A Comparison of Communication Protocols for Mobile Agents

Charu Virmani
Manav Rachna International University, Faridabad
Haryana.India

Author Email: charu.fet@mriu.edu.in

Abstract
Mobile agent is a software program that migrates from one node to another while performing given tasks on behalf of a user. Mobile agent technology is used to develop many distributed applications. A mobile agent can communicate independently with other agents, with users, and with the hosts in the network and preserves all of its state when it moves from one network node to another. However, the mobility of agents makes it more difficult to trace mobile agents and transfer messages reliably. Therefore, a reliable communication protocol that provides efficient location management and reliable message delivery is the key issue to the design of mobile agent system. Several communication protocols exists in a mobile agent communication environment, however an analytical study indicated that some problems remain unsolved and none have proved to be the best. This paper aims to present a comparison of various protocols with the intention to propose a hybrid version in future.

Keywords: *Mobile Agent, MDP, Communication Protocols.*

1. Introduction
A *mobile agent* is a software program and data which represents a user in a computer network, and is capable of migrating *autonomously* from node to node in a heterogeneous network to perform some computation on behalf of the user continuously with the features of autonomy, social ability, learning, adaptivity, reactivity, mobility etc [1][2]. They are defined as objects that have behavior, state, and location. Its tasks are determined by the agent application, and can range from online shopping to real-time device control to distributed scientific computing. It decides when and where to migrate. It can execute at any point or suspend its execution, moves easily across the network and continue its execution on another host. When a mobile agent decides to move, it saves its own state, transports this saved state to the new host, and resumes execution from the saved state. Each agent is typically composed of the agent code, the agent execution thread along with an execution stack, and the agent data part, which corresponds to the values of the agent's global variables.

Mobile agents are used in a wide area of applications like Network Management and Monitoring such as processing data over unreliable networks, Information Searching and Filtering like distributed DBMS (Database Management System), multimedia, Internet, Intrusion Detection, Military, telecommunications, Secure brokering such as untrustworthy collaborators etc. The ability of a mobile agent to personify their creator's intentions and to act and negotiate on behalf of them makes it well suited for electronic commerce For example, instead of spending a huge amount of time going through on-line bookstores to find the best deal on a book, firing up an agent to do this task would save us a considerable amount of time.

The agent would be programmed to visit a number of bookstores and find the best deals on books we need.

Mobile agents play an important role in the development of active and dynamically managed networks and distributed systems. It provides several advantages [1], most notably of which are: reduction in network traffic, asynchronous autonomous interaction, overcome network latency, robustness and fault tolerance and its support for heterogeneous environments.

This paper is being organized into 4 sections. Section 2 describes the basic architecture of mobile agent. Section 3 presents analysis and comparison of existing protocols for mobile agent communication. Section 4 provides the shortcomings of the prevailing protocols and concludes the paper.

2. Architecture of Mobile Agent

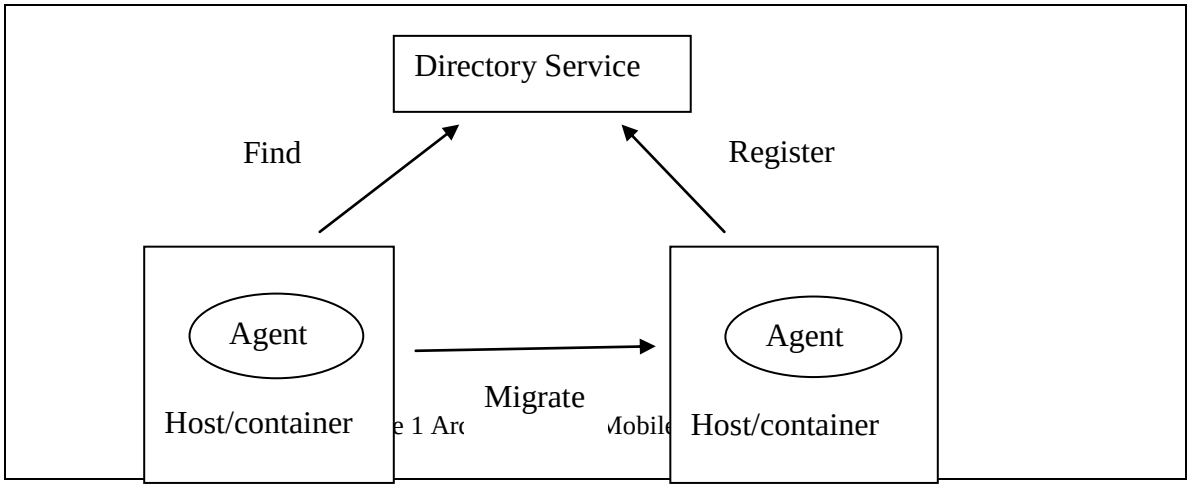
A mobile agent consists of the program code and the program execution state (the current values of variables, next instruction to be executed, etc.). Initially a mobile agent resides on a computer called the home machine. The agent is then dispatched to execute on a remote computer called a mobile agent host (a mobile agent host is also called mobile agent platform or mobile agent server). When a mobile agent is dispatched the entire code of the mobile agent and the execution state of the mobile agent is transferred to the next host.

