Location Monitoring In Mobile Agents System: A Survey

Jasmeen Kaur
Dept. of Computer Science and Engineering
Guru Nanak Dev University
Amritsar, India

Ramandeep Kaur
Dept. of Computer Science and Engineering
Guru Nanak Dev University
Amritsar, India

Rahul Hans
Dept. of Information and Technology
D.A.V. Institute of Engineering and Technology, Jalandhar, India

ABSTRACT

Mobile agent technology is emerging as a new paradigm in the area of distributed and mobile computing. Mobile agents play an important role in the development of active and dynamically managed networks and distributed systems. Mobile agents are the intelligent programs that act autonomously on behalf of a user and can migrate from one host to another host in a network in order to satisfy the requests made by their clients. In any mobile agent system, the ability to communicate with agents in real-time, as agents move from one node to another, is essential for retrieving any data or information that they have collected, and for supporting coordination and cooperation among them. So the critical problem in managing a mobile agent system is to track the location of the agents. Communication with a mobile agent incorporates the ability to locate it. Thus locating agents efficiently is the issue central to any mobile agent system. This paper surveys various mobile agent location monitoring techniques, proposed by various authors.

Keywords

Mobile Agents; Location tracking; Itinerary; Message passsing;

1. INTRODUCTION

An agent-based system is a distributed computing environment in which mobile autonomous processes called mobile agents operate on behalf of users [1]. The flexibility offered by mobile agents is quite noticeable in distributed computing environments [20]. Mobile agents are programs which are dispatched from a source computer and run among a set of networked servers until they are able to accomplish their task. Mobile agent computing paradigm is different from others because not only data but the code acting on the data is also transported among the nodes which makes the developed application more flexible. Mobile agents are proactive, reactive and cognitive [2]. An agent can suspend its execution, migrate to other node and restart its execution there at the other node.

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, multimedia, Internet, Intrusion Detection, Military, telecommunications, Secure brokering such as untrustworthy collaborators etc [19].

In a distributed system the clients have to access network resources in a prescriptive way by interacting with individual servers probe by probe in the network to accomplish a complicated task. Thus, it is very difficult to accomplish a complicated task that requires its client to interact with multiple servers intensively [4]. In mobile agents system, the ability to get the status information from the agents when an agent is moving in its itinerary and collecting information from individual hosts is an important issue in order to decide the further behavior of the agent depending upon the current situation of the agent. To request such information from an agent, the agent's location has to be known, which implies that the agent has to be located by the system [3].

The monitoring of mobile agent based distributed systems involves the collection, interpretation, and display of information concerning the interactions among concurrently executing components [16]. This information and its display can support the debugging, testing, performance evaluation, and dynamic documentation of such systems.

In general terms, monitoring is defined as the essential means for obtaining the information required about the components of a distributed system in order to make management decisions and subsequently control their behavior [15]. In such systems the components may evolve during the course of computation and problem solving; therefore static design of monitoring identified at design time may not prove appropriate. Thus it need to dynamically monitor host agents, in order to maximize the reliability and the availability of overall mobile agent based distributed applications.

After an agent is submitted into a service network, the client or the network manager may need to know its current location in order to inquire its status, or to control its execution, etc. A simple way is to send another agent called search agent, to search the original agent along the original path or to send a message to every server where the agent might have visited.

But there are some problems associated with these straightforward solutions like sending many messages over a wireless network may be too expensive, the path that an agent took may be nondeterministic such that it is difficult to trace, a sequential search may take too much time [4].

There have been many approaches proposed by various authors to deal with this problem of communication, any designed approach for location transparent communication must provide solutions to [5] agent tracking i.e. keep track of current agent location to make it possible to find the agent. and message delivery i.e. the message sent to the mobile agent must not be lost, it must be delivered to the current agent location.

The rest of the paper is organized as follows. Section 2 provides an overview of related work and discusses some of the basic methods of agent tracking in mobile agents system. Section 3 discusses some of the existing location tracking mechanisms proposed by various authors in mobile agents system and section 4 briefly gives the conclusion and future work.

2. RELATED WORK

A mobile agent migrates from one node to another while performing a task autonomously on behalf of a user. When a mobile agent migrates, it chooses the next destination according to an itinerary, which is a predefined travel plan, or dynamically according to execution results. Because of the extremely dynamic nature of mobile agents, mobile agent systems not only feature mobility, but also a technique to discover their position at any time. Locating agents efficiently is thus an issue central to any mobile agent system [6, 16].

