

## Wireless Sensor Network for Real-time Flood Monitoring Based on 6LoWPAN Communication Standard

Nuhu B. K.<sup>1\*</sup>, Arulogun O. T.<sup>2</sup>, Adeyanju I. A.<sup>3</sup>, Abdullahi I. M.<sup>4</sup>

<sup>1&4</sup>Department of Computer Engineering, Federal University of Technology Minna, Nigeria.

<sup>2</sup>Department of Computer Science and Engineering, Ladoko Akintola University of Technology, Ogbomoso Nigeria.

<sup>3</sup>Department of Computer Engineering, Federal University Oye-Ekiti, Ekiti, Nigeria

\*Corresponding author, e-mail: nuhubk@futminna.edu.ng

### Abstract

*Riverine flood is a major disaster faced by most countries and has significant adverse effect on long term economic growth of affected regions and their environments. Several systems have previously employed different technologies to monitor riverine flood but are expensive with low accuracy and consumes high amount of energy. In this paper, we proposed an energy efficient and accurate flood monitoring system. The system leverages on Internet Protocol Version 6 over Low Power Wireless Personal Area Network (6LoWPAN) technology to construct a Wireless Sensor Network (WSN) comprising of two XM1000 motes and a rule-based water level monitoring application. The motes were configured using NesC programming for flood monitoring with Basestation and water level sensing applications. The water level sensing mote samples and transmits real-time water level information to the Basestation mote which interfaces with a rule-based water level monitoring application. The application compares current water level with a predetermined threat level and alerts relevant agencies when flood is imminent via an email. The results obtained from the emulation of the developed system showed that, it achieved an accuracy of 95.3% in water level monitoring with a Mean Squared Error of 5.1. The power consumed in transmitting a packet of 2 bytes payload plus other overhead was 0.4 $\mu$ J and 0.0396mJ with and without 6LoWPAN configuration respectively.*

**Keywords:** Riverine flood, Monitoring 6LoWPAN, Mote

Copyright © 2016 APTIKOM - All rights reserved.

### 1. Introduction

One of the major problems faced by many countries around the world today is riverine flooding and it occurs due to changing weather conditions recently attributed to Global warming. According to the United Nations Educational Scientific and Cultural Organizations (UNESCO), fifty percent of water related natural disasters world-wide between 1990 and 2001 are caused by flood [1]. Riverine flood is the overflowing by water above the normal confines of a river. There are other kinds of flood such as: flash flood, single and multiple event flood, seasonal flood, coastal flood, urban flood etc. [1].

is mostly a natural disaster caused by the changing weather conditions recently attributed to global warming resulting into heavy rainfall, highly accelerated snowmelt, severe winds over water, unusual high tides or tsunamis. It can also be caused by the failure of dams, levees, retention ponds or other structures retaining water [2]. Flooding can have significant effect on long term economic growth of the affected regions and their environments [3]. It leads to loss of lives, displaces hundreds of people from their homes, washes away farmlands, destroys social amenities such as hospitals, schools, markets, poles carrying electric cables etc.

Report from Nigeria has shown that in 2012 at Lokoja, Kogi state, not fewer than 332 communities in nine of twenty-one local Government areas were affected; rendering close to two million persons homeless. According to the National Emergency Management Agency (NEMA), about 70 per cent of Kogi State population were affected [5]. Flooding also caused some health problems to the affected persons. Most of the people around a river depend on that river for drinking water, after flooding the water becomes contaminated causing cholera and typhoid among other diseases.

Despite the annual damages in Nigeria and its counterpart developing countries, few systems exists to monitor flood and the few are either expensive, require high energy, inadequate, inefficient or totally depends on complex middleware to connect with existing internet infrastructure for wider public information outreach. This research is intended to provide mechanism to monitor riverine flood employing 6LoWPAN technology to create a wireless sensor network.

IPv<sub>6</sub> packets over low power wireless personal area networks (6LoWPAN) protocol is an adaptation layer that allows the transmission of IPv<sub>6</sub> packets over IEEE802.15.4 networks.

### 1.1. Overview of IEEE802.15.4 (6LoWPAN) Technology

6LoWPAN is based on the IEEE802.15.4-2003 standard introduced by the Internet Engineering Task Force (IETF) in 2005, IETF aimed at defining header compression mechanism that allows the transport of IPv6 packets over IEEE802.15.4 links through a layer called “the adaptation layer” that stands in between IEEE802.15.4 Medium access Control (MAC) layer and the IPv<sub>6</sub> network layer (Figure 1). The adaptation between full IPv<sub>6</sub> and the IEEE802.15.4 is performed by routers at the edge of 6LoWPAN regions, referred to as edge routers.

