A Real-Time Flood Alert System for Parking Lots

Homar Báez*, Idalides Vergara-Laurens*, Luz Torres-Molina*, Luis G. Jaimes†, Miguel A. Labrador‡
* School of Engineering Universidad del Turabo Gurabo, PR 00778
† Department of Computer Science, Florida Polytechnic University, Lakeland, FL 33805
‡ Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620
hbaez2@email.suagm.edu, ivergara@suagm.edu, torresl6@suagm.edu, ljaimes@flpoly.org, labrador@cse.usf.edu

***Abstract*—Floods are a constant threat throughout the year to the United Stated and its territories like Puerto Rico. Although there are various methods of alerts available; such as the Emergency Broadcast System or sirens, none of these can alert a user remotely in an efficient and timely manner. The design goal of this project is to provide a real-time system able to monitor sudden floods in parking lots, addressing the concern of water damage to vehicles; creating a personal opt-in alert that could reach an end user through their mobile phone. In this case, the system defines two types of nodes: Sensing and Sink. Each sensing node uses a hydrostatic pressure sensor to monitor the water levels; it will then communicate with neighboring nodes via XBee radios until the data reaches the sink node. The sink node is then responsible for sending the received data from the sensors to a remote server via mobile communications network (GSM). An up to date database of users and flood levels will then be processed and handled by the server, which will send users an email alert that will reach any mobile phone as a text message (SMS).**

***Index Terms*—Flood Detection, Wireless Sensor Networks, Arduino, XBee, SMS**

I. Introduction

Application of Wireless Sensor Networks include Crowdsensing [1], [2], security [3], and health care [4],Floods are a constant threat throughout the year to the United Stated and its territories like Puerto Rico. According to National Weather Service, approximately seventy-five percent of all Presidential disaster declarations in the US are related to floods [5]. Urban areas can be affected by flash floods, coastal flooding, river floods or by the inability to drain the sufficient amount of water from the area. While there are various methods of alerts available; such as the Emergency Broadcast Service and Coastal flood warning sirens, none of these alert users remotely in an efficient and timely manner. We are proposing a real-time flood alert system able to monitor sudden flooding in parking lots, alerting users through their mobile phones via an opt-in alert. This directly addresses the concern of a vehicle suffering potentially severe water damage.

The system can be defined to have two main components, a Wireless Network Sensor (WSN) and a central server. The sensor network will monitor flood levels in the area of interest and send the gathered data to the server which will analyze and store the data, register and handle the user data and send the corresponding alerts depending on the data received from the sensors. The system aims to have full compatibility with any mobile phone by using SMS text alerts.

A difficult aspect of this project is the power consumption and management of the sensor network. The system works off the power grid and relies fully on battery powered operation with solar charging. This was addressed by using synchronous sleep cycles for the sensors to achieve a balance between rate of discharge versus the charge rate provided by the solar panel. With this approach, the system may theoretically work powered by a lithium ion 1.2Ah battery without taking into account the solar charging during the day. It is of utmost importance that we achieve more than one day of battery power available to compensate for factors like low sunlight level and night operation.

This paper presents the System architecture as well as its functionality taking into consideration the power and transmission constraints. The rest of this paper will be organized in the following manner: Section provides the related work to this paper including SMS text alerts, and also the different ways those systems monitor flood levels. Section III provides a thorough look into the proposed system and how the communications protocol impacts our systems power consumption constraint. Section IV presents the results for the system evaluation in power consumption and network metrics. Finally, Section V concludes the paper and brings upon the roadmap for the future work on the server.

II. Related Work

This section presents previous works on flood alert systems using SMS text alerts utilizing and different methods on flood sensing.

A. Flood Detection with SMS Text Alerts

There have been many successful solution on flood alerts using SMS, yet most count on either having a constant power source or relying completely on solar energy as their main source of power. One example similar to our sensor design is the system proposed by [6]. In that paper a sensing unit is built using a GSM module and a PIC18F452 micro controller with 3 different liquid level sensor as input. The system defines a level threshold and sends one SMS text alert via GSM per once the threshold reached; once the levels start receding it will also alert the user on said event. The drawbacks of this design are that the sensing and alert system are concentrated into one unit. Also, the system is designed to be only be triggered by the levels and does not store historical data. Finally, the system is supplied constant power through a 15V power supply, having the necessity of being tethered to the power grid.

