

DEVELOPING A SERVER ROOM ENVIRONMENT MONITORING SYSTEM USING INTERNET PROTOCOL ENABLED SENSORS

Ahmed Kijazi, Assistant Lecturer, College of Business Education, Bibi Titi Mohammed Road, P. O. Box 1968,
Dar es Salaam, Tanzania, Tel: +255-22-2150177, Mobile: +255 612 306240,
E-mail: ahmedkijazi@gmail.com; a.kijazi@cbe.ac.tz

ABSTRACT

Environmental condition monitoring is very important for efficient operation of electronic components and devices. Server rooms and Data centers typically house thousands of dollars' worth of electrical and computing equipment, often storing proprietary data, customer accounts, or other sources of valuable information that represents the lifeblood of the business. The aim of this paper is to explain how to develop an environment monitoring system using Internet protocol (IP) enabled sensors, for detecting temperature, humidity, water and fire in the Server rooms and Data Centers in order to protect important computing and other electronic devices which are sensitive to environmental condition changes. This is a system which is sensitive to environmental condition changes and takes necessary actions to protect monitored devices, thus ensuring safety and durability of the devices and contributes to the network availability. In this paper IP enabled sensor, Java programming language for client and server programming, SQL Server for storing collected sensor values readings, SMS gateway software for sending alert messages to network administrator and a sample configured network have been used. Additionally, server room environmental monitoring system is a working system, which was tested in a lab environment, and it seems to work fine.

Key words: SMS Gateway, IP enabled Sensors, Local area network (LAN), environmental condition;

1.0 INTRODUCTION

1.1 Background

The demands for application of sensors in monitoring environmental conditions for different purpose are increasing daily. It is possible to use sensors in agricultural fields for monitoring crops humidity, soil moisture in the land and light conditions in plants when it is concerned with some crops. Not only that, sensors can be used for monitoring environmental conditions that affect electronic equipment and systems. Such conditions include moisture, dust, vibration, humidity, temperature, water and fire in order to reduce failures and dropping maintenance costs (Nishinaka 2002). Furthermore, the readings captured from sensors can be collected to a central system for analysis purpose or sent to the users via SMS when necessary (Samrit et al. 2013). The connection between the sensors and systems can either be wired or wireless depending on the nature of the project (Meeraval & Anusha 2012). In this paper the discussion is made based on the wired sensor connection that has been used to build the system.

1.2 Statement of the Problem

Currently, there are different kinds of environmental monitoring systems. Generally, it is claimed that, despite the existence of these systems, they have not successfully performed some tasks such as: (i). To detect instantly sensors' malfunction and report to network administrator immediately via email or SMS, this ensures reliability of environmental sensors and networking devices and (ii). To have the capability of shutting down servers upon critical environment condition, e.g., high temperature, humidity, water and fire in the server room in order to protect servers. The environment monitoring system using Internet protocol (IP) enabled sensors was developed purposely to overcome the mentioned deficiencies of the existing systems. This is a server room and data center monitoring system with two extra functionalities which have been mentioned above. IP enabled sensors are sensors which can be installed within Ethernet network and act like a network host. These can therefore enable one to monitor sensor availability as well as reading current sensor values over the network. These kinds of sensors can communicate with any device within the network using their configured IP addresses. The process of detecting instantly sensors malfunction and alerting the network administrator ensures continued operation of environmental sensors which results in efficient environmental condition monitoring in the server rooms and data centers. Similarly, the process of shutting down servers upon critical environmental condition protects servers and crucial information which have been stored in servers.

2.0 LITERATURE REVIEW

Server room environment monitoring system using internet protocol enabled sensors is not a new phenomenon, since many efforts have been made to develop similar systems in the past. A few examples of these systems present include Data Center Infrastructure Management system (Raritan 2016). This is a system which detects the current

environmental condition in the data center and alerts the network administrator upon any critical environmental condition. Similarly, there is another one called ENVIROMUX-16D, this is a Server Environment Monitoring system which monitors critical environmental conditions, such as temperature, humidity, liquid water presence, power, intrusion, and smoke. From this system when a sensor goes out of range of a configurable threshold, the system will notify an operator via email, LEDs, alarm beacon and SMS (Network technologies incorporated 2016).

