Low-Cost System for Early Detection and Deployment of Countermeasures against Wildfires.

Introduction to IoT
Course Project Report
Submitted by

Amulya Varshney - 191010204
Aakash Agarwal - 191010201
Piyush Rawat - 191010227

Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur

JUNE-2020

Abstract

A wildfire is an uncontrolled fire that occurs mainly in forest areas, although it can also invade urban or agricultural areas. To fight against these wildfires, we need to adopt a comprehensive approach which has an instant response and continuous screening to wildfires. In the present scenario, wildlife and forest departments are facing this huge problem. The number of trees has reduced drastically from the forest that creates an unhealthy environment for animals to survive in the forest.

It has been found in a survey that 80% losses are caused due to fire. This could have been avoided if the fire was detected in the early stages. This project proposes a system for tracking and alarming for the protection of trees against forest fires. Nowadays IOT (Internet of Things) devices and sensors allow the monitoring of different environmental variables, such as temperature, humidity, moisture etc. along with the location of the place. Arduino platform based IOT enabled fire detector and monitoring system can be considered as the solution to this problem.

In this project, we have designed and built a wildfire detection system using an Arduino UNO and Xbee module which is interfaced with a temperature and Humidity sensor and a smoke sensor. This paper describes a hierarchical approach that aimed at early fire detection in risky areas, integrated with the fire fighting command centres, geographical information systems, communication and alert systems.

Keywords- Tracking, IOT, Arduino Uno, Xbee Module, GPS

1. Introduction

Wildfires are a serious threat to the environment and they give birth to natural disasters and produce real emergency situations which causes severe destruction of both lives and resources. Every season, not only hectares of forest areas destroyed by wildfires, but also assets, properties, public resources and facilities are destroyed. Moreover, wild animals, firefighters and civilians are at risk, with a terrible toll in human and animal lives each year.

Although progress has been made in the field of wildfire fighting in the last decades, there is still a need to strengthen the disaster response capacity, including early warning systems and improvements in real time exchange of data at all stages and levels of a forest monitoring scheme such that we can save more lives and resources.

Wildfire prevention and detection systems can be considered a main goal for conserving the environment as they can detect a wildfire in advance taking in account the behavior of the environmental variables at a particular instant of time. In addition to this, the real-time monitoring of certain environmental variables will also contribute to forest profiling and make the wildfire prevention, detection, and fighting more efficient and fast.

A wireless sensor network (WSN) based on Internet of Things (IoT) devices and sensors can be used to perform a real-time environmental monitoring of the above mentioned parameters for forest profiling. The design and distribution of the system needs to aim at covering as much forest area as possible. With respect to this, several challenges must be considered, such as the authentication and communication of sensor nodes, security of wireless communications among distributed sensor nodes, taking into account possible areas out of network coverage.

Real time values of certain environmental parameters are collected and stored in cloud servers where the collected data can be analysed aiming at performing a short-term estimation of forest fire risks. Likewise, Uncertain decrease in humidity or oxygen level will raise temperature and concentrations of gases, such as carbon dioxide and carbon monoxide, which may involve a high probability of outbreaks of recent nearby fires. Therefore, continuous monitoring and profiling may make it more efficient to detect fire spread, estimated by analysing the values of temperature, humidity and smoke concentration over nearby forest areas, since these parameters have a direct impact on relevant fire propagation factors such as dryness of vegetation and organic fuels.

The main goal is to estimate the existence of wildfire risks and to detect the recent occurrence of fire outbreaks in forest areas using the data collected all over different forest areas. A particular prototype of a system using IoT devices has been designed, such that whenever a fire outbreak is detected, a decision-making method based on analyses of already collected data is enabled to determine the neighbouring forest area that is more likely to favour fire spread as a result of its current environmental conditions. Moreover, a sms alert system is also proposed aiming at activating alerts to the concerned authorities.

2. Related Works

Recent developments in information and communication technologies are already having a huge impact, specially forest fire detection systems. A lot of focus reliant on wildfire detection has been done by experts worldwide in order to consider structures fit for looking at to control fire. **Lloret et al. in 2009** proposed a remote sensor network structure which uses IP (Internet Protocol) cameras in order to perceive and affirm fire in wild areas and send a sensor alert to a central server which then picks the closest remote cameras to the multi-sensor that raised the caution, and sends them a message in order to get ceaseless pictures from the zone. This system requires better web consideration in forest zones. Another downside is the use of IP cameras which eats up greater imperativeness and requires immense information transmission to most likely send the photos. The general cost of executing this structure is too high.

A related work by **Bolourchi and Uysal in 2013** prepared an investigation work that proposes the use of WSN for data gathering to be used as unrefined information data into a control system that they made. They picked fire acknowledgement to speak to the Intelligent Decision Making (IDM) limit of the structure and made Fuzzy Logic figuring using temperature, smoke, light, dampness and detachment as limits. Probability of fire subject to fleecy rules using the status of the limits is shown by mirroring the made structure on the Matlab programming. The weakness in this investigation is that it is just a theoretical work reliant on generation on the Matlab programming to discover the probability of occasion of fire. The structure isn't attempted in the real condition to give out some understanding on how it will continue.

