

Distributed System in Mobile Agent Communication

M. Dhinakaran*, S. Sivapriya**, K. Thirugnanasoundari*** & V. Vanmathi****

*Assistant Professor, Department of Electronics and Communication Engineering, S.K.P Engineering College, Tamil Nadu, INDIA.
E-Mail: dhina.ece2010{at}gmail{dot}com

**UG Student, Department of Electronics and Communication Engineering, S.K.P Engineering College, Tamil Nadu, INDIA.
E-Mail: sivapriya1517{at}gmail{dot}com

***UG Student, Department of Electronics and Communication Engineering, S.K.P Engineering College, Tamil Nadu, INDIA.
E-Mail: thirugnanasoundari96{at}gmail{dot}com

****UG Student, Department of Electronics and Communication Engineering, S.K.P Engineering College, Tamil Nadu, INDIA.
E-Mail: vanmathiv29{at}gmail{dot}com

Abstract—Agents are applicable to many computing areas like distributed systems because of their autonomy and Mobility factors. Reliable and efficient communication between agents in a mobile agent system is a challenging task. To deliver the messages efficiently and reliably, mobile agent communication schemes generally have two phases. The first phase is agent tracking phase which deals with locating the agent current location. Second phase deals with the delivery of message to the agent at its current location. The existing approaches suffer from high communication overhead or problems such as message chase problem, triangle problem, or delayed delivery of messages. Approaches such as I-update deliver the message very efficiently without any overhead in message delivery phase but increase the memory and network Overhead in agent tracking phase. In this paper, we present a novel approach that minimizes message delivery delay while maintaining a balance between the costs of agent tracking phase and message delivery n phase. Our approach enhances the home agency scheme by maintaining track of agent migrations and costs of message delivery through different channels at the home agency. This approach allows the home agency to send the message to an agent via optimal transmission cost path, which minimizes message delivery delay and reduces the probability of occurrence of message chase problem.

Keywords—Agent Tracking; Communication Overhead; Distributed System; Message Delivery; Mobile Agent.

Abbreviations—Mobile Agent (MA); Message Efficiently Forwarding Scheme (MEFS); Remote Evaluation (REV); Remote Procedure Call (RPC).

I. INTRODUCTION

THE term “agent” is heard frequently today. Mobile agents refer to self-contained and identifiable computer programs, bundled with their code, data and execution state, that can move within a heterogeneous network of computer systems. They can suspend their execution at an arbitrary point and transport themselves to another computer system. During this migration, the agent is transmitted completely, that is, as a set of code, data and execution state. At the destination computer system, an agent’s execution is resumed at exactly the point where it was suspended before. Mobile agents are used in a wide area of applications like network management and monitoring such as processing data over unreliable networks [Islam, 8], information searching and filter such as distributed Database Management System, Mobile Media, Internet, Intrusion Detection, Parallel Programming Paradigms such as Java

Symphony the ability of a mobile agent to personify their creators intentions 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 online 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. Although mobile agents have recently attracted widespread interest, several development obstacles remain. The technology still lacks secure and efficient execution support, standardization, appropriate programming languages, and coordination models.

II. OPENCV

2.1. A Survey on Mobile Agent Communication Protocols

An agent is a self decision making software entity which acts on behalf of the user. A mobile agent roams the internet in

order to access different services. To attain large goals different agents work and communicate together in order to achieve goal efficiently. Reliable communication between multi mobile agents is still a challenge. For reliable communication, sender must current location in order to deliver the message. Mobile location management consists of two phases. First one is tracking phase and other is message delivery phase, both of these phases have their own overheads. Agent communication failure can occur due to the triangle problem or message chase problem. The message chase problem arises in highly mobile agents. Due to the message chase problem, a message may never be delivered. To deliver the message to agent in a timely and reliable manner is one of the challenging issues in multi mobile agent environments. In this paper, we discuss and analyze existing location management schemes. The existing approaches lever up the overhead in first phase but trim down the overhead in second phase or vice versa. The overhead could be in the form of number of location update messages, memory or delay. Mobile agents work in a team to complete a given task assigned by the user. Mobile agents must coordinate and cooperate with each other for decision making and information sharing. To accomplish their task, the agents roam over the network to acquire different services. To accomplish their task, the agents roam over the network to acquire different services. The existing mobile agent location management schemes have many issues like triangle problem, and message chase problem etc., which affect the efficiency of identifying agent current location and delivering a message. In agent oriented environment there are parent agents (called master agents) and child agents (called slave agents). Master agent divides a heavyweight task into smaller subtasks and assigns them to slave agents. Master and slave agents need to communicate with each other in a reliable and timely manner to complete their assigned task. To deliver the message efficiently and reliably, mobile agent communication schemes generally have two phases. The first phase is agent tracking phase which deals with locating the agent current location. Second phase deals with the delivery of message to the agent at its current location. Several techniques have been proposed like central server, forward pointers, broadcast or multicast, home agency, hierarchical approach, intelligent update, for agent communication management systems [Ahmed & Nadeem, 1].