2.1 How the System Works?

The system consists of two parts, the hardware and software part. The hardware part is composed of sensors while the software part is developed from Java programming language with two instances, a client and server instances. The server instance is installed on the environment monitoring server while the client instance is installed on the host devices which are monitored e.g. Other Servers (Figure 1). All client instances are connected and communicate with a server instance by using TCP client and server socket programming (Harold 2004). The server instance does the following tasks: - (i) Reads the sensors captured environmental condition, status values from the XML file located inside the sensor control unit with the help of XML parser libraries (Figure 4, 6, 7, 8 & 9). The captured sensor values were stored in SQL server Database for future reference. (ii) It detects an environment sensor malfunction as soon as it happens and sends alert message to the network administrator (iii) it sends alert messages to the network administrator and commands the client instances to shut down their corresponding servers upon critical environmental condition in the server room in order to protect servers. This is done using socket programming in Java, (Figure 1 & 4)

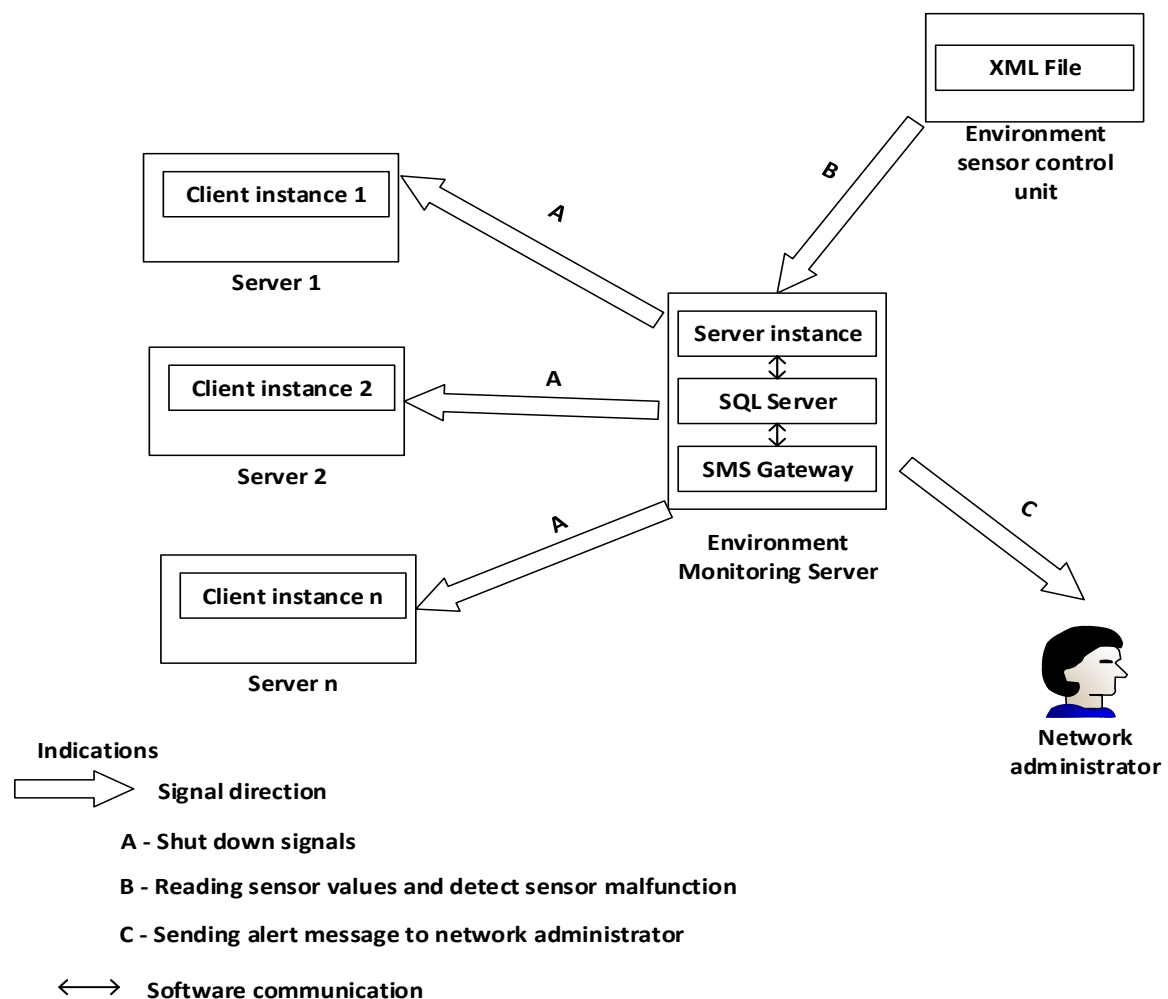


Figure 1. Block diagram of instances, operations and communication

2.2 Monitoring of sensor availability

Network sensors are connected in the network like other host, the server instance of the system contains a module which monitors sensors availability using periodical ping. This improves network monitoring activities since the availability of the network depends also on the safely operated environmental condition, (Figure 2 & 9)

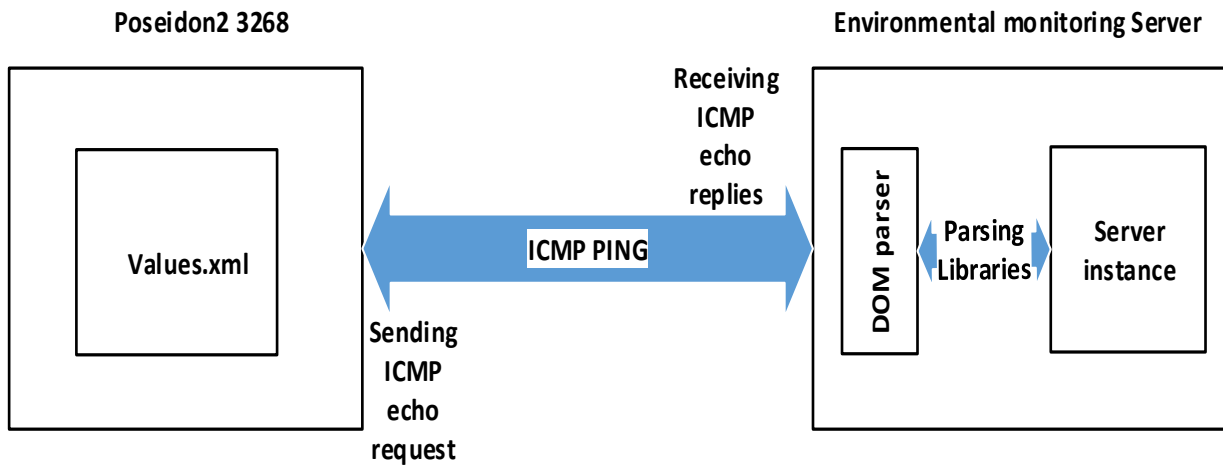


Figure 2. Environmental monitoring server detecting environmental sensor availability

2.3 Sending environmental status notifications.

The system notifies Network administrator about environmental status via SMS,(Figure 3 & 8) It is recommended that the system developer should use the SMS gateway software to minimize the complexity of developing a separate module for sending a notification message to the Network Administrator such as Diafaan SMS Server (Kijazi & Kisangiri 2015).