Summary of Related works

Sr. No.	Author and Year	Name of the solution	Methodology	Limitation
1.	**Lloret et al. in 2009**	A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification	Uses IP (Internet Protocol) cameras in order to perceive and affirm fire in wild areas and send a sensor alert to a central server which then picks the closest remote cameras to the multi-sensor that raised the caution, and sends them a message in order to get ceaseless pictures from the zone.	This system requires better web consideration in forest zones and use of IP cameras requires immense information transmission to most likely send the photos. The general cost of executing this structure is also too high.
2.	**Bolourchi and Uysal in 2013**	Forest Fire Detection in Wireless Sensor Network Using Fuzzy Logic	Use of WSN for data gathering. They picked fire acknowledgement to speak to the Intelligent Decision Making (IDM) limit of the structure and made Fuzzy Logic figuring using temperature, smoke, light, dampness and detachment as limits. Probability of fire subject to fleecy rules using the status of the limits is shown by mirroring the made structure on the Matlab programming.	It is just a theoretical work reliant on generation on the Matlab programming to discover the probability of occasion of fire. The structure isn't attempted in the real condition to give out some understanding on how it will continue.

3. Problem statement

The problem statement focuses on design and development of a low cost and effective forest fire detection system which is able to detect approaching wildfires by estimating the existence of wildfire risks and to detect the recent occurrence of fire outbreaks in forest areas using the data collected all over different forest areas. This system can be used in combination with active fire protection mechanisms, in order to alert the concerned command centres and other wildland fire fighting units in an attempt to sustain the advance of the flame front. The prime objective of this work is to develop a solution to proactively protect small areas of interest: safety of wild animals, isolated homes, small villages, camping sites, music festivals.

4. Objectives

- To detect the approaching forest fire as early as possible by measuring the level of temperature and carbon dioxide level with all measurements taken must have a time-stamp.

- Apart from the preventive measures, early detection and suppression of the fire to maintain the damage and casualties. The system needs to be run all day 24/7.

- Use of WSN and IOT-based forest fire detection systems to detect the fire by monitoring the values of carbon dioxide level, temperature and humidity.

5. Proposed Methodology

The general structure of our approach is depicted in the Figure below.