The basic need to locate an agent that has been previously created at the home server is that:

i. To stop a running agent or start a new agent.
ii. To stop a running agent or start a new agent.
iii. For the development of agent-based applications, in geographically distributed environment.
iv. For fault tolerance in mobile agents.
v. To detect the malicious hosts in the itinerary for security.

The basic solutions to the problem of location tracking involve:

i. The existence of a central host where all the movements of the agents are constantly updated.
ii. Having each agent leave a trace of its movements on its way.

Both solutions present obvious disadvantages. In the centralized solution the biggest problems are related to fault tolerance and security and the central database could crash resulting in a complete loss of the information. The second solution consists of leaving at each node the indication of the node where the agent has moved. This has to be done for each agent possibly incurring in a high space complexity to store all the information [7].

There have been many methods to keep track of the location of the mobile agents in its itinerary. The basic methods to keep track of the location of mobile agents in its itinerary are [8]:

I. Full Information Approach

In [5] every agency has full knowledge about the current location of all agents in the system. For each migration, all agencies in the network must update their local location directory. So agent tracking becomes a very expensive task, while, message delivery is an easy task, like the broadcast method.

II. No Information Approach

In this method no agent tracking is established in this method, therefore no agency has direct knowledge of the current location of any agent other than those residing locally. Agent migration is very cheap but message delivery is very expensive [5].

III. Footprints Mechanism

In this method the agent leaves the record after it was at a particular server. When mobile agent migrates to the next host, it will leave a record on the current host. The disadvantage of this approach is when it has more than one mobile agent on the network, the mechanism had to consider for the problem that one agent can delete the footprints record of another agent on agent system. If it deletes a record earlier, the user can’t track the mobile agent location [8].

IV. EMFS (efficiently message forwarding scheme) communication mode

In this mechanism [18] the Mobile Agent platform is described as a "Mobile Agent Space" with each node "Place" set with Home module and Communicate module.

Home module records the Agent name and the current location of the Agent in the process of moving generated by the local nodes. Communicate module records the Agent name which stays in the current node and the Agent name, target, time which leaves from the local node.

To achieve addressing and message forwarding through the Communicate module, and auxiliary addressing through Home module. In order to the mechanism avoids message chasing by setting "Timeout" and chasing decision parameters, and sets chasing message registration.

V. Broadcasting Approach

When mobile agent migrates to next server and it uses broadcasting to inform all mobile agents about the status information, all agents based on this information can find out its current location quickly. Its disadvantage is that when many mobile agents roam on agent system it will cause to remain more location information on network. Neither user wants to track their mobile agent current location, nor does user want to communicate with other agents. It will reduce the network bandwidth and take more memory to store every agent current location data [8].

VI. Centralized Location Register Station

In this method a centralized location register station on agent system is created which is responsible to manage all of systems current mobile agent location. This station controls all of the location information as well as processes the location information. Its disadvantage is that all of mobile agent location data handle by a station and will cause data process holdup issue [8].

3. VARIOUS LOCATION MONITORING MECHANISMS

The various mobile agents locating tracking mechanisms have been proposed by various authors a few of them are listed below:

a) Using Host Criticalities for Fault Tolerance in Mobile Agent Systems

In [16] authors present mobile agent based fault prevention and detection technique where the team of mobile agents monitor each host in mobile agent based system and considers a simple scenario of multi agent system where each service host performs some service as per roles assigned. It performs interactions with other hosts in response to query or to complete the assigned task. If some host which may have higher importance (such as initiator of task) fails, then it may disorganize the whole system. Such dynamically changing systems need constant monitoring. Author proposes distributed monitoring mechanism to improve efficiency and robustness of distributed system software by means of employing various mobile agents.

The proposed multi agent monitoring mechanism uses the *Monitor Agent (MOA$_i$), Manager Agent (MAM), Alert Agent (AlA)* mobile agents which improve the response of system to changes in behavior of the system.

This multi-agent monitoring mechanism supports mobile agent based distributed application and responds adaptively to environmental changes. These agents involved in monitoring play the following two roles:

- Supervision and control of service hosts.
- Generation of global information.

