Partial Connection Architecture for Mobile Computing

Phyoung Jung Kim, and Seogyun Kim

Abstract**—**In mobile computing environments, there are many new non existing problems in the distributed system, which is consisted of stationary hosts because of host mobility, sudden disconnection by handoff in wireless networks, voluntary disconnection for efficient power consumption of a mobile host, etc. To solve the problems, we proposed the architecture of Partial Connection Manager (PCM) in this paper. PCM creates the limited number of mobile agents according to priority, sends them in parallel to servers, and combines the results to process the user request rapidly. In applying the proposed PCM to the mobile market agent service, we understand that the mobile agent technique could be suited for the mobile computing environment and the partial connection problem management.

Keywords**—**Mobile agent, mobile computing, partial connection.

I. INTRODUCTION

A mobile computing is appeared as a new computing model by rapid development of wireless LAN, cellular and satellite communication technology [1]. A mobile host can be connected to networks from different places at different times. The characteristics of host mobility in the mobile computing need a different model from a distributed system model.

Mobile computing service must solve the problems caused by inherent characteristics of mobile computing [1, 2, 3]. First, mobile computing service has the location setting problem. Current location of a mobile host should be setup to communicate the host. Since the network connection point is continuously changing caused by the mobility characteristic of a mobile host, the IP address which serves as the connection point in internet also is continually changing. Second, mobile computing service has partial connection problem. A mobile server broadcasts messages to mobile hosts in the cell area of the mobile server through wireless media. Since wireless media are used as the communication media in mobile computing, they introduce high error rate and low bandwidth. Meanwhile, since a mobile host uses an independent power, the connection to a mobile server is often intentionally disconnected to reduce the power consumption of the mobile host. Also, an unexpectedly disconnection can be occurred during the handoff in mobile computing. Accordingly, there are some cooperative works among partial connected computers in mobile computing environment. Third, since conventional distributed algorithms utilizing the logical structure of network topology such as tree and star reconfigure the structure dynamically as needed since mobile hosts continually moving in mobile computing.

Partial connection problem of the above problems in mobile computing can be solved by a mobile agent technology. Mobile agents work by transporting themselves from one host in a network to another. Since a mobile agent is an autonomous program that can migrate under its control from host to host disconnecting a network connection to hosts, it can be efficiently utilized in unstable or heavy loaded environments [3, 4, 5]. Mobile agent system provides the software platform that manages the life cycle of mobile agents by creating, moving, executing, interpreting and disposing mobile agents. Many commercial and research mobile agent systems, for example, *D'Agent* [3, 4, 5], *Aglets* [6, 7], *Concordia* [8], *Grasshopper* [9], *JumpingBeans* [10] are introduced.

To solve the partial connection problem by utilizing mobile agent technology, some additional functions are needed. First, a mobile server has the functionality for monitoring the connection status of the server to a mobile host when the server wants to return a mobile agent to the host. Second, a mobile agent needs a directory service to choose the adequate servers according to the user's requests when the agent arrives at the host. Third, a mobile host needs the functionality for transferring mobile agents according to the list of servers provided a directory service in parallel and gathering results from the agents.

In this paper, we design PCM to solve the partial connection problem, implement and apply it to a mobile market agent service called *MMAS*, a kind of mobile computing services.

The paper is organized as follows. We survey previous works to solve the partial connection problem in section II. And we propose the mobile agent system architecture to solve the partial connection problem in section III. In section IV, we implement the proposed Partial Connection Manager and apply it to MMAS. And finally, we conclude and describe further studies in section V.

II. PARTIAL CONNECTION PROBLEM

Gray and Brewington proposed a laptop docking system as a trial to solve the partial connection problem by utilizing mobile agent technology [5]. Each mobile host has an associated dock, which is a permanently connected machine within the network. A stationary agent called the dockmaster runs on each dock machine. If an agent wants to visit the mobile host, it first tries

Phyoung Jung Kim is with the Chungbuk Provincial College, 40, Kumku-ri, Okchon, Chungbuk, 373-807, Korea (corresponding author to provide phone: +82-43-730-6351; fax: +82-43-730-6359; e-mail: pjkim@cbpc.ac.kr).

Seogyun Kim is with Korea Evaluation Institute of Industrial Technology, 58-4 Hwaam-dong, Yuseong-gu, Daejeon, 305-348, Korea (e-mail: smartlion@keit.re.kr).

to migrate directly to the host. If the migration fails, the agent transfers itself to the dock, adds itself to the dockmaster's queue of waiting agents, and goes to sleep. As soon as the host reconnects to the network, the local dockmaster forwards all waiting agents to the appropriate destination.