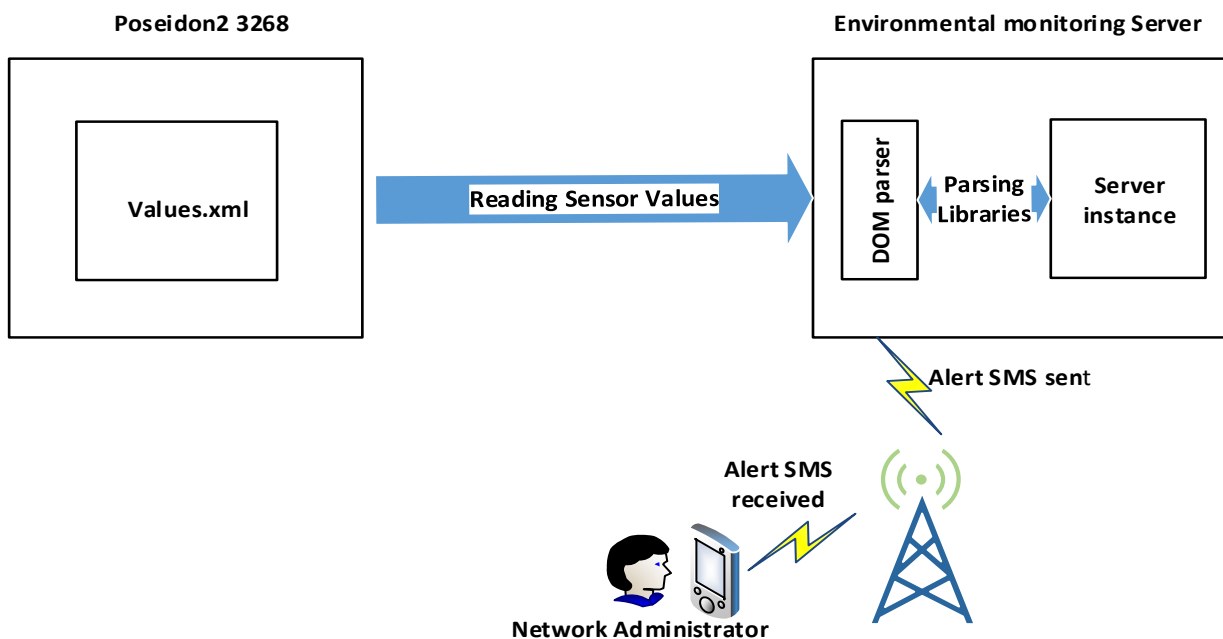


Figure 3. The environmental monitoring server sends alert SMS

2.4 Protecting Servers

The system takes action when the environmental values exceed the threshold limit.

Since all devices in the network can communicate with IP enabled sensor environmental monitoring system, the system have the capability to shut down all devices in the network when there is critical environmental condition in the server room, such as high temperature caused by air conditioning failure. This avoids unnecessary loss which might happen in the server room such as data loss as well as damage of devices due to the rise of temperature in the room etc. (Figure 4)