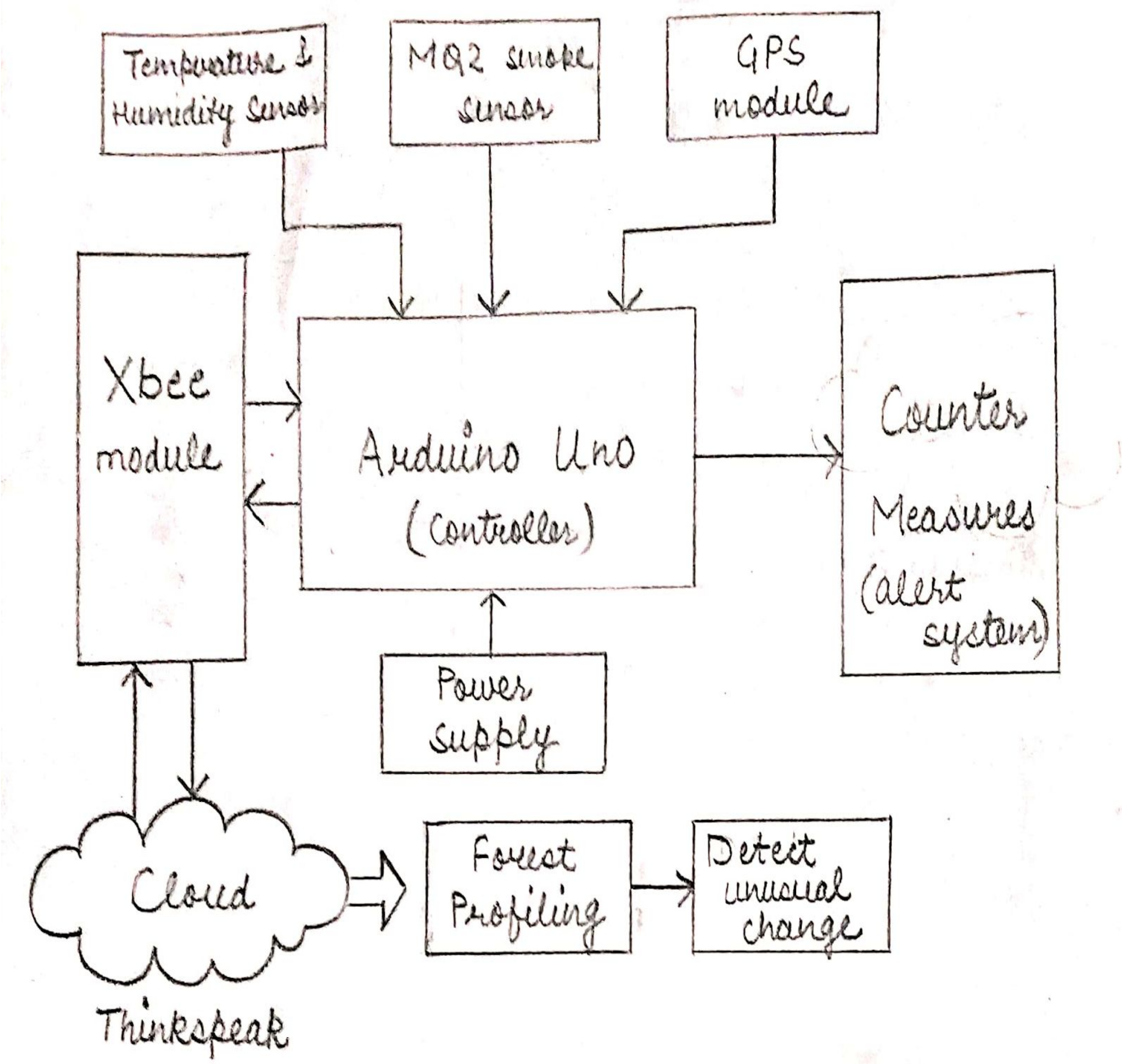
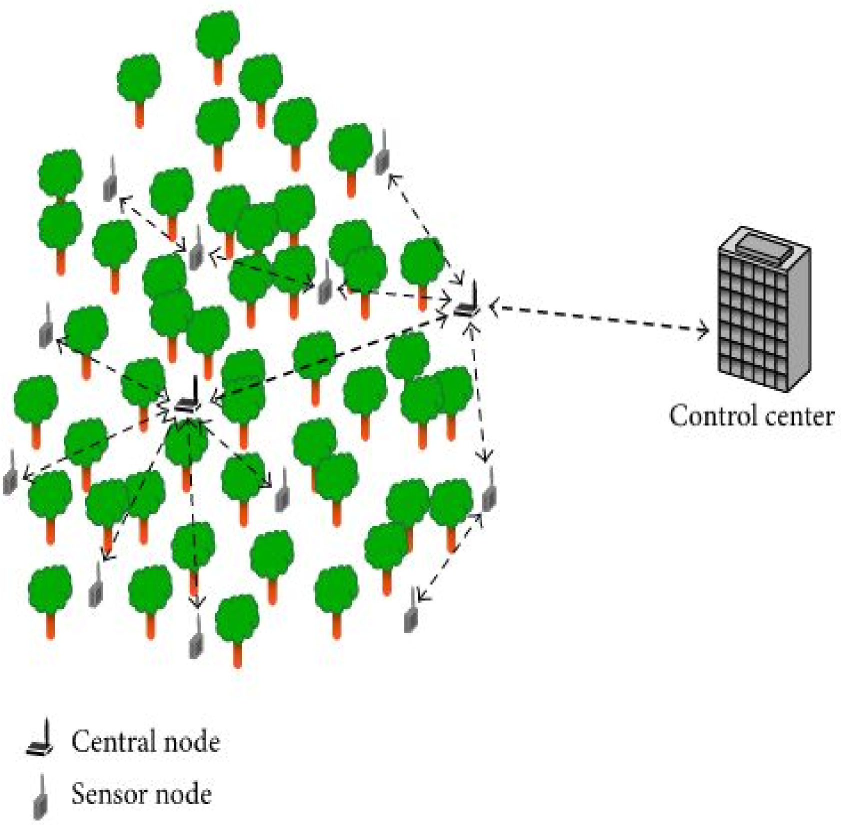


Fig 1. Conceptual view of proposed solution-IoT based Wildfire Detection System

In this project, we have designed and built a wireless wildfire detection system where each sensor node is constructed using an Arduino UNO and Xbee module which is interfaced with a DHT Temperature and Humidity Sensor, MQ2 Smoke Sensor and GPS module. The Arduino collects data from each sensor and pushes it to the cloud server (using Xbee module integrated with Ethernet) for analysis of collected data which then concludes whether there is a risk of forest fire or not.

The Xbee network is used for communication between various sensor nodes and control centres. This wireless sensor network aimed at early fire detection in risky areas, integrated with the fire fighting command centres, geographical information systems, fire simulators and fire fighting operations: fire brigades, communication systems, and aerial, coordination, and land means.

General structure of the approach proposed. Sensor nodes capture data from the environment that are uploaded to the central nodes, which transfer all the information to the system middleware.



The node structure is composed of 2 levels, central nodes, devoted mainly to short and long range communications and control purposes, and sensor nodes, used to collect data from the area under monitoring and send them to the central nodes. All node types can include environment and meteorological sensors.

In order to implement this project, we will be using an alert system which is used to provide the final alarming location to the concerned authority through the simulation program, whenever a fire occurs, the system automatically senses and alerts the authorities by sending an alert directly to the command centres and other wildland fire fighting units.