[7] proposes an approach to address the problem by using the HC-SR04 ultrasonic sensors to measure the distance of the liquid from the zero point reference. In this paper, an ATMEL AT89S51 is used as the micro controller unit with a SIM 300C as the GSM Module. The use of the ultrasonic sensor allows the system to get an accurate reading on the water level and to be able to measure up to four meters; it also allows the zero point height to be changed, making the system adaptable to various flood prone areas. However, the system can only be used by two users at a time, which input their phone numbers to the system and receive three alerts per threshold reached. This design relies on solar power as its only energy source, making it unsuitable for twenty-four hour operation. The node footprint is also quite large because in order to get a reading up to four meters, an eight-inches PVC pipe was used, a size not suitable for a large number of sensors deployed on a parking lot.

On [8] a portable flood alert system to be implemented and used by the Malaysian authorities is achieved with the use of analog water

978-1-5090-4228-9/17/$31.00 ©2017 IEEE

level sensors that emit electrical pulses when triggered. The system is capable of sensing 5 different water levels from normal to dangerous. The sensing unit sends all its data to a twenty-four hour control sever via a MAX232 serial interface. The server then stores every single level variation in a database. If water levels reach sensor 4 (dangerous level) the server will then send an SMS text message with an alert to the user or local authorities. The SMS server used was Ubuntu Server inside a virtual machine. While the system does achieves its threshold, it needs constant power to operate, also the wired serial communication with the server would be an inconvenience for remote sensor locations.

B. Radar Based Flood Alert Systems

Flood monitoring systems have been around for some time with the use of radar based flood estimation. In [9], the city of Houston established a flood alert system for the Brays Bayou. It was commissioned by the Texas Medical Center in 1998, and it gives the center real-time data on the predicted floods in the area. The system uses the National Weather Services NEXTRAD installation near downtown Houston.

This paper [10] presents a flood forecasting model using autoregressive methods with stochastic parameters. Data for these forecasts was obtained with two different types of radars for the Mayaguez Bay Drainage Basin Area. These radars (Off-the Grid and TropiNet) are able to detect rainfall events missed by the NOAAs NEXRAD. A distributed hydrologic model was used to obtain flood depths.

III. Proposed System

The proposed system consists on two main components: a Wireless Sensor Network (WSN) and a Server as presented on Figure 1.

The wireless sensor network will be used to constantly monitor the flood levels in the area of interest while the server will receive, analyze, store the data, and sends the alerts when level thresholds are reached. The sensor network defines two types of nodes: Sensor Nodes and the Sink node. Both node types will be powered and charged using a 10W Polycrystalline solar panel with a 1200mAh rechargeable battery pack. They also will be housed in a four inch diameter, five feet tall PVC pipe. We discuss the individual details bellow.

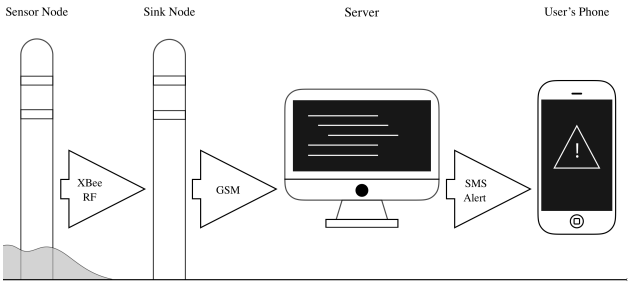

Fig. 1. Basic system design.

A. Sensor Node

The sensor nodes, showed on Figure 2, are responsible of measuring flood levels and reporting it to a sink node. The main component for these sensor nodes is the Arduino UNO R3 micro controller unit (MCU). It collects and converts the analog data collected from a Milone eTape liquid level sensor. This sensor is a hydrostatic pressure level sensor; it changes its resistive output depending on the external pressure applied by the liquid. Having a solid state sensor allows us to have a smaller node footprint. When the MCU has the data converted, it sends the gathered data to a sink node using a 2.4GHz XBee Series 1 module (XB24-DMWIT-250). After transmission of the data, the node will go to sleep for a set amount of time.

