Volume 2, Issue 5 May 2016 ISSN NO: 2454 - 7514

North Asian International Research Journal Consortium

North Asian International Research Journal Of Science, Engineering and Information Technology

NAIRJC JOURNAL PUBLICATION

North Asian International Research Journal Consortium

Welcome to NAIRJC

ISSN NO: 2454 -7514

North Asian International Research Journal of Science, Engineering & Information Technology is a research journal, published monthly in English, Hindi. All research papers submitted to the journal will be double-blind peer reviewed referred by members of the editorial board. Readers will include investigator in Universities, Research Institutes Government and Industry with research interest in the general subjects

Editorial Board

Address: - Dr. Ashak Hussain Malik House No. 221 Gangoo, Pulwama, Jammu and Kashmir, India - 192301, Cell: 09086405302, 09906662570, Ph. No: 01933-212815,
Email: nairjc5@gmail.com, nairjc@nairjc.com, info@nairjc.com Website: www.nairjc.com

A PROPOSED SERVICE MANAGER MODULE FOR MANET USING MOBILE AGENT

RAJESH KUMAR [1] & PROF. S. NIRANJAN [2]

[1] Research Scholar, Mewar University, Gangrar, Chittorgarh, Rajasthan-312 901, India
[2] Professor, Mewar University, Gangrar, Chittorgarh, Rajasthan-312 901, India

ABSTRACT:

Use of mobile agent and mobile phone technologies are rapidly increased in the current scenario. The mobile agent has a revolution in the industry to meet the requirements of users because every aspect of life have direct or indirect link with Mobile agents. Unfortunately, this technology faces some security problems from mobile agents and hosts sides. Two of the fundamental issues that rise are in designing protocols for message passing between mobile agents (MAs) are tracking the migration of the target agent and forwarding messages to it. Even with an ideal fault-free network-transport mechanism, messages can be dropped during MA migration. Therefore, in order to provide reliable message delivery, protocols need to overcome message loss caused by asynchronous operations of agent migration and message forwarding. In this paper we propose message forwarding approaches in this concern, namely push and pull, are explored to design adaptive and reliable message delivery protocols that are based on the RMST (Reliable Multi Segment Protocol. Also location aware services scheme by using an agency, which helps in discovering the required services for the mobile users. The agency consists of five agents: Manager Agent (MA), Route Monitoring Agent (RMA), Local Service Management Agent (LSMA), Global Service Management Agent (GSMA) and Area Monitoring Agent (AMA).

***Keywords:* Host Protection, Distributed Systems, Mobile Computation, Mobile agents (MAs), pull, push, reliable message.**

I. INTRODUCTION

Mobile Agent is thought to be the main distributed computing model based on the next generation of Internet. It is widely accepted that the mobile agent has the following key features: autonomy, collaboration, activity and especially mobility [1]. To exhibit these features, mobile agent must interact and cooperate with each others. Although the most important purpose of a mobile agent paradigm is to establish a communication link and exploiting local access to resource on a remote server. A mobile agent system is a new technology allows computers to communicate and facilitates the mobile computation. This technology is sub area of distributed system. It comes from the approach of the Remote Procedure Call (RPC). A mobile agent is an entity that has ability to travel from host to another under itself decision. By using its itinerary table, the mobile agent visits hosts to request services [2]. This technology faces a big challenge in the security area and many security aspects should be considered when a mobile agent system is designed. The system should protect hosts against malicious hosts and protect the mobile agents against malicious hosts. But the agent mobility presents distinct challenges to communication frameworks. If the agents keep stationary, we can ensure that messages will be received within a limited time in a fault-free network. However, due to the autonomy and mobility of the agents, the communication object may move from one host to another at any time in the mobile agent framework. The physical location change of agents will result in a problem of communication failure, that is, before a message gets to one host, the target agent has left away, making it unable to receive this message. Even if we use connection-oriented protocols such as TCP, we still cannot guarantee the message delivery to mobile agents all the same. Because message cannot be instantly delivered, the problem widely exists in mobile agent systems.

