Efficient Fire Detection and Tracking Using Mobile Agents in a Wireless Sensor Network

Contact: Chien-Liang Fok (liangfok@wustl.edu)

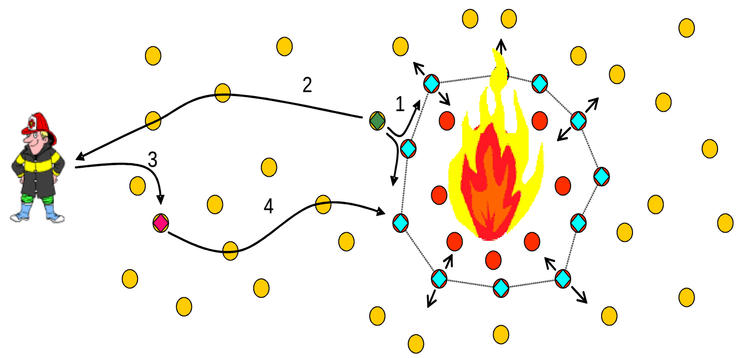

Figure 1Fire detection and tracking using mobile agents in a wireless sensor network

Objective: Wildfires are notoriously difficult to predict endangering the lives of firefighters and people living nearby. Sensor networks can help reduce this problem by allowing us to detect the position of the fire at previously unattainable resolutions. There are many projects on developing algorithms for efficiently detecting and tracking fire using a sensor network (e.g., Firebug http://firebug.sourceforge.net/). These efforts, however, focus on traditional fixed-installation infrastructures that cannot easily adopt to a changing context and prevents the network from being used for other purposes like habitat monitoring when no fire is present. Agilla is a middleware for sensor networks that utilizes mobile agents as the basic unit of execution and modularity. Instead of writing an application as a single monolithic block of code, applications written in Agilla consist of several intelligent mobile agents that can move and spread throughout the sensor network and coordinate through local tuplespaces on each node to achieve the desired application behavior. The ability to inject agents into the network and for the agent to move across nodes and coordinate through tuplespaces significantly increase application flexibility and enables greater network utilization by allowing multiple applications to co-exist. The goal of this project is to use mobile agents to monitor a network and track a fire as it spreads and retreats.

Challenges: When developing an Agilla application, the first challenge is to determine what types of agents to use. For example, one possibility is to use a single agent for detecting and tracking a fire. Initially, this agent can scour the network looking for fire. Once detected, it can clone itself to form a dynamic barrier around the fire for tracking purposes. There are many questions that must be answered even for this simple design. For example, how should an agent scour a network? Should it move back and forth along pre-defined routes in the network, or divide the network into regions and clone itself such that each region is monitored by one agent? After detecting a fire, how can an agent reliably track it to ensure the barrier is not breached for a prolonged period of time? The agents must periodically check the position of the fire and check its neighbors to see if the fire has moved, and if so, ensure the barrier has not been breached. Other challenges include how to model and detect a fire, and deliver tracking data to the base station.

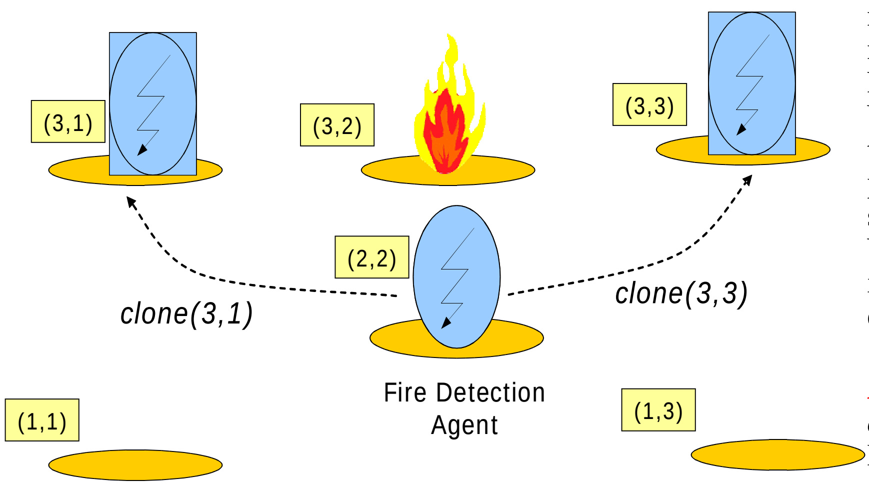

Figure 1Once a fire is detected, an agent must clone itself to neighbors to form a barrier for tracking the fire

Approaches: To model fire, we create *fire agents* that insert special *fire tuples* into the local tuplespace of the nodes that are supposed to be on fire. A fire detection agent detects fire by looking for these tuples. We have created several types of fire and fire detection agents. They differ in the way they model the fire (e.g., some are hard coded to spread to a particular set of nodes, others are designed to randomly spread starting from one corner), and detect the fire (e.g., some randomly move about while others follow fixed paths like those specified by the genetic algorithm for Multi-Mobile Agent Deployment http://newport.eecs.uci.edu/~dmassagu/dsm/MMA/MMA.html). We are still in the process of investigating their effectiveness and refining their implementation. .

COLLABORATORS	**Distributed Systems Middleware Group @ UCI** URL: http://www.ics.uci.edu/~dsm/
	This research has been supported in part by the US Office of Naval Research under MURI research contract N00014-02-1-0715 and by the Generalitat de Catalunya, the Patronat of Escola Politecnica Superior at the Universitat de Girona and the Balsells Program at UCI through a Girona fellowship.