Forest Fire Detection System Reliability Test Using Wireless Sensor Network and OpenMTC Communication Platform

Anton Herutomo, Maman Abdurohman, Novian Anggis Suwastika, Sidik Prabowo, Catur Wirawan Wijiutomo
School of Computing,
Telkom University
Bandung, Indonesia
Email: {antonh, abdurohman, anggis, pakwowo, caturwijiutomo}@telkomuniversity.ac.id

***Abstract* — Machine to Machine (M2M) communication system has started gaining its real world momentum by the introduction of Internet and mobile technology into this system. Several works have come up to use this integrated system into many different vertical solutions and tried to bind one solution platform for many monolithic systems. One recent innovation is to use HTTP REST architecture as communication platform. This work tackles the first step of implementing OpenMTC as M2M and IoT communication platform. Involved sensors are carbon monoxide gas concentration, temperature, and humidity, combined with Zigbee and Arduino microcontroller to make up 2 Device Application components (or DAs, as proposed by ETSI M2M standard) connected to OpenMTC GSCL. Reliability tests were performed to measure sensor accuracy and system response in sensor-gateway communication. From the experiments in Kamojang Forest (West Java, Indonesia), we conclude that the system is sufficient to operate, considering 341 meters optimal range for sensor and gateway communication for the related area and 1250 ms average data transmission between sensor and gateway.**

Keywords — Forest Fire Detection, Wireless Sensor Network, ZigBee, OpenMTC, Smart City, M2M.

I. INTRODUCTION

The machine to machine (M2M) communication system has started to get its down-to-earth implementation in 2014, according to [1]. M2M technology has been gradually releasing its illusions and meeting realistic implementation with the adoption of Internet and mobile technology. Physical world or even dubbed as 'things' will become even closer to the digital world in this era.

One of the bridges which catalyzes blurrer boundary between physical and digital infrastructures is the M2M communication platform. It refers to the intermediary software and hardware by which we can connect a vast number of devices (eg. sensors, actuators) and expose them with a set of services [2]. The existence of M2M communication platform opens wide innovation and business models and as a consequence, it raises a need to define a standard for IT governance purpose.

The European Telecommunication Standard Institute (ETSI) among well-known standard bodies has started its way in M2M governance by developing a standard specializing in M2M and has matured between 2012 and 2013. One of its proof-of-concept, the OpenMTC, has been developed by Fraunhofer Fokus [3] to showcase how M2M will provide solutions in the world of open, many-verticals, and monolithic implementations [4, 5].

OpenMTC strength lies in the simple yet powerful groups of functions for different vertical solutions: the Network Service Capability Layer (NSCL) and Gateway Service Capability Layer (GSCL). Both groups of functions or services relies on common protocol widely used today in Internet and mobile world: HyperText Transfer Protocol (HTTP) Representational State Transfer (REST). It enables asynchronous communication between components and ensure the data and information are available at any convenience point of time.

In this paper, the objective is to present an approach to test the reliability of fire detection system use case in West Java Province in Indonesia using OpenMTC platform. The platform was chosen as part of the prototype because of its potential to become horizontal approach to store and retrieve data as well as information related to forest fire prevention data. The fire detection system was chosen as a use case because of its potential benefits to effectively reduce the damage caused by forest fire, as suggested by [6].

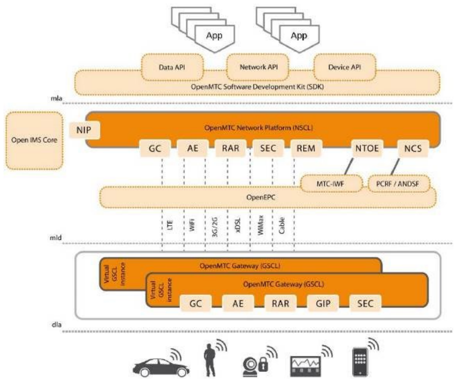

Fig. 1. OpenMTC RESTful Architecture [10]

II. RELATED WORKS IN FOREST FIRE DETECTION SYSTEM

Traditional approach to prevent forest fires usually relies on satellite imagery and/or visual observation by forest guards. Further improvement on forest fire danger management arises since then, involving measurements of stable elements (such as vegetation, topography, altitude, climate, sun radiation, team organization) as well as non-stable elements (such as temperature, humidity, wind, rain level) [7].