In some extreme situations, like the agent moves frequently, whenever a message reaches a host where the target agent used to reside, the agent has just left away so it cannot get the message all the time and the message keeps chasing the recipient around the system but never gets delivered, resulting in a race condition which is called Message Chasing. If the agent cannot receive messages sent to it in time, the collaboration will fail and the system may even crash. In this paper, we analyze that how different agent send message to each other and then describe the different communication protocol for the agents, and after this we describe a general-purpose framework [3]. A reliable communication mechanism for mobile agents named MPFP (Message proficiently Forwarding plan) is also describe. Our MPFP implements the delivery process in an effective and efficient manner and gets high adaptability by configuring protocol parameters dynamically at runtime. For the convenience, we always assume that the network consists of fault-free channels. That is to say messages will be transferred from one side of the channel to another orderly without any transmission fault. In this we also describe Agent as a Service Manager for the Location Based Services (LBS) which provide personalized services to the subscriber based on their current position. Location information is becoming increasingly important in many persistent computing applications ranging from human-oriented information appliances to distributed sensor networks and robotic colonies. Location based services employ accurate, real-time positioning to connect users to nearby points of interest. Location aware applications are becoming increasingly attractive due to the widespread distribution of wireless networks and the emergence of small and cheap locating technologies. Some of the service categories for LBS include emergency, safety, entertainment, information navigation, tracking and monitoring, m-commerce, mobile yellow pages, etc.

The satellite based, Global Positioning System (GPS) is the most widely used RF system, providing global outdoor coverage. Emerging higher-rate radios such as 802.11a hold great promise in providing accurate position data due to their inherent timing accuracy. Managing the mobility or roaming of the wireless devices becomes a major issue.

The location-based services architecture must support self organization in several ways:

1) Locations should be determined with minimal user inputs,
2) Multiple location information sources should be combined to increase accuracy.
3) Seamless switching between indoor and outdoor operation.
4) Cooperation between neighboring clients to determine position or increase accuracy [3].

II. PROPOSED MODEL

The proposed model is the assumption in the existence of an agent platform in the mobile agent nodes. An agent platform at a mobile node offers following services: security, communication, persistence, agent mobility, agent tracking, agent creation, and agent execution. However, in case of unavailability of an agent platform support for mobility, agents employ traditional message passing method for communication. The proposed model uses a set of static and mobile agents. Mobile agents offer asynchronous and disconnected task execution, which is very much required in a wireless mobile network [4].

2.1 RELIABLE MESSAGE DELIVERY FOR MOBILE AGENTS:

PUSH OR PULL

A. Push and Pull Approaches

Push and pull are two possible modes in the relay communication model to forward messages from the relay station to the MA. In general, during the execution of an MA, there can be one or several relay stations serving the agent. To simplify the discussion, however, we assume that each MA is associated with only one relay station. This can be easily extended to situations where more than one relay station is used for forwarding messages to an agent.

Push: In the push mode, the relay station maintains the location information of the MA. Incoming messages destined to an agent are pushed to the current location of the target agent. Upon migration, after the MA reaches to the destination site, it registers the new location with its relay station [5]. The relay station will update the agent's location information maintained in its database. Subsequent incoming messages destined to this agent are pushed to the agent's new address. The simple push mode, although achieving location transparency, cannot guarantee reliable message delivery. It is possible for a message to be sent from the relay station toward the MA, and for the MA to move away before the message is delivered. That is, when a message is forwarded to the address as kept in the relay station, the target agent may have left for another host. Although it can be further forwarded, the message may keep chasing the target agent.

To avoid message loss and the chasing problem caused by agent mobility, we propose a *synchronized push* mode. Synchronization between message forwarding from the relay station and agent migration is implemented in the following way. Before migration, the agent deregisters its current location with the relay station and waits for the ACK message. After it receives the ACK message, the agent migrates to the new location and registers its new location with the relay station upon arrival. As shown in Fig. 1, messages can be forwarded to the MA when it is in "stationary" and "waiting" states and must be blocked when it is in the "moving" state. Since the agent will not move until it receives the ACK messages from the relay station, messages forwarded before the ACK message will have reached the target agent before its migration. No message will be forwarded during the migration of the target agent, i.e., during the interval between the ACK message and the next register message. Therefore, message loss and the chasing problem cannot occur [5].

Pull: In the pull mode, the relay station simply buffers the messages to the MA and does not need to keep its location information. The MA queries the relay station periodically for messages. Upon receiving a query message, the relay station forwards the buffered message to the agent. If there is no message in the buffer, a "null" message is sent to the agent as a reply. The MA can use either a synchronous or an asynchronous query operation. Synchronous query means the agent suspends its execution after issuing a query until it receives the reply from the relay station.