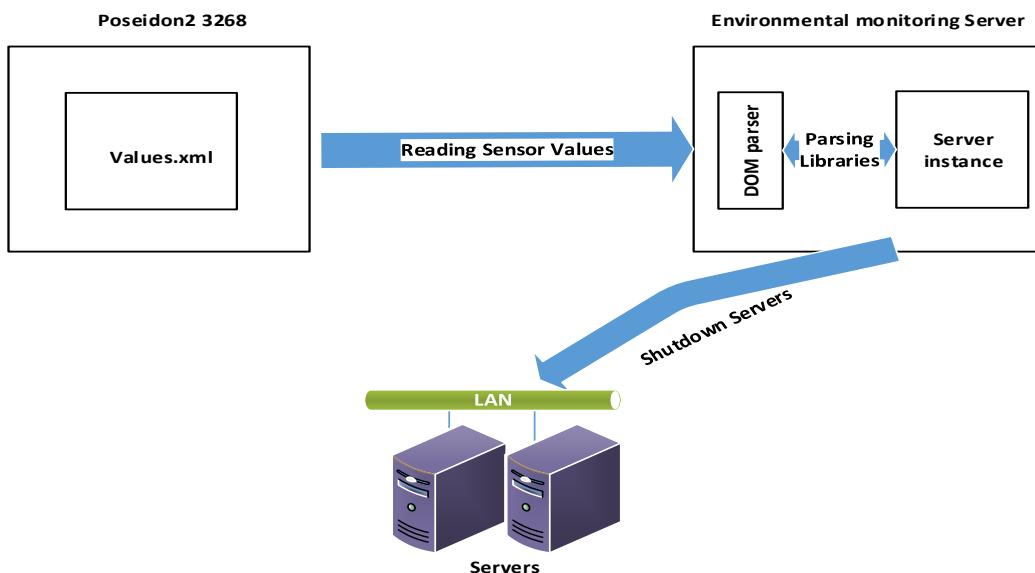


Figure 4. Environmental monitoring server shutdown servers

3.0 METHODOLOGY

Before the development of the system a preliminary study was conducted to observe the functionalities and working principles of different server rooms and data center environment monitoring systems. A few examples of these are ENVIROMUX-16D system which was developed by Network technologies incorporated company, Data Center Infrastructure Management system by (Raritan 2016) and Falcon F200 Monitoring_Solution which was developed and owned by the SRO Company (SRO 2017). Neither of these systems have the capability to protect the electronic devices upon critical environmental condition in the server rooms and data centers.

The environment monitoring system using Internet protocol (IP) enabled sensors consists of software module as among the components. Consequently, the waterfall methodology of software development has been used for developing this module. The waterfall model involves the following steps, requirement gathering, system design, implementation, integration and testing, deployment of the system and maintenance (Waterfall model for software development, 2015).

3.1 GATHERING OF SYSTEM REQUIREMENTS

During the development of the system the following software and hardware components have been used:- Sensor control unit (Poseidon 3268), Fire sensor, Temperature sensor, Humidity sensor, Water sensor, SMS gateway software, Modem, Data cable, Netbean IDE (Java SDK), SQL Server, Database, Ethernet Switch and three PCs installed with windows OS.

3.1.1 Fire sensor

The fire sensor continuously detects fire in the server room and informs the sensor control unit by changing the fire status of the XML file in a sensor control unit to be binary data 1 when there is a fire, or maintaining the default binary data 0 when there is no fire.

3.1.2 Temperature sensor

Temperature sensor continuously detects the critical temperature in the server room and stores the current temperature readings to the XML file in a sensor control unit with the aid of XML parser libraries which are downloaded and imported in the Java netbean, during software development. The same procedure is used to capture the readings of Humidity and water sensors, (Figure 7).

3.1.3 SMS Gateway software.

The purpose of SMS gateway software is to send alert SMS upon critical environmental condition or sensor malfunctions in the server room. This is done as soon as after receiving commands from the server instance.

3.1.4 Three PC installed with windows OS

One computer acts as an environmental monitoring server on which the Database, SMS Gateway, Modem and server instance are installed. Similarly, on the remaining computers client instances are installed. The server instance communicates with the sensor control unit and client instances over the network.

3.1.5 Ethernet Switch

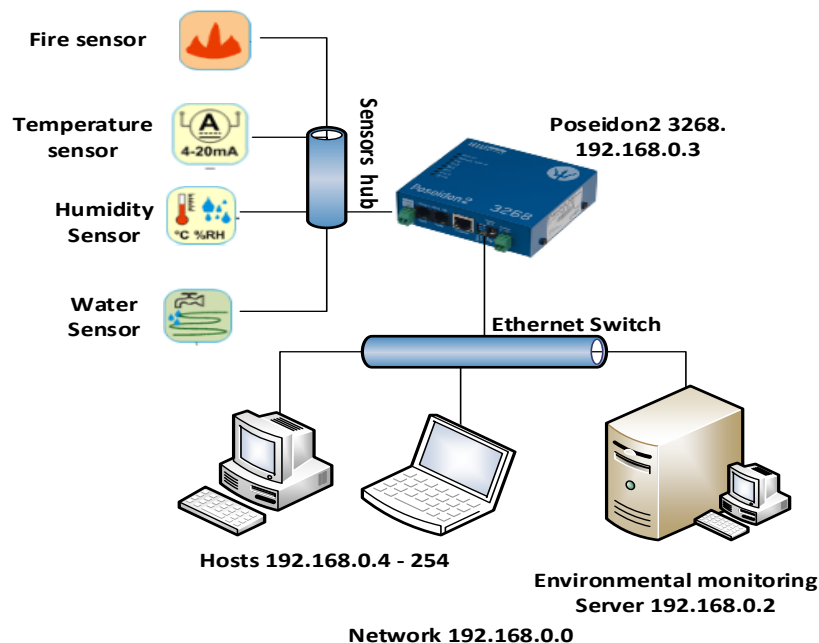
An ethernet switch connects all devices within the network. The Combination of an ethernet switch, sensor control unit, and PCs form up the configured network.