HARDWARE AND SOFTWARE REQUIREMENTS

1) Arduino Uno :- Arduino Uno is a microcontroller based on the **ATmega328**. It has 14 digital input/output pins in which 6 can be used as PWM outputs, a 16 MHz ceramic resonator, an ICSP header, a USB connection, 6 analog inputs, a power jack and a reset button. This contains all the required support needed for microcontrollers**.**

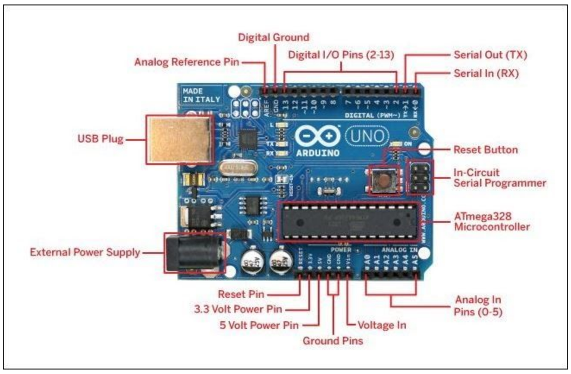


Fig 3. The schematic diagram to represent the various pins in Arduino Uno.

We are using the most official version **Arduino UNO R3**. All the sensors and modules are dirctly connected to it and it controls the whole sensor node.

2) Temperature and Humidity Sensor :- The **DHT11** is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

Fig 4. a DHT temperature and Humidity sensor.

3) Smoke Sensor :- The **MQ-2 Gas sensor** can detect or measure gasses like LPG, Alcohol, Propane, Hydrogen, CO and even methane. The module version of this sensor comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas. When it comes to measuring the gas in ppm the analog pin has to be used, the analog pin also TTL driven and works on 5V and hence can be used with most common microcontrollers.

Fig 5. a MQ2 Smoke sensor.

4) GPS Module :- The GPS receiver gets a signal from each GPS satellite. The satellites transmit the exact time the signals are sent. By subtracting the time the signal was transmitted from the time it was received, the GPS can tell how far it is from each satellite. The GPS receiver also knows the exact position in the sky of the satellites, at the moment they sent their signals. So given the travel time of the GPS signals from three satellites and their exact position in the sky, the GPS receiver can determine your position in three dimensions – east, north and altitude. The GPS system helps us to get a time stamp attached to each signal so that each signal can be synchronised effectively. We are using the NEO **6M GPS module** here.

Fig 6. a NEO 6M GPS module.

5) Xbee Module :- Xbee modules use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking. They are designed for high-throughput applications requiring low latency and predictable communication timing with the speed of 250 kbits/s. XBee devices communicate with each other over the air, sending and receiving wireless messages. However, they can communicate with intelligent devices via the serial interface. XBee devices transmit data coming from the serial input over the air, and they send anything received wirelessly to the serial output. It acts as the medium of communication between various sensor nodes, cloud servers and control system.

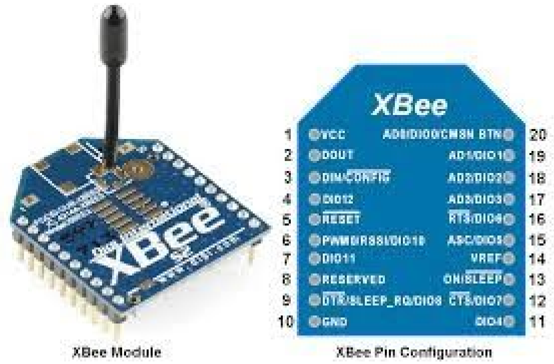


Fig 7. a XBee module and its pin representation.

6) Battery :- In our project, we are using a 12 volt battery (for each sensor node) to give power to the components of the vehicle.

7) Thingspeak Cloud :- ThingSpeak is the open IoT platform with MATLAB analytics. Our device will communicate with ThingSpeak using an API provided by it, and we can either keep your data private, or make it public. Then we use ThingSpeak to analyze and act on our collected data to aggregate, visualize and analyze live data streams in the cloud.