Brewinton proposed a mobile agent planning system to solve the Traveling Agent Problem, which is similar to the Traveling Salesman Problem [4]. The mobile agent planning system is to determine the list of servers to which a mobile agent visits to minimize the total waiting time until the needed information is found. The system is not developed for a mobile computing environment, but is used to solve the Traveling Agent Problem consisted of a planning module, a network monitoring module, and a yellow page module.

III. Partial Connection Manager Architecture

In the paper, we added PCM on the top of Mobile Agent System (MAS) in a mobile server to solve the partial connection problem. PCM is consisted of three stationary agents that are, Parallel Planning Module (PPM), Network Monitoring Module (NMM), and Directory Service Module (DSM) as in Fig. 1.

Fig. 2 shows data flows among modules of the proposed Partial Connection Manager.

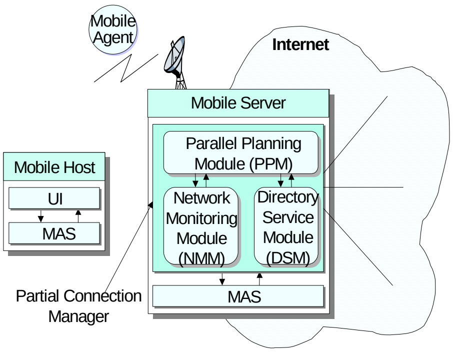

Fig. 1 Partial Connection Manager in the mobile server

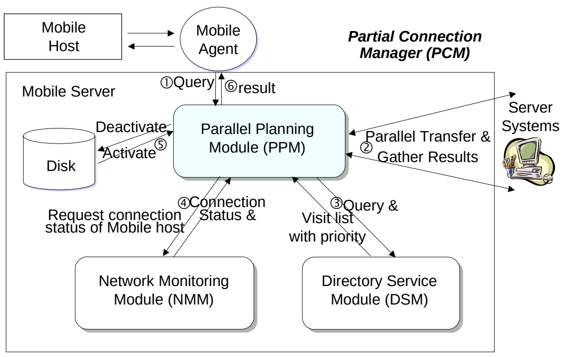

Fig. 2 Data flows of Partial Connection Manager

When a mobile agent with user's requests arrives at a mobile server from a mobile host, it first transfers user's requests to PPM. PPM inquires the list of servers of the mobile agent to visit to DSM. DSM returns the list of servers to be visited with priorities. The priority is a probability to find the needed information. In this paper, we assume the probability can be measured. PPM creates mobile agents of which number is determined by a priority, and transports them in parallel.

PPM receives mobile agents from mobile hosts, and creates the visiting list to find the servers adequate to satisfy user's requests through DSM. A mobile server transports mobile agents in parallel with servers in the visiting list, and waiting results from mobile agents during the limited period. The mobile server gathers and analyzes the results, which mobile agents return until the designated time is over, and returns the analyzed results to mobile hosts. At this time, the mobile server monitors the network connection status between the server and the mobile host through query to NMM, returns the mobile agent to the mobile host or adds the agent to the associated queue deactivating the agent according to the connection status. The status of the network connection between the mobile server and the specific mobile hosts is monitored through the periodic polling. Meanwhile, the mobile host sends a new IP address to PCM whenever it reconnects to the network. When the disconnected mobile host is reconnected to network by either case, PPM solves the partial connection problem by activating all waiting agents in queues and returning the gathered results to the associated mobile hosts. In the paper, we assume that IETF IP mobility support [11] is able to solve a location setup problem.

A. Parallel Planning Module

In this section, we describe parts of PPM in detail.

1) Directory Service Interface Part

The directory service interface part receives the user's request message *UREQ* from a mobile host through *receiveUser(Message req)* function as in Fig. 3. PPM sends *DREQ* message containing title through *sendDService(Message req)* function to DSM, and receives *DRES* message through *receiceDService(Message msg)* function. *DRES* message contains domain names and priorities of servers to be visited. In this paper, PPM creates mobile agents according to the number of servers with priority more than 50%, and transports them in parallel. We assume that DSM has the ability to calculate a priority as a probability like *Yahoo* or *Lycos*.