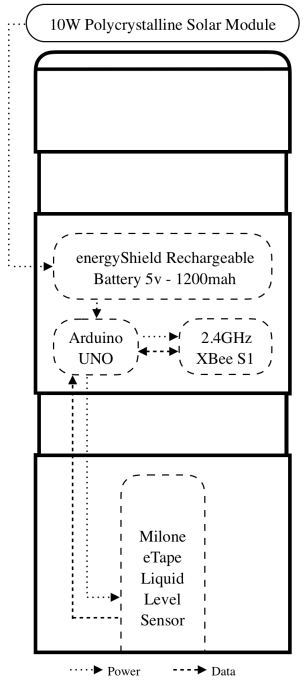

Fig. 2. Sensor node architecture.

B. Sink Node

The sink node will gather all data from the sensor nodes and send it to the server for processing via GSM. The main component in the sink node is the Arduino MEGA 2560. It uses the same XBee module (XB24-DMWIT-250) as the sensor nodes for communication with the WSN. Once it has aggregated the sensed data it sends said data via internet using a SIMCOM SIM900 GSM Module. Then sink node will also provide sleep coordination for the rest of the WSN. We can see this node's architecture in Figure 3.

C. Server

The systems server will be responsible for the processing of the acquired data sent by the sink nodes. Our design allows the server to manage multiple networks located in different locations. This server will keep and updated database with the registered users and the flood data. After it has processed all of the data it will send a text message notification to the registered users in the area when the flood level threshold has been reached. To have a clear understanding on the servers actions upon receiving the data from the network, we can take a look at Figure 4.

In order to register for the service, the users will send a text message with the parking area to the email belonging to the lots network as can be seen on Figure 5. Registered users will be kept on the database for a total of 12 hours before being removed from the alert service. The server will send the text message alerts (SMS) to users via email; letting it be compatible with any phone with an SMS service, not just smartphones. The server will also make the flood data available online.

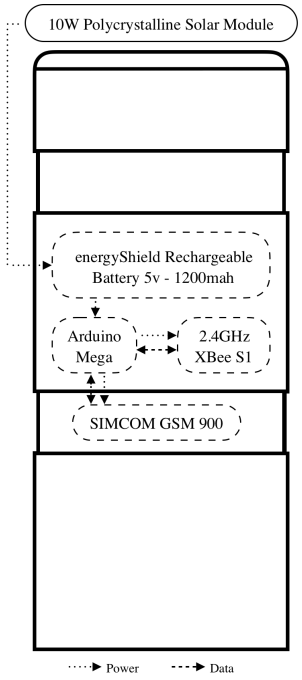

Fig. 3. Sink node architecture.

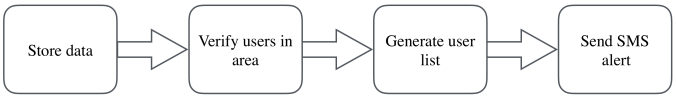

Fig. 4. Server actions upon arrival of data.

IV. Sensor Network Comunication and Power Management

The approach taken uses a Mesh network to deliver the sensor information to the sink node. It uses the DigiMesh networking protocol, a proprietary protocol based on a modified version of the AODV (Ad-Hoc On Demand Distance Vector) routing algorithm [11]. It allows the formation of a peer to peer mesh network with self healing routes, and the ability to put routing nodes to sleep [12]. XBee 802.15.4 Series One radios are used to interface the nodes with each other and implement the protocol. These radios provides a low latency as well as predictable communication timing according with [13]. In addition, the use of a mesh network for the system improve the network reliability given the fact that it will recompute another route in case of a node failure [14].