From the protocol stack in Figure 1, it can be seen that the PHY and MAC layers of 6LoWPAN are defined by IEEE802.15.4 and the upper layers are defined by IETF. The network layer is based on IPv6 packets. At the Transport layer contrary to the conventional IP network, 6LoWPAN uses only User Datagram Protocol (UDP) and Internet Control Messaging Protocol (ICMP); Transport Control Protocol (TCP) is not commonly used by 6LoWPAN for complexity reasons [13].

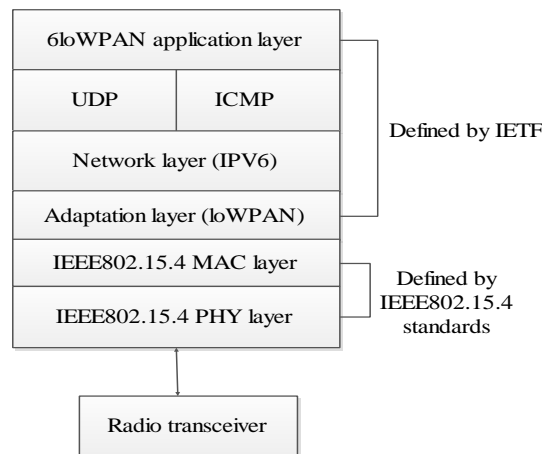


Figure 1. 6LoWPAN protocol stack

## 2. Flood Monitoring

Flooding has been studied under various considerations and technologies such as WSN, embedded system with a middleware; internet based real-time data acquisition, flood modelling and forecasting [14]. Three areas has been identified as the basic aspects of flood management, they are flood monitoring, flood prediction/forecasting and flood alerting/warning. The most important of the three is flood monitoring, through monitoring, data are gathered and used to model future possibilities of flood occurrence which is the main focus of this research to monitor flood through Wireless Sensor Network (WSN). In the next section WSN is discussed because it was identified as the most appropriate and reliable system used in recent years for monitoring applications where a number of sensors are planted to read and transmit wirelessly the hydro/metrological parameters to be used in analysing flood.

### 2.1. Wireless Sensor Network (WSN).

Wireless sensor networks are low power, low cost, multi-hopping systems that are independent of external service providers, can form an extendable network without line of sight coverage; but have self-healing data paths [12]. Wireless sensor networks consist of distributed, wirelessly enabled embedded devices capable of employing a variety of electronic sensors. Each node in a wireless sensor network consists of one or more sensors in addition to a microcontroller, wireless transceiver, and energy source (Figure 2). The microcontroller functions with the electronic sensors as well as the transceiver to form an efficient system for relaying small amounts of important data with minimal power consumption [4]. Wireless sensors are used to acquire real time meteorological/hydrological data and transmit it wirelessly to a desired node in the network for flood analysis.

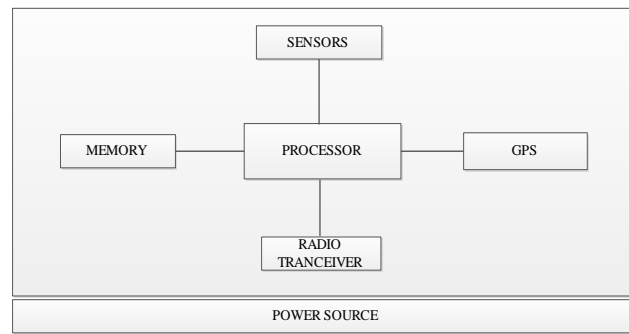


Figure 2. Components of a WSN node [10]

## 2.2. Related Works

In literature, the existing riverine flood monitoring systems that employed WSN used one of the existing wireless communication technologies; hence their limitations are inherent in the technologies they employ. For instance, in the works of [8] (shown in Figure 3), [14] (shown in Figure 4), [11] and [9] they employed GPRS technology in designing a WSN for relaying water related data to the place where flood computation was to be carried out. The authors in [14], Incorporated a middleware called VirtualCOM (Figure 5) to support the GPRS communication. The systems have their individual capabilities ranging from simplicity and cost efficiency. However, the GPRS technology they employed made them to consume higher energy than other existing low power technologies like Zigbee and 6LoWPAN. Again they are faced with transmission delay that leads to untimely information which could cause distrust on the part of its potential users. Finally, they require a gateway or proxies to connect with the existing internet infrastructure.

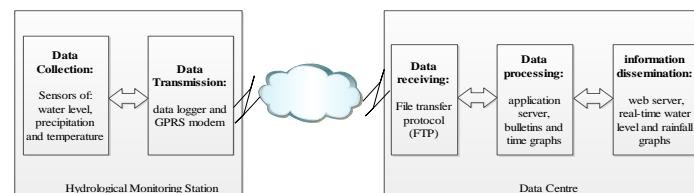


Figure 3. The high level architecture of flood monitoring system [8]

## 2. Research Method

Explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode or other), how to test and data acquisition [1], [3]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [2], [4].

