Forest Fire Detection

Danika Gaviola
Department of Computer Science
Binghamton University
Binghamton, NY

Matt Rotlevi
Department of Computer Science
Binghamton University
Binghamton, NY

Emileigh Bonzek
Department of Computer Science
Binghamton University
Binghamton, NY

ABSTRACT

Forest Fires lead to the loss of billions of dollars, thousands of buildings and multiple species each year. These fires, when left undetected, have to potential for widespread detection. We created a Forest Fire Detection system that detects community destroying forest fires, and alerts emergency services so that they can stop these fires before they get out of hand.

KEYWORDS

Forest Fire, IOT, Detection, TelosB, AWS

1 Introduction

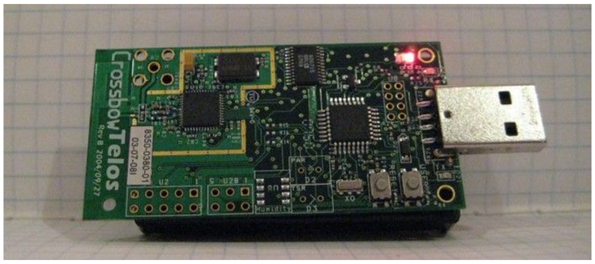
Every year, it seems as though there is a community destroying forest fire somewhere in the world. These forest fires have been known to cause widespread destruction. In 2018, California experienced forest fires that accumulated $400 Billion in damages, 25,000 buildings (including homes, schools, churches, and more) lost, and irreversible loss of biodiversity. More recently, Australia is experiencing a community crushing forest fire to this day. Many animal species are coming close to extinction due to the loss of their entire habitats. Namely, over 1000 koalas have died, along with a vast majority of their main food source, eucalyptus. We may be the last generation to have witnessed many of these animals that are facing extinction.

Although forest fires happen nearly everyday, not all forest fires are community destroying forest fires. Most forest fires burn out on their own, and are not a huge cause of concern. We propose a solution to detect these community destroying forest fires, and stop them in their tracks before they become so widespread that they risk becoming uncontrollable. Our system will detect forest fires, and upon reaching certain levels of heat and humidity, send out alerts to the fire department. This system will allow forest fires to be detected before they reach dangerous levels.

2 Design

Our design requires a TelosB Mote, and two laptops. One laptop is used to collect the TelosB data, and the other is used to collect that data using AWS and send out a ping to the fire department.

2.1 Hardware: TelosB Mote



The data is collected in a TelosB Mote. The TelosB has built in heat and humidity sensors, which we have utilized to predict dangerous fire levels.

2.2 Software

2.2.1 VMWare/TinyOS

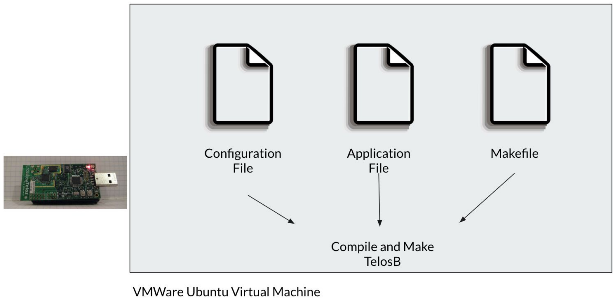
The TelosB mote is plugged into a laptop, which is running a TinyOS Ubuntu virtual machine with VMWare. This VM is used to interface with the TelosB Mote and collect the data from it.

2.2.2 Amazon Web Services

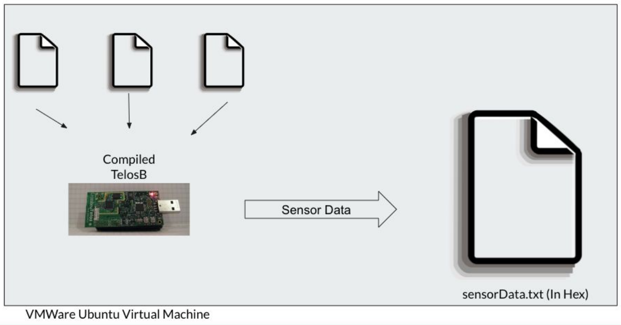
We utilized three EC2 instances and a Kinesis stream. These AWS services helped us to collect and analyze the data collected by the TelosB.