A. Digimesh Network Protocol

The DigiMesh network protocol is used to form a peer to peer mesh network for the transfer of sensor data to the sink node. In this case Digimesh was chosen over other XBee compatible protocols like ZigBee due to its ability of allowing routing nodes to be put to sleep. It also has an advange over ZigBee's star topology in that it treats every node as a routing node, taking away the single point of failure and forming a self healing mesh as we can see on Figure 6.

The XBee radios have two different inteface protocols: AT (transparent/command modes) and API. API-2 Mode was used over Transparent mode for various reasons. In order to set up a network with more than 3 radios API mode is needed. Since transparent mode

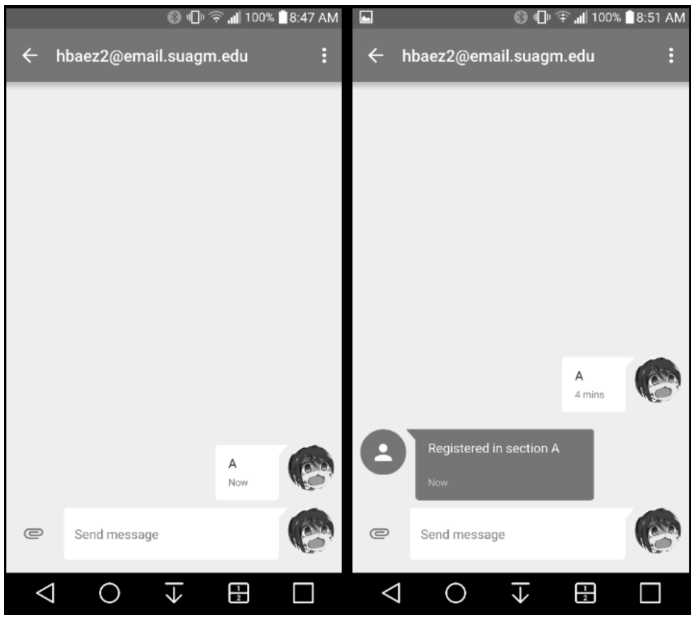

Fig. 5. User registration example.

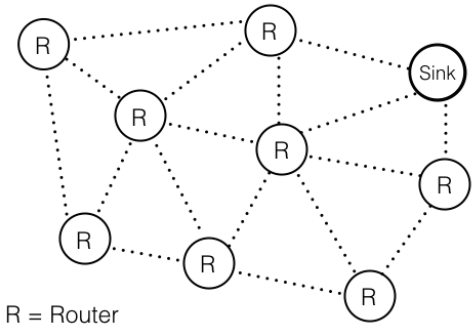

Fig. 6. Digimesh network topology.

is non packaged, meaning the data is sent in the exact format as it is input; with time, it causes the mixing of data when used with multiple radios. API mode is used to transmit highly structured data quickly, predictalbly and reliably [15].

Algorithm 1 Sensor Node Program

```
Initialize
while true do
    sensorVal ← (1023/analogRead) − 1
    res ← (refRes/sensorVal)
    payload ← res
    Create a Transmit Packet with updated payload
    Send TX Packet to the sink
    Sleep for 3 minutes
end while
```

To interface the Arduino microcontroller with the XBee radio and DigiMesh, a version of Andrew Rapp's XBee library for ZigBee modified to contain DigiMesh specific options for the construction of the Transmit Request was used. As we can see in algorithm 1; when

active, the Arduino collects the analog data from the level sensor. It then converts it from analog to a digital reading, finally getting a resistance value by dividing the obtained value by the reference resistor, in this case 560 ohms. That value is then encapsulated in the payload of a Transmit Packet with the sink node's destination address. Once the packet is sent, the DigiMesh firmware takes charge of routing the packet to the destination.

B. Power Management and Sleep

One of the main constraints for any wireless sensor network is power consumption. Since the sensor node will be battery powered, we have the need to maintain a viable charge/discharge rate. We use a cyclic approach to keep the sensors sleeping when communication is not required [16]. As stated before, one of the reasons for the use of DigiMesh in this network is the availability to have routing nodes sleeping. This is done via the Protocols radio firmware which gives us various sleep option divided into Asynchronous and Synchronous sleep. Synced sleep was better in this case due to the whole network being available on wake up, reducing the chances of the packet not having a route to the sink node. The whole network wakeup is then managed by a set Sleep coordinator, in this case the Sink node. The coordinator controls the timing of the other radios by sending a control message notifying if they are off schedule keeping the network in a synchronous state.