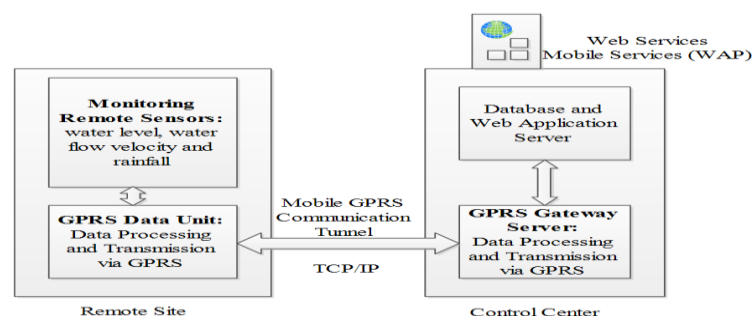


Figure 4. System Architecture of real-time flood monitoring and warning system [14]

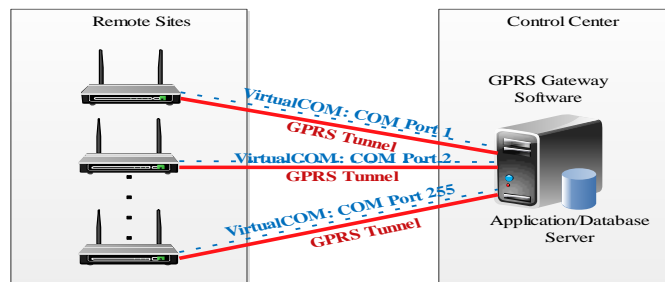


Figure 5. Logical Structure of the Implemented VirtualCOM [14]

Authors in [6] and [15] developed a WSN monitoring systems that incorporated Zigbee technology for transmitting sensor readings. In [6], a wireless gateway as shown in Figure 6 was used to connect the sensing module with the surveillance module where flood computation was carried out. The systems are robust and give timely alert. However, the incorporation of a gateway between the field sensors and surveillance centre means more energy consumption and to integrate with the existing internet infrastructure, the systems may require a middleware for efficiency.

In the work of [7], titled 'Remote Detection and Monitoring of a Water Level Using Narrow Band Channel'; they proposed a difference image based Joint Photographic Expert Group (JPEG) communication scheme using sparsely sampled images in time domain. The system comprised of a master system and a slave system (Figure 7). The slave system consists of two subsystems: the image acquisition subsystem with storage and the image encoding and transmission subsystem. The image acquisition subsystem comprised of a camera that captures the water images with the size 640×840 and stores them in the memory to be used in finding the difference image when compared with a reference image initially captured when the camera is been calibrated.

The image encoding system encodes the difference using JPEG encoder; this reduced the image to about 10 Kbits to be sent through a narrowband Ultra High Frequency (UHF) modem with a speed of 9600bps. The master system provides control to the remote slave system; it also receives the transmitted data from the slave system. It first receives the reference image then followed by the difference JPEG image. The difference image is decomposed using a JPEG decoder. The decoded reference image is added to the reference image to reconstruct the original field image and then passed through a low pass filter to remove noise due to light reflection and ripple on the water surface during capturing. The image is finally used by an algorithm developed to measure the water level. The system shows some remarkable accuracy level as compared to ultrasonic sensor and can be used to monitor not only the river but the surrounding situation of the river. However, the system is complex and lacks power management strategy; the medium of communication used can also lead to transmission delay due to its narrowness.

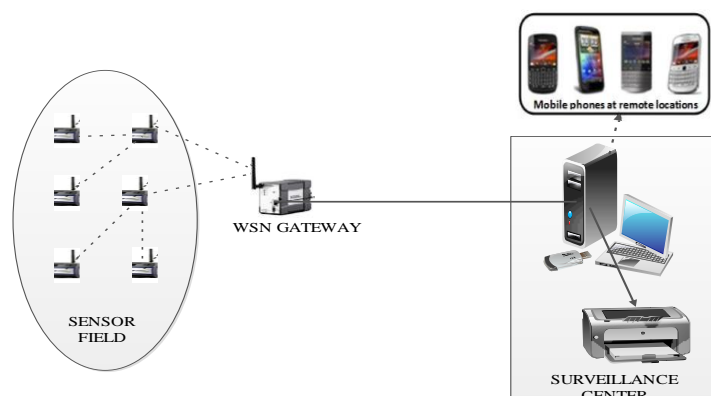


Figure 6. Architecture of the FMDS [6]

