

Reliable Schemes in Jini Middleware for Mobile Ad Hoc Networks

Sung-Joon Lee, Jai-Hoon Kim, and Young-Bae Ko

College of Information and Communication
Ajou University, Suwon, 443-749, SOUTH KOREA
{piro00, jaikim, youngko}@ajou.ac.kr

Abstract—Jini system, the middleware for service oriented computing in wired and wireless networks, may operate well in a static environment where all the components are not changing their initial locations. However, it may not always true in a dynamic networking environment such as ad hoc networks that are formed dynamically by mobile nodes. Thus, the provision of services in the network is unreliable. In this paper, we investigate problems that can occur when mobility is considered in the current Jini system and propose reliable schemes for adapting to the mobility of Jini components.

I. INTRODUCTION

Jini [1] system is the middleware for service oriented computing [2] in wired and wireless networks. The system consists of three entities; lookup services [3], service providers, and users. Service providers are generally required to register their services to lookup services in a form of the proxy. Users request their necessary services to the corresponding lookup services. Lookup services then send the service proxy to users so that users can be offered the wanted services from service providers. This system may operate well in a static environment where all the components are not changing their initial locations in a network topology. However, this may not always true in a dynamic networking environment such as ad hoc networks [4] that are formed dynamically by mobile nodes.

The mobility is the core of problems. We believe, at least three problems can occur when mobility is considered in the current Jini system. Three possible problems are the absence of lookup service, the incorrect information about service provider, and the poor service provider selection. In Figure 1 (a) we present the mobility of lookup services. When the lookup service moves or disappears, Jini system may not work correctly because it does not consider mobility in its current spec. The effect of moving service providers is depicted in Figure 1(b). When the service provider moves, the information about the service previously registered into the lookup service becomes stale. If there are clients using the service, they cannot be served any more due to the absence of the service provider.

This research was supported by University IT Research Center Project and by a grant of the 21C Frontier R&D Program, Ministry of Science & Technology in South Korea.

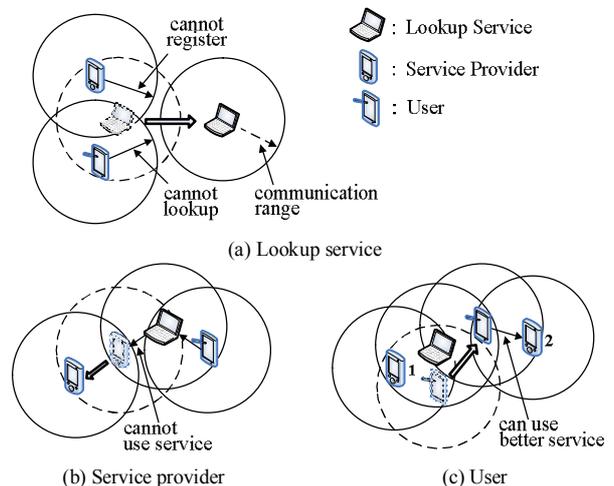


Fig. 1. Three possible problems caused by the mobility of all the components in Jini system.

Figure 1(c) depicts the effect of moving users. When users move away from their current position, any available service (provided by the service provider no. 1 in the Figure 1(c)) will no longer be available. Instead, a new service provider no. 2 is now closer to the users. Let us assume that the service quality by the service provider 2 is superior to that by the provider 1. Therefore, the user should be able to change its connection to the better service provider. We now look at each problem more in detail, and see how those problems of Jini can be solved in Section 2.

II. THE PROPOSED SCHEMES

A. Lookup service election algorithm

In Figure 2, any node detecting the absence of lookup service floods a message to inform its neighbors of the information. The message has a hop count field, which increases whenever it is forwarded by intermediate nodes. The message will eventually reach to nodes located at the edge of the network. Then these edge nodes are required to reply with a message that contains the current hop counts to the initial sender. On receiving multiple reply messages, the sender selects the

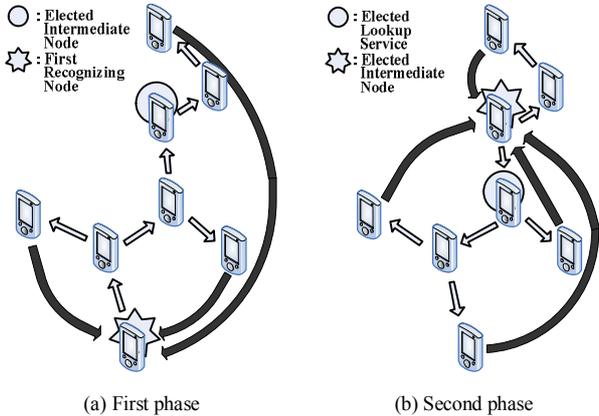


Fig. 2. Our proposed lookup service election algorithm.

