CAUSALLY ORDERED MESSAGE DELIVERY PROTOCOL FOR MOBILE AGENT GROUP CONSIDERING RELIABILITY

JINHO AHN

Dept. of Computer Science, Kyonggi University, Republic of Korea
E-mail: jhahn@kgu.ac.kr

Abstract- A swarm of mobile agents may be used as a vital medium for dynamically collaborative distributed applications based on unreliable networks requiring reliable causally ordered message delivery. However, the existing message delivery protocols for mobile agent groups may not satisfy this important requirement. In this paper, we propose a new causally ordered mobile agent group communication protocol to achieve all the requirements mentioned above as follows. First, to guarantee agent communication reliability to a maximum despite agent location managers' failures, the location information of each agent can be replicated in an effective way to degrade its scalability to a minimum despite agent mobility. Second, it guarantees all messages sent to a mobile agent group to be reliably delivered to its surviving group members in a consistent causal order.

Keywords- Distributed System, Group Communication, Causal Message Ordering, Mobile Agent.

I. INTRODUCTION

Mobile agent group concept is a versatile software paradigm capable of providing group task execution flexibility and collaborative adaptability in various application fields such as P2P file replication for high assurance, multimedia games, video conferencing, message-based social networking, and so on [4, 5, 8]. However, as the size of these fields is rapidly increasing, improving the performance of the agent group communication in Internet-scale infrastructures should be reconsidered to be suitable for their scale. For this purpose, some appropriate inter-agent group communication protocols are required in distributed agent-based systems [1, 6, 7, 9].

Also, each member of the agent group may encounter the situation where it has to make a decision on its own actions by itself only with all the messages received from the group. At this time, in order to locally choose the best among them, the inter-message causality relation information broadcast from its group may be very crucial.

For this purpose, causal order delivery to a broadcast group [2, 3] can be used and satisfied if send events of any two messages have causality relation and the same destination, their corresponding delivery events should occur on the destination in their sending order. In order to ensure this ordering constraint, most of conventional protocols behave in common as follows; if a group member receives a message capable of violating the constraint, the message delivery to the application is forced to wait for releasing the restriction caused by its predecessors [2, 3].
Generally, they are operating differently according to their pursuing preferences; optimizing the amount of the control information piggybacked on each application message to a group or reducing the delivery time of the message to the corresponding application.



Fig. 1. The important properties of the proposed protocol.

A swarm of mobile agents may be used as a vital medium for dynamically collaborative distributed applications based on unreliable networks requiring reliable cause-effect ordered message delivery. However, the existing message delivery protocols for mobile agent groups [1, 6, 7, 9] may not satisfy this important requirement. In this paper, we propose a new causally ordered mobile agent group communication protocol to achieve all the requirements mentioned above like in Fig. 1. First, the protocol allow each mobile agent to autonomously replicate trails of forwarding pointers only on a small number of appropriate visiting nodes. Second, it ensures delivering reliably and causally each message to not only one mobile agent, but also an agent group regardless of agent mobility and failures of location managers. In addition, it provides short-cuts to its target group by each sending agent keeping its effective group location information table.

II. SYSTEM MODEL

This paper considers an asynchronous distributed agent basedsystem where there is no global memory, no global clock and nobound on message delay. The system consists of a set of agentservice nodes. Each service node supports an environment in whichagents can operate safely and securely, and provides a uniform setof services through which visiting agents

can access its local resources ina limited way regardless of their locations. An agent is initiallycreated on a service node, called home node of the agent, andis given a unique identifier within the node. So, each agent canbe identified as a globally unique object in the system by usingthe combination of its local identifier and the identifier of itshome node. When an agent migrates in the system, its code andstate information are captured and then transferred to the nextnode. After arriving at the node, the mobile agent resumes andperforms its task, if needed, by interacting with other agents. Inorder to perform an assigned task on behalf of a user, a mobileagent AID executes on a sequence of l(l>1) service nodesaccording toits itinerary, I_{AID}=[DN_{home}, DN_1, …, $DN_{(l-1)}$], which may be statically determined before the mobile agent is created on its home node or dynamically whileprogressing its execution. It is assumed that communicationchannels support standard asynchronous message passing with the reliable FIFO ordering property and areimmune to partitioning. Mobile agents canmigrate 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 memoriesand stop their executions [17].