3 Implementation

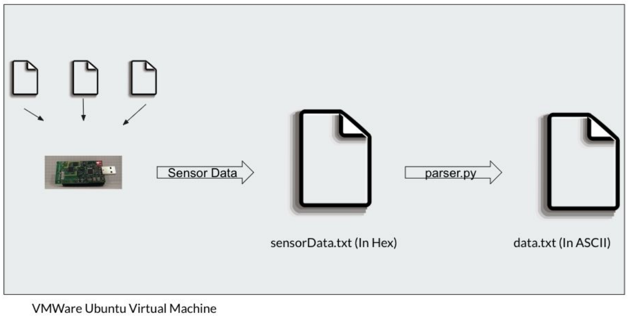
3.1 TelosB Data Collection



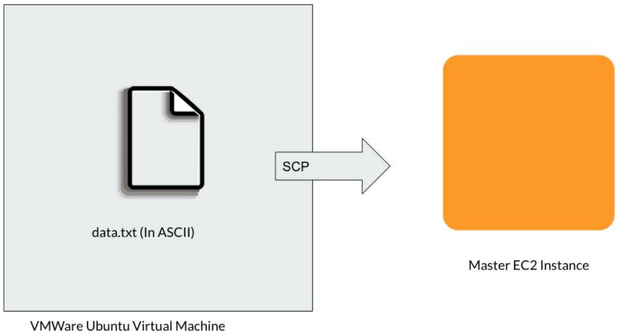
In order to collect the data, there are three main files, a configuration file, an application file, and a makefile. After compiling and installing on the telosB, the telosB is collecting temperature and humidity sensor data and printing to a serial port. All of this happens within a Ubuntu VM running TinyOS.



We have another file, a simple bash script that automates these commands and uses a java command to redirect the serial port printing to a file which it writes to in hex.

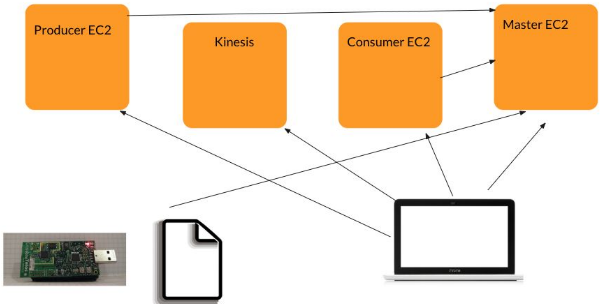


Lastly, there is a python file called parser.py that converts this hex to readable ascii and writes the output to another file called data.txt that contains the heat and humidity data.



Finally the bash script uses scp to transfer data file to the master EC2 instance.

3.1 AWS Data Analysis



On a second laptop, there are 4 AWS resources being built: A Kinesis Stream, a Master EC2 instance, and Producer EC2 instance and Consumer EC2 instance. The kinesis stream is built first, and then the master EC2 instance is created second. Once the master is successfully built, the code(infra.py) uses scp to transfer the files put_data.py and get_data.py to the /var/www/html directory where there is an Apache web server running. At the same time, telosB should be collecting and sending a text file with the temp and humidity data, called data.txt, to the master EC2.

Then, infra.py creates the 2 additional EC2 instances: the Producer and Consumer. Each EC2 instance has a startup script called user_data.sh that runs when each EC2 instance is being booted, respectively. In the producer's user_data.sh script, a program called wget is used to download the put_data.py file. In the consumer's user_data.sh script, wget is used to download the get_data.py file. Next, in infra.py we utilized a python library called paramiko which uses the SSH protocol to remotely log in to the producer EC2 and run the put_data.py. This code then does the same thing with the consumer EC2 and runs get_data.py.

get_data.py utilizes formulas that calculate the threshold for a forest fire. Given the collected heat and humidity data, and when the data reaches this threshold, we ping the master EC2 instance. This ping is a "proof-of-concept" that the consumer can detect when a community destroying wildfire is going to begin and has the ability to communicate anywhere on the internet.

Formulas Used to Calculate Forest Fire Threshold

$$P = 0.61078 \exp\left(\frac{17.27T}{T + 237.3}\right), \tag{1}$$

Formula 1, Teten's Formula, is used to calculate saturation vapor pressure of liquid and ice given the temperature. We use the temperature data gathered from the TelosB and input it into the formula to get the Saturation Vapor Pressure.

$$VPD = vp_{\text{sat}} \times (1 - \text{relative humidity}/100) \tag{2}$$

Formula 2 then uses that calculated Saturation Vapor Pressure, along with the current humidity, to find the Vapor Pressure Deficit (VPD). Upon reaching a VPD of greater than .66, the threshold for a forest fire, we gather that there is a forest fire in the process.

4 Future Work

Our project utilizes a single TelosB sensor that can be used to detect very small fires, such as a lighter or a small campfire. However, we would like to expand this project to detect large scale forest fires that have the potential to be community destroying and take over entire towns, states, or even nations. In order to accomplish this goal, we would have several TelosB sensors throughout an entire forest. If enough of these sensors are collecting environmental data, we can declare a potentially dangerous forest fire an emergency and alert the appropriate emergency services so that they may take the appropriate measures to stop the forest fire in its tracks.

ACKNOWLEDGMENTS

We would like to take the time to thank Professor Mo Sha and Teaching Assistant Di Mu for dedicating time to helping us with our project.

REFERENCES

[1] https://en.wikipedia.org/wiki/Tetens_equation
[2] http://www.publish.csiro.au/wf/Fulltext/WF19023
[3] https://en.wikipedia.org/wiki/Vapour-pressure_deficit