By having the radios control the sleep cycle, we avoid having to add additional clock circuitry to the Arduino node by just waking it up via pin interrupt on pin 2. One downside is that the radio must be in a powered state even if there is no data to report back to the sink [17]. After the radio wakes up, it will send a high output via its wakeup pin to the Arduinos interrupt pin (either pin 2 or 3). When the Arduino wakes up, it senses and processes the data and sends it to the sink node. After a small delay it goes into powered down sleep. The cycle chosen will have the network waking up for 2 seconds every 3 minutes.

V. Evaluation

Two key aspects of the system were evaluated: power consumption and battery life, and the packet delivery ratio based on node distance.

A. Power Consumption

In order to determine autonomous battery life, the sensor node's current consumption was measured in both active and sleeping states to determine the average hourly current consumption. Here are some key terms for this process:

- Ia - Current while awake
- Is - Current while sleeping
- Sa - Seconds spent awake
- F - Sensor cycle frecuency

The sensor node consumes 83.7mA while sensing (Ia) and 22.3mA when sleeping (Is). The cycle used was of 2 seconds of activity (Sa) and 3 minutes of sleep (F). With these values we could determine the hourly average in the following manner:

$$AvgCurrent = \frac{(Ia \cdot Sa + Is \cdot (F - Sa))}{F} \quad (1)$$

From this, an hourly average current consumption of 23.9mA was obtained. Using a 1200mAh battery, that gives us around 52.4 hours of autonomous operation.

B. Packet Delivery

In order to determine the largest distance we could have between the nodes, we ran packet delivery tests using the XCTU (i.e., An application for the configuration and testing of the XBee radios). We looked to get a high enough delivery rate as to avoid the packet retransmission. Since the network is in a synchronized sleep cycle, the message might not reach the destination as it might run the risk of the network going to sleep during its route. We can see the results of these tests on Figure 7.

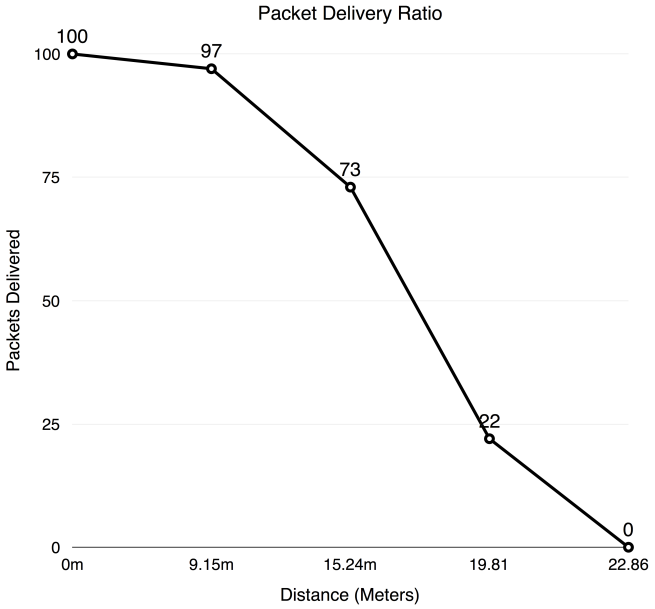

Fig. 7. Packet delivery test results.

Based on our testing for the worst case scenario, the distance between the nodes needed to have more than 95% of the packets delivered was 9.15 meters. This is heavily dependent on external factors such as line of sight and channel congestion. Since the radios on the DigiMesh firmware do not Channel Hop, a channel spectrum test should be performed at the location of choice. With line of sight, the XBee radios used should be able to have close to 30 meters of range in urban areas.

