ForestFire Detection

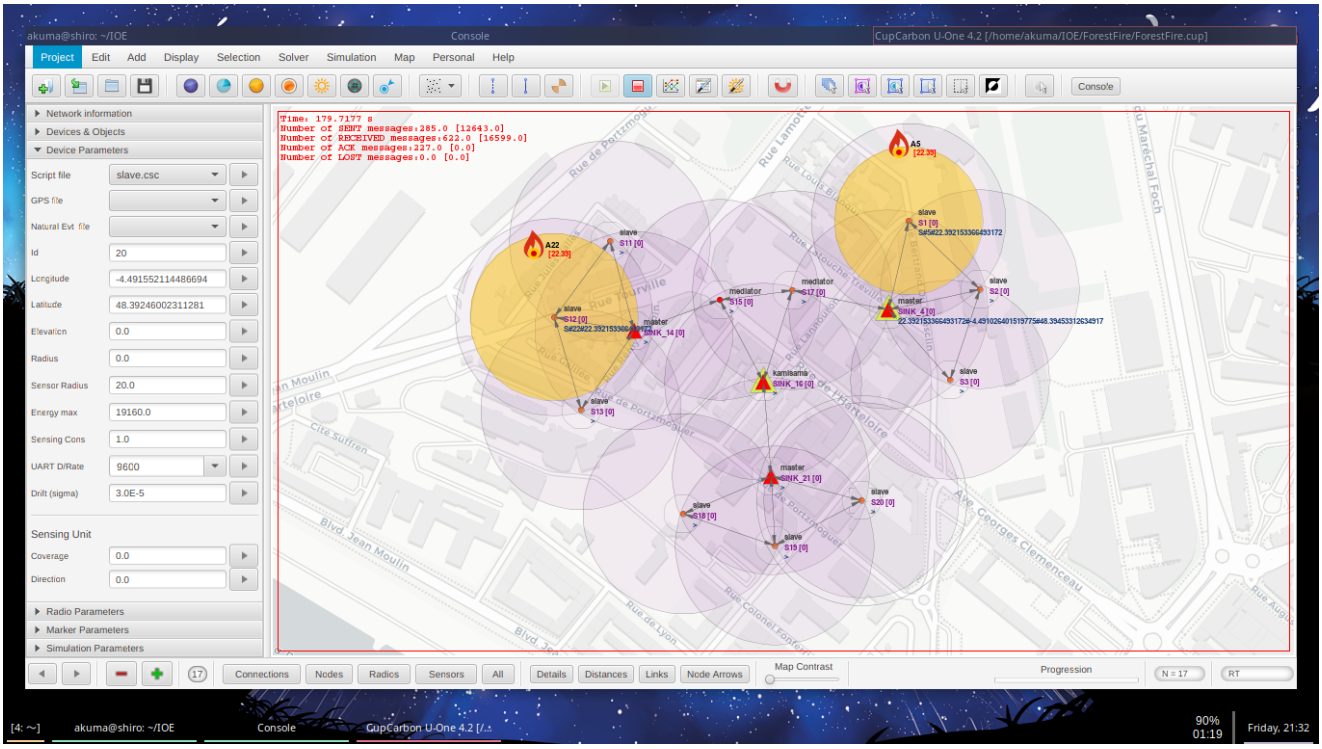
Introduction

CupCarbon is a multi-agent and discrete event Wireless Sensor Network (WSN) simulator. Networks can be designed and prototyped in an ergonomic user-friendly interface using the OpenStreetMap (OSM) framework by deploying sensors directly on the map. It can be used to study the behaviour of a network and its costs. The sensors are composed of a microcontroller, a battery, an antenna and a sensor unit. The main objectives of CupCarbon are both educational and scientific. It can help trainers to explain the basic concepts and how sensor networks work and it can help scientists to test their wireless topologies, protocols, etc. The current version can be used only to study the power diagram of each sensor and the overall network. The sensor programming is done individually in each sensor as it must have its own communication program. In this version, only sending, waiting commands, and mobility on the OSM map are implemented so far. The power diagrams can be calculated and displayed as a function of the simulated time.