Bouabdellah et al [6] suggested that there are at least two best-known fire detection which could be applied in fire detection system:

1. Canadian approach
   The calculation uses data collected from temperature, humidity, wind, and rain sensors to calculate fire behavior (initial spreads and build ups) and make up a Fire Weather Index (FWI) [2].
2. South Korean approach
   The calculation uses data collected from humidity, solar radiation, and rain sensors to directly calculate index in Forest Fire Surveillance System (FFSS) [2].
3. Other approach
   Some calculation methods also exist such as National Fire Danger Rating System (NFDRS) and Distributed Fuzzy Logic Engine Rule-based WSNs (D-FLER).

The above research suggested that CPU time used for calculation affects the energy consumption of the Wireless Sensor Networks (WSNs).

Other research presented in [8] focuses on the energy consumption behaviour of the WSNs by using 3 stages for minimizing power consumption: sleep mode, sensor-microcontroller-transceiver on mode, and transceiver-only on mode.

III. SYSTEM DESIGN

A. System Overview

In this research, the overall architecture of the system is defined to fulfill the following objectives:

1. Develop a prototype for forest fire detection system based on WSN which can communicate to OpenMTC M2M platform using the Internet.
2. Test the prototype's reliability by measuring sensor accuration and system response time in early detection of forest fire.

Based on the related works explained above, we chose to use these 3 variables for our forest fire detection: carbon monoxide gas concentration, temperature, and humidity. Variables were chosen after the discussion with Natural Resources Conservation Agency in West Java Province, Indonesia, to suit the practical condition in related area.

Data from variables was acquired from sensors and Arduino Uno microcontroller, which then sent to gateway computer for later transmission to OpenMTC GSCL component.

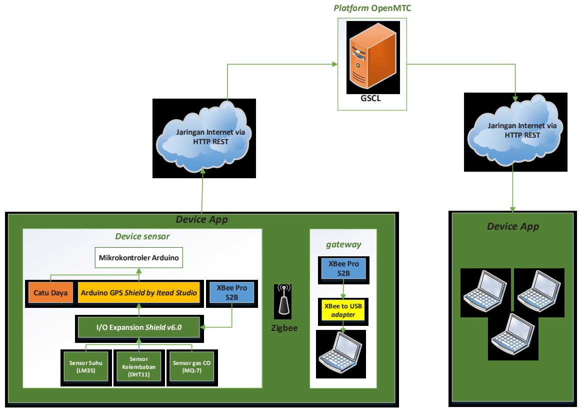

Fig. 2. System Architecture

B. Software Part

From the GSCL point of view, this system has 2 Device Applications (DAs):

1. The one for data acquisition and transmission to the GSCL;
2. The other for data visualization and reporting.

The device app responsible for data acquisition will capture the data via Zigbee and send the data in JSON format via public Internet to OpenMTC GSCL component using HTTP with RESTful Application Programming Interfaces (APIs).