In this way, the agent can ensure that no message will be forwarded to it during its migration. If asynchronous query is used, the agent can continue its execution after a query. However, to avoid message loss, the agent cannot migrate to other hosts until all the replies arrive. The agent always knows the location of its relay station and initiates the request for messages, so location registration is unnecessary in the pull mode. Since the agent will not leave for the next host without receiving the response to its current query, there is no message loss and also the chasing problems cannot occur [6].

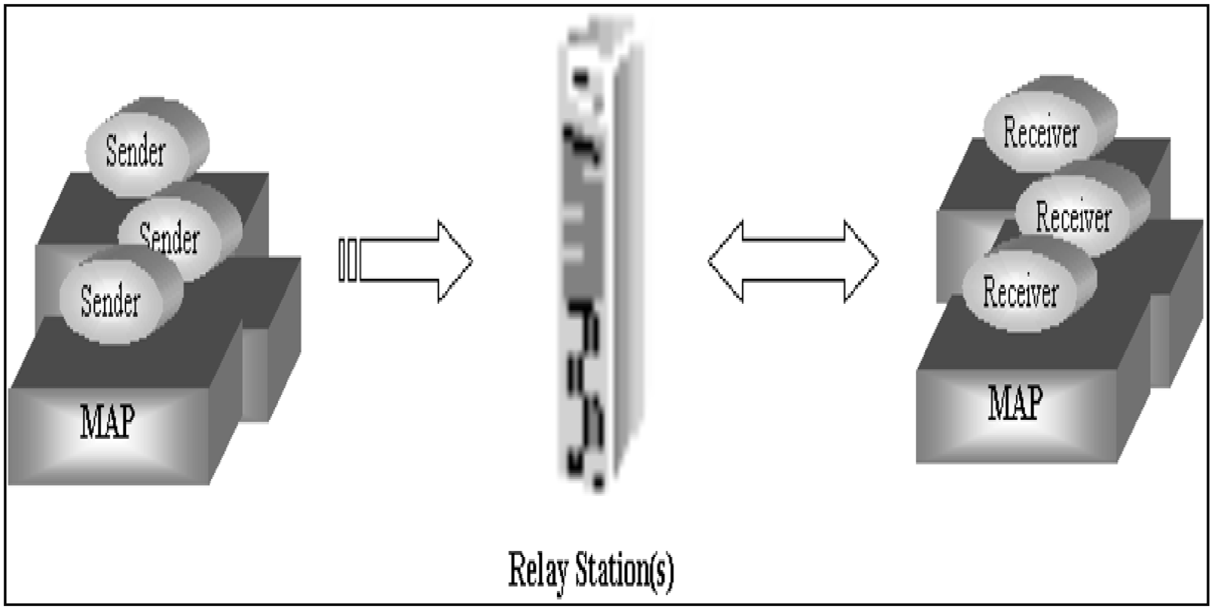

Figure.1 Mechanism for relay stations

B. Properties of Push and Pull

The properties of push and pull modes are analyzed in terms of reliability, resiliency to failures, constraint on agent mobility, support for real-time processing, communication overhead, and flexibility.

1) Reliability:

By reliability, we mean that the messages can be routed to its target agent within a bounded number of hops. As discussed above, message loss and the chasing problem may occur under the simple push mode without synchronization. The synchronized push mode can avoid these problems and thus guarantee reliable message delivery. In the pull mode, since the receiver agent takes the initiative to request messages from its relay station, the agent can ensure that no message will be forwarded to it during its migration. Therefore, the requirement of reliable message delivery can easily be satisfied [7].

2) Resiliency to Failures:

In the push-based approach, the location and status (e.g., stationary, moving, and waiting as shown in Fig. 2) of the agent must be maintained at the relay station during the agent's life cycle. The state of the agent is lost if the relay station fails. After recovery, the relay station may have lost the trace of the agent. Moreover, the agent cannot detect the failure of the relay station and reregister with it until its next migration. In contrast, the pull-based relay station is resilient to failures due to its stateless nature. By periodically querying the relay station, the agent can easily detect failure of the relay station.

3) Constraint on the Agent Mobility:

In the synchronized push mode, the agent has to deregister with its relay station and wait for the ACK message before its migration, therefore the agent mobility is constrained and the migration time is increased. In the pull mode, if synchronized query operation is used, the agent can leave for next host as soon as it finishes its execution at this host, but the execution time is increased. For asynchronous query, the agent also has to wait for the arrival of all the response to its query before migration [7]. However, by deciding the time and number of queries, the agent can flexibly reduce the constraint on its migration.

