IOT based Flood vigilance system

J.Fenila Naomi
Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India

Kavitha M
Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India

Dinesh E M
Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India

Brahadeesh. B
Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India

Akhil. M
Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology, Coimbatore, Tamilnadu, India

Abstract- **For several thousand years, human population has experienced the catastrophes brought by the natural disasters. Since the disasters occur without any prior warning to people, there is no enough time to evacuate the area of disaster. Even though the disasters are suddenly occurring, it is possible to predict the occurrence of some of them. Flood is one these disasters. It is one of the most commonly occurring natural disasters in the world and accounts for the highest loss that it brings to humanity. In this paper, we present a system which we designed to provide alert using short message system before flood. The system is also designed to store the history of water level in its database. The database is present on a centralized server which does all the jobs from data collection to sending message before flood.**

Keywords – natural disasters, humanity, flood, water level

I. INTRODUCTION

In the year 2013, the northern part of India was stricken by a devastating flood that affected parts of Uttarakhand, Himachal Pradesh and Uttar Pradesh. Over 4,200 villages were severely affected, killing over 5,700 people [1]. This was the country's worst natural disaster since the 2004 tsunami. The disaster was caused due to a multi-day cloud burst, which caused the melting of Chorabari glacier and thus causing the overflow of Mandakini River. Heavy rainfall for four consecutive days as well as melting snow aggravated the floods. Warnings by the India Meteorological Department predicting heavy rains were not given to public, causing thousands of people to be unaware of the situation, thus leading to huge loss of life and property.

If people in these areas were provided information regarding the continuous rain in advance, they would have got time to evacuate the area and also save their movable properties. So a disaster early warning system is necessary to provide proper alert before disaster occurs. India has been a victim to floods since a long time. More number of floods occurs than any other disasters in the country. In Fig. 1, the chart shows a statistics of major disasters that has occurred in India from 1980 to 2010. Clearly, India is mostly affected by flood.

It would also be useful if the level of water in a water body is stored in a database along with the date and time of the change in the level. This data can be used for future reference. For example, the history of water level can be used in scientific studies. There are certain spots in water bodies were we can measure the water level. These are the spots were the water is still or the current is not very strong. To determine whether the water level is safe or not in a certain area, we can use these spots to do so. Sensors can be planted in these areas. Then the readings of the sensors in these areas must be stored in the database with respect to the locations of the sensors. In this way, we can keep track of the water levels in all these areas. For this, we will have to manage the database effectively. Since natural disasters are very common these days, it has become a need to continuously monitor the weather and climatic changes. So if these changes over a long period of time are recorded, it can prove to be effective.

So looking at these problems, we have designed a system which can be used to monitor record and send alert to the people before disaster strikes. Since almost every common man is using a mobile phone these days, it makes best sense to send alert to these devices using SMS. People can easily approach their mobile devices and get the knowledge of the future disasters. Looking at advancement in today's technology, we can provide alert in alternate modes such as e-mail or using websites that can display the exact coordinate of the area of disaster on a map.

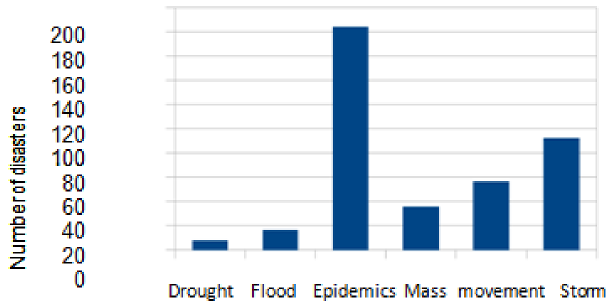

Fig. 1 Disasters in India for a period of 1980-2010

Space systems from their vantage position have unambiguously demonstrated their capability in providing vital information and services for disaster management [2]. The Earth Observation satellites provide comprehensive, synoptic and multi temporal coverage of large areas in real time and at frequent intervals and thus have become valuable for continuous monitoring of atmospheric as well as surface parameters related to natural disasters. Geo-stationary satellites provide continuous and synoptic observations over large areas on weather including cyclone-monitoring. Polar orbiting satellites have the advantage of providing much higher resolution imageries, even though at low temporal frequency, which could be used for detailed monitoring, damage assessment and long-term relief management. The vast capabilities of communication satellites are available for timely dissemination of early warning and real-time coordination of relief operations. The advent of Very Small Aperture Terminals (VSAT) and Ultra Small Aperture Terminals (USAT) and phased - array antennae have enhanced the capability further by offering low cost, viable technological solutions towards management and mitigation of disasters. Satellite communications capabilities-fixed and mobile are vital for effective communication, especially in data collection, distress alerting, and position location and coordinating relief operations in the field [3].

