gets delivered, resulting in a race condition which is called **Message Chasing** .If the agent cannot receive messages sent to it in time, the collaboration will fail and the system may even crash.

In this paper, we first analyze different ways to designing a communication protocol, and then describe a general-purpose framework. A reliable communication mechanism for mobile agents named **MEFS** (Message Efficiently Forwarding Schema) is also brought forth. Our MEFS implements the delivery process in an effective and efficient manner and gets high adaptability by configuring protocol parameters dynamically at runtime. For the convenience, we always assume that the network consists of fault-free FIFO (First In, First out) channels. That is to say messages will be transferred from one side of the channel to another orderly without any transmission fault. The remainder of this paper is organized as follows. Section 2 contains an overview of proposed mobile agent communication protocols. Section 3 discusses ways to settle Communication Failure and describes the design space for inter-agent communication protocol. Section 4 presents our MEFS with its mechanism to solve the problem of message chasing in brief. We analyze the performance of MEFS in Section 5 and finally conclude with a summary of our work in Section 6.

2. Review of existing literature

Although a few Mobile Agent Systems, such as Ajanta [4] and Odyssey [5], adopt RMI or RPC for communication at a higher layer, still many research works propose their inter-agent message passing schemes respectively. In ICM [6], *store-and-forward* is the fundamental architecture. There is a CS (i.e. Communication Server) on each host taking charge of the communication details. The CS will store messages and try to forward them to corresponding receivers. If the recipient cannot be located or has moved away, CS will reject the message or keep it for some period of time before discarding it. It is clear that ICM is not a reliable system for message passing. “Session-Oriented Communication” [7] implemented in Mole system [8] uses the method of “request-and-reply” to establish communication between a pair of agents. When communication failure happens, Mole will inform the sender and discard the message. Mole cannot offer a location transparent and reliable communication. Each message sender in Epidaure [9] knows the “home” of the target object and each host on the migration path of an agent keeps a forwarding pointer pointing to the next host on the path. Messages are sent to home and then forwarded to the recipient along the path directed by the pointer. However, a racing condition may occur if the target agent moves frequently while Epidaure provides no solutions to it. The same matter exists in Voyager [10], so they both are not reliable. MDP [11] organizes a tree-like hierarchy of servers to locate agents and route messages, but the hierarchy is difficult to be built and, more important, message chasing cannot be avoided here. MStream introduced in [12] presents a resending—based mechanism. A *Location Manager* broadcasts agent's new location with some strategy and if a message is sent to an outdated address of the target agent, it will be retransmitted. Because there is no upper bound of the number of message resending, Mstream cannot meet the requirement of reliability when agents migrate frequently. A reliable communication mechanism is stated in [13] based on the idea of *snapshot* .By broadcasting the to-be-sent messages and associating different status value with every incoming channel for each node in the mobile agent system, an agent can receive all of the messages sent to it. In this algorithm, the traffic overhead is unaffordable when there are a large number of hosts and agents in the network. Mailbox-based scheme like ARP [14] and JATLite [15] is another solution in this field. All messages are sent to a mailbox (in JATLite, it is called AMR) and wait for agent to check. If the mailbox pushes the message to the receiver, communication failure and message chasing problem still exists. When a "pull" method is adopted, synchronization is needed and the receiver will be blocked every time checking the mailbox.

3. Ways to Approach

Two fundamental issues must be addressed in any communication protocol for mobile agents:1) tracking the location of target mobile agent, and 2)delivering message to the agent. If we look at the agent tracking in the process of sending message as a *reading* operation and the change of location after the recipient migrates as a *writing* one, the *reading-writing* concurrent access collision on the shared variable "current agent location" leads to the "dirty reading " problem, i.e. agent changes its location when the message is en-route so message is sent to a host where agent does not reside any more. It is the presence of mobility, not the possibility of fault in the network that causes communication Failure. We know that there are four main tracking methods for mobile objects [16] all of which have been applied in the mobile agent systems aforementioned:

* **Broadcast/Multicast scheme (BM):**The sender sends query messages to all hosts for location of the receiver or broadcasts agent message directly in the system. Another way is that the agent broadcasts its new location after each migration. The BM works efficiently in local network domain, especially in bus-based multiprocessor systems. But it is impractical in large-scale network because of large communication overhead.
* **Hierarchical scheme (HS):**A tree-like hierarchy of servers forms a location directory here. For each agent there is a unique path of pointers that starts from the root and ends at the leaf that knows the actual address of the agent. The HS supports the locality of mobile object migration and communication well. However the hierarchy is not always easy to construct, especially in the Internet environment.
* **Central Server scheme (CS):**A location server is used to keep the track of the physical location of a mobile agent. Although CS is easy to implement, the location server in it is really a potential bottleneck of performance.
* **Forwarding Pointer scheme (FP):**Each host on the migration path of an agent

has a forwarding pointer pointing to the next host on the path so that messages can be forwarded to the recipient along the path. The FP has less reliance on a location server and incurs no location registration overhead. But it may be difficult to guarantee message delivery and shorten the forwarding path if a communication protocol adopts FP

We can classify ways to solve Communication Failure into three categories:

Ostrich: Discarding messages in Mole and ICM when a host does not know where the recipient locates is an instance of Ostrich. The Ostrich ignores the problem and does nothing for Communication Failure so it cannot contribute to a reliable protocol in mobile agent systems.