4) Support of Real Time Message Processing:

In the push mode, unless the agent is in "moving" state, messages are forwarded to their target agents immediately after they arrive at the relay station. The sender has greater certainty that the message will reach its target within an appropriate timeframe. However, in the pull mode, the transmission time of a message depends not only on the network delay, but also on the frequency at which the receiver queries its relay station. Therefore, the delayed time for the message getting processed by the receiver is longer in the pull mode [8].

5) Communication Overheads:

In the pull mode, two messages are needed for each query, namely, the query message and the response from the relay station. Moreover, to decrease the delay of message processing, the receiver may query at a higher frequency than the frequency of the message arrival at the relay station. Therefore, the pull-based approach is liable to impose a larger load on the network. On the other hand, three extra messages, namely, deregister, ACK and register, are needed for each agent migration in the push mode. In the cases where the agent migrates frequently but seldom communicates, the communication overhead of the push mode is significant.

6) Flexibility:

Since the agent has the autonomy to decide on the time and frequency of the queries for messages, more flexibility is introduced in the pull mode. For example, the agent can adjust its query frequency dynamically. If it is in urgent need of information from its coordinator, it may query at a higher frequency. Otherwise, a lower frequency is adopted. Distance can be another factor of concern [9]. If the current location of the agent is very far from its relay station, it can query the relay station at a much lower frequency or does not query at all. When it migrates to a host nearer to its relay station, it can query more frequently and process more messages buffered in the relay station.

III. COMMUNICATION PROTOCOLS FOR MOBILE AGENTS

There are mainly two issues must be addressed in any communication protocol for mobile agents:

1) Tracking the location of target mobile agent,
2) Delivering message to the agent.

If we look at the agent tracking in the process of sending message as a reading operation and the change of location after the recipient migrates as a writing one, the reading-writing concurrent access collision on the shared variable “current agent location” leads to the “dirty reading” problem, i.e. agent changes its location when the message is en-route so message is sent to a host where agent does not reside any more. It is the presence of mobility, not the possibility of fault in the network that causes Communication Failure. We know that there are four main tracking methods for mobile objects all of which have been applied in the mobile agent systems aforementioned:

- **Broadcast/Multicast scheme (BM)**

The sender sends query messages to all hosts for location of the receiver or broadcasts agent message directly in the system. Another way is that the agent broadcasts its new location after each migration. The BM works efficiently in local network domain, especially in bus-based multiprocessor systems. But it is impractical in large-scale network because of large communication overhead [10].

- **Hierarchical scheme (HS)**

In this a tree-like hierarchy of servers will be used. For each agent there is a unique path of pointers that starts from the root and ends at the leaf that knows the actual address of the agent. The HS supports the locality of mobile object migration and communication well. However the hierarchy is not always easy to construct, especially in the Internet environment.

- **Central Server scheme (CS)**

A location server is used to keep the track of the physical location of a mobile agent. Although CS is easy to implement, the location server in it is really a potential bottleneck of performance [11].

- **Forwarding Pointer scheme (FP)**

Each host on the migration path of an agent has a forwarding pointer pointing to the next host on the path so that messages can be forwarded to the recipient along the path. The FP has less reliance on a location server and incurs no location registration overhead [11]. But it may be difficult to guarantee message delivery and shorten the forwarding path if a communication protocol adopts FP. We further classify three ways to solve Communication Failure into three categories:

- **Ostrich**

Discarding messages in Mole and ICM when a host does not know where the recipient locates is an instance of Ostrich. The Ostrich ignores the problem and does nothing for Communication Failure so it cannot contribute to a reliable protocol in mobile agent systems [12].

- **Avoidance**

Avoidance, which is widely adopted, establishes some mechanism to prevent delivering messages to a host on which the recipient does not reside so that the communication failure will never happen [13]. Synchronous methods are mostly used here, but agents are disallowed migrating until having collected all ACK messages needed, which makes Avoidance economically ineffective and technically inefficient, especially, in the Internet.