The rest of the paper is organized as follows. System design is explained in section II. Implementation and performance are presented in section III and IV. Concluding remarks are given in section V.

II. Proposed Algorithm

2.1 Data Collection Module

The first step would be to use a sensing device which can supply the information about the water level. In our system, we have used a level gauge to do this job. The voltage across this gauge changes if water level changes. This gauge is connected to the microcontroller unit which is present inside the client side system described in the next sub-section.

The data collection module of our FAS consists of a level gauge, Arduino MCU board and Raspberry pi. The level gauge is actually a variable resistor whose resistance changes as the water level changes, resulting in voltage variation across it. This voltage is fed to the analog input pin of the Arduino board. The MCU on this board converts this signal to digital value and then sends it to Raspberry pi. A program running in Raspberry pi opens the device file associated with Arduino Uno and sends a request to it. The Arduino then responds by sending the sensor reading. This happens for every fixed interval. Once the sensor reading is received, it is ready to go to the server, where the decision is taken on what to do with the data.

2.2 Client System

When the client system is placed online in the flood alert system (FAS), it requests for connection with the server and starts sending sensor reading to it. The components used in the client side system are described below. The first component is used to read the analog voltage from the level gauge. This component would be a microcontroller unit (MCU). For this purpose, we have used Arduino Uno board which houses Atmel ATmega328 microcontroller. The sensor is connected to analog port of the MCU. This board can be programmed using Arduino IDE. The data collected by the sensor must be sent to the server over the internet. For this, we needed a system which could be connected to the Arduino board and also has an interface to connect to the internet.

For this purpose, we chose Raspberry pi model-B, which is a pocket-sized computer. This computer has a 700 MHz SoC with 512 MB RAM and is based on ARM architecture. It runs ARM version of many lightweight Linux distributions. We have chosen Raspbian OS, a Debian OS derivative, which since it has lots in common with Ubuntu, which is the base operating system that we used. This system also has Ethernet interface, which can be used to connect the system online. Since Raspberry pi is a complete computer based on Linux, we can program it using the programming techniques used to program a typical Linux-based system and the standard API's. We connected the Arduino board to the USB port of Raspberry pi and then to the server using Ethernet. Raspbian OS has a complete set of tools used for programming the system.We have used the GNU (recursive synonym for GNU's Not UNIX) GCC (GNU Compiler Collection) for compiling our program. Figure 2 shows a Raspberry pi Model-B.

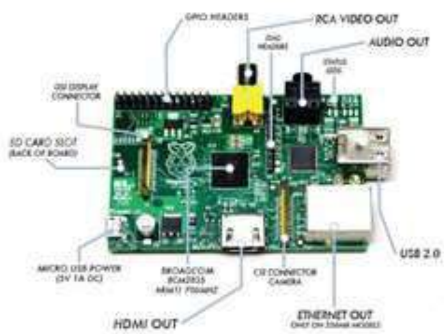

Figure. 2 Raspberry pi Model-B

The server is connected to a system which can send SMS upon request. This can be a GSM module or a mobile phone itself. To send commands to the SMS module, we used gnokii, which is a suite of programs used to communicate with mobile phones. It was originally designed for Nokia mobile phones running on symbian OS. Newer version of this software provides support for more phones. Using this tool, we can use different features of mobile phones such as phonebook, SMS, calendar, etc. We used the SMS feature of this tool. The software works fine with GSM SIM900 module as well.

2.3 Server System

The client system must be connected to a server. This server reads data from the client and then stores it in its database. This data can then be used for future reference. In our system, we used a computer as a server. This computer waspowered with Ubuntu, an open source operating system based on Linux. We used MySQL for managing the database. The server does a set of jobs. These jobs are described below.

- Receive sensor reading from the client and store it in the database.
- Register mobile number of the user upon user request. This is done using a web page.
- Unregister a mobile number from the database on user request. This is also done using a web page.
- Send alert SMS to the users registered in the database if the water level is critical.

The server runs common gateway interface (CGI) program to create dynamic web pages. This program is used to insert or delete phone numbers on user request. The server also displays tabulated sensor data on the web page as soon as the user requests for it. Fig 3 shows the layout of the components connected in the FAS in block diagram. The system is completely written in C. We have chosen C because it is completely supported by default in a Linux system and it is also friendly and robust when it comes to network programming.