WORKING AND ALGORITHM USED

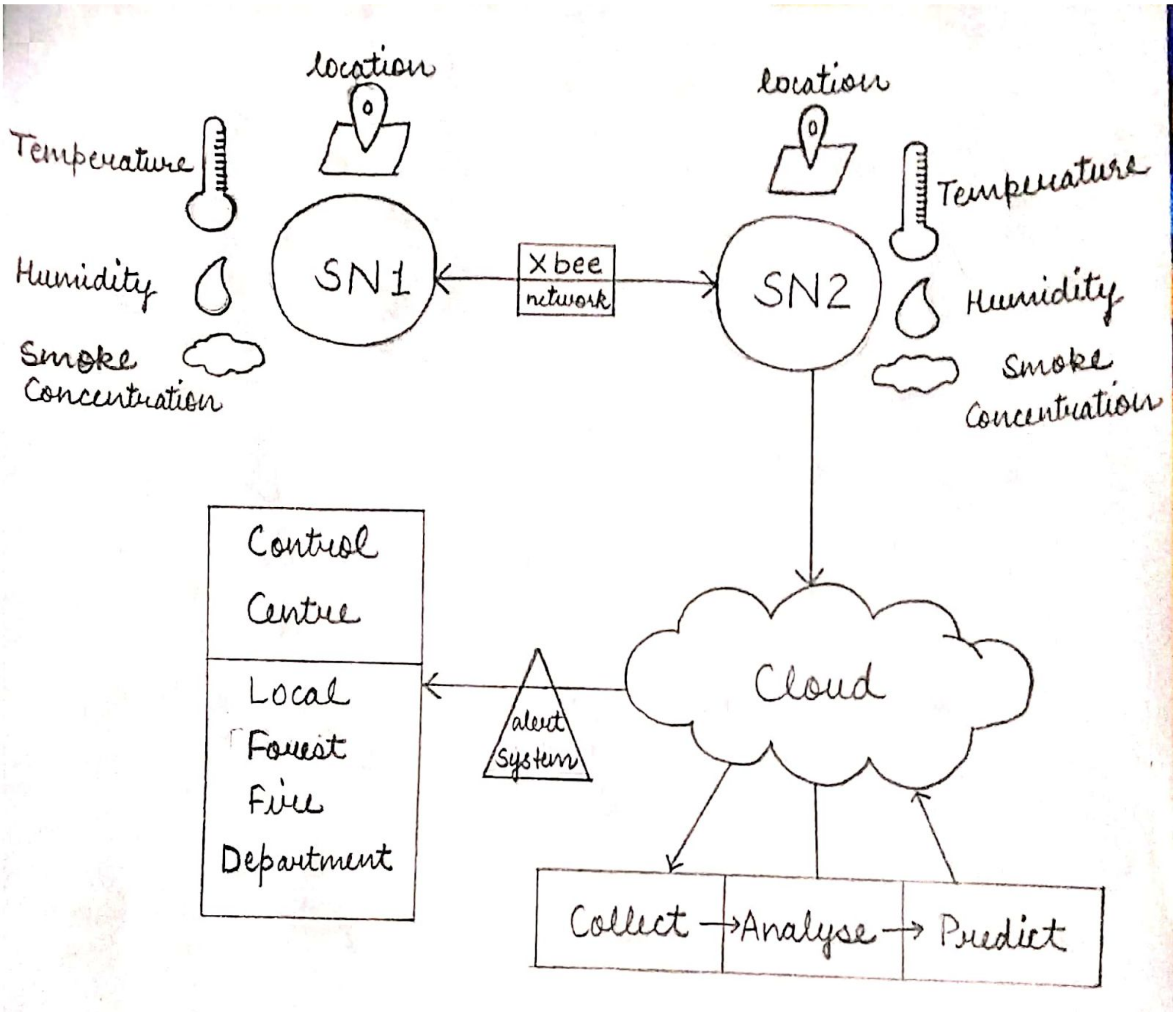


Fig 8. Block diagram of proposed solution-IoT based Wildfire Detection System

- This project comprises a WSN for early detection of forest fires. This network can be easily deployed at areas of special interest or risk. There are two types of nodes from the physical structure point of view: SNs, to collect data from the environment, and CNs, to gather data from the SNs and transmit the information to a Cloud server.

- First we'll create a sensor node which will have all the sensors, namely **DHT11 Temperature and Humidity sensor**, **MQ2 Smoke sensor** and **NEO 6M GPS module** for time stamping. **Arduino Uno** is used as the microcontroller. All the connections are done using **Jumper wires**.

- Next we connect the **Xbee Module** with Arduino and set up using **XCTU** where we plug the Xbee module into an Explorer module. After setting up, selecting the same port, channel number and PAN ID for all Xbee modules, we pick a Coordinator Enable (CE) value which is set to be 1 or 0 based on the way of communication.

- Log in to the **Thingspeak Cloud server** and push all data through the sensors using different channels and API to aggregate, visualize and analyze live data streams in the cloud. Upon computation of any risk of wildfire, a signal is transmitted directly to the central node and an alert to the control system as a preventive measure.

- We will push the data recorded by each sensor node to the cloud server where it will be analysed with respect to the location of the sensor node in the forest area. Several graphs will be generated figuring out the flow of environmental parameters with time.

- This enables a proper and seamless configuration of the network, provides redundancy, and ensures there will be full temporal and geographical coverage in the deployment zone.

- Since only a single sensor node will not be enough to cover the whole forest area, we have to deploy multiple sensor nodes with a single central node as a gateway between control centre, cloud services and other sensor nodes.

- Once a network is established between the sensor nodes and cloud server, we can deploy the system over the entire forest area. And each sensor node will have a GPS module installed so that it pinpoints the exact location of risk.

- We can also get more data from other sources for better analysis and can also use previous outcomes to better the system. These samples contain data values during a normal day and during wildfires.

- The information gathered is related not only to early detection purposes but also to environment monitoring to maximize the WSN usage. This environmental data can also be employed to fire fighting preventive tasks such as vegetation modeling, microclimate studies, and propagation model parametrization.

- For each recorded data that is pushed to the cloud, will now be compared with the already stored data and statistics against the pre-recorded data. And based on that if the data shows any resemblance to the pre-recorded data during a wildfire, we can assume that there is a wildfire.

- In the next step, we will trace back the risky location by finding which sensor node is sending the data using GPS module so that we can get an accurate location in the forest.