In multi agent system agents and hosts may need to communicate with other service hosts from time to time, either for required competencies or resources. These interactions are usually in form of message exchange.

The service host is in charge of storing this information in log tables. In order to calculate dependence of other service hosts on their host, mobile agents use several parameters such as executed tasks and type and number of messages exchanged.

This calculation is made by use of weights assigned to messages exchanged between such hosts. These weights may be useful in detecting which host becomes too heavy or if system is burdening few hosts. At occurrence of weight threshold or at passage of Δt time threshold, the MoA_i executes the weight updating algorithm. The service hosts continue to act according to their goals and requirements.

The advantage of this approach is that it focuses on building an automatic, adaptive and predictive determining policy where critical host agents are identified in advance by monitoring agents, to avoid their failures and the technique increases system's fault tolerance beside effective recognition of vulnerabilities in system. The monitoring mobile agents act together to detect undesirable behaviors and also provide support for restoring the system back to normalcy.

b) A Hierarchical Dynamic Monitoring Mechanism for Mobile Agent Location

In [9] the authors have proposed a hierarchical dynamic monitoring mechanism for mobile agent location, the method is proposed to use distributive mechanism for delivering MMA (Mobile Monitor Agent) to the network node (Agent Server) of mobile agent system.

MMA is responsible for collecting the mobile agent location in its mobile agent server.MMA is delivered by the centralized monitoring station. The centralized monitoring station along with the network environment can decide the delivery how many MMAs to come up to its network. It delivers on the agent servers to take a region the location monitoring station, reduces the operation and the data process bottleneck which is the centralized monitoring station. It is responsible to collect the mobile agent's information instantly in its monitoring area. Furthermore, the users can also inquire their own mobile agent's current location and working status.

The above mentioned mechanism is to distribute MMA dynamically to solve the hierarchical monitoring mechanism scalability issue, and still keeps the advantages of hierarchical monitoring mechanism to decrease information processing bottleneck issue which the centralized monitoring mechanism encounters.

The drawback of this approach is that the evaluation for dynamic monitoring mechanism is how every agent server becomes a region monitoring station, and how it still keeps the service, which originally provides also the agent servers provide the MMA working environment and how to confirm security between MMAs and servers.

c) Novel Monitoring Mechanism for Distributed system Software Using Mobile Agents.

In [17] Authors introduces the mobile agents based monitoring technique where monitor agents constantly collect and update the global information of the system using antecedence graphs. Updating weights of these graphs further help in evaluating host dependence and failure vulnerability of these hosts. These graphs help monitoring mobile agents to detect their unwanted behavior.

The novelty of proposed scheme is that the use of antecedence graphs for storing the component information by monitoring mobile agents. Antecedence graph is a directed acyclic graph which contains the information of the events occurred before a state interval. When global information needs to be collected, all relevant monitoring mobile agents are identified and informed concurrently which dramatically decreases the time latency of tracing dependencies.

The advantage of this approach is that the technique makes full use of the computation and storage ability of mobile agents and then reduces synchronization message transmission overhead. The technique uses antecedence graphs which exploits a new way to design cost-effective agent-driven software monitoring system.

d) A Fully Distributed Mechanism for Locating Mobile Agents

In [7] the authors have proposed fully distributed mechanism for locating mobile agents. The method is based on distributed hashing to locate mobile agents in a system. The mechanism is entirely transparent to the client. The idea is to treat mobile agents and hosts similarly to the way peers are treated in a peer-to-peer system, and to implement a lookup protocol that is based on that principle, while taking care of the mobile nature of the agents. Each agent is associated to a coordinator to whom it periodically reports its location. When a client needs to locate an agent, it contacts an arbitrary host with the key corresponding to the agent.

The client may not know any information about the coordinator or the agent's ID in advance. It just needs to send the request about what kind of services it needs, and it will be assigned a matching agent to accomplish the task. The system is dynamic, a coordinator peer can leave the system at any time and does not need to call back or destroy the agents it controlled, which are automatically assigned to another coordinator.

The advantage of this approach is that the notion of Agent Pool agents that permits the reuse of agents. In fact, idle agents can be reused until they stay idle for a long period of time and terminate themselves. There is no central control, the coordination of the mobile agent's activities is fully distributed as it occurs in P2P systems.