2.2. Applications of Mobile Agents in Healthcare Domain: A Literature Survey

Mobile agent technology is emanating as a new paradigm in the area of distributed and mobile computing, and has been engaged in many areas from network management tasks to information management. It makes considerably simpler to model, implement, and maintain distributed systems, including sensor networks and cloud computing. It provides an infrastructure not only for executing autonomous agents but also migrating them between different computers. In the recent years, mobile agent paradigm has emerged as a viable approach for the evolution of autonomic systems in the

healthcare domain. This paper reviews the applicability and usefulness of mobile agents in a wide variety of healthcare applications such as medical data management, medical information retrieval, health data integration, decision - making support, telemedicine, securing medical information and coordination of distinct medical activities.

The word —Mobile Agent‡ unifies two words—Mobility‡ and —Agent‡. A mobile agent is a set of executable programs that migrate from one machine to another (hence mobile) in heterogeneous networks representing users in various tasks (hence agent). Consequently, mobile agents are not limited to the computer where they are created, but they can move freely among computers [10]. The mobile agent can defer its execution at any arbitrary point, jumps to another machine and recommences the execution there [9]. Prior to the appearance of Mobile Agents, many approaches have been proposed and developed for communication between client and server such as message passing, remote procedure call and remote evaluation. In Remote Procedure Call (RPC), the client sends the data as parameters to a procedure that resides at the server. The procedure will be executed on the server and the results will be sent back to the client. In Remote Evaluation (REV), the procedure itself is sent from the client to the server to be executed rather than calling a remote procedure at the server and server returns the result. In contrast, a mobile agent (MA) is a program (encapsulating code, data and state) sent by a client to the server [1]. The basic process of three paradigms is depicted diagrammatically in figure 1. A basic difference between client-server and mobile agent paradigm is (figure 1): in client-server paradigm, there are number of interactions between client and server in form of requests and replies which increases the load of network but this load reduces in mobile agent paradigm as the only interaction of client is with mobile agent which migrates to server site and make local interactions with it [Braun & Rossak, 6].

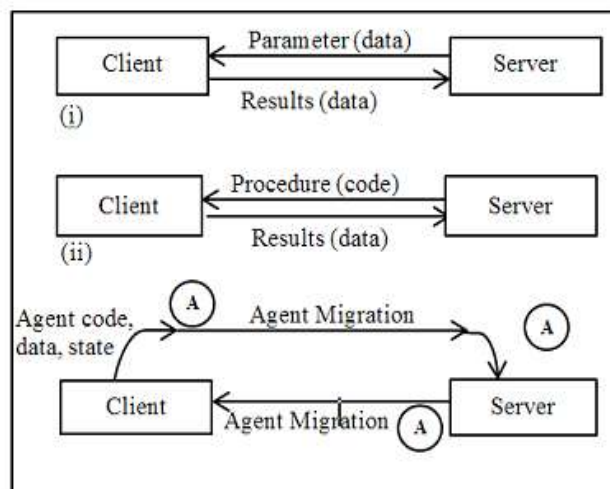


Figure 1: Basic Process

III. EXISTING

In previous work, we used in Message Efficiently Forwarding Scheme approach (MEFS) framework. In Message Efficiently Forwarding Scheme approach MEFS mobile agents are allowed to communicate seamlessly regardless of their location, each agent is assigned a global unique name when it initializes from which its home places address can be easily resolved. When migrating, agent must unregister, leaving a forwarding pointer, when it leaves a node and register when arrives at a new one. To solve the message chasing problem, they use synchronous communication when racing occurs but constrain agent autonomous migration as little as possible. In MEFS, each agent must calculate the velocity of 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 V_0 , the agent should establish a connection with its home place, 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. The drawback of this approach is the blockage of agent at time of fast mobility which affects the execution performance of agent. This approach suffers from really inefficient phase when the mobility and the number of messages increases for an agent. The above mentioned approaches manage the problem of location management system. Each approach discussed above has their overheads while managing agent tracking and message delivery phase. I-update approach efficiently manages the message delivery phase but create extra memory and network overhead by storing transmission cost of every host in network to each host in the network [Baumann, 4].