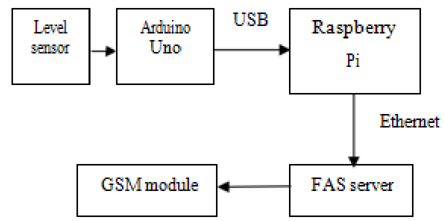

Figure. 3 Layout of components in FAS

Arduino can be programmed using Arduino IDE. This IDE is designed to provide complete support in programming the microcontroller. Raspberry pi can be programmed using the tools which are already present in the system. It is very convenient to use these tools for programming purpose.

This system can be connected as shown in Fig. 4 with little modification in programming. This setup enables us to monitor water level in more than one location and place the data in the same database. Each of these clients contains a level sensor, a MCU, Raspberry pi. This setup is connected to the server through internet.

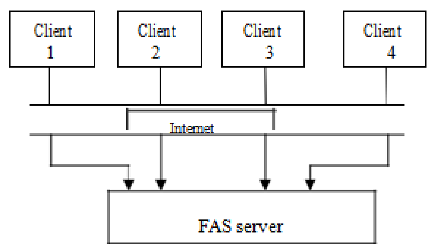

Figure. 4 Multiple FAS clients connected to a server

The sample database entries made by our program are shown in Fig. 5. This is done with the MySQL APIs which is designed to embed database queries inside a C program. Since the tools like MySQL, Gnokii, Arduino, etc. are open source and there is plenty of open source software that can fulfill the system design of FAS, the design cost is also reduced.

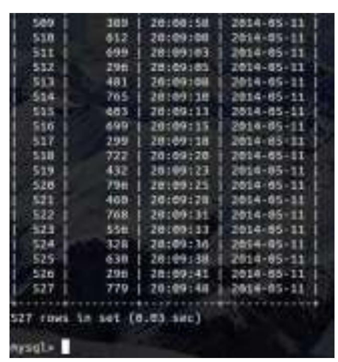

Figure.5 Sample database entries made by FAS

III.IMPLEMENTATION

We have implemented the system for one client and server system, although we could make it by using multiple clients. This is just for the ease of implementing a prototype of the system. For example, instead of using a physical server, we have used a service that acts as a server. The different tools used to develop this system are:

GNU Compiler collection (GCC) to compile our program

- Gnokii software for sending SMS.
- MySQL database to store data and user numbers.
- LAMPP (MySQL/ Apache server).
- Arduino UNO

- Raspberry PI
- Float Sensor

IV.PERFORMANCE

To test the performance of the system, we kept it under observation for ten hours continuously. We observed that the sensor reading was accurately transmitted from the client to the server. There was not even a slight fluctuation or variation in sending the data and the information also arrived in time. The database where the sensor data is stored was updated only if there was a change in two consecutive values that arrived to the server. This ensured that there were no multiple copies of one type of reading. The message was sent on time as soon as the sensor triggered the rise in water level. When in operation, the Raspberry pi was heated, but it was normal for the system to heat up that much. So, the hardware worked efficiently throughout the period and also worked continuously.

V.CONCLUSION

Flood warning is closely linked to the task of flood forecasting. The distinction between the two is that the outcome of flood forecasting is a set of forecast time-profiles of channel flows or river levels at various locations, while "flood warning" is the task of making use of these forecasts to make decisions about whether warnings of floods should be issued to the general public.Flood alert system forecasts are extremely effective in reducing flood damage. Advanced warnings for floods can mean the difference between life and death and in reducing property losses. As little as one hour of lead-time can result in up to a 10-percent reduction on flood damages. Flood forecasting has proven to be a vital link in providing economic benefits to a Nation and must continue to improve.

REFERENCES

[1] "India raises flood death toll to 5,700 as all miss ing persons now presumed dead". CBS News. 16 July 2013.

[2] Rao, D.P. 1998, Remote sensing & GIS for sustainable development: An overview. Proc. Int. Symp. on Resource and Environmental Monitoring : Local, regional and global. Sept. 1-4, 1998 Budapest, Hungary.

[3] Vogel, A., Contribution of Space Technology to Earthquake Prediction, Research, Adv. Earth Oriented Application. Space Technology, 2012.

[4] Jyoti Purohit and C.R. Suthar, Disaster statistics in Indian scenario in the last two decades, Department of Public Administration, M L S University, Udaipur, India, 2012