PROGRAM A

A Sensor Network Detection Simulation System

B. Chu
Computer Science & Software Engineering, University of Western Australia, WA

Supervisors: G. J. Milne, J. Kelso, and P. Johnson.
Computer Science & Software Engineering, University of Western Australia, WA

1. Introduction

Wireless sensors can be deployed in remote areas to detect bushfires through observation of temperature, moisture, and other environmental phenomena. However, the success of a bushfire sensor network is dependent on a large number of variables, most of which are difficult or impossible to test in real-world experimentation due to cost and environmental damage. Accordingly, we use a simulation environment to examine the optimal configuration of sensors in the most common scenarios.

These sensors communicate wirelessly to share gathered data from remote locations to user accessible locations.

2. Wireless Medium Model

Wireless transmissions use the air (or the *ether*) as a medium to carry data. However, due to the many factors which affect the ether (such as background noise and moisture in the air), low-powered wireless transmissions have been found unreliable. To test the feasibility of network configurations, we must model how wireless signals spread.

We have chosen the Interacting Automata approach to capturing the wireless medium. In this method, we divide the ether landscape into multiple 'cells'. Each Cell represents the state of the ether at that location. These Cells then interact with each other to determine how far signals can travel. This means local information (such as obstacles) can be used to determine the effect on a signal as it spreads through the cells.

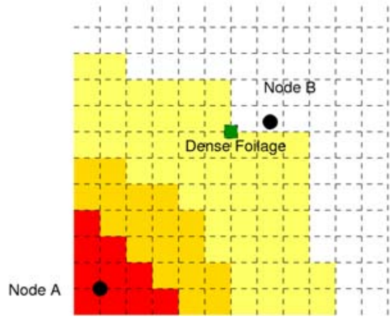


Figure: An Ether Landscape divided into Cells. The local information (the obstruction) is used to determine the overall effect on data transmission

3. Structure of the Simulator

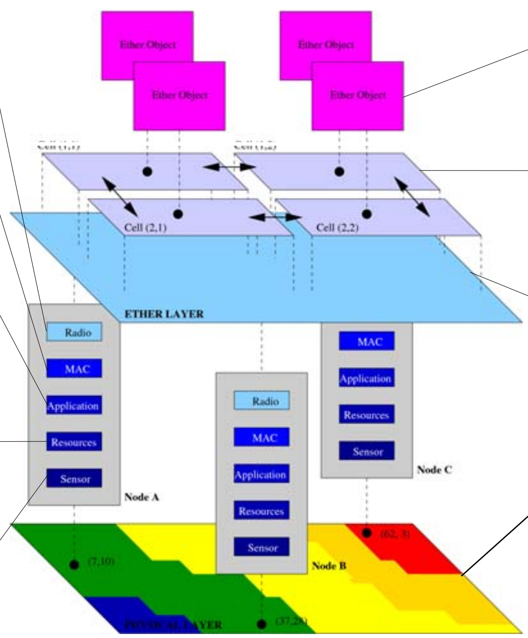
The **Radio** sub-module represents the physical antenna that is used to communicate with other nodes. This differs from the MAC module, as it deals with the physical transmission of packets.

The **MAC** sub-module controls access to the node's Radio. Packets to be transmitted can be stored here if the radio is busy.

The **Application** sub-module handles the node's programmed reaction to detected events such as sensor readings and received packets from other nodes.

The **Resource** sub-module keeps track of the power the node has remaining, the memory consumption, and CPU usage. All other sub-modules of the node send messages to this sub-node when they perform operations so it can manage available resources.

The **Sensor** sub-module represents the node's way of detecting changes in the environment. It receives messages from the Physical Layer indicating that events of note have occurred in the environment



The **Ether Object** modules each represent a portion of the ether landscape. We use a Finite State Machine to explicitly define the possible states (such as signal strength) and how to react to incoming signals.

Each **Ether Cell** contains the rules for how the signal can spread. This is kept separate from the Object so spread models can be fine-tuned separate from the state model.

The **Ether Layer** module is used to facilitate communication from Nodes to the Ether and vice-versa. This is a separate module so that cell sizes and structures can be adjusted without affecting the Node modules.

The **Physical Layer** represents the environmental factors that nodes monitor (such as heat and moisture). This Module can be a fully independent model of fire behaviour, or it can be an interface to a file containing pre-generated fire values. This allows the simulator to be coupled with fire spread data from other source (such as fire simulators or recorded data).

4. Operation

A sample simplified chain of operation follows.

1) Firstly, within the Physical Layer, a high temperature event occurs. The Physical Layer transmits a message to the Sensor of Node A.

2) The Sensor messages the Resource (the power consumed) and the Application (the temperature reading detected).

3) The Application determines that this data needs to be shared with another node and messages the MAC about transmitting this data.

4) The MAC checks if the Radio is operational and ready for transmissions before it notifies it of the outgoing data.

5) The Radio sends the data to the Ether Layer.

6) The Ether Layer determines which Cell corresponds to the location of Node A and passes the data to the appropriate Ether Cell.

7) The Ether Cell passes the data to the Ether Object, which determines the change in state.

8) The Ether Object changes state from Empty to Filled (indicating a radio signal present) and notifies the Cell of the state change.

9) The Cell checks whether there are any obstacles to its neighbours and passes the data to Cells with no obstructions.

10) Each Cell which becomes Filled will message the Ether Layer, which will pass the data to the Nodes corresponding to the Cells which became Filled.

11) The Nodes process the received data.

CRC AUSTRALIA

THE UNIVERSITY OF WESTERN AUSTRALIA