Enabling direct communication between mobile agents in Wireless Sensor Networks

Shunichiro Suenaga
The Graduate University for Advanced Studies,
Nihon Unisys, Ltd.
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430 Japan
ssuenaga@grad.nii.ac.jp

Shinichi Honiden
The University of Tokyo,
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo 101-8430, Japan
honiden@nii.ac.jp

Abstract
Mobile agent technologies are effective ways of deploying applications in wireless sensor networks (WSN). When mobile agents are put on a WSN that need to communicate, there are two ways to establish the communication between agents. The first is specifying the location of agents, and second is to use shared memory. Having to specify the location of agents restricts the freedom of movement of the agent. Using shared memory which is not suitable for large-scale WSN. Therefore, we need a solution by which agents can search the locations of other agents on demand to communicate directly. In this paper, we describe how to use multiple “landmarks” for memorization and provide a mapping showing the agents and their locations.

1. Introduction
When mobile agents which want to communicate with other agents are put on a WSN, they need to know the location of their targets agent. Hard-coding the agents location restricts the freedom of movement of the agent. Centralized or decentralized shared memory over the nodes is often used to overcome this problem. A unique identification of the agent and the location are written in this memory, so that agents can know the location of their target agent. However, using centralized shared memory does not scale well for large WSN. When distance from the agents location to centralized shared memory grows, agents must communicate via this single node, meaning that latency and reliability become a serious consideration for ensuring the stability of the WSN’s network connectivity. On the other hand, decentralized shared memory solves this issue, but requires communication overhead to share information and consumes a lot of energy. Therefore we need a solution which is applicable to and feasible for large-scale WSN to execute efficient agent communication.

2. Related work and an example
Most past studies on WSNs focused on only one application, not how to implement two or more applications. Moreover, when the application changes, reprogramming is necessary (Deluge [1]). Multiple applications can be realized by mobile agents, while at the same time, allowing for reprogramming. SensorWare[2] and Agilla[3] are two studies on putting agents on WSNs. Their biggest difference is between their platforms. SensorWare works on the iPAQ, and Agilla works on MOTE. Agilla is designed so that it can work on a machine with little power.

3. Approach
We use landmarks to manage the agent’s name and the agent’s location within the landmark’s coverage. The agent can send a request and get the location of the target agent. Note that the names of the agents are not unique identifications, they are the name of a type of agent which has some functionality and code. A Detector sends requests to the landmark to find out the locations of the nearest Tracker. After the landmark receives the request, sends the location of the nearest Tracker to the Detector who sent the request. Then the Detector can initiate direct communication with the Tracker. Figure 1 shows this protocol. When agents migrate between nodes, agents update their map information on the landmark.

The target agent can always be discovered within the coverage of a landmark, if target agent is located within it. In this condition there are two issues: First, the discovered agent is not necessarily the nearest one to the requesting agent. Second, if the agent migrates beyond the coverage of the landmark, it may not find the target agent within the coverage of the landmark.

To overcome these problems, we implement two functionalities. First, the agent can register their name with multiple landmarks which are close to the agent. This is appropriate because communication between landmarks requires more distance.
Second, if the agents want to migrate beyond a landmarks' coverage, they are automatically duplicated. In addition, it would be inefficient if inquiries always required requests to be sent to the landmark. To avoid this, a cache table is also installed in each node.

4 Implementation
We expanded Agilla. The table that maintained the mapping between the agent’s name and the landmark location is shared and updated by the agents who were on nodes within the landmark’s coverage. When the node is not a landmark node, the table is used as a cache table. If communication with the other agent succeeds, the locations and the agent names are registered in this cache table. It is possible for the agent to periodically get the location of the target agent and store it in the cache table. The use of a cache table enables agents to initiate direct communication with each other and reduce the communication overhead to landmarks. Figure 2 shows the sequence to discover the location of another agent. When the cache misses, the agent sends a request to the landmark to get updated information. Agents can also delete the information from the cache table.

The communication between landmarks is implemented, however we only use this communication to call the target agent when there is no target agent within the coverage area. This primitive is used when the programmer deploys agents and makes deployment of agents easy.

5 Evaluation
We have evaluated in TOSSIM. We apply the 12feet lossy model because there is an infra-red sensor whose range covers 16feet radius. The latency and accuracy of decoupled communication, this approach and this approach using a cache are shown in Figure 3. Latency stands for the duration between when a Detector detects the intruder and a Tracker completes the migration to the node where the detector is located. Accuracy stands for the success rate of this process. Figure 4 shows that if we make 5x5 cluster and agents are deployed within the coverage area of every landmark, we can assure 80% accuracy in a large scale WSN which is composed of many nodes, while decoupled communication can not scale to large scale WSN. The latency increase is because of the communication overhead to the landmark. It is possible to improve the latency by using cache.

6 Conclusion
The approach described in this paper enables agents to communicate with the target agent which locates a neighborhood on demand. We propose the direct communication model within large-scale WSN which assure a constant level of agent interaction quality in large scale WSN. In addition the programmer doesn't have to worry about the agent's location and specifying any location regarding communication.

7 REFERENCES