The data transmission from gateway to GSCL is illustrated in Fig. 4. Data will be stored on GSCL's MongoDB database and it is accessible as part of its exposed service. The device app responsible for reporting function, which is based on Java, will periodically retrieve the data from GSCL database. As depicted above, the reporting application showed the temperature data in degree Celcius, humidity in %Rh, and CO gas concentration in PPM. It also shows current location of sensors and gateway computers in map imagery and longitude-latitude form

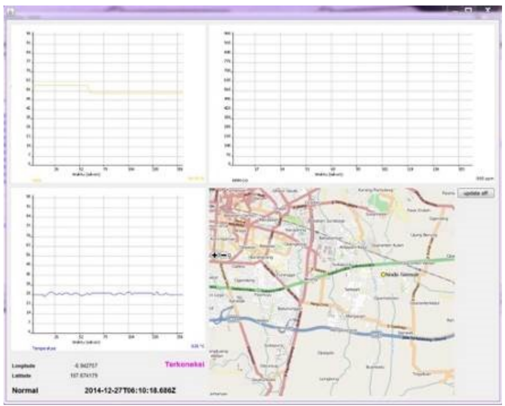

Fig. 3. Reporting DA for Forest Fire Detection System

```
/* 1 declare sensors: GPS: LM35, MQ7, DHT11 */
/* 2 print sensor data */
/* 3 get CO sensor data with curve fitting method */
float CO = 1.666 * analogRead(coPin) + (-583.5);
        if (CO < 20.0) CO = 0.0;

/* 4 get temperature sensor data with 10 sample averaging */
        float temp=0.0;
        int i = 0;
        while(i<10){
        temp = temp + ( (5.0 * analogRead(tempPin) * 100.0 ) / 1024.0 );
        i++;
        smartDelay(1);
        }
        temp = temp / (float)i;


/* 5 get humidity sensor data */
        DHT11.read();
        int rh = DHT11.humidity;

/* 6 calculate index if needed */
/* 7 print all sensor data */


/* Program entry point, print sensor and halt for 1 second */

void loop(){
        printSensor();
        smartDelay(1000);
}

public class ForestFireGateway {
        public static void main(String[] args) {
                /// 1 prepare serial port
                        /// 2 input stream serial port
                        /// 3 get default comm port
                /// 4 detect and wait comm port until available
                /// 5 check whether port is "serial port/comport"
                /// 6 test and connect to port
        }
}
/// code below is to acquire data and transmit to openMTC
SimpleOMTCPusher omtc = new SimpleOMTCPusher(
            "http://serverAddress/appFolder");
            /// 7 parse data
            raw = raw.replace("\r", "").replace("\n","");
            System.out.println(raw);
            String[] splittedRaw = raw.split(" ");
            if(splittedRaw[0].charAt(1)=='*'){
                    splittedRaw[0]="NULL";
            }
            if(splittedRaw[1].charAt(1)=='*'){
                    splittedRaw[1]="NULL";
            }

            /// 8 transmit data to openMTC
            /// format data string
            /// data to be sent: long lat co temp rh stat
            JSONObject push = new JSONObject();
            push.put("long",splittedRaw[0]);
            push.put("lat",splittedRaw[1]);
            push.put("co",splittedRaw[2]);
            push.put("temp",splittedRaw[3]);
            push.put("rh",splittedRaw[4]);
            push.put("stat",splittedRaw[5]);
            System.out.println(push.toJSONString());
            System.out.println(omtc.pushData(push.toJSONString()));
            /// 9 finish and restart
} catch(Exception e){}
```

Fig. 4. Acquire Data from Sensors, Convert and Push Data from Gateway to GSCL

C. Hardware Part

Main hardware components of this system consist of MQ7 Carbon Monoxide gas sensor has features on high sensitivity to carbon monoxide. The application of this sensor is used in gas detecting equipment for carbon monoxide in family and industry or car. It can detect carbon monoxide on 20 ppm – 2000 ppm [9]. LM35 temperature sensor is a precision centigrade temperature sensor. It has range from -55 c to 150 c and using analog interface [10], and DHT11 humidity sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. It has 20% - 90% RH [11].

Fig. 5. DHT11 humidity sensor, LM35 temperature sensor, and MQ7 Carbon Monoxide gas sensor

Sensors were then connected to Arduino Uno board equipped with XBee adapter and Arduino GPS shield, to enable location tracking of the sensor in the forest field.