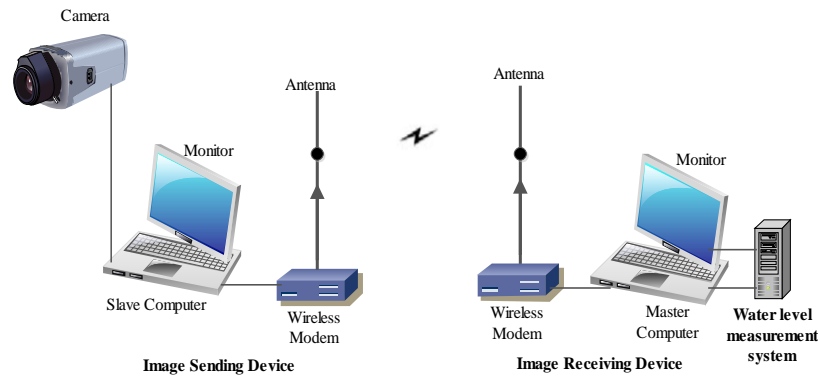


Figure 7. System configuration of remote detection and monitoring of water level

### 3. Method

The system is composed of both software and hardware resources, the block diagram of the system is shown in Figure 8 and the system architecture is in Figure 9, the system comprised of four modules namely: sensing, transmission, database/monitoring and alert modules. All the modules were configured with both software and hardware resources as can be seen in the subsequent sections.

#### 3.1. Hardware Component

The main hardware component of the design is the XM1000 sensor platform from Advanticsys Company; it is the new generation of mote modules, based on "TelosB" technical specifications, with upgraded 116Kb-EEPROM, 8Kb-RAM and integrated temperature, humidity and light sensors. Other features include: IEEE 802.15.4 compatible, TI MSP430F2618 Microcontroller with CC2420 RF chipcon for transmission, TinyOS 2.x and ContikiOS compatible, user and reset buttons, 3 LEDs, USB interface, powered by 2xAAA batteries of 1.5volts each. These made it an ideal choice for this work.

Another hardware used was Parallax "PING)))" ultrasonic distance sensor which served as the water level sensor, it provides precise, non-contact distance measurements from about 2 centimetres (0.8 inches) to 3 meters (3.3 yards). It can be interfaced with the XM1000 mote requiring only one I/O pin. The "PING)))" sensor works by transmitting an ultrasonic burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. The echo pulse width corresponds to the distance between the device and the target (water surface), as the distance decreases it signifies a rise in the water level.

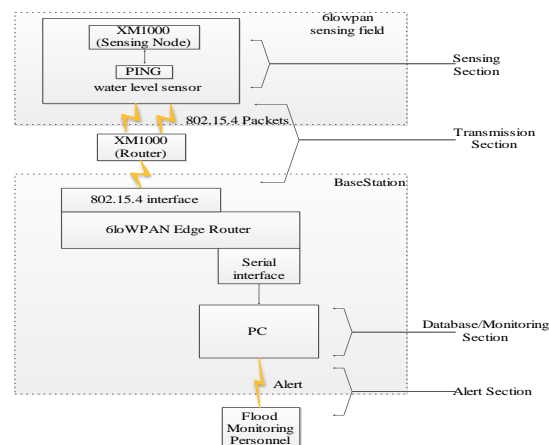


Figure 8. Block Diagram of the 6LoWPAN Based Flood Monitoring System

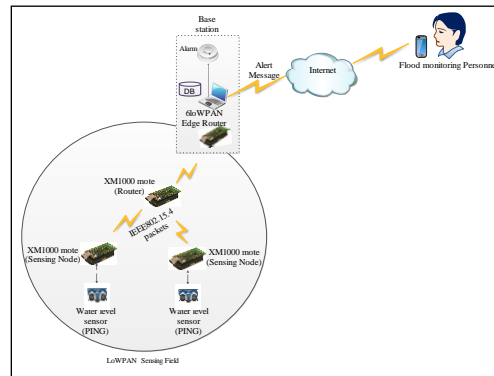


Figure 9. System Architecture of the 6LoWPAN Based Flood Monitoring System

### 3.2. Software Component

The major software component of the system is the Operating System (OS) called ‘TinyOS’ which is written in a language called Network Embedded Systems C (nesC), a dialect of C programming language. This OS is configured and installed on the XM1000 sensor mote to perform the required operation. The configuration and installation is done on the Linux virtual machine called UBUNTU installed on Workstation version 10.0, on my windows 8 Personal Computer (PC). The TinyOS version used was version 2.1.2.

#### 3.2.1. Sensing Module

The sensor used for water level measurement (PING sensor) was interfaced with the XM1000 through its General Input/Output Pin (GPIO) expansion pin (port 2, pin 6), and thus it used the HplMsp430GeneralIO component and interface. The flowchart on how the sensor and the mote were interfaced to achieve the water level measurement event is shown in Figure 10.