III. THE INTER-MESSAGE CAUSALITY ENSURING MOBILE AGENT GROUP COMMUNICATION PROTOCOL

In order to satisfy the first requirement, the following method is used in the proposed protocol. 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 k (k≥1) node crash failures, (k+1) locaters of each mobile agent should exist. Our location updating algorithm is designed to guarantee reliable delivery to mobile agents despite both agent mobility and k failures of forwarders and minimize location information replication overhead like in [1] except for assuming unreliable network and there is no group leader for sequencing order of messages to a group. The second requirement is satisfied in the protocol as follows. Suppose there is a mobile agent group consisting of 4 agents, p, q, r and s, sending 3 messages, m1, m2 and m3, to all members in order (by executing Module B-SEND(m)). In the previous inter-message causality ensuring protocols [2, 3], r cannot receive m1 and m2 except for delivering m3 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, $DLVD_Q_{rcvr}$ (by executing Module B-RECV(m, $MVector_{sndr}$)). 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 (by executing Module CHECK-MSGS()). If a member r receives a message m3 from s without the receipt of its predecessors m1 and m2 like in Fig. 2, it requests m3's sender s gives the predecessors to itself by sending a solicitation message with m3's dependency vector, $MVector_{rcvr}$, and $MVector_{sndr}$ piggybacked on m3 received from s (by executing Module SOLICIT-RECV($MVector_{rcvr}$, $MVector_{upper}$)). After having obtained m1 and m2 from s, r can deliver all three messages to their corresponding application despite the message loss like in the figure (by executing Module RPY-RECV(MSG_Q)).

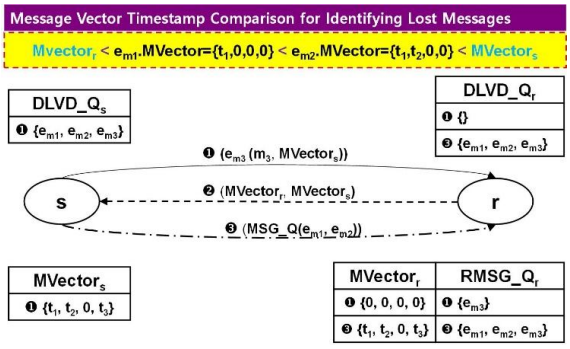


Fig. 2. Interaction procedure for ensuring causality relation.

Module B-SEND(m) OF P_{sndr}
$MVector_{sndr}[sndr] \leftarrow$ current time value of P_{sndr} ;
broadcast (m, $MVector_{sndr}$) to all the other members ;

Module B-RECV(m, $MVector_{sndr}$) OF P_{rcvr}
if($MVector_{sndr}[sndr] \leq MVector_{rcvr}[sndr]$) then
 discard message m;
elseif(($MVector_{sndr}[sndr] > MVector_{rcvr}[sndr]$)∧
($\forall i \neq sndr: MVector_{sndr}[i] \leq MVector_{rcvr}[i]$)) then
 $\forall i$: $MVector_{rcvr}[i] \leftarrow \max(MVector_{rcvr}[i], MVector_{sndr}[i])$;
 deliver m to its corresponding application ;
 insert (m, $MVector_{sndr}$) into $DLVD_Q_{rcvr}$ in m's sending time order ;
 call CHECK-MSGS() ;
else
 insert (m, $MVector_{sndr}$) into $RMSG_Q_{rcvr}$ in m's sending time order ;
 send solicitation($MVector_{rcvr}$, $MVector_{sndr}$) toP_{sndr} ;

Module CHECK-MSGS() OF PROCESS P_p
for all e ∈ $RMSG_Q_p$ in FIFO order do
 if(e.$MVector_j[j] > MVector_p[j] \wedge \forall i \neq j$:

e.MVector$_j$[i] ≤ MVector$_p$[i]) then
∀i:MVector$_p$[i] ← max(MVector$_p$[i], e.MVector$_j$[i]) ;
deliver e.m to its corresponding application ;
insert e into DLVD_Q$_p$ in e.m's sending time order ;
remove e from RMSG_Q$_p$;

Module SOLICIT-RECV(MVector$_{rcvr}$, MVector$_{upper}$) OF P$_{sndr}$
MSG_Q ←Φ ;
for all e ∈ DLVD_Q$_{sndr}$ in FIFO order st ((e.MVector < MVector$_{upper}$)∧(e.MVector > MVector$_{rcvr}$)) do
insert e into MSG_Q in e.m's sending time order ;
send reply(MSG_Q) toP$_{rcvr}$;

Module RPY-RECV(MSG_Q) OF P$_{rcvr}$
RMSG_Q$_{rcvr}$ ← RMSG_Q$_{rcvr}$∪MSG_Q in m's sending time order ;
call CHECK-MSGS() ;

Fig. 3. Modules for inter-message causality ensuring mobile agent group communication protocol.

