ISSN: 2454-1311

Causally-Ordered Message Delivery Protocol for Mobile Agent Groups

Jinho Ahn

Department of Computer Science, Kyonggi University, Republic of Korea

Abstract—*Agent group communication is one of the most important functionalities that should be reconsidered for highly dynamic and large-scale distributed agent systems. Although previous researches on this topic have actively been performed, there is little research work that addresses the various important issues or properties such as reliability, scalability, causal order and so on harmoniously due to their respective limitations. In this paper, we introduce our novel work that have already been done to achieve the goal mentioned above, and compare it with the existing ones in detail.*

Keywords— causal ordering, distributed system, group communication, mobile agent, reliability.

I. INTRODUCTION

Software agent programming paradigm with agent mobility has several beneficial features such as reduction of network traffic, overcoming of network delay, enabling asynchronous execution and enhancement of dynamic adaptation [5, 6, 10, 19]. These advantages may enable this technology to be very widely used in distributed systems. But, as their entire size rapidly becomes larger, several research issues related to this technological paradigm such as communication, security, dynamic adaptation, etc., should be reconsidered to overcome this sort of new demand [1]. Among them, the most important functionality is improving the mobile agent communication performance even though the systems are based on very large-scale infrastructures.

Although previous researches on this topic have actively been performed, there is little research work that address the various important issues or properties such as reliability, scalability, causal order and so on harmoniously due to their respective limitations. In this paper, we introduce our novel work [1] that have already been done to achieve the goal mentioned above like in Fig. 1. They allow each mobile agent to replicate its location information only on a few among its visiting nodes adaptively depending on its preferences. They enable all messages sent to a mobile agent group to be reliably delivered to its surviving group members in a consistent causal order. In addition, they provide short-cuts to its target group by each sending agent keeping its effective group location information table. Then, we attempt to analyze it against the other previous works in various aspects in detail.

Fig. 1. The important properties to satisfy for our goal.

II. SYSTEM MODEL

This paper considers an asynchronous distributed agent based system where there is no global memory, no global clock and no bound on message delay [8]. The system consists of a set of agent service nodes. Each service node supports an environment in which agents can operate safely and securely, and provides a uniform set of services through which visiting agents can access its local resource in a limited way regardless of their locations. An agent is initially created on a service node, called home node of the agent, and is given a unique identifier within the node. So, each agent can be identified as a globally unique object in the system by using the combination of its local identifier and the identifier of its home node. When an agent migrates in the system, its code and state information are captured and then transferred to the next node. After arriving at the node, the mobile agent resumes and performs its task, if needed, by interacting with other agents. In order to perform an assigned task on behalf of a user, a mobile agent AID executes on a sequence of $l(l>1)$ service nodes according to its itinerary, $I_{AID}=[DN_{home}, DN_1, \ldots, DN_{(l-1)}]$, which may be statically determined before the mobile agent is created on its home node or dynamically while progressing its execution. It is assumed that communication channels support standard asynchronous message passing with the reliable FIFO ordering property and are immune to partitioning [8].

ISSN: 2454-1311

Mobile agents can migrate and messages be passed along these channels. Finally, we assume that service nodes and mobile agents crash based on the fail-stop model, in which they lose contents in their volatile memories and stop their executions [17].

III. CAUSALLY ORDERED MESSAGE DELIVERY FOR MOBILE AGENT GROUPS

In this section, we describe how our work [1] can handle each issue stated earlier in detail.

3.1 Reliability

3.1.1 Agent Mobility and Directory Node Failure

Before introducing our work, let us explain the definition of two important terms, forwarding node and locator, according to the role of each directory service node. Forwarder of an agent is a directory service node that maintains a forwarding pointer of the agent on its storage. Thus, there may be the various number of forwarding nodes of each agent in the system according to which agent communication protocol is used. Locator of an agent is a special forwarder managing the identifier of the service node that the agent is currently running on. Assuming every node is failure-free our protocol requires only one locator for each mobile agent to address agent mobility. But, if the protocol intends to tolerate up to F ($F \geq 1$) node failures, (F+1) locaters of each mobile agent should exist. Our location updating module can ensure reliable delivery to mobile agents despite both agent mobility and F failures of forwarders as follows. Then, we assume that the value of F is 1 in all examples shown later for explaining. Initially, when the agent is created, its home node becomes the first forwarder and locator of the agent. This module allows each mobile agent to autonomously replicate trails of forwarding pointers only on a small number of appropriate visiting nodes unlike the existing forwarding pointer-based protocols [13, 14]. In this way, it can control adaptively the tradeoff between location updating and message delivery costs depending on agent's preferences.