#### 3.2.2. Transmission Module

Transmission of the data is handled by the radio component which is the CC2420, as a way of preserving energy, because transmission of packets consumes more energy than any other operation of a mote. It is very important to have control over the radio, such that it only becomes active when a packet is ready to be sent, this was handled through the NesC coding as shown in the flowchart of Figure 10, thus, the radio is called only when the sensor was read and the structure of the message is completed.

#### 3.2.3. Database/Monitoring Module

At the Basestation is another XM1000 mote serving as the Edge router which interfaces the sensing network with a rule-based water level monitoring application through the serial port of a Personal Computer (PC).

The rule-based water level monitoring application whose flowchart is shown in Figure 11, compares the current water level with a predetermined threat level, so that, when it is less than or equal to the threat level, it alerts the flood monitoring authorities that flood is imminent and the river should be under close watch. A web server application was also designed in compliant with ASP.NET Internet Information server (IIS) which is a standard for web based services. The server served as a central database which stores the water level information with a timestamp and allow users to access the current and historical information at anytime from anywhere using any internet enabled device through a web based user interface designed using C# programming language.

#### 3.2.4. Alert Module

There are two types of alert employed: alarm and email. The email alert contains the information “River Chanchaga is under flood risk, water level is xcm or less to the surface of the river bank, please watch closely” sent to the flood monitoring authorities, where ‘x’ stands for the threat level. The alarm was designed to sound at the basestation location. Both the email and alarm were designed to be activated whenever the river level is equal or greater than the threat level but while the email based alert required an active internet to send the message, the alarm does not depend on the internet.

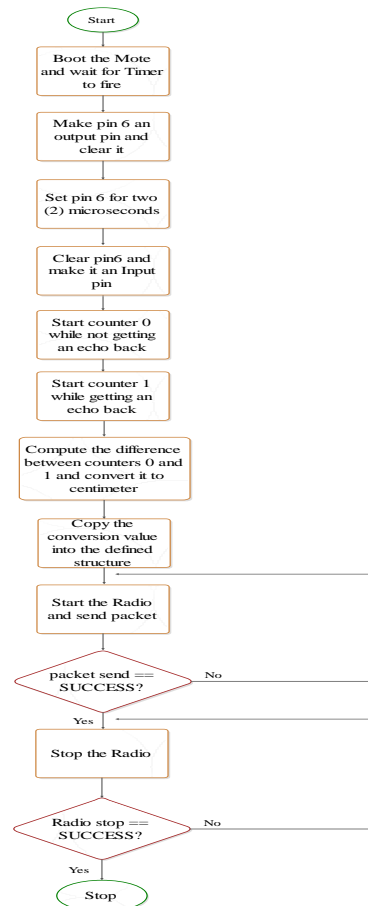


Figure 10. Flowchart of sensing and transmission events for the flood monitoring

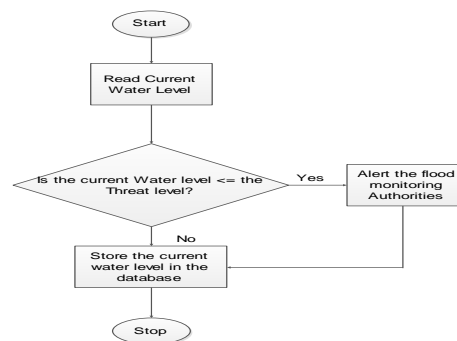


Figure 11. Flowchart of the rule-based model

### 3.3. Power Consideration

The XM1000 mote at the sensing field (river site) is powered by 2 AA batteries of 1.5V each. From experiments carried out, the adopted IEEE802.15.4 (6LoWPAN) communication standard helped to save a lot of energy consumed by the system as compared to conventional communication schemes during packet communication. The XM1000 edge router at the basestation interfaced with the PC through a Universal Serial Bus (USB); hence it received its power from the PC. The PC is powered from Alternating Current (AC) mains through an inverter which provides power cover in the case of power outage and automatically charges itself when power is restored.

#### 4. System Implementation/Emulation

The system was implemented in two configurations, with 6LoWPAN and without 6LoWPAN. Compiling a TinyOS application with 6LoWPAN support for XM1000 was done through Berkeley Low Power IP (BLIP) stack implementation using the terminal command `'sudo make xm1000 blip'` where xm1000 is the target and blip is the IP stack implementation for the constrained device. Before executing this command, the necessary header files were included to the configuration and module files of the NesC codes installed on the device, they are: `<IPDispatch.h>`, `<lib6lowpan/lib6lowpan.h>`, `<lib6lowpan/ip.h>`, `<lib6lowpan/ip.h>` and `"blip_printf.h"`. In the MAKEFILE, some flags were set to define the routing protocol which is RPL, the prefix address that all nodes in the network will have is fec0, the channel used which is 21 and the RPL root address which is 1, among others. These were necessary to compile with 6LoWPAN.