Avoidance: Avoidance, which is widely adopted, establishes some mechanism to prevent delivering messages to a host on which the recipient does not reside so that the communication failure will never happen. Synchronous methods are mostly used here, but agents are disallowed migrating until having collected all ACK messages needed, which makes Avoidance economically ineffective and technically inefficient, especially, in the Internet.

Detection: In Detection, the system must be able to detect communication failure and take some measures to deal with lost messages. For the "nature of agents", Detection always uses asynchronous ways to implement communication within mobile agents. However, it is difficult to design a reliable Detection with its side effects, like Message Chasing, well settled. Some protocols presume different conditions such as having known the topology of the network or destination of migration in advance, while these assumptions are not always reasonable and we will not consider here.

4. MEFS: A Reliable and Efficient Communication Mechanism

To allow mobile agents communicate seamlessly regardless of their location, each agent is assigned a globally unique name when it initializes from which its homeplace's address can be easily resolved. Meantime, each node in the network keeps table recording information about the agents currently residing in it and the agents initialized on it. When migrating, agent must unregister, leaving a forwarding pointer, when it leaves a node and register when arrives at a new one. Assuming agent b migrates from some node Pb to P 'b as an example: When b prepares to migrate, it sends a control message to its homeplace Hb ,telling its target P'b .After informing Pb to update its status table, b moves to node P 'b and registers to P 'b by adding or updating an entry in the status table stored in P 'b. Actually, control messages are generated after agent having made the migration decision, so here we have no preknowledge of agents action and also make no limitation to migrations. Forwarding messages asynchronously is the core of MEFS. To deliver message efficiently with less limitation to agent migration, a module "Communicator" in every node processes each incoming messages with responsibility for registration. If agent a resides on the same node as b, messages can be sent to b just within the node; otherwise Communicator will look for the status table maintained in the same node or query b 's homeplace for b 's location and then forward the messages to the desired next node. Even though agent b is roaming in the Space freely, we can easily forward messages because b has left a pointer linking to the next host every time it migrates. Messages are forwarded along a chain logically constructed by the recipient and we need not query b's homeplace every time leading homeplace to be a potential bottleneck of the system.

Since it is believed that asynchronous methods cannot fully solve the message-chasing problem, we use synchronous communication when racing occurs but constrain agents autonomous migration as little as possible. MEFS achieves synchronous by using **Chasing Message Register** and **Over-speed Agent Blocking.** Chasing Message Register means that if a message has been forwarded with a given times **C0**, the message is then thought as a "chasing message" and its serial number will be sent to the recipient's homeplace to store. On the other hand, each agent must calculate the velocity itself once reaching a node. The velocity can be the average speed in a given period of time. If the velocity exceeds a given maximal **V0**, the agent should establish a connection with its homeplace, getting and then deleting the chasing message number list for it. Agents will be blocked at current node until all chasing messages are received. This is so-called Over-speed Agent Blocking. If agent's velocity is low, the message will catch their recipient affirmatively. While the agent migrates frequently, the strategy of over-speed agent blocking can surely avoid message chasing. Besides being blocked when over-speed, each agent in MEFS must contact its homeplace and wait for chasing messages at intervals. We name the interval **TS**. By synchronizing regularly, we can guarantee that the recipient, no matter how the network condition changes, will receive the message eventually. With Chasing Message Register, Over-speed Agent Blocking and Regular Synchronization, Message Chasing problem is eradicated and a reliable communication is finally achieved in MEFS. From the experiment, we observe that all the messages will be delivered within a limited period of time.

6. Performance Evaluation

The MEFS scheme was pitted against a normal non-MEFS to identify the performance benefits of our strategy over a conventional framework. NS2 was used to simulate this comparative performance evaluation environment wherein a simulated environment consisting of a random graph possessing nodes of the order of 50 to 1000 were generated. The mobile agents were then allowed to migrate from node to node and the MEFS algorithm was used to compare the number of messages that could not be delivered to the target agent. Our measure of success in this simulation has been the number of failures in catching up with the target agents. The number of messages that could not be delivered to the target agents form the true performance evaluation of our work. Figure 1. attached below comprehensively details the benefits of using our approach vis-à-vis the non MEFS scheme.