The drawback of this approach was that this approach is not suitable for multiple service hierarchy where services are organized hierarchically in a tree.

e) Design of Mobile Agent Communication Mechanism Based on SPF

In [18] the authors proposes approach which is in accordance with the design of EMFS (efficiently message forwarding scheme) mode. The Mobile Agent will report its location to its Home when arrive a node each time, also check whether needs to receive messages to decide to migrate or wait. The particular of this design is based on SPF algorithm and generating Assistant Agent which belongs to Mobile Agent to help receive message to ensure the message delivered reliably.

For example, Mobile Agent A sends a message M to Mobile Agent B. To avoid message chasing, this approach plans to use this program when the Mobile Agent A and B are at the same node , the Mobile Agent A and Mobile Agent B are not at the same node, but B had reached the node which A is at and when Mobile Agent A and Mobile Agent B is not at the
same node.

The SPF (shortest path first) algorithm basic thought is that in a graph, a node v0 is determined as a starting point,

calculate the shortest path from this node to others. First of all, add up each distance from v0 to other nodes, the distance is the weight on the edge which between v0 and other nodes. Fill in a weight table of all the weight values.

Then compare all weights from v0 to other nodes, choose the minimum corresponding node vi. Take vi and v0 into a set S, keep this minimum weight record, the rest nodes included into set T. For each node in S, has been calculated the shortest path to itself from v0. Secondly, when vi joins to set S, the each path from v0 to each node in set T may change.

Such as vk in set T, the original path is (v0,vk), the weight of the corresponding edge is c(0,k), but when vi joins S, the just mentioned path may change to (v0,vi,vk), the corresponding weight may change to c(0,i)+c(i, k). If c(0,i)+c(i, k)< c(0,k), then amend the shortest path from v0 to vk to be the new smaller value, else retain the old value. Thirdly, in set S, take the new added node as intermediate node which link v0 to rest nodes in set T, recalculate and compare the weight in the weight table. Then perform the same operation in the second step. In the end, get the shortest path and corresponding weight from v0 to other nodes.

The advantage of this approach is that it combines SPF Algorithm and uses the intelligence of Agent to generate the Assistant Agent to get message to give a improvement program. The purpose is to improve communication efficiency in short time for Mobile Agent, and ensure the message delivered reliably but the drawback is that it does not consider security problems.

f) A Scalable Hash-Based Mobile Agent Location Mechanism

In [6] Author proposed a novel mobile agent tracking mechanism based on hashing. To allow our system to adapt to variable workloads and dynamic rehashing is supported. The proposed mechanism scales well with both the number of agents and the number of moving and querying operations. Author proposes a two-tier mechanism. Special agents, called Information Agents (IAgents), are responsible for maintaining the current location of a set of mobile agents. Which set of mobile agents is associated with each IAgent is determined through a hash function. This mapping changes over time as new IAgents are created or existing IAgents are merged depending on the current system workload.

To locate a mobile agent A, in the first phase, we determine which special agent (IAgent) is responsible for maintaining the precise current location of A. This is done by applying the hash function on A’s id. In the second phase, the responsible IAgent is contacted. For the system to adapt to the changing number of mobile agents and the varying system workload, the number of IAgents changes over time. Specifically, when an IAgent becomes over-loaded, it splits its load by creating a new IAgent. Similarly, under-loaded IAgents are merged by assigning their load to other existing IAgents.

The advantages of this mechanism is that the approach scales well with both the number of agents and their mobility rate and provides almost constant search time for locating an agent independently of the system workload [6].

The system is currently extended in two ways. First, author studies a dual problem, the placement of the IAgents so that locality is exploited. Second, author investigates means for enhancing the fault tolerance of our mechanism. Currently, it supports the primary copy mechanism for the hash function, thus making the HAgent that keeps this copy a vulnerability point.

g) A Consensus Algorithm for Synchronous Distributed Systems using Mobile Agent

In [10] author presents a consensus algorithm for synchronous distributed systems using cooperating mobile agents. The algorithm is designed within a framework for Mobile Agent Enabled Distributed Server Groups (MADSG), where cooperating mobile agents are used to achieve coordination among the servers. Being autonomous and cooperative, cooperating mobile agents exchange information among them to achieve consensus.