Porting the application to the same target was using the terminal command `'sudo make xm1000 blip install.2 bsl,/dev/ttyUSB0'` where 2 is the node ID and /dev/ttyUSB0 refers to the port where the mote is plugged which can be obtained by typing `'motelist'` from the terminal. To compile without 6LoWPAN the blip at the end of the command is simply removed including the added header files.

Riverine flood was simulated in order to emulate the functionality of the system using both the hardware and software resources mentioned earlier. A plastic container was used to represent the river with the system deployed around it as shown in Figure 12, Water was continuously added to the container to depict the rising of the river water level, the response of the system was noticed through sending of the email and alarm sounding as the condition in the rule is met. Figure 13 is a schematic of the setup.

Thirty of such tests were carried out to ensure that the system responded as expected. In each of the simulated water levels, a tape was used to measure the actual value. The actual water level was recorded side by side with the simulated value for correlation analysis and other computations for performance evaluation purpose.

The system was implemented in two configurations, with 6LoWPAN and without 6LoWPAN. Compiling a TinyOS application with 6LoWPAN support for XM1000 was done through Berkeley Low Power IP (BLIP) stack implementation using the terminal command `'sudo make xm1000 blip'` where xm1000 is the target and blip is the IP stack implementation for the constrained device. Before executing this command, the necessary header files were included to the configuration and module files of the NesC codes installed on the device, they are: `<IPDispatch.h>`, `<lib6lowpan/lib6lowpan.h>`, `<lib6lowpan/ip.h>`, `<lib6lowpan/ip.h>` and `"blip_printf.h"`. In the MAKEFILE, some flags were set to define the routing protocol which is RPL, the prefix address that all nodes in the network will have is fec0, the channel used which is 21 and the RPL root address which is 1, among others. These were necessary to compile with 6LoWPAN.

Porting the application to the same target was using the terminal command `'sudo make xm1000 blip install.2 bsl,/dev/ttyUSB0'` where 2 is the node ID and /dev/ttyUSB0 refers to the port where the mote is plugged which can be obtained by typing `'motelist'` from the terminal. To compile without 6LoWPAN the blip at the end of the command is simply removed including the added header files.

Riverine flood was simulated in order to emulate the functionality of the system using both the hardware and software resources mentioned earlier. A plastic container was used to represent the river with the system deployed around it as shown in Figure 12, Water was continuously added to the container to depict the rising of the river water level, the response of the system was noticed through sending of the email and alarm sounding as the condition in the rule is met. Figure 13 is a schematic of the setup.

Thirty of such tests were carried out to ensure that the system responded as expected. In each of the simulated water levels, a tape was used to measure the actual value. The actual water level was recorded side by side with the simulated value for correlation analysis and other computations for performance evaluation purpose.



Figure 12. Actual water level measurement during system emulation

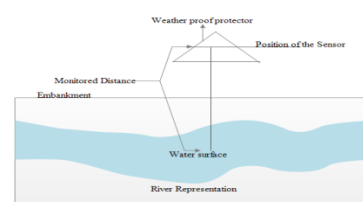


Figure 13. Schematic of the emulation setup



The setup as shown in Figure 14 was used to compute the energy consumed by the system in transmitting a packet. The instruments involved were a digital oscilloscope (Sefram 5062DC), an instrumentation amplifier (AD620AN) to amplify the voltage which was too small to be analysed on the oscilloscope. Figure 15 is the circuit arrangement of the instrumentation amplifier connection.

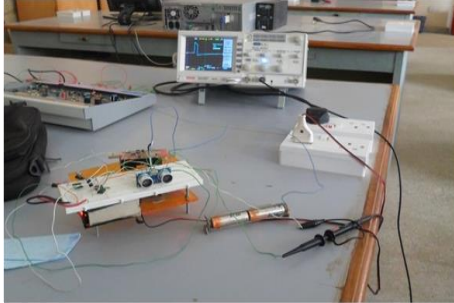


Figure 14. Energy consumption experimental setup

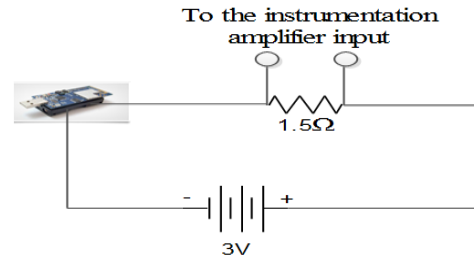


Figure 15. Energy consumption experimentation setup

From the Figure, it can be seen that a 1.5 ohms resistor was inserted in series with the battery powering the sensing mote. The voltage drop across this resistor was measured through the instrumentation amplifier with a gain of 106 and the energy consumed by the mote in sending one packet of the water level to the edge router was computed using equation 1

$$E(\text{joules}) = (V_{\text{scope}} \div (106 \times 1.5)) \times 3 \times \Delta t \quad (1)$$

Where: 3 is the value of the 2 AA batteries powering the mote

$\Delta t$  is the time division read from the oscilloscope.