2) Parallel Transfer Part

Fig. 4 shows the details of parallel transfer part of PPM. The parallel transfer part of PPM sets up the time out through *setTime(int timeOut)* function, and gathers only the returned results during the designated period. The number of mobile agents to be transferred in parallel is determined by the priority in DRES message. To discard the results of mobile agents arrived after the designated time an address of the mobile host, a Master ID, and an Aglet ID are saved in a table. The transport of a mobile agent is performed by invoking *dispatch (URL url)* function. The parallel transfer part of PPM transfers the history of visited servers to DSM to provide further efficient directory services before combining the results of mobile agents.

3) Network Monitoring Interface Part

Fig. 5 shows the network monitoring interface part of PPM. The network monitoring interface part inquires about the current status of network connection of the recipient through *sendNMonitoring (URL url)* function to return the result of a mobile agent to the associated mobile host. If connected, PPM dispatches the mobile agent to the mobile host immediately. If connection is down, PPM deactivates the mobile agents, and saves it in the associated queue through *deactivateAglet(long duration)* function. When the mobile host reconnects and requests the result, PPM should activate all waiting agents to be transported to the mobile host. For this, PPM has an associated queue for each mobile host.

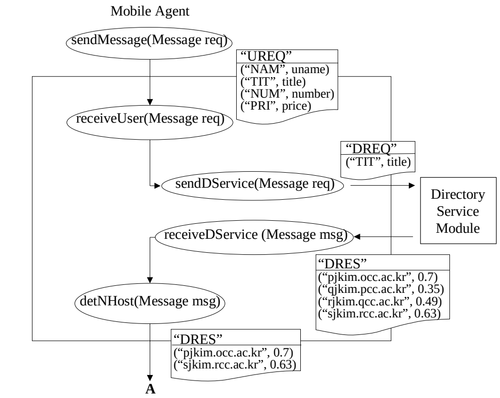

Fig. 3 Directory Service Interface Part of PPM

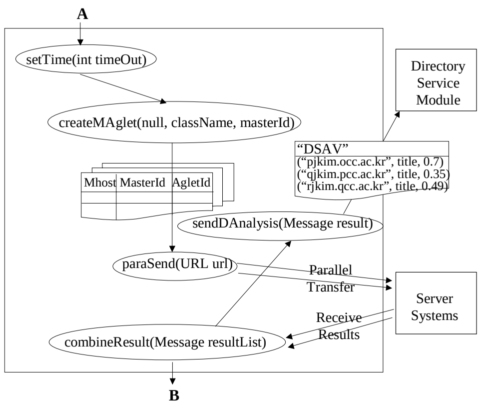

Fig. 4 Parallel Transfer Part of PPM

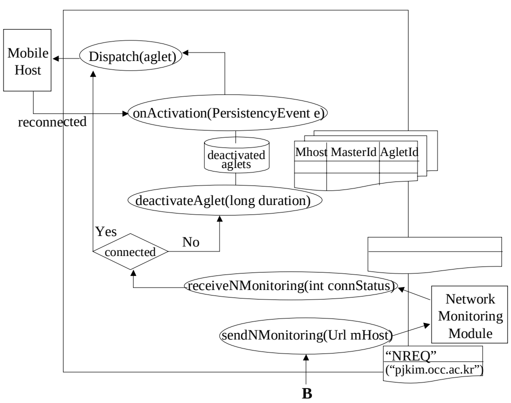

Fig. 5 Network Monitoring Interface Part of PPM

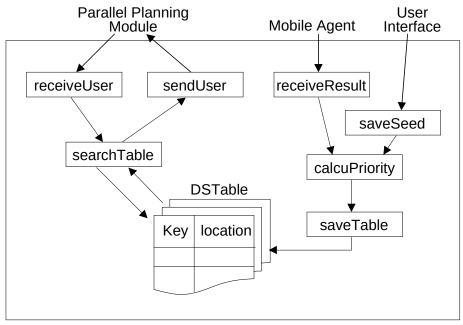

Fig. 6 Directory Service Module

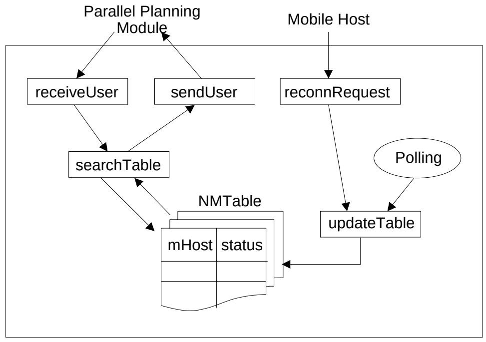

Fig. 7 Network Monitoring Module

B. Directory Service Module

DSM receives requests from PPM, searches the directory service table called *DSTable*, and then returns the list of servers to be visited and the priority. *DSTable* is constructed with two methods. One method is that the manager of a mobile server inputs a seed address of the server to be visited via user interface at the initial setup phase. Another is that *DSTable* is constructed with analysis of the searching results after transferring mobile agents in parallel. DSM analyzes the visiting results and stores them. The more the number of visited servers increases, the closer the stored information is to user requests. Fig. 6 illustrates the details of DSM.