Fig. 6. Arduino Uno and GPS Shield

D. Sensor Accuration

First reliability test were conducted to check the accuracy of MQ7, LM35, DHT11 sensors in measuring its environment:

1. MQ7 sensor was tested in a closed glass box filled with CO gas from a motorcycle for 3 seconds. Comparison data were captured from CO alarm detector. We got normal variance of 13-16 PPM from standard measurement tool.
2. LM35 sensor was tested in low, medium, and high temperature and compared to infrared thermometer results. We got the variance of 1 degree Celcius from standard measurement tool.
3. DHT11 sensor was tested by comparing its results with digital hygrometer HTC-2 for one hour measurement. We got normal variance of 0.61-0.67% from standard measurement tool.

Fig. 7. Forest Condition in Kamojang Forest

Fig. 8. Sensor Placement in Kamojang Forest, West Java, Indonesia

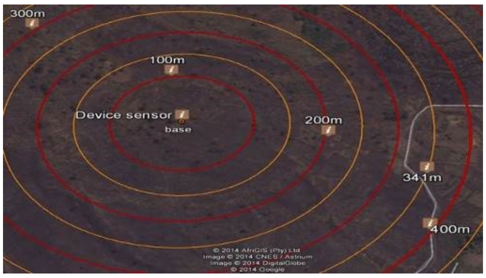

Fig. 9. Sensor Placement in Kamojang Forest, West Java, Indonesia

E. System Response Time

TABLE 1. SYSTEM RESPONSE TIME FOR 100-400 M RANGE

Test	System Response Time (ms) range between sensor and gateway				
	100m	200m	300m	341m	400m
1	1188	1180	1188	1180	Loss
2	1183	1179	1179	1446	Loss
3	1446	1457	1183	1184	Loss
4	1184	1184	1451	1187	Loss
5	1179	1184	1327	1183	Loss
6	1184	1184	1040	1450	Loss
7	1449	1456	1192	1184	Loss
8	1184	1186	1442	1180	Loss
9	1184	1240	1183	1184	Loss
10	1180	1119	1180	1450	Loss
11	1454	1456	1180	1182	Loss
12	1187	1173	1457	1184	Loss
13	1180	1184	1183	1180	Loss
14	1183	1183	1201	1454	Loss
15	1450	1467	2871	1184	Loss
16	1188	1180	45	1180	Loss
17	1184	1175	885	1187	Loss
18	1180	1293	1183	1447	Loss
19	1449	1337	1184	1182	Loss
20	1184	1220	1446	1180	Loss
Average	1250	1251.9	1250	1249.4	0

Second reliability test of forest fire detection system was conducted by measuring time difference between adjacent continuous data transmission from sensor to gateway.

Tests were conducted by separating sensor between 10, 20, 30, 40, 50, 60, 70, 80, 100, 200, 300, 400 meters from gateway. Obstacles were in the form of bush, trees, and geographical contour with 1.700 meters elevation above sea level.

We got the results between 1249.4-1250 ms response time and maximum range of 341 meters in the surface condition described above, which is acceptable for forest fire early warning system.

IV. CONCLUSION AND FUTURE WORKS

From this experience on developing forest fire detection system, it can be concluded that reliability of fire detection system using Zigbee WSN depends on the sensor being used to monitor the environment and their placement in widepread forest area. Considering its 341 meter effective range and approximately 1250 meter transmission time, placement should be decided carefully in coordination with forest guard or operator.

It also noted that Internet is not always accessible in the forest area where gateway is to be placed, therefore placement of gateway as node coordinator should also be carefully decided in order to optimize Internet connection to OpenMTC platform. Sensor and gateway physical security should also be taken into account as forest fire could damage the overall system operation.

Suggestions for further researches are: to conduct further reliability test by measuring overall transmission time from sensor to GSCL (over the Internet). Where the number of sensors or gateways is still not sufficient for load test, further research is also possible to develop a testing tool to simulate connections from sensors to gateways and platforms.