- **Detection**

In Detection, the system must be able to detect communication failure and take some measures to deal with lost messages. For the "nature of agents", Detection always uses asynchronous ways to implement communication within mobile agents. However, it is difficult to design a reliable Detection with its side effects, like Message Chasing, well settled. Some protocols presume different conditions such as having known the topology of the network or Destination of migration in advance, while these assumptions are not always reasonable and we will not consider here.

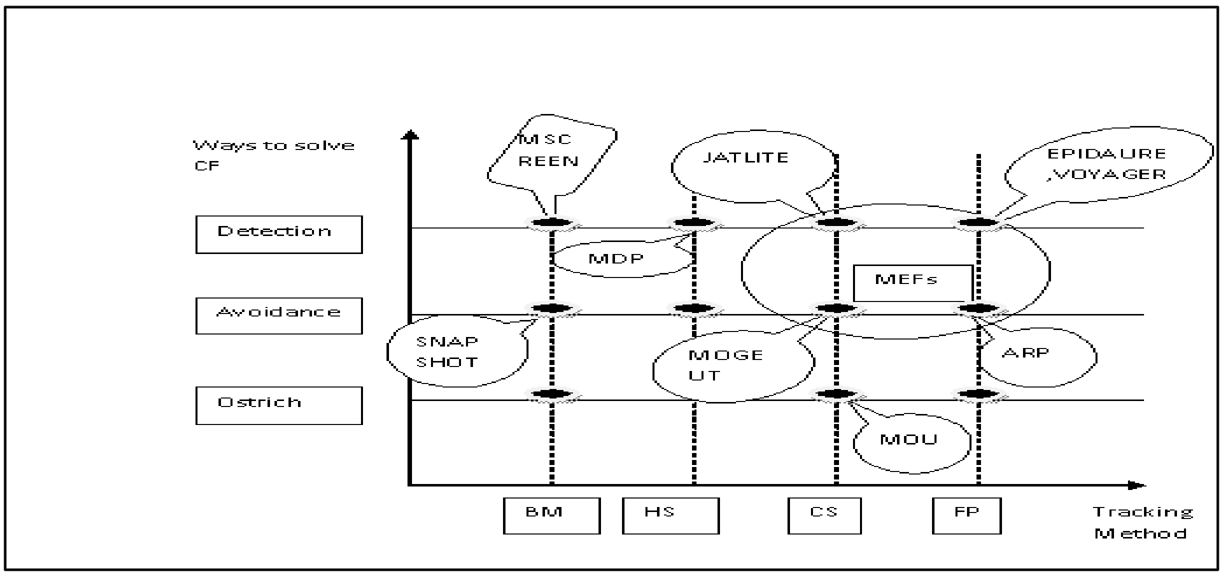

Figure 2 Communication Protocols for MA

We now describe the design space for inter-agent communication protocol [13]. As we have seen in Figure 3, every intersection in this figure provides one choice to design a type of protocol, and is actually a combination of an agent tracking method and a way to solve Communication Failure. Currently existing solutions are also labeled in this two-dimensional model. To establish a reliable and efficient communication protocol, we propose our MEFS (Message Efficiently Forwarding Schema), which needs no pre knowledge of the agent behavior or system configurations. The MEFS combines the Detection and Avoidance techniques and uses the tracking methods of Forward Pointer and Center Server scheme.

- **MPFP: A RELIABLE AND EFFICIENT COMMUNICATION MECHANISM**

To allow mobile agents communicate seamlessly regardless of their location, each agent is assigned a globally unique name when it initializes from which its home place's address can be easily resolved. Meantime, each node in the network keeps table recording information about the agents currently residing in it and the agents initialized on it [14].

When migrating, agent must Unregister, leaving a forwarding pointer, when it leaves a node and register when arrives at a new one. Taking agent A migrates from some node 1A to 2A as an example: When A prepares to migrate, it sends a control message to its home place HA, telling its target 2A. After informing 1A to update its status table, A moves to node 2A and registers to 2A by adding or updating an entry in the status table stored in 2A. Actually, control messages are generated after agent having made the migration decision, so here we have no pre knowledge of agents' action and also make no limitation to migrations. Forwarding messages asynchronously is the core of MPFP.

To deliver message efficiently with less limitation to agent migration, a module "Communicator" in every node processes each incoming messages with responsibility for. If agent B resides on the same node as A, messages can be sent to A just within the node; otherwise Communicator will look for the status table maintained in the same node or query A's home place for A's location and then forward the messages to the desired next node. Even though agent A is roaming in the Space freely, we can easily forward messages because A has left a pointer linking to the next host every time it migrates [14].