C. Network Monitoring Module

NMM monitors the current status of network connection to the mobile host to which a mobile agent returns. For this, NMM should maintain the correct network status between a mobile host and a mobile server through periodic monitoring. NMM maintains current network connection statuses of the interested mobile hosts in *NMTable*. Although delay, bandwidth, current CPU load, and packet loss rate can be considered as network connection status, if network between a mobile host and a mobile server is connected is considered in this paper. Fig. 7 shows the detailed processing functions of NMM. Reconnection to the specified mobile host from a mobile server can be performed in two methods. First, whenever a mobile host requests the reconnection to the mobile server, NMM returns current network address of the mobile host. Second, NMM polls the mobile host through the known address. Although a mobile host is actually connected to a mobile server, the reconnection of the mobile host to the mobile server may not be realized because of network congestion. In case that a mobile host requests the reconnection to a mobile server, the mapping between current address of a mobile host and the old address of the same host is needed for transparent mobility. We assume that IETF IP mobility [11] can solve the address mapping.

IV. Applying Partial Connection Manager to Mobile Market Agent Service

As a basic system to implement the proposed PCM in a mobile server, Java-based Aglets is used. And, to apply PCM to a mobile computing service, Java APIs are developed on Java Virtual Machine under Windows NT system. We implemented a mobile market agent service called MMAS as a mobile computing service.

Overall system configuration of *MMAS* is shown in Fig. 8. A MAS and user interface called *MMmaster* are installed in a mobile host. A MAS and a PCM are installed in a mobile server, and a MAS and *MMserver* are installed in a server system. *MMserver* provides a simple commercial trading simulation.

Fig. 9 shows user interfaces of MMAS in the mobile host. The user inputs the user and goods information. This requests either the buy or the sell by pressing either the buy- button or the sell-button. And then MMAS creates the mobile agent of having the inputted user and goods information. The mobile agent is migrated to mobile server and is played the given role.

Fig. 10 shows the mobile market server in the server system. It shows the arrival message of mobile agent in the Market Status List. The mobile agent registers either Buying Item List or Selling Item List and waits either Buyer Waiting Place or Seller Waiting Place according to either the buying role or the selling role. If it trades among the mobile agents in the server system, it informs Status List, Item List, and Waiting Place of indicating the appropriate messages. And then, the mobile agents go home.

If the mobile agent goes back home with the processing result, it shows its state in the Status List, and shows the dealing result in the Mmresult screen in the Fig. 11.

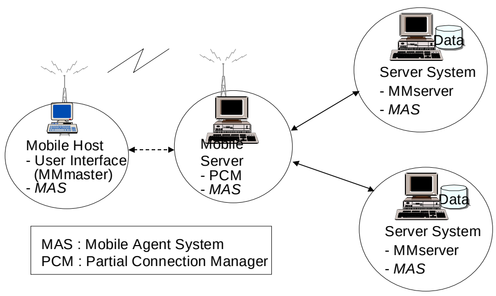

Fig. 8 System configuration of MMAS

Fig. 9 User Interface of *MMAS*

Fig. 10 Mobile Market Server of *MMAS*

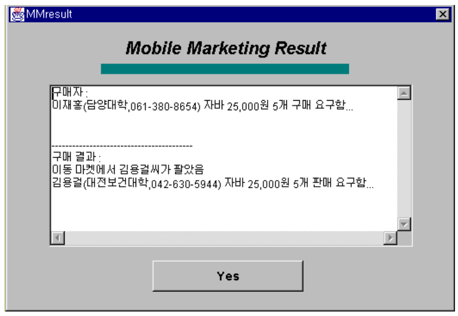

Fig. 11 Result Dialog Box of *MMAS*

V. Conclusion and Further Study

In this paper, we proposed the architecture of a mobile agent system for providing mobile computing services efficiently. Especially, to solve the partial connection problem we added PCM to MAS of a mobile server.