In this algorithm, some of the n hosts will be designated as Coordinating Hosts (CHs). The proposed algorithm is executed by a mobile agent $Agent_{i,j,}$ on behalf of host h_i traveling to $CH_{i,j,}$. $V_{i,j}$ denotes the value set carried by $Agent_{i,j,}$ and V_h denotes the value set derived from all the value sets carried by the mobile agents at a host *h*. *Ts* denotes the maximum time required by an agent to migrate between any two hosts. $f_{i,j,}$ denotes the *Altered_Flag* of Agent $Agent_{i,j,}$ which indicates whether some new values are added into its value set during last trip. The algorithm terminates as soon as the agent is destroyed by its home host.

In each iteration of the loop, the mobile agent $Agent_{i,j,}$ waits for the signal for its migration . After received signal from its home host, it then copies the value set maintained by the master agent and migrates to CH_j . After arrival at CH_j the agent gives its value set to $CH_{i,j,}$. Then it waits for a signal from CH_j .When signal is received, every agent, which is dispatched to CH_j, must have arrived at $CH_{,j}$, and V_{CHj} will contain complete values of all arriving agents. $Agent_{i,j,}$ will contain complete values of all arriving agents. $Agent_{i,j,}$then compares V_{CHj} and the value set $V_{i,j}$ and sets the flag $f_{i,j}$ to indicate whether they are different. Finally, the agent copies the value set of CH_j and returns to its home host.

The advantages of this approach is that algorithm is designed within the MADSG framework where cooperating mobile agents are used as an aid to structuring distributed server groups. The correctness of the proposed algorithm is proved and the performance is evaluated by analysis and simulation.

The drawback of this work is that the proposed work in this paper represents only a specific example of the MADSG algorithm family. Improvement of this algorithm is needed for synchronous distributed systems and design mobile agent enabled consensus algorithms for asynchronous distributed systems.

h) Message Efficiently Forwarding Scheme (MEFS)

In [11] authors firstly discuss ways to solve communication failure, and then describe a two-dimensional solution space for designing inter-agent communication protocols. Authors 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 and condition in a fault-free network.

This mechanism combines two techniques, Central Server and Forwarding Pointer Schemes, to solve the problem of location tracking as well as two other techniques, Avoidance and Detection, for delivering the message.

MEFS offers location-transparency to tackle the problem of location tracking. In this system, each host has the information about agents created on it and agents residing on it at present. This information is always updated by forcing a migrating agent to unregister itself and put a forwarding pointer in the current host, informing the destination to its initial host, as well as register itself in the new host. Maintaining this procedure makes each agent can be easily tracked within a network. MEFS introduces Chasing Message Register, Over-speed Agent Blocking and Regular Synchronization to particularly resolve the problem of message chasing. A message chasing happens when a message is forwarded more than a certain number of times (Co). When it happens, then the message will be stored in recipient's home host. Over-speed Agent Blocking method forces an agent to synchronize itself with its home host whenever its migration velocity is over a certain specified level (Vo). It then waits for chasing messages for an amount of time (Ts), so that it can receive them.

The advantage of this approach is that this mechanism guarantees that each message will be received by its recipient. It completely solves the problem of location tracking and message delivery as well. However, forcing the agent to perform a regular synchronization with its home host will limit the mobility of the agent itself.

This mechanism has a drawback that the communication overhead caused by the synchronization occurs as well, especially in an environment with a high number of agents. Moreover, it is quite difficult to configure the suitable level of Co, Vo and Ts, at which the system can optimize the tradeoff between reliable delivery, limitation of agent mobility, time for forwarding the message and communication overhead caused by synchronization process.

i) Distributed Sendbox Scheme

In [12] authors have proposed a two-modes mobile agent communication mechanism termed as distributed sendbox scheme. It manages the message delivery in the mobile agent based on the standard of Session Initiation Protocol (SIP).

To deal with location-tracking, this scheme deploys a centralized server dedicated to maintain the path of all agents within a mobile agent system. To be able to migrate, each agent should register its destination host and un-register its current host to the server, so that its location can be updated. Inside the server, each agent is identified by its SIP-standard identity, formatted as agent_id <agent_id@hostname>.