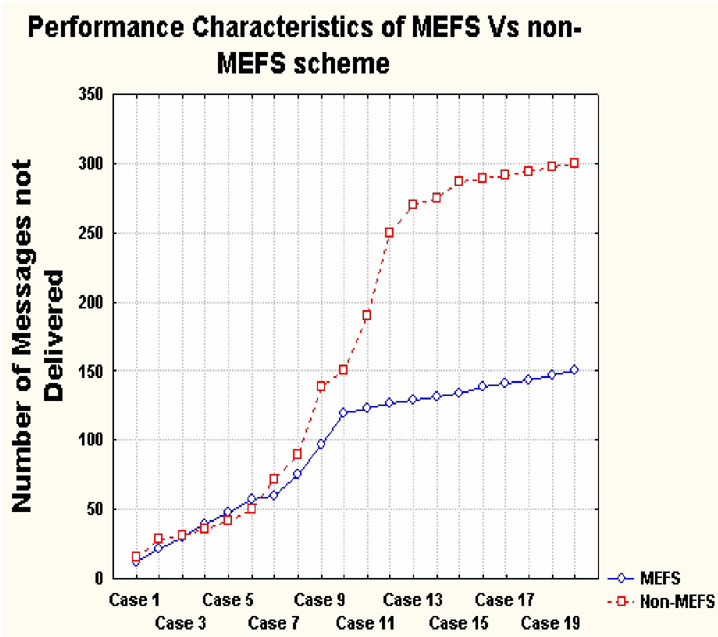

Figure 1. A performance comparison of the MEFS Vs non-MEFS scheme to gauge the number of messages that could not be delivered to the target agent.

6. Conclusions and Future Work

The communication infrastructure plays an important role in Mobile Agent Systems. In this paper, after describing a general framework for designing message delivery protocols in mobile agent systems, we briefly propose an adaptive mechanism **—MEFS (C0,V0,TS)**, which not only realizes a reliable communication but also achieves high efficiency for the system. In MEFS, agents can roam with little restriction in the system and receive all messages in a limited period of time. Although agents will also be blocked at the time of over-speeding, this happens with little probability. In addition, maintaining information and forwarding messages at each host in the system will not cause great overhead in most cases. There are some problems yet to be settled for MEFS, such as dynamic configuration of system parameters and path compression technique for message forwarding. We will try to solve these problems in the future work.

References

[1] FENG Xinyu,CAO Jiannong,et al.“An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems ”.*G.P.Picco(Ed.),Mobile Agents,LNCS 2240, Spintger-Verlag* ,2001,pp.135-151.
[2] Yannis Labrou and Tim Finin,“A Proposal for a new KQML Specification”, *Technical Report CS97-03* ,UMBC,1997.
[3] Foundation for Intelligent Physical Agents,“FIPA 97 Specification Part 2:Agent Communication Language ”, *http://www.fipa.com* ,Oct.1998.

[4] Neeran Karnik and Anand Tripathi,“Agent Server Architecture for the Ajanta Mobile-Agent System ” ,In *Proceedings of the 1998 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98)*,July 1998,pp 66~73.
[5] General Magic Inc.,“Introduction to the Odyssey ”, *http://www.genmagic.com/agents* ,1997.
[6] F.G..Mcabe “Inter-Agent Communication Reference Model ”, *http://www.nar.fujitsulabs.com/documents* ,2000.
[7] Joachim Baumann,Fritz Hohl,Nikolaos Radouniklis,Markus Straßer and Kurt Rothermel,“Communication Concepts for Mobile Agent Systems ”,In *Mobile Agents,Proc.1st Int. Workshop,MA`97.Springer* ,1997.
[8] J.Baumann,F.Hohl,M.Straßer and K.Rothermel, “Mole-Concepts of a Mobile Agent System ” ,In *WWW Journal, Special Issue on Applications and Techniques of Web Agents*, 1998.
[9] Jocelyn Desbiens,Martin Lavoie and Francis Renaud, “Communication and Tracking Infrastructure of a Mobile Agent System ”,In *Proc.31st Annual Hawaii Intl.Conference on System Science* ,Vol.7,1998,pp54-63.
[10] ObjectSpace Inc.,“Objectspace voyager ” , *http://www.objectspace.com/voyager* ,1997.
[11] S.Lazar,I.Weerakoon and D.Sidhu,“ A Scalable LocationTracking and Message Delivery Scheme for Mobile Agents ”,In *Proc.7th IEEE Intl.Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* ,1998.
[12] M.Ranganathan,M.Beddnarek and D.Montgomery,“A Reliable Message Delivery Protocol for Mobile Agents ”,In *Proc. of the Joint Symposium ASA/MA ’2000* ,Sept.2000.
[13] Amy L.Murphy and Gian Pietro Picco,“Reliable Communication for Highly Mobile Agents ”,In *Agent Systems and Architecture/Mobile Agents(ASA/MA)’99* ,Oct 1999.
[14] CAO Jiannong,FENG Xinyu,et al.Design of Adaptive and Reliable Mobile Agent Communication Protocols. In *Proceedings of 22nd International Conference on Distributed Computing Systems(ICDCS 2002)*,July,2002,Vienna,Austria.
[15] H..Jeon,C.Petrie and M.R.Cutkosky,“JATLite:A Java Agent Infrastructure with Message Routing ”,*IEEE Internet Computing* ,Mar/Apr,2000.
[16] Andrew S.Tanenbaum and Maarten Van Steen,“Distributed Systems Principles And Paradigms ”,*Prentice Hall.Inc.*,2002.