$V_{\text{scope}}$  is the voltage of the oscilloscope.

## 5. Results and Discussion

The results obtained from the 6LoWPAN based flood monitoring system during system emulation, are divided into two: emulation result and experimentation result. Figure 16 is a screenshot showing the alert sent during system's emulation.

### 5.1. Emulation Results

The emulation result presents the 30 samples of water level reported by the system versus the actual water level measured during the system emulation for accuracy, Mean Squared Error and the correlation analysis for performance evaluation. The result is contained in Table 1.

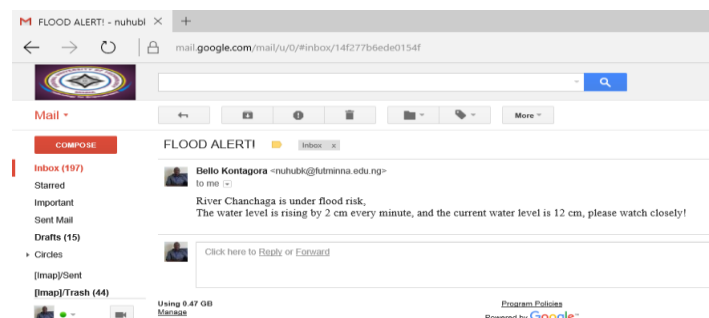


Figure 16. Screenshot showing the flood alert during system emulation

Table 1. Actual versus simulated water levels recorded during emulation

S/NO	ACTUAL WATER LEVELS (cm)	SIMULATED WATER LEVELS (cm)	ACTUAL ERRORS (cm)	PERCENTAGE ERRORS (%)	SQUARED ERRORS
1	5.9	6	0.1	1.7	0.01
2	7.0	7	0.0	0.0	0.00
3	7.8	8	0.2	2.6	0.04
4	9.8	10	0.2	2.0	0.04
5	11.5	12	0.5	4.3	0.25
6	12.1	13	0.9	7.4	0.81
7	13.1	14	0.9	6.9	0.81
8	14.5	15	0.5	3.4	0.25
9	15.2	16	0.8	5.3	0.64
10	16.5	17	0.5	3.0	0.25
11	17.9	18	0.1	0.6	0.01
12	21.5	22	0.5	2.3	0.25
13	22.1	23	0.9	4.1	0.81
14	24.3	26	1.7	7.0	2.89
15	25.8	27	1.2	4.7	1.44
16	27.5	29	1.5	5.5	2.25
17	29.0	30	1.0	3.4	1.00
18	32.9	34	1.1	3.3	1.21
19	34.3	36	1.7	5.0	2.89
20	35.4	37	1.6	4.5	2.56
21	38.0	40	2.0	5.3	4.00
22	39.0	41	2.0	5.1	4.00
23	39.2	42	2.8	7.1	7.84
24	43.1	46	2.9	6.7	8.41
25	44.4	47	2.6	5.9	6.76
26	45.1	48	2.9	6.4	8.41
27	53.4	57	3.6	6.7	12.96
28	61.2	65	3.8	6.2	14.44
29	66.5	70	3.5	5.3	12.25
30	87.6	95	7.4	8.4	54.76
	Mean		1.6cm	4.7%	5.1

### 5.2. Energy Measurement Experiment Results

The experimentation result, presents the energy consumed by the system in sending a single packet. Table 2 contained the summary of the parameters read from the oscilloscope during the experiment and the computed energy consumed for two scenarios.

Table 2. Experimentation results

Parameter	Without 6loWPAN	With 6loWPAN	Difference
Vscope	1.00V	960mV	0.9904V
Time Division ( $\Delta t$ )	21ms	21ms	0
Energy Consumed	0.0396mJ	0.4 $\mu$ J	0.0392mJ

### 5.3. Performance Evaluation Results

In this section, the Mean Squared Error (MSE), Mean Percentage Error and the correlation between the simulated and actual water level values for the 30 samples as shown in Table 1 were computed using equations 2, 3 and 4 respectively

$$MSE = \frac{\text{Sum of squared errors for the 30 samples}}{30} \quad (2)$$

$$MPE = \frac{\text{Total PE for the 30 samples}}{30} \quad (3)$$

$$\text{Correl}(X, Y) = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}} \quad (4)$$

where  $\bar{x}$  and  $\bar{y}$  are Averages of simulated and actual water level values respectively

The MPE and MSE were computed to be 4.7% and 5.1 respectively while the correlation coefficient (R) between the simulated and actual water levels during the emulation was found to be 0.999749 and the squared of it is 0.9995. Figure 17 shows the correlation plot.