Messages are forwarded along a chain logically constructed by the recipient and we need not query A's home place every time leading home place to be a potential bottleneck of the system. Since it is believed that asynchronous methods cannot fully solve the message-chasing problem, we use synchronous communication when racing occurs but constrain agents' autonomous migration as little as possible. MPFP achieves synchronous by using Chasing Message Register and Over-speed Agent Blocking. Chasing Message Register means that if a message has been forwarded with a given times t0, the message is then thought as a "chasing message" and its serial number will be sent to the recipient's homeplace to store. On the other hand, each agent must calculate the velocity 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 V0, the agent should establish a connection with its homeplace, 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. If agent's velocity is low, the message will catch their recipient affirmatively. While the agent migrates frequently, the strategy of over-speed agent blocking can surely avoid message chasing [14]. Besides being blocked when over-speed, each agent in MPFP must contact its homeplace and wait for chasing messages at intervals. We name the interval TS. By synchronizing regularly, we can guarantee that the recipient, no matter how the network condition changes, will receive the message eventually. With Chasing Message Register, Over-speed Agent Blocking and Regular Synchronization, Message Chasing problem is eradicated and a reliable communication is finally achieved in MPFP.

IV. SYSTEM MODEL FOR LOCATION BASED SERVICE

The system environment considered for designing the proposed scheme comprises of the following.

1. A wireless network with several clusters (all clusters are interconnected by wired network, i.e., each cluster is a local network).
2. Each cluster has an active fixed node, which comprises of proposed agency.
3. Nodes residing in any of the cluster will advertise their services to neighboring clusters and its active node.
4. Nodes requiring services will ask the active node with service parameter requirements. Parameters may be bandwidth, delays, service location, etc.
5. Mobile nodes move from one cluster to another.
6. GPS services will be available in all the mobile nodes [15].

I. AGENTS AS A SERVICE MANAGER FOR LOCATION BASED SERVICES

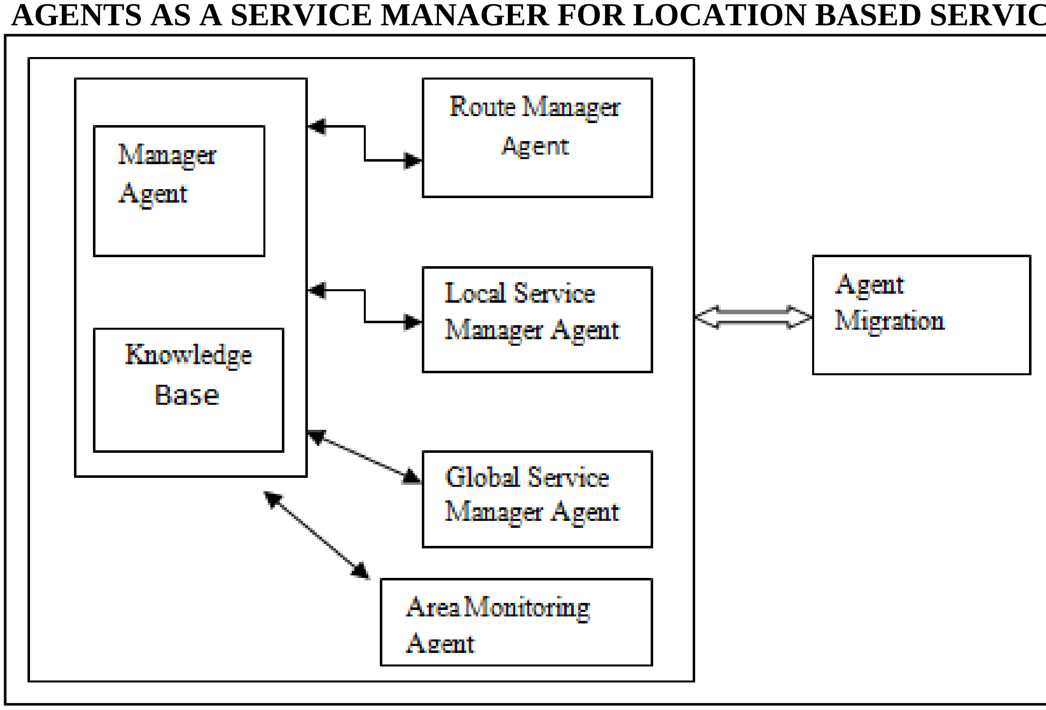

Figure 3 Location aware service agencies

The proposed model consists of an active node in every network. The active node is a special kind of a node consisting of a set of static and mobile agents. The location identification of all the nodes is found by using GPS system at the active node. Figure 1 depicts the proposed agency for providing location-based services. It consists of five different agents and a knowledge base as described below.