PCM is consisted of three stationary agents, such as PPM, NMM and DSM. PPM receives mobile agents from mobile hosts, and creates the visiting list to search the servers adequate for satisfying user's requests through DSM. A mobile server transports mobile agents in parallel to servers in the visiting list, and waiting results from mobile agents during the limited period. The mobile server combines results to mobile agents return until the designated time is over, and returns the analyzed results to mobile hosts. At this time, the mobile server monitors the current status of the network connection of recipients through query to NMM, returns the mobile agent to the mobile host or adds the agent to the associated queue deactivating the agent according to the connection status.

To evaluate the performance of the proposed PCM in a mobile computing environment, we implemented a cyber-market called *Mobile Market Agent Service*. We found that a mobile agent technology was adequate to a mobile computing service through the implementation of *MMAS*, and found out that the proposed PCM could solve the partial connection problem caused by a mobile computing environment.

Compared with Gray's laptop docking system, the proposed system has many differences. First, the proposed system has the asymmetric structure by installing a PCM in a mobile server only. Second, DSM utilizes the visiting history for providing better directory service. Third, the network connection of only mobile hosts, which request service, is monitored in the proposed system. Fourth, mobile agents are transported in parallel.

The choice of parallel or sequential search of a mobile agent, and the scheme to filter off the needed data only by analyzing the search results from web servers efficiently are under study.

REFERENCES

[1] K. Brown and S. Singh, "A Network Architecture for Mobile Computing, " *in Proc. of INFOCOM'96*, IEEE, pp.1388-1396, March 1996.

[2] G. H. Cho, "Location and Routing Optimization Protocols Supporting Int ernet Host Mobility," *Ph.D. dissertation, Univ. of Newcastle,* U.K., Dec. 1995.

[3] R. Gray, D. Kotz, S. Nog, D. Rus and G. Cybenco, "Mobile Agents for Mobile Computing," *in Proc. of the 2nd Aizu Int'l Symposium on Parallel Algorithms/ Architectures Synthesis (pAs97),* Fukushima, Japan, p.17, M arch 1997.

[4] B. Brewington, R. Gray, and K. Moizumi, "Mobile Agents in Distributed Information Retrieval", *in Intelligent Information Agents, Springer Verla g*, 1999. http://agent.cs.dartmaouth.edu/papers/

[5] Robert S. Gray, Agent Tcl: A flexible and secure mobile-agent system, *P h.D. dissertation, Dartmouth College*, Hanover, N.H., June 1997.

[6] D. B. Lange and M. Oshima, Programming and Deploying Mobile Agent s with Aglets, Addison-Wesley, 1998.

[7] M. Oshima, G. Karjoth and K. Ono, Aglets Specification 1.1 Draft, Septe mber 1998. http://www.trl.ibm.co.jp/aglets/spec11.html

[8] David Wong, Noemi Paciorek, Tom Walsh, Joe DiCelie, Mike Young, Bi llPeet, " Concordia: An Infrastructure for Collaborating Mobile Agents," *In Mobile Agents: First International Workshop,* LNCS Vol. 1219, Sping er-Verlag, Berlin, Germany, 1997.

[9] Markus Breugst and Thomas Magedanz, "Mobile Agents-Enabling Tech nology for Active Intelligent Network Implementation,*" IEEE Network Magazine, Special Issue on Active and Programmable Networks,* Vol. 12, No. 3, May/June 1998.

[10] Ad Astra Engineering, Inc., "Jumping Beans," *White Paper,* October 199 9http://www.JumpingBeans.com/

[11] C. E. Perkins and D. B. Johnson, "Mobility Support in IPv6," in *Proc. of 2nd Int'l Conf. on Mobile Computing and Networking (Mobicom'96)*, 199 6.

Phyoung Jung Kim received the B.S. degree in Computer Science & Statistics, from Chungnam National University, Korea in 1985, M.S. degree in Computer Science from Korea Advanced Institute of Science and Technology, Korea in 1995, Ph.D. degree in Computer Science from Chungnam National University, Korea in 2000. He had been with Electronic Telecommunication Research Institute, Korea from 1988 to 1998. Since March 1998, he has been with Dept. of Computer Information, Chungbuk Provincial College, Korea. He is currently interested in mobile computing, mobile intelligent agent, and network computing.

Seogyun Kim received the B.S., M.S., and Ph.D. degrees in electronics engineering from Chonnam National University, Kwangju, Korea, in 1992, 1996, and 2002, respectively. Since April 2002, he has been with the Institute for Information Technology Advancement. He is currently interested in IPTV services, mobile services, broadcasting systems, storage systems and network devices.