3.2 SYSTEM DESIGN

3.2.1 System architecture

In order to develop an IP enabled sensors environmental monitoring system, there was a need of configured network connection for communication between the software and hardware part of the system. The network is composed of IP enabled sensor control unit, networking devices, hosts, environment monitoring server, SMS gateway and SQL server Database. SMS gateway and SQL server are installed on the environment monitoring server also, (Figure 5).

Figure 5: Sample Configured Network



3.3 SYSTEM IMPLEMENTATION

3.3.1 Sensor control unit configurations (Poseidon2 3268)

In order to configure a sensor control unit, the device must be connected to the network first. Any computer in the network can be used to configure the device since it has already been assigned the default IP address. The person configuring the sensor control unit may access the device configuration panel by typing its IP address on the address bar of the web browser and press enter button (HW- Group, 2017).

3.3.2 Network and host devices configuration

The normal procedure of configuration of IP version 4 in the networking devices and host should be followed.

Note that: As usual, in order for poseidon2 3268 to communicate with the environmental monitoring server and any other device in the network both must be in the same subnet, not otherwise.

3.3.3 Reading of sensor values

IP enabled sensor environmental control system, is implemented using Java codes. Poseidon2 3268 has built in values.xml file which periodically store values captured from the sensors. The task of the Java programmer is to write the Java codes which can read the values.xml file in poseidon2 3268 using the device IP address. In order for Java code to read .xml files, xml parser concept should be used (Tutorial point 2016). The xml parser must be downloaded as a separate Java library and imported into the netbean IDE.

There are several Java, XML parsers such as DOM, SAX and StAX Parser in Java. In this paper DOM parser has been used. DOM parser acts as a translator to Java codes in order to understand the content of values.xml file, (Figure 4, 6, 7, 8 & 9)

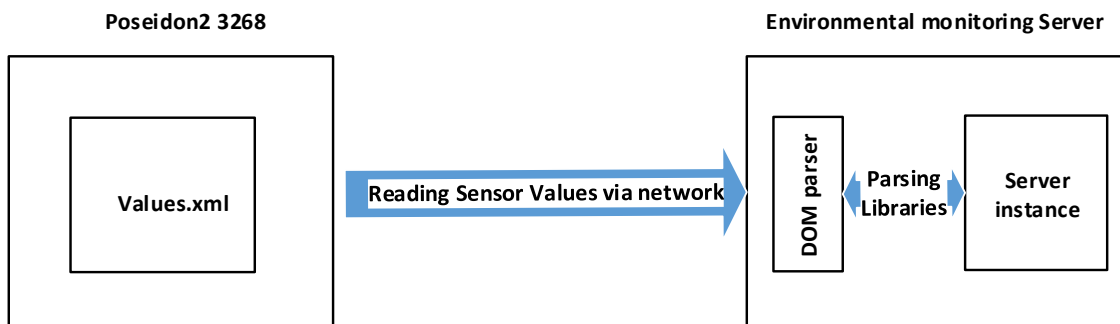


Figure 6. The server instance communicates with the sensor control unit

Poseidon2 model 3268

Dry Contact Inputs			
Name	ID	Current Value	Alarm Alert
Binary 1	1	0 (Off)	Disabled
Binary 2	2	0 (Off)	Active if on
Binary 3	3	0 (Off)	Active if off
Binary 4	4	1 (On)	Disabled
Comm Monitor 1	123	0 (Off)	Active if on

Relay Outputs			
Name	ID	Current Value	Mode
BinOut 1	151	0 (Off)	Manual
BinOut 2	152	0 (Off)	Manual

Sensors					
Name	ID	Current Value	Safe Range	Hysteresis	Alarm Alert
Rack1 Temp	42538	31.8 °C	18.0 .. 24.0	0.0	SMS and SNMP trap