• ***Knowledge Base (KB)*****:** It consists of the different services available and their related information in a local network that is updated by manager agent. The service related information stored is as follows [16]:

- service providing node, type of services, service location, availability,
- status of breakdown, status of link and the service provider,
- clients accessing the service, number of times service disconnected for a particular client,
- service reliability, service refused clients,
- Bandwidth available in local network, delays in local network, traffic density, etc.

It also comprises of the services available in other clusters and their related information along with route to reach the services. The route-monitoring agent updates route information.

• ***Manager Agent (MA)*****:** This is a static agent. The MA creates knowledge base and all other agents of the agency and acts as a coordinator of all the agents in the active node. Functions of the agent are as follows.

- It creates the agents as and when required.
- It is also responsible for updating the knowledge base by looking at the advertised services from local network [17].
- Any updating to knowledge base by other agents is done through this agent.

• ***Route Monitoring Agent (RMA)*****:** This is a static agent, which communicates with the arrived global service management mobile agents to find the services available and its location in the host node as well as visited nodes of mobile agent. Functions of the agent are as follows:

- The travel information from the arrived mobile agent will be picked up by the route monitoring agent to decide several nearest services of same kind and updates knowledge base with service related information and its route. This information can be used to access the services of other networks for the nodes in a cluster.

- In the event of breakdown of a service node or link failure, RMA tries to provide another service node and its reachable path to continue with the service to service requested node [18].

• ***Local Service Management Agent (LSMA)*:** This is static agent, which is created by MA to identify the services in its cluster when a mobile user in the cluster requests for location based service with certain quality of service (QoS) requirements such as news, video games, movies, songs, etc. QoS is specified in terms of bandwidth and delays. It scans the knowledge base to check the availability of services with given QoS requirements. If available, service node, service id and other related information is given to requested user to connect to service providing node (which may be residing in its cluster or another cluster). If not available, it requests MA to generate a global service management agent to locate the required services by the requested user [19].

• ***Global Service Management Agent (GSMA)*:** This is a mobile agent. When the MA finds that the requested service is not available within its local network range, it creates the GSMA to roam around the networks and locate the nearest service with given QoS requirements. GSMA clones (cloning is a process of creating a copy of GSMA and sending it neighbors) to its nearest active nodes and so on until it locates the services in the networks. Once GSMA finds the services of same kind and the route to it, GSMA informs MA. Later MA can inform the requesting node to access the services. While roaming, GSMA can interact with the visited nodes to exchange the information.

• ***Area Monitoring Agent (AMA)*:** This is a static agent. It monitors the parameters such as delays and bandwidth available in the local network [20]. It also tracks the service breakdown, node/link failures for service offering. The monitored information is updated in knowledge base. The agents in the agency interact with one another in the following sequence in order to provide services to the service requesting nodes as follows.

1. The mobile client requests for the service to MA with QoS requirements.
2. MA informs LSMA to find out the nearest location of service providing node that satisfies the QoS requirements. (Assume that service is not available in local network knowledge base.)
3. LSMA informs MA to generate GSMA.
4. MA creates GSMA with service types and QoS requirements.
5. GSMA roams in the network generating clones and updating and exchanging information with visited nodes.
6. GSMA locates a required service in some network and informs the MA, who has created GSMA.
7. MA informs service requesting node about the service available and its location in the networks and route to it.

8. Requesting node connects to service providing node and the session continues until the session is completed or service breaks down due to node/link failures or service failure. If service breakdown occurs, RMA attempts to find the similar kind of service by itself or by interacting with neighbor RMAs [21]. If it is successful in finding the service, service may be resumed for the requesting user by connecting to different service provider.

V. CONCLUSION

The proposed work service manager module for Manet using mobile agent gives us a reliable utility for mobile agent that works under different environment. It comes from the approach of the Remote Procedure Call (RPC). In this module we are using the Push and Pull Approaches that is used for migration of agents as well as concerned with the all kind of Mobile agent services as a service manager in an effective manner as well as it also concerns all the security points that is must for a mobile agent platform. This module also integrate with the other modules to work combinable for reaching its goal state as well as according to the nature of problem in which works going on.