The host provides a suitable execution environment for the mobile agent to execute. The mobile agent uses resources (CPU, memory, etc.) of the host to perform its task. After completing its task on the host, the mobile agent migrates to another computer. Since the state information is also transferred from the host, mobile agents can resume the execution of the code from where they left off in the previous host instead of having to restart execution from the beginning. This continues until the mobile agent returns to its home machine after completing execution on the last machine in its itinerary.

The basic architecture of the mobile agent comprises of an agent and its execution platform. The agent may have to travel various platforms in order to complete its designated task. Thus, there is a need of an efficient and flexible protocol which allows the agent on the different platforms to communicate with each other.

The architecture consists of Agent Manager, Security Manager, Inter-agent Communication, Directory Manager and the language for the efficient transfer of data. Most current agent systems are implemented on top of the Java Virtual Machine (JVM), which provides object serialization and basic mechanism to implement weak mobility.

The agent manger sends agents to and receives agents from remote hosts. It prepares agents for transport by serializing the agent and creates the agents execution context. The Security Manager protects the host and agent from unauthorized access. It authenticates agents before allowing execution. It is automatically invoked when the agent's tries to use any system resource or tries for any unauthorized activity.



Names and addresses of services and agents are registered in Directory Service. The agent first migrates to remote container and registers itself to the Directory Services. When any other agent needs to find some agent it contacts the Directory Services for help.
Inter-agent Communication is used to allow the agents to communicate through message passing mechanism. This part is still under research work where agents from different agent systems can communicate to each other. Till now all those agent system which follow FIPA (Foundation for Intelligent and Physical Agent) standards are able to exchange messages as they follow a standard format for sending and receiving messages. Inter communication is still an issue among heterogeneous agent system.

3. Existing Communication Protocols for Mobile Agent

Although the mobile agent paradigm provides great potential advantages over traditional approaches in distributed computing applications, there are still several issues to be addressed before the technology can be widely accepted. The performance of the communication protocol is one of the critical issues in mobile agent systems. A basic communication protocol for mobile agents must provide the desired degree of location transparency, reliability and efficiency [1]. Although many communication protocols have been proposed [2][3] for mobile agent systems and most of them are location transparent, these protocols usually compromise some aspects of reliability and efficiency. Early research attempts to solve these three problems using agent remote procedure call (ARPC) [13], which is analogous to the traditional RPC. With ARPC, programmers have to explicitly handle agent allocation and message delivery.

Home-server schemes (HSS) [2][11] are the most popular communication protocols for multi-agent systems because they are compatible with the current Internet Protocol. In this approach, each mobile agent associates with a stationary agent, which is called the mobile agent's home agent. Each home agent has a database to store the addresses of all agents that use this host as home agent. A central naming server, called home server, maintains a binding between mobile agents' names and their home agents' addresses. In the home-server approach, a mobile agent must inform the home agent of its new location after each migration. To contact a mobile agent, the sending agent delivers the message to the home server. Then, the home server routes the message to the receiving agent's home agent and after that the home agent forwards the

message to the receiving agent's actual location. The home-server scheme is simple to implement and works well for small-to-medium distributed systems. This protocol has some drawbacks. If a mobile agent moves far from its Home Node (HN) or specific server, the costs of location update and message delivery become relatively high. When the number of mobile agents grows and mobile agents frequently migrate, an HN or a specific server becomes a bottleneck. If the location information of an HN does not have the latest data, or if a mobile agent has left prior to a message arriving at the destination, message delivery fails. In this case, the tracking problem occurs. However, in the triangular routing, central server constraints as well as the message loss problems in the home-server protocol affect the system performance [16]. Mobile agent systems such as Aglets and SPRINGS support this protocol.

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 the 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 problem persists in Voyager [10], making both of them unreliable.

The Forwarding–Proxy (FP) protocol [11][17][18] adopts the Forwarding pointers approach for location management and the Forwarding approach for message delivery. Location information in the FP protocol is stored at nodes that the mobile agent has visited. When a mobile agent migrates to a different location, a forwarding proxy that maintains information of the next location of the mobile agent is created at each node. Message delivery is performed by following the chain of proxies, referred to as path proxies. In contrast to the Home Proxy (HP) protocol, location management and message delivery are distributed in the FP protocol. The FP protocol does not involve a remote update during migration. However, every node within path proxies must participate in the message delivery procedure. If path proxies are long, the cost of message delivery increases. In addition, if path proxies are broken, message delivery fails. Suppose that path proxies are long and a mobile agent moves frequently. If a mobile agent leaves before a message arrives, and thus, a message does not catch up with the agent, the tracking problem will occur. Accordingly, the cost of communication increases. The Voyager mobile agent system supports the FP protocol.