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 [18, 21], 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 conceptualdefinition 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. 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.

Atomic mobile agent group communication protocol [1] was proposed to allow messages destined to a group of agents to be reliably delivered to all surviving agents in the same order despite their mobility and directory nodes' failures. Also, it replicates paths on which messages should be transmitted to their targeting mobile agents in an effective manner. Furthermore, mobile agent group location cache each sending agent keeps can considerably speed up message delivery to a group of agents and lower message forwarding load imposing on directory nodes.

CONCLUSION

We presented a new causal order ensuring mobile agent group communication protocol to achieve all the requirements mentioned earlier. First, the protocol allow each mobile agent to autonomously replicate trails of forwarding pointers only on a small number of appropriate visiting nodes. This feature maylower message forwarding load imposing on forwarders. Second, it ensures delivering reliably and causally each message to not only one mobile agent, but also an agent group regardless of agent mobility and failures of location managers. In addition, it provides short-cuts to its target group by each sending agent keeping its effective group location information table. This feature may speed up message delivery to a group of agents.

ACKNOWLEDGEMENT

This work was supported by Kyonggi University Research Grant 2014(Project Number: 2014-011).

REFERENCES

[1] Ahn, J.: Lightweight Fault-tolerance Mechanism for Distributed Mobile Agent-based Monitoring, In Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference. IEEE Press, Las Vegas, NV, USA, 1232-1236. (2009)

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

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

[4] Cao, J., Feng, X., Lu, J., Das, S.: Mailbox-based scheme for mobile agent communications. IEEE Computer, Vol. 35, No. 9, 54-60. (2002)

[5] Cimino, M., Marcelloni, F.: Autonomic tracing of production processes with mobile and agent-based computing. Information Sciences, Vol. 181, No. 5, 935-953. (2011)

[6] Cooper, C., Klasing, R., Radzik, T.: Locating and repairing faults in a network with mobile agents. Theoretical Computer Science, Vol. 411, No. 14-15, 1638-1647. (2010)

[7] Defago, X., Schiper, A., Urban., P.: Total Order Broadcast and Multicast Algorithms: Taxonomy and Survey. ACM Computing Surveys, Vol. 36, No. 4, 372-421. (2004)

[8] Elnozahy, E. N., Alvisi, L., Wang, Y. M., Johnson, D. B.: A Survey of Rollback-Recovery Protocols in Message-Passing Systems. ACM Computing Surveys, Vol. 34, No. 3, 375-408. (2002)

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

[10] Lin, K., Chen, M., Zeadally, S., Rodrigues, J.: Balancing energy consumption with mobile agents in wireless sensor networks. Future Generation Computer Systems, Vol. 28, No. 2, 446-456. (2012)

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

[12] Milojicic, D., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B., Kosaka, K., Lange, D., Ono, K., Oshima, M., Tham, C., Virdhagriswaran, S., White, J.: 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, 50-67. (1998)

[13] Moreau, L.: Distributed Directory Service and Message Router for Mobile Agents. Science of Computer Programming, Vol. 39, No. 2-3, 249-272. (2001)

[14] Moreau, L.; A Fault-Tolerant Directory Service for Mobile Agents based on Forwarding Pointers. In Proceedings of the 17th ACM Symposium on Applied Computing. ACM New York, Madrid, Spain, 93-100. (2002)

[15] Murphy, A. L., and Picco, G. P.: Reliable Communication for Highly Mobile Agents. Journal of Autonomous Agents and Multi-Agent Systems, Vol. 5, No. 1, 81-100. (2002)

[16] Perkins, C.: IP Mobility Support for IPv4, RFC 3344, (2002)

[17] Schlichting, R. D., Schneider, F. B.: Fail-stop processors: anapproach to designing fault-tolerant distributed computingsystems. ACM Transactions on Computer Systems, Vol. 1, No. 3, 222-238. (1985)

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

[19] Su, C., Wu, C.: JADE implemented mobile multi-agent based, distributed information platform for pervasive healthcare monitoring. Applied Soft Computing, Vol. 11, No. 1, 315-325. (2011)

[20] Xu, W., Cao, J., Jin, B., Li, J., and Zhang, L., GCS-MA: A group communication system for mobile agents. Journal of Network and Computer Applications, 30 (2007), 1153–1172.

★★★