3.1.2 Mobile Agent Failure

Assuming this kind of mobile agent platforms, there may occur two sorts of mobile agent failures, general member's failures and group leader's failure [1]. First, if some agent members except for their group leader crash, their identifiers and locators information have only to be removed from the other surviving agent members' membership table GMInfo in our work. But, if the group leader's failure is detected, the corresponding module must execute the following complicated procedure. Among the other surviving members, a new group leader is chosen that has delivered the largest number of messages destined to their group. Then, the leader should force the other live agent members to receive the global delivery sequence numbers of all the messages which it has delivered, but they haven't yet, and deliver the messages in order.

3.2 Causal Message Ordering

To satisfy causal ordering property on message delivery, each agent group member i should have the following variables.

- $MVector_i$: a vector where $MVector_i[j]$ is the timestamp of the last message sent from each member j having affected the current state of agent i.
- $RMSG\_Q_i$: a message buffer for maintaining all the messages which have been already received by agent i, but are waiting until the causally ordered delivery condition is satisfied with itself. Each element e consists of two fields, msg and MVector, where MVector was piggybacked on message msg when agent i has received the message. So, if j is the sender of msg and $((e.MVector_j[j] > MVector_i[j]) \wedge (\forall k \neq j: e.MVector_j[k] \leq MVector_i[k]))$, msg can be delivered to the corresponding application and then e is copied into $DLVD\_Q_i$ and removed from $RMSG\_Q_i$.
- $DLVD\_Q_i$: a message buffer for keeping every message which has been already delivered to the corresponding application. Each element e consists of two fields, msg and MVector, where MVector was piggybacked on message msg when agent i has received the message. It is used for giving lost messages to another mobile agent member on its request for ensuring reliable message delivery even on unreliable networks. When agent i knows every other group member has received the message, it may be garbage collected from the buffer.

Suppose there is a mobile agent group consisting of 4 agents, p, q, r and s, sending 3 messages, m_1, m_2 and m_3, to all members in order. In the previous inter-message causality ensuring protocols [2, 3], r cannot receive m_1 and m_2 except for delivering m_3 in this example due to their loss on the unreliable network assumed in this paper. In order to deliver all messages to a group of mobile agents reliably, our proposed protocol allows each member like p, q, s to buffer received messages in its memory, DLVD_Q. Also, to keep the causal ordering requirement, the protocol makes each member check causality relation violation. In case the received message arrives at a process later than its successors, all consecutive successors after the message in order are delivered to the corresponding application. If a member r receives a message m_3 from s without the receipt of its predecessors, m_1 and m_2, like in Fig. 2, it requests m_3's sender s gives the predecessors to itself by sending a solicitation message with m_3's dependency vector, $MVector_r$, and $MVector_s$ piggybacked on m_3 received from s. After having obtained m_1 and m_2 from s, r can deliver all three messages to their corresponding application despite the message loss like in the figure.

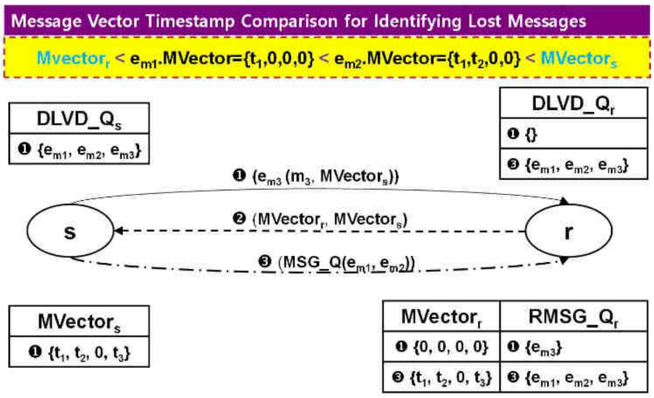

Fig. 2. Procedure for ensuring causality relation.

IV. COMPARISONS

Broadcast-based protocol [15] guarantees transparent and reliable inter-agent communication and can also provide multicast communication to a set of agents. But, to locate the message destination, the protocol has to contact every visiting node in the network. Thus, its large traffic overhead makes broadcasts impractical in large-scale distributed agent systems.