The Broadcast protocol [17][19] is another communicating protocol which does not maintain any location information on mobile agents. The Broadcast protocol broadcasts a message to all nodes within a network in order to deliver it to a mobile agent. If a node has the mobile agent,

it delivers the message to the mobile agent directly or through its mailbox (or message dispatcher). The Broadcast protocol can be used to send a message to a group of mobile agents. In this case, the message contains multiple IDs or names of receiver agents. The cost of communication becomes very high as the number of nodes and regions increases.

The Shadow protocol [22] adopts the Location server and Forwarding pointers approaches for location management and the Forwarding approach for message delivery. The Shadow protocol uses a placeholder (namely, shadow) that records the locations of all dependent agents, similar to a Location server. It also uses forwarding proxies at nodes that the mobile agent has visited. A mobile agent updates the current location to the associated shadow according to a Time To Live (TTL). If TTL still remains, a mobile agent leaves behind forwarding proxies at the nodes that it has visited. After TTL expires, it updates its current location to the shadow. Accordingly, path proxies are shortened. Message delivery is performed through the associated shadow. If there are path proxies, a message is delivered following the path proxies. The Mole mobile agent system provides the Shadow protocol.

The Search-by-Path-Chase (SPC) protocol [20] adopts the Location server and Forwarding pointers approaches for location management and the direct approach for message delivery. The SPC protocol takes into account a multiregion mobile agent computing environment. Location information is stored in a distributed way at a Region Agent Register (RAR) or a Site Agent Register (SAR). RAR is responsible for maintaining location information about all the agents within a region. SAR maintains information about the agents at a node, or a reference (that is, forwarding pointer) about the agents that have visited the node. Location management and message delivery are achieved by tracking the links that a mobile agent has left at two registers. The SPC protocol has the tracking problem and high communication cost.

The Adaptivity and Reliable Protocol (ARP) protocol [21] adopts the Location server and Forwarding pointers approaches for location management and the Mailbox approach for message delivery. The ARP protocol delivers message s using a mobile mailbox. An agent keeps track of the location of its mobile mailbox. When an agent migrates, its mobile mailbox moves from one node to another according to the cost of message delivery. A message is first sent to the mobile mailbox. A mobile agent retrieves messages from the mailbox whenever needed. The ARP protocol has a high overhead when transferring the mailbox. If the node that has a mailbox fails, the messages are lost. Furthermore, both mobile agents and home nodes must keep track of mobile mailboxes.

Message Delivery Protocol (MDP) [6] offers a mechanism to track the location of an agent accurately. It maintains all agents' addresses in a tree-like hierarchy structure. Agents are categorized into groups, each of which will be put in a domain. Each domain has a gateway and the top of all domain gateways is the root. Each server within a domain updates the address of an agent when it migrates to another domain. This procedure is logically simple; however building the hierarchical tree is a complicated task, especially in a very large distributed network such as internet. Furthermore it can not address the situation when a message is continuously chasing a highly mobile agent. 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.

Two fundamental issues that must be addressed in any communication protocol for mobile agents are tracking the location of target mobile agent, and delivering message to the agent. There are four main tracking methods for mobile objects namely Broadcast/Multicast scheme, Hierarchical Scheme, Central Server scheme and Forwarding pointer scheme

In Broadcast/Multicast scheme, a sender agent broadcasts a request of recipient's location or the message to all the hosts in the system. It 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. Whereas, Hierarchical Scheme (HS) uses a hierarchy to store the location of all agents. Each agent has a path of its location from the root to the leaf, which is a group domain in which it resides. It 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) utilizes a dedicated server to maintain the location path of a mobile agent. Although CS is easy to implement, the location server in it is really a potential bottleneck of performance. In contrast to this, in the 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.

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. This change in physical location of the agents will result in Communication Failure, that is, before a message gets to one host, the target agent has left away, making it unable to receive this message. Ostrich, Avoidance and Detection are the three different ways to solve this problem. The Ostrich ignores the problem, whereas, Avoidance 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 and in Detection, the system must be able to detect communication failure and take some measures to deal with lost messages.

Ostrich is not a reliable protocol because it simply discards the message when the host does not know the recipient's location and does nothing for the communication problem. Avoidance, which is widely adopted, employs synchronization mechanism in which agents are disallowed migrating until having collected all ACK messages needed, which makes avoidance economically ineffective and technically inefficient, especially, in the Internet. 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 technique that could overcome its side effects, like Message Chasing.