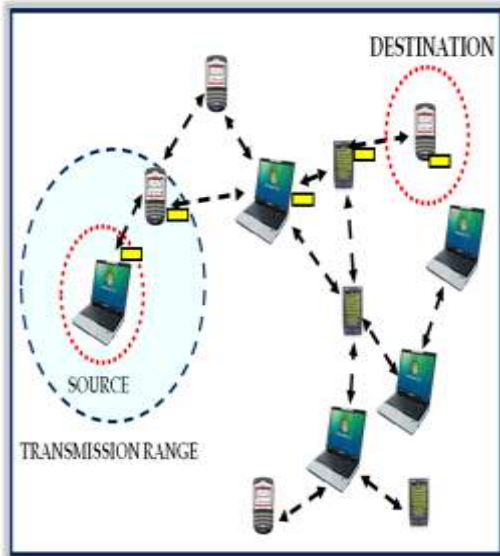


Figure 2: Existing System

3.1. Disadvantages

The drawback of this approach is the blockage of agent at time of fast mobility which affects the execution performance of agent. This approach suffers from really inefficient phase when the mobility and the number of messages increases for an agent.

The existing approaches suffer from high communication overhead or problems such as message chase problem, triangle problem, or delayed delivery of messages. Approaches such as I-update deliver the message very efficiently without any overhead in message delivery phase but increase the memory and network overhead in agent tracking phase [Desbiens et al., 7].

IV. PROPOSED SYSTEM

In this paper, we present a novel approach that minimizes message de-livery delay while maintaining a balance between the costs of agent tracking phase and message delivery phase. Our approach enhances the home agency scheme by maintaining track of agent migrations and costs of message delivery through different channels at the home agency. This approach allows the home agency to send the message to an agent via optimal transmission cost path, which minimizes message delivery delay and reduces the probability of occurrence of message chase problem. Our proposed approach uses home based approach (home agency and forward pointing) with some modification to reduce the overheads of home agency and forward pointing approaches. Home agency normally keeps the current location of the agent in the network. But in addition, proposed approach will keep the record of agent visited locations. Due to the long forward chain towards agent current location, and sending message over high transmission cost path, the cost of transmitting interaction messages becomes high. The purpose of our approach is to send the message over optimal path towards agent's current location. Home agency will calculate the optimized path to send message to its current location. In tracking phase our approach performs two tasks on each migration agent. First, agent sets a pointer to its next location on which agent wants to migrate, at its current location, just before departure to its next host. Second, agent sends one location update message about its current location to its home agency, just after arrival to the next host, describes the agent migration from one host to another in the network and exchange of messages between hosts on each migration. The small dotted lines in, the location update messages and solid smooth lines show the agent migration path as well as the forward pointers direction set by the agent on each migration [Li & Lam, 10; Marzougui & Barkaoui, 11].

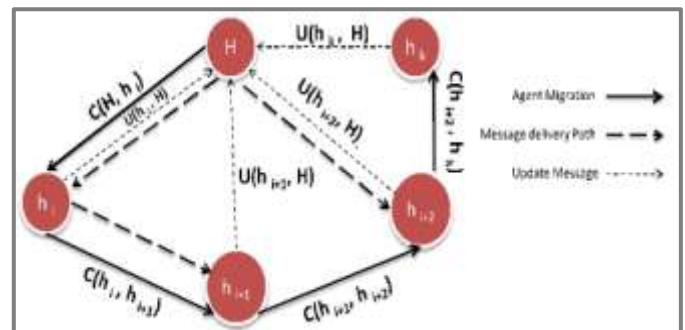


Figure 3: Proposed System

4.1. Advantages

- In mobile agent communication it is necessary to keep a balance between the costs of both phases.
- Approaches such as I-update deliver the message very efficiently in message delivery phase but it increases the memory and network overhead in agent tracking phase.
- Similarly in message broadcast approach the message delivery phase
- Have large network overhead than agent tracking phase.
- Increase of cost in one phase of agent communication affects the overall performance of the communication.

V. BLOCK DIAGRAM

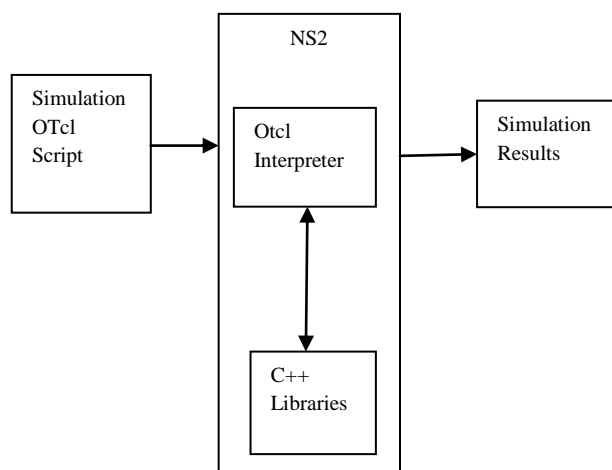


Figure 4: Block Diagram

VI. INTRODUCTION TO UBUNTU

Ubuntu is a Debian-based Linux operating system, with Unity as its default desktop environment. It is based on free software and named after the Southern African philosophy of Ubuntu (literally, “human-ness”), which often is translated as “humanity towards others” or “the belief in a universal bond of sharing that connects all humanity”. According to some metrics, Ubuntu is the most popular Linux distribution.