In home-based protocol [12] inspired by Mobile IP [16, 18], every mobile agent has a home node and should register its current location with the home node whenever it moves. Thus, when some messages are sent to a mobile agent currently located at a foreign node, the messages are first directed to its home node, which forwards them to the agent. This protocol is simple to implement and results in little mobile agent locating overhead. However, it is unsuitable for highly mobile agents in distributed agent based systems because every agent location updating and message delivery are all performed around the home node, which introduces centralization. Additionally, in the distributed agent-based systems, the home node may be disconnected from the network.

Forwarding pointer-based protocol [13] forces each node on a mobile agent's movement path to keep a forwarding pointer to the next node on the path. Thus, if a message is delivered to an agent not being at the home node, the message must traverse a list of forwarders. Thus, this protocol can avoid performance bottlenecks of the global infrastructure, and therefore improve its scalability, particularly in large-scale distributed agent-based systems, compared with the home based one. Additionally, even if a home node is disconnected from the rest of the network, the forwarding pointer based protocol allows agents registering with the node to communicate with other agents. However, as highly mobile agents lead to the length of their chains of pointers being rapidly increasing, its message forwarding overhead may be significantly larger. Furthermore, the number of forwarding pointers each service node needs to keep on its storage may exponentially increase if a large number of mobile agents are running in the systems. In a previous work [13], a type of update message called inform message was introduced to include an agent's current location for shortening the length of trails of forwarding pointers. In this case, a node that receives the message is allowed to update its table if the received information is more recent than the one it had. However, it introduces no concrete and efficient solutions for this purpose, for example, when update messages should be sent, and which node they should be sent to. To consider failures of forwarders, a fault-tolerant directory service for mobile agents using redundancy of forwarding pointers [14] was proposed.

Mailbox-based protocol [4] was proposed to provide location-independent reliable message delivery. It allows messages to be forwarded at most once before they are delivered to their receiving agents. Also, the movement of agents can be separated from that of their mailboxes by determining autonomously whether each mailbox is migrated to its owner agent. However, uncertainty of message delivery to mailboxes may result in useless early polling. On the other hand, even if urgent messages are forwarded to a mailbox on time, they can be delivered to its corresponding agent very late depending on the agent's polling time. Moreover, whenever each mailbox moves, its new location information should be broadcasted to every node which the mailbox has visited. This may incur high traffic overhead if assuming most agents are highly mobile.

Macêdo and Silva in the work [11] defined mobile agent group concept and its membership mobility for the first time. However, the protocol just provides the conceptual definition of mobile agent group communication services and a sort of very limited virtual synchrony forcing each mobile agent to install and update a group view every migration. Also, it doesn't guarantee any message delivery order and incurs high overhead caused by agent mobility.

Sama [9] is a distributed and scalable application level group communication mechanism for large scale mobile agent applications which delivers messages in a considerably low time. It uses Message Dispatcher Objects (MDOs), which are special objects on every host, to parallelize and speed up message delivery to the group members. It can also detect host failures and restore from them. However, it doesn't address any total order or causal order message delivery requirement.

Hierarchical group communication system for mobile agents [20] was introduced to support totally ordered multicast and membership management functions [7]. In this system, a group of mobile agents form several agent clusters, and each agent cluster consists of all mobile agents residing in the same sub-network and is managed by a special module, named coordinator. Then, all coordinators form a ring-based overlay for interchanging messages between clusters.

In summary, all the protocols mentioned before cannot satisfy every properties required when designing reliable mobile agent group communication with causal message delivery order [2, 3]. Fig. 3 shows analysis results of the existing mobile agent communication protocols including ours. As you can see in this figure, our work can have all the five beneficial features unlike the others.

Protocol / Properties	Ours	GCS-MA [20]	SGCM [9]	MAG [11]	Broadcast [15]	Mobile IP inspired [12]	Forwarding pointer [13-14]	Mailbox [4]
Agent Mobility	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Fault-tolerance	Very High	High	High	High	Medium	Very Low	High	High
Causal Order	Yes	No	No	No	No	No	No	No
Group Concept	Yes	Yes	Yes	Yes	Yes (Partially)	No	No	No

Fig. 3. The important properties to satisfy for our goal.

V. CONCLUSION

In this paper, we analyzed several representative previous mobile agent communication protocols with our work in terms of reliability, scalability, causal ordering and group concept support. Also, we can verify our protocol has a lot of strengths on every issue against the existing ones. For future works, we should perform various experiments to precisely and quantitatively evaluate their scalability, adaptability and reliability by simulation. Moreover, we will extend our protocol to satisfy soft real-time requirement to run deadline-constrained collaborative applications on IoT systems.