An analysis of the existing protocols is drawn in the Table 1. MDP provides mechanisms of tracking problem. But, however it does not guarantee reliability when a mobile mailbox migrates. The HSS [2] and Broadcast [19] protocols provide asynchronous delivery when a message broker or a dispatcher is used. In the FP and Shadow protocols, a message is delivered asynchronously, following path proxies or through a shadow. However, the FP and Shadow protocols do not provide timeliness because they deliver messages by following path proxies. The ARP protocol does not achieve timeliness because messages are not retrieved from a mailbox until a mobile agent desires message delivery. The Broadcast protocol does not use location information to send messages, thereby engendering high cost because it transmits multiple copies of a message to all the nodes within a network. In contrast, the HSS, FP, Broadcast, Shadow and ARP protocols do not provide scalability. The HSS protocol suffers from server bottleneck. The FP protocol has a long path proxy problem. The SPC protocol does not support scalability when path proxies a used during communication, but does so otherwise. ARP has high migration overhead of a mobile mailbox and synchronization between an agent and its mobile mailbox.

Table 1 Analysis of Existing Protocols

Items		**HSS**	**ICM**	**FP**	**Broadcast**	**Shadow**	**SPC**	**ARP**	**MDP**
Reliability	Tracking problem	No	No	No	In some cases	No	No	No	Yes
	Message delivery	No	No	No	In some cases	No	No	In some cases	In some cases
Asynchrony		in some cases	yes	Yes	In some cases	Yes	No	yes	Yes
Timeliness		Yes	Yes	No	Yes	No	Yes	No	Yes
Location Dependency		Yes	yes	Yes	No	Yes	Yes	Yes	Yes
Scalability		No	no	No	No	No	In some cases	No	In some cases

4. Conclusion & Future Scope

Several Communication protocols have been proposed in a mobile agent environment: HSS, FP, Shadow, Broadcast, SPC, ARP, ICM and MDP. However, some problems remain unresolved. First, existing protocols apart from the SPC protocol do not consider multiregion computing environments. Second, they do not guarantee the delivery of messages. In other words, a tracking problem occurs; a message follows a mobile agent without being delivered to the agent. A message is just sent to the nodes that the mobile agent left without delivery. Third, no protocols deal with location management and message delivery of cloned mobile agents and parent-child mobile agents. In a mobile agent computing environment, a mobile agent can be cloned or a child mobile agent can also be created because a mobile agent is a software program. To solve these problems, a new reliable Protocol is required for multiregion mobile agent computing environments which will fulfill the following design goals: reliability, asynchrony, timeliness, location dependency, scalability, and communication cost.

References

[1] J. White, "Mobile Agents," General Magic White Paper, 1996.

[2] "Ip mobility support," Network Working Group, Tech. Rep. RFC2002, 1996.

[3] Object Management Group, "Mobile Agent System Interoperability Facilities Specification," OMG TC Document orbos/97-10-05, 1997.

[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/voyage ,1997.

[11] J. Baumann, "A Comparison of Mechanisms for Locating Mobile Agents," IBM Research Report 3333, Aug. 1999.

[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] F. Bellifemin, G. Caire, A. Poggi, and G. Rimassa, "Jade - a white paper," EXP - in search of innovation, vol. 3, no. 3, pp. 6–19, Sept.2003.

[17] P.T. Wojciechowski, "Algorithms for Location Independent Communication between Mobile Agents," Technical Report 2001/13, Comm. Systems Dept., EPFL, Mar. 2001.

[18] D. Deugol, "Mobile Agent Messaging Models," Proc. Fifth Int'l Symp. Autonomous Decentralized Systems, pp. 278-286, Mar. 2001.

[19] H. Jafarpour, N. Yazdani, and N. Bazzaz-zadeh, "A Scalable Group Communication Mechanism for Mobile Agents," J. Network and Computer Applications, vol. 30, no. 1, pp. 1153-1172, Jan. 2007.

[20] A.D. Stefano and C. Santoro, "Locating Mobile Agents in a Wide Distributed Environment," IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 8, pp. 153-161, Aug. 2002.

[21] J. Cao, L. Zhang, J. Yang, and S.K. Das, "A Reliable Mobile Agent Communication Protocol," Proc. 24th IEEE Int'l Conf. Distributed Computing Systems (ICDCS '04), pp. 468-475, Mar. 2004.
[22] J. Baumann and K. Rothermel, "Shadow Approach: An Orphan Detection Protocol for Mobile Agents," Proc. Second Int'l workshop Mobile Agents, pp. 2-13, 1998.
[23] J. Baumann, "A Comparison of Mechanisms for Locating Mobile Agents," IBM Research Report 3333, Aug. 1999.