Figure 7. Values.xml file in poseidon2 model 3268 graphical user interface
Source: HW-Group (2017)

```

1 <Name>Rack1 Temp</Name>
2 <SenId>28230a700120084c</SenId>
3 <PortId>42538</PortId>
4 <Units>C</Units>
5 <Value>31.8</Value>
6 <Calib>0.0</Calib>
7 <Min>18.0</Min>
8 <Max>24.0</Max>
  
```

Figure 8. Values.xml file in poseidon2 model 3268 XML codes
Source: HW-Group (2017)

```

1
2      String sensoripaddress=192.168.1.20;
3      sip= "http://"+sensoripaddress+"/values.xml";
4      DocumentBuilderFactory factory=DocumentBuilderFactory.newInstance();
5      DocumentBuilder builder=factory.newDocumentBuilder();
6      Document root=builder.parse(sip);
7      NodeList value=root.getElementsByTagName("Value");
8      root.normalize();
9      Element tempread=(Element)value.item(1);
10     temp =Float.parseFloat(tempread.getFirstChild().getNodeValue());
11
12     Element humread=(Element)value.item(0);
13     hum=Float.parseFloat(humread.getFirstChild().getNodeValue());
14
15     Element waterread=(Element)value.item(2);
16     water=Float.parseFloat(waterread.getFirstChild().getNodeValue());
17
18     if(temp<tempmin-3 || temp>tempmax-3)
19     {
20         System.out.println("Critical temperature in the Server Room");
21         System.out.println("Send Message to system administrator");
22         System.out.println("Shut down other devices in the server room");
23     }
24     }
25     else
26     {
27         System.out.println(" temperature is Normal in the Server Room");
28         System.out.println("Send Message to system administrator");
29     }

```

Figure 9. A Server instance Java module reads values.xml file

```

1
2      String sensorip=192.168.1.20;
3      pingcmd = "ping " +sensorip;
4      r = Runtime.getRuntime();
5      p = r.exec(pingcmd);
6      BufferedReader in = new BufferedReader(new InputStreamReader(p.getInputStream()));
7      String inputLine;
8      int k=0;
9      while ((inputLine = in.readLine()) != null)
10     {
11         a[k]=inputLine;
12         k=k+1;
13     }
14     String ad=a[2].concat(a[3]);
15     boolean adl=ad.toLowerCase().contains("ttl");
16     if(adl==true)
17     {
18         System.out.println("*****Sensor device is Up*****");
19         System.out.println("Send Message to system administrator");
20     }
21     }
22     else
23     {
24         System.out.println("*****Sensor device is Up*****");
25         System.out.println("Send Message to system administrator");
26     }

```

Figure 9. A Server instance Java module detects sensor malfunctions

3.4 SYSTEM INTEGRATION AND TESTING

The system was installed in the LAB with the network setup as shown in the (Figure 5). The server instance was installed on the environmental monitoring server while the client instances were installed on the other two computers. The server instance was able to communicate with both client instances and environmental sensors control unit via the network. The testing process was divided into two parts as follows :-(i) Testing for sensor malfunction (ii) Testing for shutting down of the servers upon critical environmental condition in the server room.

3.4.1 Testing for devices protections

The constant environmental condition was maintained by switching on the air cooling devices in the server room while keeping windows and doors closed. The environmental condition threshold values were set on the server instance control panel as indicated in the (Figure 8). These values were assumed as the safety environmental condition values which the monitored devices in the server room can operate without fail as indicated by the device manufacturer's data sheet. The amount of temperature was increased by switching off the air cooling systems for some time. When the temperature rose beyond the threshold values, the system sends alert messages to the system administrator as well as shutting down all servers in the serve room.

3.4.2 Testing of sensor malfunction and availability

This part was tested by disconnecting sensor control unit in the network and observed the system reaction. Upon detection the absence of sensors, the server module alerted network administrator via SMS informing about sensors malfunction.