ACKNOWLEDGEMENTS

This work was supported by Kyonggi University Research Grant 2013.

REFERENCES

[1] J. Ahn, "Mobile Agent Group Communication Protocol Ensuring Causal Order Semantics," In Proceedings of the 2014 International Conference on Smart Computing. IEEE Press, Hong Kong, China, pp. 269-274, 2014.

[2] R. Baldoni, "A Positive Acknowledgment Protocol for Causal Broadcasting," IEEE Transactions on Computers, vol. 47, pp. 1341-1350, 1998.

[3] R. Baldoni, S. Bonomi, M. Platania, and Querzoni. L., Dynamic message ordering for topic-based publish/subscribe systems. The 26th IEEE International Parallel and Distributed Processing Symposium, IEEE Press, Shanghai, pp. 909-920, 2012.

[4] J. Cao, X. Feng, J. Lu, S. Das, "Mailbox-based scheme for mobile agent communications," IEEE Computer, vol. 35, no. 9, pp. 54-60, 2002.

[5] M. Cimino, F. Marcelloni, "Autonomic tracing of production processes with mobile and agent-based computing. Information Sciences," vol. 181, no. 5, pp. 935-953, 2011.

[6] C. Cooper, R. Klasing, T. Radzik, "Locating and repairing faults in a network with mobile agents," Theoretical Computer Science, vol. 411, no. 14-15, pp. 1638-1647, 2010.

[7] X. Defago, A. Schiper, P. Urban., "Total Order Broadcast and Multicast Algorithms: Taxonomy and Survey," ACM Computing Surveys, vol. 36, no. 4, pp. 372-421, 2004.

[8] E. N. Elnozahy, L. Alvisi, Y.M. Wang, D.B. Johnson, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems," ACM Computing Surveys, vol. 34, no. 3, pp. 375-408, 2002.

[9] H. Jafarpour, N. Yazdani, and N. Bazzaz-zadeh, "A scalable group communication mechanism for mobile agents," Journal of Network and Computer Applications, vol. 30, pp. 186–208, 2007.

[10] K. Lin, M. Chen, S. Zeadally, J. Rodrigues, "Balancing energy consumption with mobile agents in wireless sensor networks," Future Generation Computer Systems, vol. 28, no. 2, pp. 446-456, 2012.

[11] R. Macêdo and F. Silva, "The mobile groups approach for the coordination of mobile agents," Journal of Parallel and Distributed Computing, vol. 65, pp. 275–288, 2005.

[12] D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, and J. White, "MASIF: The OMG Mobile Agent System Interoperability Facility," In: Rothermel, K., Hohl, F. (eds.): Mobile Agents. Lecture Notes in Computer Science, vol. 1477. Springer-Verlag, Berlin Heidelberg New York, pp. 50-67, 1998.

[13] L. Moreau, "Distributed Directory Service and Message Router for Mobile Agents," Science of Computer Programming, vol. 39, no. 2-3, pp. 249-272, 2001.

[14] L. Moreau, "A Fault-Tolerant Directory Service for Mobile Agents based on Forwarding Pointers," In Proc. of The 17th ACM Symposium on Applied Computing, pp. 93-100, 2002.

[15] A. L. Murphy and G. P. Picco, "Reliable Communication for Highly Mobile Agents," Journal of Autonomous Agents and Multi-Agent Systems, vol. 5, no. 1, pp. 81-100, 2002.

[16] C. Perkins, IP Mobility Support for IPv4. RFC 3344, August 2002.

[17] R. D. Schlichting and F. B. Schneider, "Fail-stop processors: an approach to designing fault-tolerant distributed computing systems," ACM Transactions on Computer Systems, vol. 1, pp. 222-238, 1985.

[18] T. C. Schmidt, M. Wahlisch and G. Fairhurst, Multicast Mobility in MIPv6: Problem Statement and Brief Survey. IRTF Internet Draft, no. 08, August 2009.

[19] C. Su, C. Wu, “JADE implemented mobile multi-agent based, distributed information platform for pervasive healthcare monitoring,” Applied Soft Computing, vol. 11, no. 1, pp. 315-325, 2011.

[20] W. Xu, J. Cao, B. Jin, J. Li, and L. Zhang, “GCS-MA: A group communication system for mobile agents,” Journal of Network and Computer Applications, vol. 302, pp. 1153–1172, 2007.