6.1. Benefits of Ubuntu

- Low cost
- Performance
- Flexibility
- Compatibility
- Choice
- Fast and easy installation
- Multitasking
- Security
- Open Source

VII. CONCLUSION

The existing approaches to agent communication suffer from high communication overhead or problems such as message chase problem, triangle problem, or delayed delivery of messages. Recent approaches like I-update and ratio-based approach address these problems but they do not perform efficiently in both phases. In mobile agent communication it is necessary to keep a balance between the costs of both phases. Approaches such as I-update deliver the message very efficiently in message delivery phase but it increases the memory and network overhead in agent tracking phase. Similarly in message broadcast approach the message delivery phase have large network overhead than agent tracking phase. Increase of cost in one phase of agent communication affects the overall performance of the communication.

Existing communication models are application specific. This means that the choice of communication model/protocol is based on the high or low MRR in the applications. In some applications agents send more interaction messages with less migration. But in some applications agent send fewer messages with extensive migration. So it is the task of an application designer to select an appropriate communication technique at design time. Even if the designer selects a communication protocol at design time, later changes in the nature of application may needs to change the communication protocol to improve performance of application. The specific problem of our interest is how to design a generalized or balanced approach for all applications regardless of MRR. In addition, the proposed design is also able to minimize message delivery delay and reduces the occurrence of message chase problem while maintaining a balance between the costs of agent tracking phase and message delivery phase, when two or more than two mobile agents communicate with each other. The proposed approach that we discussed in this paper is a balanced approach and allows the home agency to send the message to an agent via optimal transmission cost path, which minimizes message delivery delay and reduces the probability of occurrence of message chase problem [Jingyang et al., 9].

7.1. Future Enhancement

In future, we have planned to test the proposed approach on some real agent oriented application to check its reliability and scalability. For this, our first step will be to implement proposed approach in any mobile agent based framework like JavaSymphony [Aleem et al., 2] or JADE (Java Agent Development Framework) [Bellifemine et al., 5]. Further, for testing the proposed approach, we will implement a health care application [Bagga & Hans, 3] over these Java based mobile agent frameworks.

REFERENCES

- [1] S. Ahmed & A. Nadeem (2012), “A Survey on Mobile Agent Communication Protocols”, *International Conference on Emerging Technologies (ICET)*, Pp. 1–6.

- [2] M. Aleem, R. Prodan & T. Fahringer (2012), “The Javasympphony Extensions for Parallel GPU Computing”, *41st International Conference on Parallel Processing (ICPP)*, Pp. 30–39.
- [3] P. Bagga & R. Hans (2015), “Applications of Mobile Agents in Healthcare Domain: A Literature Survey”, *International Journal of Grid and Distributed Computing*, Vol. 8, No. 5, Pp. 55–72.
- [4] J. Baumann (1999), “A Comparison of Mechanisms for Locating Mobile Agents”, *Technical Report 1–25. Univ. of Stuttgart, Faculty of Computer Science*.
- [5] F. Bellifemine, F. Bergenti & G. Caire (2005), “Jade — A Java Agent Development Frame-work”, In: Chapter 9, 15. *Springer US*, Boston, MA, Pp. 125–147.
- [6] P. Braun & W. Rossak (2005), “Mobile Agents Basic Concepts, Mobility Models, and the Tracy Toolkit”, *Morgan Kaufmann Publishers Inc*.
- [7] J. Desbiens, M. Lavoie & F. Renaud (1998), “Communication and Tracking Infrastructure of a Mobile Agent System”, *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, Vol. 7, Pp. 54–63.
- [8] M.A. Islam (2012), “Routing Issues in Opportunistic Networks: Evolution of Delay Tolerant Networks from Mobile Ad-hoc Networks to Opportunistic Networks”, *LAP Lambert Academic Publishing*, Germany.
- [9] Z. Jingyang, J. Zhiyong & C. Daoxu (2003), “Designing Reliable Communication Protocols for Mobile Agents”, *Proceedings of 23rd International Conference on Distributed Computing Systems Workshops*, Pp. 484–487.
- [10] T.Y. Li & K.Y. Lam (2002), “An Optimal Location Update and Searching Algorithm for Tracking Mobile Agent”, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2. ACM*, New York, NY, USA, Pp. 639–646.
- [11] B. Marzougui & K. Barkaoui (2013), “Interaction Protocols in Multi-Agent Systems based on Agent Petri Nets Model”, *International Journal of Advanced Computer Science & Applications*, Vol. 4, No. 7, Pp. 166–173.
- [12] K. Obraczka & F. Silva (2000), “Network Latency Metrics for Server Proximity”, *IEEE Global Telecommunications Conference*, Vol. 1, Pp. 421–427.