- This location will be immediately forwarded to the **Local forest fire department** (Control Centre) with a warning message and all the nearby residents are also alerted as a protective measure.

6. Results & Discussion

The wireless sensor network(WSN) for wildfire monitoring systems is very useful for wireless instrumentation and communication in fire-prone areas to detect and stop the spread of destructive, environmentally hostile wildfires.

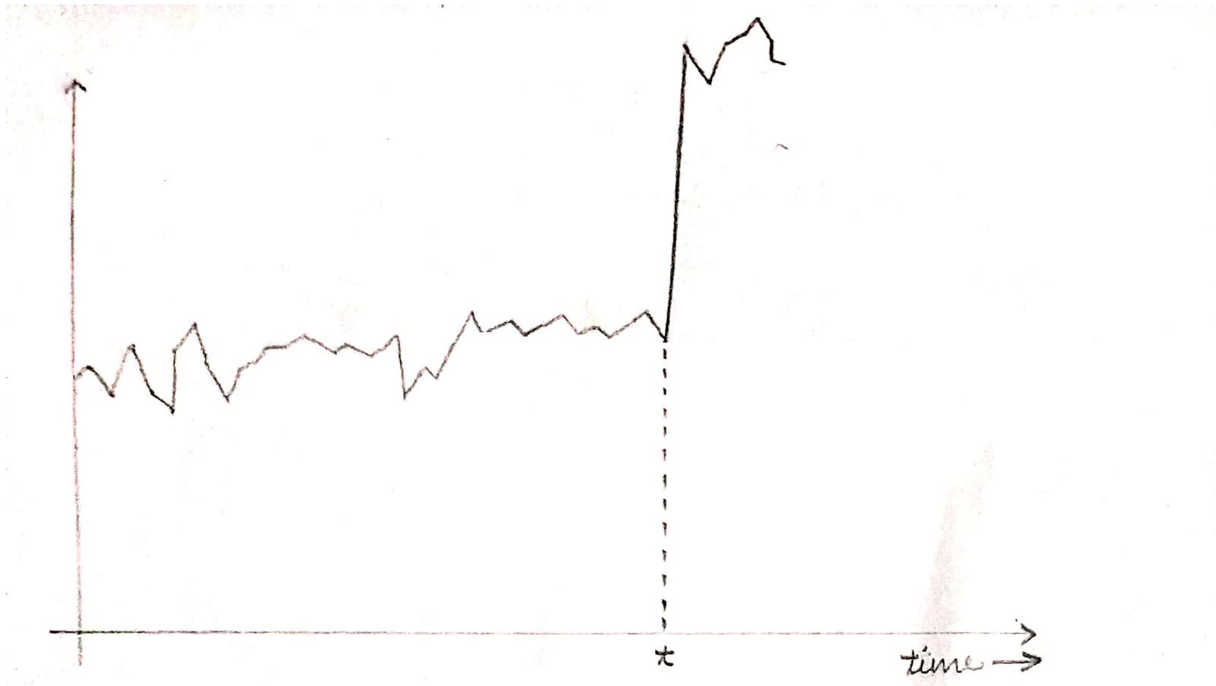
- We created a wireless network between sensor nodes which measures major environmental variables such as temperature, Relative humidity, Carbon Monoxide and Carbon dioxide.

- **time synchronization**: The network system is able to get the current time from the GPS module and the data collection and analysis is instant using cloud services.

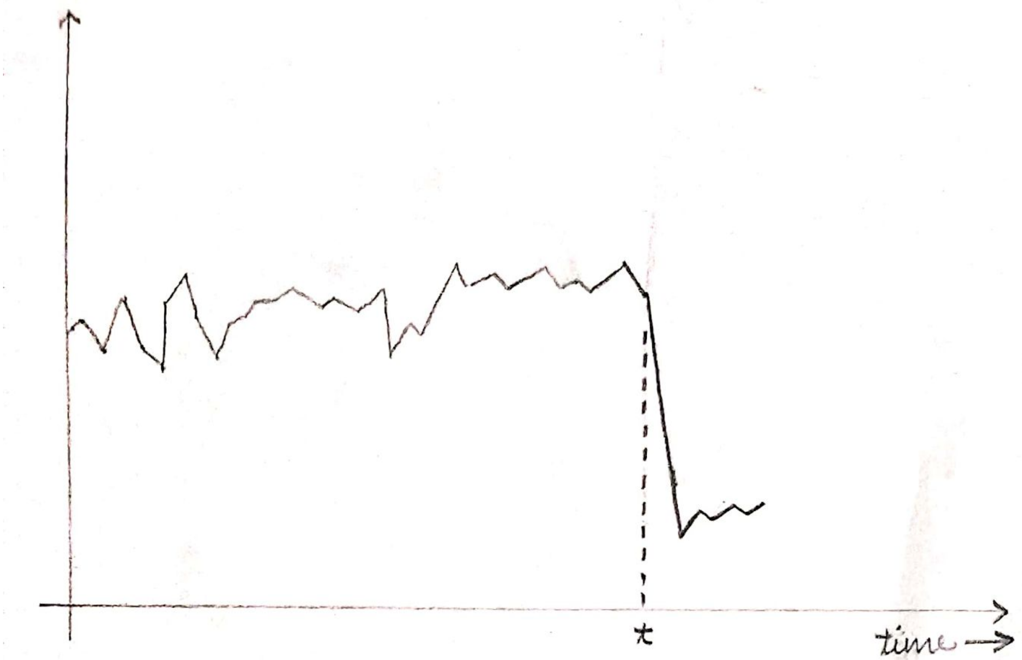
- **dual functionality**, i.e., environment monitoring and early detection of wildfire as deployment of networks can be costly for the only purpose of fire detection. Even at high risk areas, the probability of fire may be very low, and a WSN can be underexploited and become a waste of resources. That's why environmental monitoring contributes to optimize the usage of the network.