Our project is meant to simulate the conditions of Fire in a forest region and use a primitive Algorithm design to Notify the respective authorities. Cupcarbon is our base tool which aids us in generation of random natural events and sensor nodes along with the base station

Architecture

The Entire system can be implemented in a subsystem as in our project. In this so subsystem, we have highlighted about 4 types of nodes.



They are:

- **Slaves** : Slaves are very basic nodes, they are the ones that are spread throughout the forest region and are *Chatty* by design, meaning they tell Everything that happens to their Master.

- **Master** : Masters are civilised nodes and non chatty by design, they are the ones that contain the logic for comparison and analysis. These analysis are preliminary by nature and are meant to be judged as the primitive methods for comparative analysis. Masters report to *kamisama* which are special nodes which monitor more than X number of masters.

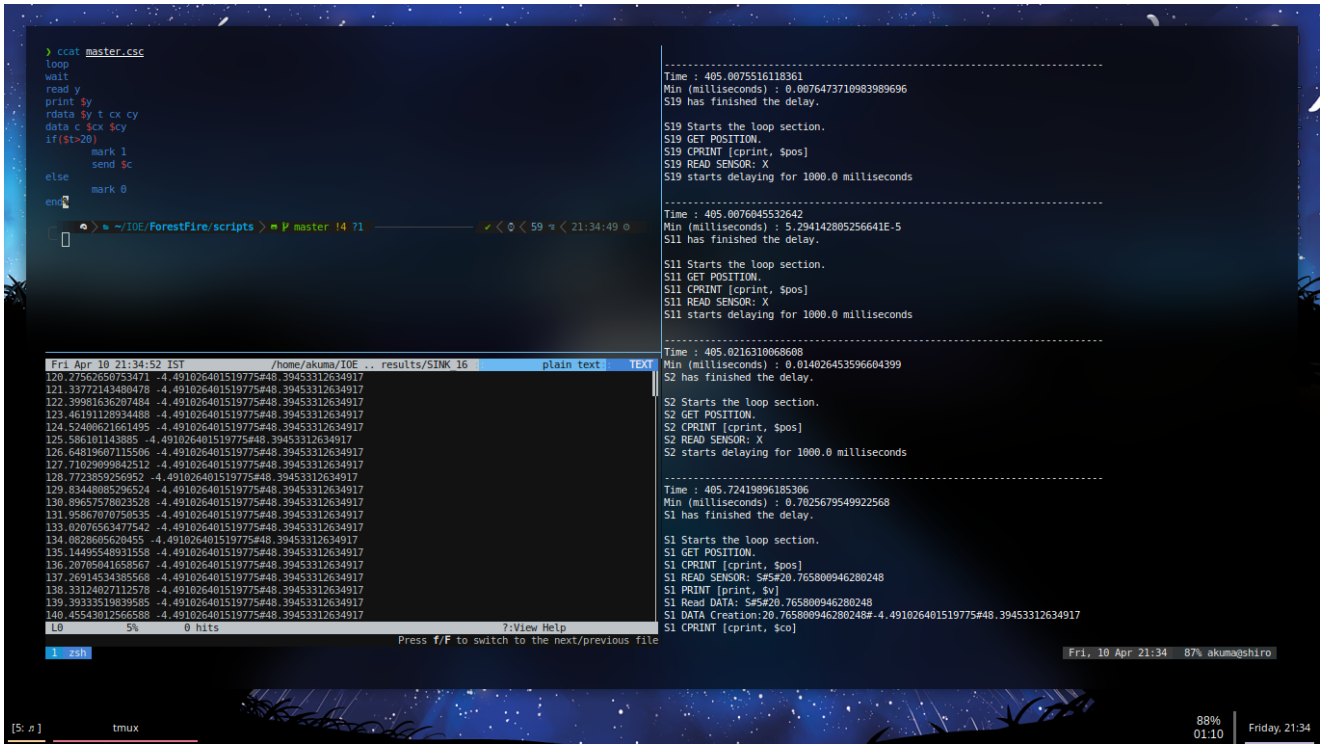
- **Kamisama** : Kamisama is the legit owner of all the masters, so as for the simplicity in the given project all it does is write in SINK_16 file (which can be further analysed/parsed by a script to Notify/Generate message).