ACKNOWLEDGEMENT

This paper is a part of an effort fully funded by Hibah Riset Internasional 2014-2015 scheme at Telkom University.

Special thanks for Alief Pascal Taruna for supporting this research with data and testing tools.

Authors would like to give appreciation to Telkom University, Fraunhofer Fokus, and PT Telekomunikasi Indonesia Tbk for their continuous support and cooperation in Internet of Things research area.

REFERENCES

[1] Gartner, Inc. (2014, August 11). *Gartner's 2014 Hype Cycle for Emerging Technologies Maps the Journey to Digital Business* [Online]. Available: http://www.gartner.com/newsroom/id/2819918

[2] A. Elmangoush, H. Coskun, S. Wahle, T. Magedanz, "Design Aspects for a Reference M2M Communication Platform for Smart Cities," in *9th International Conference on Innovations in Information Technology*, Al Ain, 2013

[3] S. Wahle, T. Magedanz, F. Schulze, "Demonstration of OpenMTC – M2M Solutions for Smart Cities and the Internet of Things," in *37th IEEE Local Computer Networks Conference*, Florida, 2012

[4] V. Suryani, A. Rizal, A. Herutomo, M. Abdurohman, T. Magedanz, A. Elmangoush, "Electrocardiogram Monitoring on OpenMTC Platform," in *IEEE 1st International Workshop on Machine to Machine Communications Interfaces and Platforms*, Sydney, 2013, pp.843-847.

[5] M. Abdurohman, A. Herutomo, V. Suryani, A. Elmangoush, T. Magedanz, "Mobile Tracking System Using OpenMTC Platform Based on Event Driven Method," in *IEEE 1st International Workshop on Machine to Machine Communications Interfaces and Platforms*, Sydney, 2013, pp.856-860.

[6] K. Bouabdellah, H. Noureddine, S. Larbi, "Using Wireless Sensor Networks for Reliable Forest Fires Detection," in *Procedia Computer Science 19*, 2013, pp. 794- 801.

[7] A. Thoha, *Forest Fire Climate and Its Relation with Forest Fire Prevention in Indonesia* (in Indonesian) [Online]. Available: http://repository.usu.ac.id/bitstream/123456789/1030/1/hutan-achmad.pdf

[8] 8.A. Bayo, D. Antolín, N. Medrano, B. Calvo, S. Celma, "Early Detection and Monitoring of Forest Fire with a Wireless Sensor Network System," in *Procedia Engineering 5*, 2010, pp. 248-251.

[9] Hanwei Electronis . *MQ7 Datasheet,* [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf

[10] Texas Instrument . *LM35 Datasheet,* [Online]. Available: http://www.ti.com/lit/ds/symlink/lm35.pdf

[11] DFRobot. *DHT11 datasheet,* [Online]. Available: http://www.dfrobot.com/image/data/DFR0067/DFR0067_DS_10_en.pdf

[12] J. Mwangama, A. Elmangoush, J. Orimolade, N. Ventura, R. Steinke, A. Willner, A. Corici, T. Magedanz. (2014, October). *Prototyping Machine-to-Machine Applications for Emerging Smart Cities in Developing Countries* [Online]. Available: http://daad-unifi.org/wp-content/uploads/2014/10/1_Final_Paper.pdf

[13] OneM2M Technical Specification (2015, January 30). *Reference: TS 0001, Version: 1.6.1, Functional Architecture* [Online]. Available: http://www.onem2m.org/images/files/deliverables/TS-0001-Functional_Architecture-V1_6_1.pdf

[14] Fraunhofer FOKUS (2013, March 2). *OpenMTC White Paper* [Online]. Available: http://www.open-mtc.org/Downloads/2013_03_OpenMTC_Whitepaper.pdf

[15] ETSI (2015). *Machine-2-Machine Communications* [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/m2m