In this scheme, each agent's state is defined as either active, waiting or moving. When an agent (S) resides in a host (PS) wants to communicate with another agent, it must send a request about recipient (R) agent's location to the home server (H), which will then check the recipient's location and state. If the recipient's state is "active" in its host (PR), then the server asks it to turn into "waiting" state and replies the sender's request with the recipient's location. Both agents are now synchronized and the communication between them can be performed. This is called as direct mode.

When sender does not directly send the message (MSR) to the migrating recipient, instead, it sends the message to the sendbox of the host on which it resides. The home server, accordingly, creates an entry in its table containing the information about the host saving the message and the recipient's identity. When the recipient finishes its migration and registers itself from its new host (PR'), the server asks it to move to the waiting state. It then notifies the hosts that save messages destined to this recipient to retrieve and forward them.

The advantage of this approach is that this scheme offers a reliable communication mechanism and it also gives an advantage on the low network load. But the drawback of this scheme is that this scheme is vulnerable to communication failures.

j) Virtual Agent-Based Communication Mechanism

In [13] authors have proposed a virtual agent-based mechanism for mobile agent communication. It introduces the use of virtual agent to provide a reliable message transmission. In addition, instead of deploying a server, this mechanism uses multicasting to track the location of each agent within the multi agent system.

In this mechanism each host in the system keeps the information about all agents in three tables. The HOME_T table consists of five columns: Locality for the name, Addr for its address, State for its status, either static or migrating, Onway_Count for total on the way messages destined to it and VA for the presence of its virtual agent. The next is AddrLocal_T which records the agent migrating to this host, and the last is AddrRemote_T that keep the information of other agent located in other host. Each agent is identified by P_ID:L_ID, of which P_ID is the host where it resides and L_ID is a logical name given by the owner. Once it is created in home host, it should be multicast to other hosts in order to inform this agent's presence.

This communication mechanism provides location-transparency. It also saves network bandwidth when it is implemented in a mobile agent system, since the migrating agent only needs to register itself to its new and home hosts, which then store all information using three tables.

The main drawback of this mechanism is that it is not efficient to be used in a large network with a lot of agents migrating frequently. One reason is because there will be high communication overhead caused by agents and their virtual counterparts to update their location and send messages.

k) *Active Distributed Monitoring for Dynamic Large-scale Networks*

In [14] author proposes a much simpler distributed algorithm and show that it performs as well as existing near optimal but expensive, centralized algorithms. As network conditions change, e.g. through faults or persisting congestion, agents can adapt and migrate to new locations. Author quantified the benefits of this approach against both the centralized and centrally-computed static distributed approaches.

In the proposed solution, the location of area managers is neither fixed nor pre-determined at design time. Area managers are realized with mobile agents, simple autonomous software entities that, having access to network routing information, can adapt and roam through the network.

The distributed monitoring system is deployed by progressively partitioning the network and by populating each partition with monitoring agents. Authors assume the existence of an agent system supporting agent cloning and assume that MA hosts i.e. locations in which mobile agents are able to run are evenly distributed within the network. This, in other terms, means that for each router there is always a MA host that is located relatively close to it and, for each LAN, the number of MA hosts is proportional to the number of MOs that need to be monitored in that LAN.

The advantage of this approach is that the application of code mobility to network monitoring is feasible, and generally results in increased scalability and steady-state performance.

But the drawback of this approach is that there are conditions in which agents cannot compete with the simplicity and efficiency of the conventional centralized approach like in very short or very simple monitoring operations. Therefore, the ideal solution is to integrate code mobility into existing systems.

4. CONCLUSIONS AND FUTURE WORK

This paper reviews the various location tracking mechanisms proposed by various authors. Location tracking mechanisms are needed to the client or the network manager to know the current location of the mobile agent in order to inquire its status, or to control its execution.

From this paper we conclude that every mechanism has its own pros and cones. The selection of mechanism should be carried out based on the characteristics of the network. Several factors should also be considered like network size, number of agents and their level of mobility, the use of hosts' resources, message chasing .

Communication Mechanism Based on SPF is a good mechanism as it also deals with the problem of message chasing by setting timeout and chasing decision parameters, and sets chasing message registration but suffers from security problem. On the other hand MEFS mechanism is also a good mechanism for location monitoring as it offers location-transparency and guarantees that each message will be received by its recipient and it completely solves the problem of location monitoring but it suffers from some communication overheads.