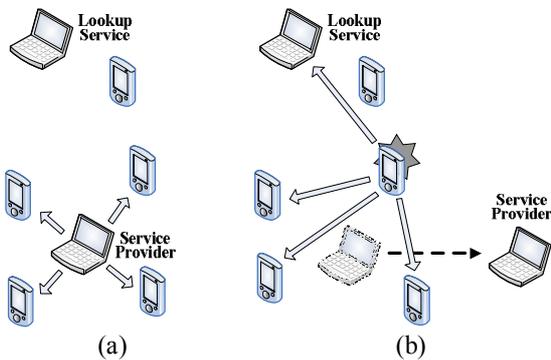


Fig. 3. (a) Service Provider informs neighbor nodes of its presence. (b) Neighbor node informs other neighbors and lookup service of service provider's absence.

largest hop counts and elects the intermediate node that is located in the half of that largest hop counts. The elected intermediate node is requested to repeat the same operation, until it can calculate the network size. This phase is to locate the lookup service in the middle of the network. The elected intermediate notifies the elected node of the election information, and the elected node performs the function of lookup service.

B. Consistency mechanism for lookup service

In Jini system, service providers inform the lookup service of its presence through the leasing mechanism [5]. However, the lookup service may not have the correct information about service providers in leasing time. Figure 3 shows our consistency mechanism. The service provider informs its one-hop neighbors of its presence periodically before the leasing time is ended. Assuming that the service provider moves away, if a neighbor node doesn't receive the message from the service provider in a determined period, the node sends the query message to the service provider in order to verify if the service provider is still alive. If there is no reply from the service provider, the node recognizes the absence of the service provider. The node then informs other neighbors and its lookup service of this information.

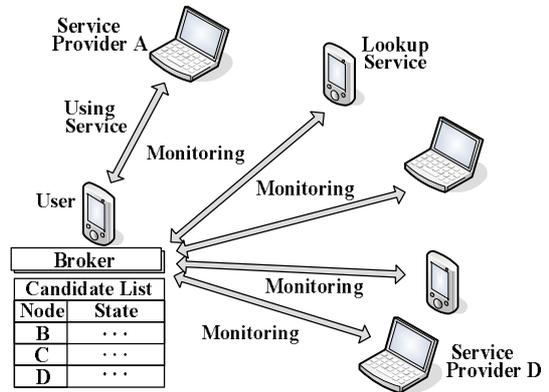


Fig. 4. Broker mechanism.

C. Broker mechanism for seamless service

In our mechanism, each user has a broker managing the candidate list which has nodes' id and their service states. When a user queries a service to the lookup service, it responds with the list of nodes' id which is able to provide the service. This user selects the node which is likely to offer the service with the best quality and maintain all the candidate lists. While the user is being provided with the service from the best node, it monitors candidate nodes' service state. In Figure 4, if the connected service provider *A* moves away, the user no longer can be offered the service. In this case, the broker selects one of the nodes in the candidate list and requests the service. The user is able to be provided the service not from the beginning, but from the stopped point. If the user is mobile, the service state can be worse and other service provider's service state can become better along with the mobility. Then the broker stops the existing connection and connects the user with better service provider.

III. CONCLUSION AND FUTURE WORK

Mobility in ad hoc networks can cause some problems to Jini system, so we proposed reliable schemes in Jini middleware for mobile ad hoc networks. We are currently implementing our schemes by modifying the existing Jini and adding new functions to it. The implementation will be tested on AODV (Ad hoc On-Demand Vector) routing protocol in Linux environment.

REFERENCES

- [1] Jini specifications <http://www.sun.com/jini/specs/>.
- [2] Rohan Sen, Radu Handorean, Gruia-Catalian Roman, and Christopher Gill, "Service Oriented Computing Imperatives in Ad Hoc Wireless Settings," Technical Report WU-CSE-2004-05, Washington University, Department of Computer Science, St. Louis, Missouri.
- [3] Wen-Hsien Tseng and Hsing Mei, "Inter-Cluster Service Lookup Based on Jini," Proceedings of the 17th IEEE International Conference on Advanced Information Networking and Applications, Xian, China, March 2003, pp. 84-89.
- [4] Radu Handorean and Gruia-Catalian Roman, "Service Provision in Ad Hoc Networks," Proceedings of the 5 International Conference on Coordination Models and Languages, 2002, pp.207-219.
- [5] Jim Waldo, "Constructing Ad Hoc Networks," IEEE International Symposium on Network Computing and Applications, October, 2001.