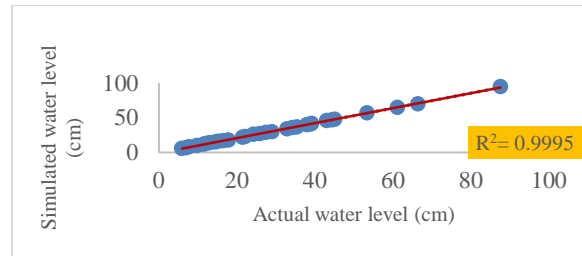


Figure 17. Correlation graph between the actual and simulated water levels recorded during system emulation

Similarly, from the experimentation results obtained in Table 2, the system was found to have saved a lot of energy during sending of one packet when configured with 6LoWPAN as compared to when the system was configured without 6LoWPAN.

### Acknowledgements

The authors wish to acknowledge the technical staff of the central engineering laboratory, Federal university of technology Minna for their understanding in making their equipments available during experimentation.

### References

- [1]. WMO (2011), "Manual on flood forecasting and warning", Geneva, Switzerland, retrieved at [http://www.wmo.int/pages/prog/hwrp/publications/flood\\_forecasting\\_warning/WMO%201072\\_en.pdf](http://www.wmo.int/pages/prog/hwrp/publications/flood_forecasting_warning/WMO%201072_en.pdf)
- [2]. Bariweni P. A., Tawari C. C. & Abowei J. F. N. (2012), "Some Environmental Effects of Flooding in the Niger Delta Region of Nigeria", *International Journal of Fisheries and Aquatic Sciences*, Vol. 1, Issue 1, pp35-46, 2012.
- [3]. Adedeji, A. A., & Salami, A. W. (2011), "Environmental hazard: Flooding and Its Effects on Residential Buildings in Ilorin, Nigeria", *Department of Civil Engineering, University of Ilorin, PMB 1515, Ilorin, Nigeria*.
- [4]. Chanin Jonathan Isaac & Andrew R. Halloran (2010), "Wireless sensor network for monitoring applications", *A major qualifying project report, submitted to the University of Worcester polytechnic institute United Kingdom*.
- [5]. Daily news report of 24th September, 2012 from: <http://www.sunnewsonline.com/new/cover/river-niger-sacks-332-villages-renders-2-million-homeless/>
- [6]. Edward N. Udo & Etebong B. Isong (2013), "Flood monitoring and detection system using wireless sensor network", *Asian journal of computer and information systems*, Vol. 01, Issue 04, pp108-113.
- [7]. Jaehyoung Y. & Hernsoo H. (2010), "Remote detection and monitoring of a water level using narrowband channel", *Journal of information science and engineering*, Vol. 26, pp71-82
- [8]. Keoduangsiue Sayaoth & Robert Goodwin (2012), "A GPRS-based data collection and transmission for flood warning system: The case of the lower Mekong river basin", *International journal of innovation, management and technology*, Vol. 3, No. 3, pp217-220.
- [9]. Khedo Kavi Kumar (2008), "Real-time flood monitoring, using wireless sensor networks", *The journal of the institution of engineers Mauritius*, pp59-69.
- [10]. Krishnamachari, B. (2005), "Networking wireless sensors", *Cambridge University Press*.
- [11]. Rafael Marin-Perez, Javier Garcia-Pintado & Antonio Skarmeta Gomez (2012), "A real-time measurement system for long-life flood monitoring and warning applications", *Sensors 2012*, pp4213-4236
- [12]. Seal Victor, Arnab Raha, Shovan Maity, Souvik Kr Mitra, Amitava Mukherjee & Mrinal Kanti Naskar (2012), "A simple flood forecasting scheme using wireless sensor networks", *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, Vol.3, No.1, pp45-60
- [13]. Shelby, Z., & Bormann, C. (2011), "6LoWPAN: The wireless embedded Internet", (Vol. 43), John Wiley & Sons.
- [14]. Sunkpho Jirapon & Chaiwat Ootamakorn (2011), "Real-time flood monitoring and warning system", *Songklanakarin Journal of Science and Technology*, Vol. 33, No. 2, pp227-235.
- [15]. Yamut C. & Kilaso S., (2011), "A wireless sensor network for weather and disaster alarm systems", *International Conference on Information and Electronics Engineering, IPCSIT*, Vol.6, pp155-159