4.0 RESULTS

4.1 Environmental monitoring system control panel

The system contains the interface which allows Network Administrators to enter the device's IP address and Network Administrator's mobile numbers and to set the sensor threshold values. Also the interface allows a network administrator to view the values captured from the sensors, (Figure 7 & 11).

Device Name	IP Address	Port Number
DNS Server	192.168.0.4	53
Mail Server	192.168.0.5	110
Web Server	192.168.0.6	80
DHCP Server	192.168.0.7	9
Gateway	192.168.0.1	
Shared Device	192.168.0.4	
Environment Sensor	192.168.0.3	

	Min Value	Max Value	
Operational Temperature	17.0	30.0	oC
Operational Humidity	46.0	50.0	%RH

	Value	Status
Temperature Reading	17.3	Normal
Humidity Reading	47.3	Normal
Water Detection	No Water	Normal

Network Admin	Mobile Number
Network Admin Mobile Number1	0712307240
Network Admin Mobile Number2	0712307240

Configuration number 2007

Refresh Update

Figure 11. Environmental monitoring system control panel

4.2 DISCUSSION

During system development, any software development model can be used depending on its suitability to software developer. But in this paper waterfall model has been used. During software testing both white and black box testing were used for system testing of interfaces as well as coding side. But this does not limit software developers to use other types of testing when necessary.

This paper explains how to develop an environment monitoring system which can be used in server rooms and data centers. This system apart from reporting environment condition status, also detects the malfunction of environmental sensors and protects devices by shutting down them automatically upon critical environmental condition.

This paper explains only protecting servers. Unfortunately, protection of switches, sensors and routers were not considered, because they need a strong knowledge of simple network monitoring protocol (SNMP). The implementation of this part is somehow challenging because the Management information base (MIB) structure is not the same in all SNMP enabled devices. They differ depending on device manufacturers. This difference in MIB structure, forcing the system developers to develop the network monitoring software, that monitor only parameter which are common to all devices are very few. Consequently, if the system developer needs to develop a software which can monitor many parameters he was supposed to stick on only one dealer, which is impossible in business.

5.0 CONCLUSION

This paper tries to explain how we can use IP enabled sensors to build an environmental condition monitoring software which can be used to monitor environmental conditions, sensors themselves, sending alerts via SMS and takes action to shut down all servers in the network when there is critical environmental condition in the server room. Consequently, these can be considered as advantages of using IP enabled sensors compared to non IP enabled sensors. It is recommended to install environmental sensor in the server's rooms and data centers in order to monitor the operation environmental condition, since the electronic systems such servers and routers can be affected by environmental conditions such as water, fire and humidity. If there is no proper way of monitoring the server rooms working environment, the devices could be destroyed and cause data loss.

REFERENCES

- Ciprian, R. and Lehman, B. (n. d). Modeling effects of relative humidity, moisture, and extreme environmental conditions on power electronic performance. *Dept. Elect. &Comp. Eng. Northeastern university Boston and diversified technologies Inc. Bedford.*
- Harold E.R., (2004). Java network programming. *O' REILLY, United States of America.* HW- Group Company, 2017. Ethernet I/O control and sensor monitoring system. Retrieved from www.hwgroup.com/products/poseidon/poseidon2_3268_en.html
- Kijazi, A & Kisangiri, M. (2014). Development of Multifunctional Network Monitoring System Using SMS. *IEEE*, 143 – 146. Doi: 9.198/SCAT.2014.6044148.
- Meeravali S., Anusha A., (2012). Detection of gas leak and its location using wireless sensors. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 1 Issue 9, November- 2012
- Network technologies incorporated, (2016). ENVIROMUX-16D. Retrieved from www.networktechinc.com
- Nishinaka, (2002). The effects of the environmental conditions of some Zener voltage references. *Conference Digest 2002 Precision Electromagnetic Measurements*, pp. 166.
- Raritan, (2016). Data Center Infrastructure Management (DCIM). Retrieved from www.raritan.com
- Samrit, D., Parate, M. & Gawai, N., (2013). Development of a data acquisition and greenhouse control system. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2 Issue 4, April 2013.
- Tutorial point, 2016. JAVA XML parsers. Retrieved from www.tutorialspoint.com