VI. Conclusion

This paper presents the implementation of the sensor network for a real-time flood alert system for parking lots. With the use of DigiMesh we were able to implement a mesh network with a synchronized sleep cycle. The ability to sleep gives ample time for the battery to recharge using the nodes solar panel. With this approach, we managed to get up to 52.4 hours of constant operation out of a 1200mAh battery. Better results could be obtained by building a more efficient voltage regulator for the Arduino since it is one of the main factors for the high current consumption while in sleep mode. Also a bare bones Arduino could be used, removing unnecessary LEDs and unused pins, this would allow us to lower the overall power consumption. Another way to improve on the system would be change to an event trigger instead of a cyclic approach, which would have the system in a shut down state and turn it on in the case of rain or water being detected. As for the range of the nodes, the radio's antenna should be located on the outside of the housing in order to improve the line of sight and reduce dropped packets.

VII. Acknowledgments

This research has been partially supported by the National Science Foundation under grant No. 1458928, An REU Site on Ubiquitous Sensing. Special thanks to the Ana G. Mendez University System.

References

[1] L. G. Jaimes, I. J. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *PerCom*, pp. 103–108, 2012.

[2] L. G. Jaimes, I. J. Vergara-Laurens, and A. Chakeri, "Spread, a crowd sensing incentive mechanism to acquire better representative samples," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom 2014 Workshops, Budapest, Hungary, March 24-28, 2014*, pp. 92–97, 2014.

[3] A. Amouri, L. G. Jaimes, R. Manthena, S. D. Morgera, and I. J. Vergara-Laurens, "A simple scheme for pseudo clustering algorithm for cross layer intrusion detection in manet," in *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1–6, IEEE, 2015.

[4] L. G. Jaimes, M. Llofriu, and A. Raij, "Preventer, a selection mechanism for just-in-time preventive interventions," *IEEE Trans. Affective Computing*, vol. 7, no. 3, pp. 243–257, 2016.

[5] "Nws flood safety home page," 2005.

[6] I. A. Aziz, I. A. Hamizan, N. S. Haron, and M. Mehat, "Cooperative flood detection using gsmd via sms," in *2008 International Symposium on Information Technology*, vol. 3, (Kuala Lumpur), pp. 1–7, IEEE, 2008.

[7] E. Kuantama, L. Setyawan, and J. Darma, "Early flood alerts using short message service (sms)," in *2012 International Conference on System Engineering and Technology (ICSET)*, (Bandung), pp. 1–5, IEEE, 09 2012.

[8] M. S. Baharum, R. A. Awang, and N. H. Baba, "Flood monitoring system (myfms)," in *2011 IEEE International Conference on System Engineering and Technology (ICSET)*, (Shah Alam), pp. 204–208, IEEE, 06 2011.

[9] P. B. Bedient, A. W. Holder, and B. E. Vieux, "A radar-based flood alert system (fas) designed for houston, texas," in *Ninth International Conference on Urban Drainage (9ICUD)*, 2002.

[10] L. S. T. Molina, E. W. Harmsen, and S. Cruz-Pol, "Flood alert system using rainfall forecast data in western puerto rico," in *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, (Melbourne), pp. 574–577, IEEE, 07 2013.

[11] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," tech. rep., 2003.

[12] "Wireless mesh networking white paper - zigbee vs. digimesh," tech. rep., 2015.

[13] V. Khedekar, "A review on xbee technology," *International Journal of Emerging Technologies in Engineering Research*, vol. 4, no. 4, 2016.

[14] V. Mayalarp, N. Limpaswadpaisarn, T. Poombansao, and S. Kittipiyakul, "Wireless mesh networking with xbee," ResearchGate, 10 2014.

[15] R. Faludi, *Building Wireless Sensor Networks - A Practical Guide to the ZigBee Networking Protocol*. United States: O'Reilly, 1st ed., 01 2012.

[16] S. Saxena, S. Mishra, A. Kumar, and S. Chauhan, "Efficient power utilization techniques for wireless sensor networks – a survey," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, pp. 905 – 925, 02 2011.

[17] "Designing a sleeping xbee sensor - digi developer," 2011.