- **Mediator** : Mediators are basic routers which routes the packet to kamisama and or masters if necessary. These are designed so as to decrease proximity between the Master and slave, which is to make sure that in bad scenario Master doesnt dies with slaves.

Working

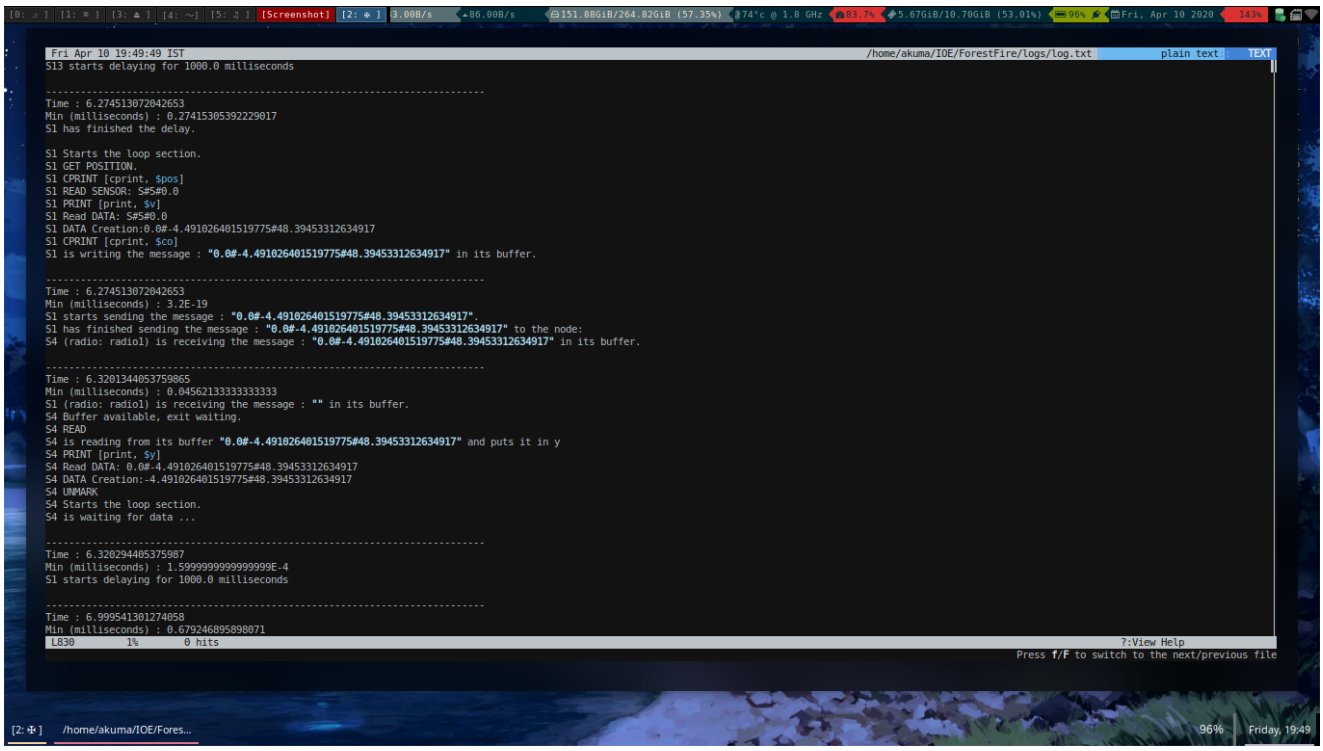
Slaves are the nodes which basically observe the temperature and notify it to master.

This simulation is achieved using Natural event generator in CupCarbon.