- **integration with the operation centre**: the network proposed system collects and analyses data over the cloud and shares it with command centres and other wildland fire fighting units. This enables a full picture of the emergency in real time.

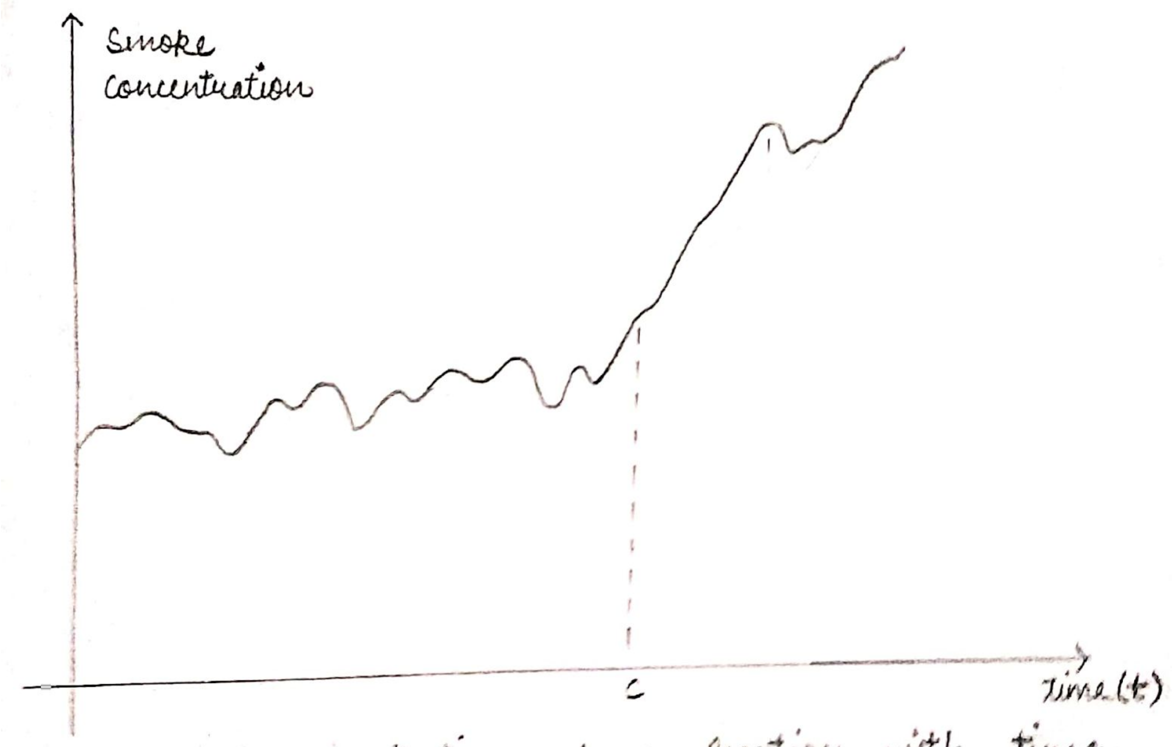
The collected data by the sensor nodes at different locations will look as follows when we will analyse it. The peak in Temperature and Smoke Concentration gives suspicion of wildfire at a location where the sensor node is located.



Variation of Temperature at a location with time



Variation of Humidity at a location with time



Smoke Concentration at a location with time

When the temperature, humidity and smoke concentration rises above the threshold(c) and the humidity at that location reduces, that region will be considered suspicious and an alert will be sent to the responsible authorities and forest fire department to take immediate action.

Using this system for wildfire monitoring has several advantages over traditional systems requiring wired sensors:

- Individual nodes are relatively inexpensive, allowing more data to be collected per unit cost;

- Standardized, freely available and modular software components reduce cost of developing, modifying and maintaining the system;

- The nodes are easy to deploy and use, works over cloud services for easy deployment;

- The deployed system allows near real-time response to events such as rapid temperature rise reported by deployed sensors.

7. Summary & Future Work

This study firstly reveals that WSN technology is a very promising green technology for the future in detecting efficiently the wildfires in our country. The more data recovered by WSN about forest fires means the more effective fire management by forest authorities. Hence, introducing the remote sensor based detection of wildfires seams to be a good solution for the future in which data from environment parameters in the surrounding areas can be collected by sensors and based on their analyses important conclusions should be drawn. Such WSN based detection systems are cost effective and can be easily implemented. They can run all day 24/7 and the alert system also alerts the control system directly so the possibility of false alarms is also reduced.