REFERENCES:

[1] M. Straßer, J. Baumann, and F. Hohl, “Mole—Ajava based mobile agent system,” in *Proc. 2nd ECOOP Workshop Mobile Object Syst.*, 2009, pp. 327–334.

[2] C. E. Perkins, “IP mobility support,” presented at the RFC 2002, Oct. 2009.

[3] G. Kunito, Y. Okumura, K. Aizawa, and M. Hatori, “Tracking agent: A new way of communication in a multi-agent environment,” in *Proc. IEEE 6th Int. Conf. Universal Personal Commune.*, vol. 2, 2009, pp. 903–907.

[4] Yannis Labrou and Tim finin, “A Proposal for a new KQML Specification”, *Technical Report CS97-03*, UMBC, 2009.

[5]General Magic Inc. “Introduction to the Odyssey”. *http://www.genmagic.com/agents*, 2010.

[6] A. Pham and A. Karmouch, “Mobile software agents: An overview,” *IEEE Commune. Mag.*, vol. 36, pp. 26–37, July 2010.

[7] D. B. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets*. Reading, MA: Addison-Wesley, 2010.

[8] Foundation for Intelligent Physical Agents, “FIPA 97 Specification Part 2: Agent Communication Language”, *http://www.fipa.com*, Oct. 2010.

[9] Neeran Karnik and Anand Tripathi, "Agent Server Architecture for the Ajanta Mobile-Agent System", In *Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2010),* July 2010, pp 66~73.

[10] F.G...Mcabe "Inter-Agent Communication Reference Model", *http://www.nar.fujitsulabs.com/documents*, 2011.

[11] Ludden B."Location Technology", http://www.locationforum.org/about-LIF/documents (2011)

[12] T. Kindberg,"People, Places, Things: Web Presence for the Real World", Technical Report HPL-2000-16, Internet and Mobile Systems Laboratory, HP Laboratories, 2011.

[13] FENG Xinyu, CAO Jiannong, et al. "An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems". *G. P. Picco (Ed.), Mobile Agents, LNCS 2240, Spinger-Verlag*, 2011, pp. 135-151.

[14] S. M. Das, H. Pucha, and Y. C. Hu, "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing", *Proc. of IEEE INFOCOM*, Miami, FL, USA, 2012.

[15] X. Feng, J. Cao, J. Lu, and H. Chan, "An efficient mailbox-based algorithm for message delivery in mobile agent systems," in *Proc. 5th IEEE Int. Conf. Mobile Agents,* Atlanta, GA, Dec. 2012, pp. 135–151.

[16] Sharad Chandra Agarwal, "Location Based Services", http://www.tcs.com/0 whitepapers/htdocs/atc/location based services sep03.pdf, 2013

[17] Object space Voyager Core Technology [Online]. Available: http://www.objectspace.com

[18] Shijun Yu1, Marie-Aude Aufaure, Nadine Cullot3, Stefano Spaccapietra1, "A Collaborative Framework for Location-Based Services", http://lbdwww.epfl.ch/ yu/files/caise03.pdf.

[19] "Location Based Services: Considerations and Challenges", White paper, http://www.palowireless.com/lbs/docs/LocationBased Services.pdf

[20] Roy Friedman and Gabriel Kliot, "Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey", *Technical report*, www.cs.technion.ac.il/users/wwwb/cgi-bin/trget. cgi/2006/CS/CS-2013-10.pdf

[21] Allan Brimicombe and Yang Li, "Mobile Space- Time Envelopes for Location-Based Services, *Transactions in GIS*, Vol. 10, No. 1 pp. 523, 2014 www.blackwell-synergy.com/doi/pdf/10.1111/j.1467- 9671.

Publish Research Article

Dear Sir/Mam,

We invite unpublished Research Paper,Summary of Research Project,Theses,Books and Book Review for publication.

Address:- Dr. Ashak Hussain Malik House No-221, Gangoo Pulwama - 192301
Jammu & Kashmir, India
Cell: 09086405302, 09906662570,
Ph No: 01933212815
Email:- nairjc5@gmail.com, nairjc@nairjc.com , info@nairjc.com
Website: www.nairjc.com