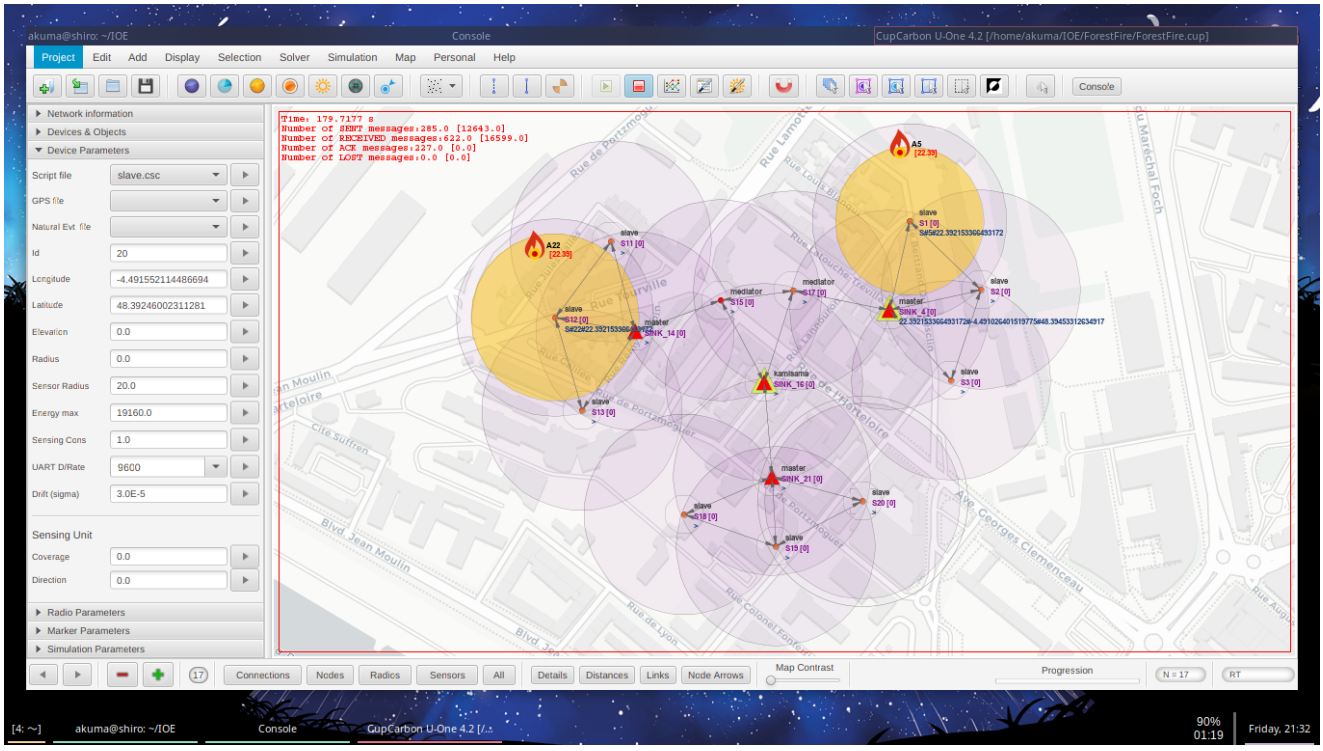
Masters simply compare them to a static value which can be predefined before deployment (**Note :** A remote synchronisation and setup protocol is under study which will certainly take

time to be developed.) Upon comparing they pass it onto kamisama, kamisama simply writes into a file for further purpose.



Simulation

Simulation is performed with two natural event at separate ends in the cluster and they highlight the Kamisama upon being received.



Assumptions

- There is no physical destruction of nodes and links right away.

- Power level consumptions havent been taken into consideration (yet!)
- Fires occur slowly and spread eventually, rather than happening in bursts. Which basically implies that there is enough time for notice of abnormal behaviour.
- The values can be hardcoded before deployment
- Notification have been done using Notify my device.

Notification System

The Notification has been achieved using API system provided by notify my device.

The simulation program writes the content (the coordinates and time) received by *kamisama* into a file SINK_16, using a python program we monitor the file and observe the time interval change.

Upon change in time, we send a notification and wait for some time between 20-30 minutes. After 30 minutes the file is again observed for changes.

Currently there is no way to mute the *kamisama* but in neat future upon implementation of synchronisation protocol, the same shall be implemented as well.

The notifications received on an android device, look like this.

(Please note that the time and coordinates are as per the cupcarbon simulator.)

![](/home/akuma/IOE/ForestFire/Screenshots/WhatsApp Image 2020-04-10 at 22.42.42 (1).jpeg)

Conclusion:

Henceforth, we have successfully simulated the cluster based network for forest fire detection with dual layer of check and compatibility. Certainly there are many more things that can be done and enchancement with the help of AI/Big data can be made, this lays down the foundation for understanding the basics of communication of clusters and masters.