For future work, we aim to do further testing to assess the behaviour of the system in real world conditions as well as to develop algorithms to pinpoint the fire front in real-time, by using multiple sensor data analysis. The potential of this sensor network to serve as a source of fire risk information, integrated with an alert system, and the system still needs to be investigated. Alternative wireless technologies should also be evaluated in order to provide redundant communication methods between Master and Slave nodes and the cloud server. We can also attach Water Sprinklers at each sensor as a productive measure to make the system more automated and advanced.

8. Source Code

```
#include <SoftwareSerial.h>
#include <dht.h>
#include <TinyGPS++.h>

#include "ThingSpeak.h"
```

```
#include <Ethernet.h>
#include "secrets.h"

byte mac[] = SECRET_MAC;      // static IP address to use if the DHCP fails to assign
IPAddress ip(192, 168, 0, 177);
IPAddress myDns(192, 168, 0, 1);

EthernetClient client;

unsigned long myChannelNumber = SECRET_CH_ID;
const char * myWriteAPIKey = SECRET_WRITE_APIKEY;

SoftwareSerial serial_connection(3, 4);    //RX=pin 10, TX=pin 11
TinyGPSPlus gps;                           //This is the GPS object that will pretty much
    do all the grunt work with the NMEA data


const int dht_apin =A0;    //Analog Pin sensor is connected to (dht)int smokeA0 = A5;
const int smokeA0=A5;


int sensorThres = 400;      // Our threshold value


dht DHT;

float lati = 28.5458,longi = 77.1703;       // create variable for latitude and longitude
    object
SoftwareSerial gpsSerial(3,4);              //rx, tx


void setup()
{
  serial_connection.begin(9600);            //This opens up communications to the GPS
  Serial.println("GPS Start");              //Just show to the monitor that the sketch has
    started
  pinMode(smokeA0, INPUT);

  Serial.begin(9600);
  Serial.println("The GPS Received Signal:");

   Ethernet.init(10);
  Serial.begin(115200);  //Initialize serial

  // starting the Ethernet connection:
  Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
```

```
    // Checking for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
      Serial.println("Ethernet shield was not found.  Sorry, can't run without hardware. :(");
      while (true) {
        delay(1); // do nothing, no point running without Ethernet hardware
      }
    }
    if (Ethernet.linkStatus() == LinkOFF) {
      Serial.println("Ethernet cable is not connected.");
    }
    // trying to configure using IP address instead of DHCP:
    Ethernet.begin(mac, ip, myDns);
  } else {
    Serial.print("  DHCP assigned IP ");
    Serial.println(Ethernet.localIP());
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);

  ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop()
{
  temp=get_temperature();

  hum=get_humidity();

  latitude=getlatitude();

  longitude=getlongitude();

  Serial.print("The Position is ",longitude, latitude);

  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  if(x == 200) {
    Serial.println("Channel update successful.");
  }

  else {
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }


  if (smoke()>400)
  {
    Serial.print("Smoke Alert")
```

```
  }
}


int smoke()
{
  int analogSensor = analogRead(smokeA0);

  return(analogSensor);
  //Threshold Value is 400
}


float getLatitude(){
  while(serial_connection.available())      //While there are characters to come from the
     GPS
   {
    gps.encode(serial_connection.read()); //This feeds the serial NMEA data into the
     library one char at a time
   }

  if(gps.location.isUpdated())                  //This will pretty much be fired all the time
     anyway but will at least reduce it to only after a package of NMEA data comes in

   {
    return(gps.location.lng(), 6);
   }
}


float getLongitude()
{
  while(serial_connection.available())      //While there are characters to come from the
     GPS
   {
    gps.encode(serial_connection.read()); //This feeds the serial NMEA data into the
     library one char at a time
   }

  if(gps.location.isUpdated())//This will pretty much be fired all the time anyway but will
     at least reduce it to only after a package of NMEA data comes in
   {
    return(gps.location.lat(), 6);
   }
}

float get_temperature()
{
```

```
  DHT.read11(dht_apin);
  return DHT.temperature;
  }

float get_humidity_()
{
  float k=0;s
  DHT.read11(dht_apin);
  k= DHT.humidity;
  return(k);
  }
```

9. References

1. B. Leblon, J. San-Miguel-Ayanz, L. Bourgeau-Chavez and M. Kong, "3 - Remote Sensing of Wildfires" in Land Surface Remote Sensing, Elsevier, pp. 55-95, 2016.

2. K. Bouabdellah, H. Noureddine and S. Larbi, "Using Wireless Sensor Networks for Reliable Forest Fires Detection", Procedia Computer Science, vol. 19, pp. 794-801, 2013.

3. Antunes, Miguel, et al. “Low Cost System for Early Detection and Deployment of Countermeasures Against Wild fires.” 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). IEEE, 2019.

4. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," Computer Networks (Elsevier) 38 pp. 393-422, 2002.