From future point of view we can remove the performance drawback of Communication Mechanism Based on SPF by dealing with security problems and also of MEFS approach by controlling the overheads due to communication. Also we can integrate Centralized approach and Footprints approach to develop a new approach to deal with location monitoring so that if due to any fault we lose our location data on a centralized server we can recover it from footprints of the agents.

5. REFERENCES

[1] H. lee, J. Alves and S. Harrison, ,2004. The Use of Encrypted Functions for Mobile Agent Security .In Proceedings of the 37th Hawaii International Conference on System Sciences.

[2] M. J. Wooldridge and N. R. Jennings,1994. Agent theories, Architectures and Languages: A survey. In ECAI Workshop on Agent Theories, Architectures and Languages, Springer.

[3] J.Baumann and Joachim, ,1999. A Comparison of Mechanisms for Locating Mobile Agents.

[4] Y.N. Lien and C.W.R. Leng. On the Search of Mobile Agents. Department of Computer Science, Taipei,Taiwan,R.O.C.

[5] A.Y. Yousaf and A.Y. Hammo, 2012,Developing a New Mechanism for Locating Mobile Agents. In Journal of Engineering Science and Technology Vol. 7.

[6] G. Kastidou, E.Pitoura and G.Samaras, 2003. A Scalable Hash based Mobile Agent Locating Mechanism. In Proceedings of the 23rd International Conference on Distibuted Computing systems Workshops

[7] P. Flocchini and M.Xie ,2007.A Fully Distributed Mechanism for Locating Mobile Agents. Proceedings of the International Multi Conference on Computing in the Global Information Technology.

[8] Ye Din-Yuan and Wang T.I ,2000.The Mechanism of Tracking Mobile Agent. Taiwan Cheng Kung University Department of Engineering Science Master paper.

[9] Y. H. Wang, H. C. Keh, T. C. Hu and C. H. Liao, 2005. A Hierarchical Dynamic Monitoring Mechanism for Mobile Agent Location .In Proceedings of the 19th International Conference on Advanced Information Networking and Applications.

[10] J.Cao, X. Wang, S. Lo and S. K. Das ,2002. A Consensus Algorithm for Synchronous Distributed Systems using Mobile Agent. In of Proceedings Pacific Rim International Symposium on Dependable Computing.

[11] Zhou J, Ja Z and Chen D, 2003. Designing Reliable Communication Protocols for Mobile Agents. In Proceedings of 23rd International Conference on distributed computing system.

[12] Fang-Yie L, Hsiao-Hong T and Wen-Kui C, 2005. Distributed Sendbox Scheme for Mobile Agent Communication, In Proceedings of International Conference on Mobile Business ICMB.

[13] Z. Jian-ming , X. zhong and H-Ying , 2006. A New Communication Mechanism Based on Virtual Agent for Mobile Agent. In Proceedings of 10th International Conference on Computer Supported Cooperative Work in Design.

[14] A. Liotti, G. Pavlou and G. Knight,2001.Active Distributed Monitoring for Dynamic Large-scale Networks .In Proceedings of IEEE International Conference on Communications.

[15] Z. Guessoum , N. Faci and Jean-Pierre Briot, 2005. Adaptive Replication of Large-Scale Multi-Agent Systems - Towards a Fault-Tolerant Multi-Agent Platform. In the ACM Electronic Proceedings of the SELMAS'OS at ICSE'05, St. Louis, Missouri, USA, ACM Software Engineering Notes, vol No 4, pp.I-6.

[16] R. Singh and M. Dave, , 2012.Using Host Criticalities for Fault Tolerance in Mobile Agent Systems", In Proceedings of 2nd IEEE International Conference on Parallel, Distributed and Grid Computing.

[17] R. Singh and M.Dave , 2013. Novel Monitoring Mechanism for Distributed system Software Using Mobile Agents",Intelligent Informatics Advances in Intelligent Systems and Computing Volume 182.

[18] J. Zhang, Z. Yang and R. Wei, ,2012.Design of Mobile Agent Communication Mechanism Based on SPF. International Conference on Computer Science and Service System.

[19] C.Virmani , 2012.A Comparison of Communication Protocols for Mobile Agents",International Journal of Advancements in Technology,Vol. 3 No.2.

[20] M.Dave , R.Singh, 2013. Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems",IEEE Transactions on Computers, vol. 62, no. 2.