Design and Development of a Satellite Based Water Resources Monitoring System

Harry Mafukidze
Department of Electrical and
Electronic Engineering
University of Stellenbosch
Email: 16736893@sun.ac.za

Riian Wolhuter
Department of Electrical and
Electronic Engineering
University of Stellenbosch
Email: wolhuter@sun.ac.za

***Abstract*—The Faculty of Forestry and Woodscience at Stellenbosch University has a requirement to monitor and record water resources and environmental data at remote sites, not within reach of any mobile services. The current solution consists of a standalone data logger based monitoring system. This system, however, is not ideal as it does not provide data in real time and has high costs and other problems servicing the particular sites. This paper presents an alternative satellite based WSN (Wireless Sensor Network) solution to this problem. The system described in this paper comprises a WSN with a three-part framework. The first part consists of sensor nodes monitoring rainfall, air temperature, air humidity, ambient light, wind speed, wind direction, soil temperature and soil moisture. Communication from these nodes to the central gateway is based in the wireless ISM band. The second part contains an Iridium satellite communications module, a gateway with a Linux based SBC (Single Board Computer) for collecting, storing and sending data from sensor nodes and forwarding such data via the SBD (Short Burst Data) satellite messaging service. The third part consists of the MS (Master Station), which is used for displaying sensory and site information. The system is solar powered and measurements indicate that the system meets an overall standby time of at least three days, as stated in the project requirements. It has been tested continuously in an actual deployment situation and is performing well. This new satellite based monitoring system is certainly an improvement and a reliable alternative to the one used up to now.**

I. Introduction

Efficient water resources monitoring is a major concern for the Department of Forestry. South Africa has a semi-arid climate and is a relatively dry country with a mean annual rainfall of 480mm. Of that value, only 9% is converted to river runoff [1]. The Western Cape, however, has a Mediterranean type climate which receives its rain in winter. As a result of climate change, available water resources, runoff and groundwater resources will be affected [1]. Information on rainfall, air temperature, air humidity, ambient light, wind speed, wind direction, soil temperature and soil moisture is of paramount importance to researchers, hence the need to put in place monitoring mechanisms. With the need for remote monitoring, focus has turned to satellite based monitoring methods. Taking this as the motivation of our project, we use a satellite link to transfer sensory information from the GS (Ground Station) to the MS. Features of our system include a Zigbee based WSN to send data to the GS. The WSN consists of four different sensor nodes which monitor site-specific data mentioned above. The GS comprises of a Linux based SBC which stores data and communicates with the IR modem. The MS is a web based application which displays the sensory data.

II. Background

Previous work on monitoring water resources has been discussed in a number of articles and some of them include systems which employ GSM and blue-tooth systems. CEDEC [2] developed a fresh water real-time monitoring system based on WSN and GSM. The system uses RF XBEE 802.15.4 which operates in the 2.4GHz spectrum, PIC16F886 MCU and the coordinator device is interfaced with a GSM/GPRS modem. [3] developed a Virtual Instrument for Radio Telemetry. In their work, they used a GP300 transceiver from Motorolla to transmit environmental conditions to a base station. The authors in [11] proposed a WSN for temperature monitoring. They developed the system to manage air conditioning systems at their institute. In their work, they developed a system which consisted of three main blocks: data acquisition, data collection and data display. Similarly, our system incorporates these blocks. As need for infrastructure-less remote monitoring increases, [4] developed an Oceanic drifter based on Iridium. In his work, drifting buoys provide information about surface currents, position and sea temperature. To provide power, [9] examined the possibilities of using solar energy to power the sensor networks. [10] implemented and utilised solar power in their WSN and found the solution to be feasible and reliable. Likewise, our project will harness power from the sun.
The objective of this research is to design and develop a dedicated network to monitor water resources in any remote area and deliver data in near real-time.

Previous solutions included setting up USB/Micro SD data loggers. The data would be collected manually after some time. Ideally, an automatic system which works in any remote area without the need for existing telecommunication infrastructure is required and this system should send data automatically to the master station (MS) without human intervention. This will help reduce the costs of site visits and it also covers areas which are not serviced by current GPRS/GSM techniques.

After introducing related work in monitoring water resources, this paper defines the requirements of the specific system in section III. Section IV presents the methodology. Section V discusses the project implementation and section VI presents the results of the experiments conducted. Finally, conclusions and future work are discussed in section VII.

 ISBN: 978-0-7972-1533-7

III. Requirements

As discussed in the previous section, a proposed solution should be capable of remote placement, measure a number of parameters reliably and be self sustaining energy wise for worst case winter scenarios. At least three days of autonomy is required. It should also be flexible in deployment to add additional measurement nodes. The network should manage disappearance or addition of nodes without any problem.
The design is affected by hardware constraints such as low power, and the components of each node should fit in a small to medium IP65 rated plastic enclosure. The size and the architecture of the network are determined by the sensing area and the parameters to be monitored.

IV. Methodology

To address the challenge at hand, available open-hardware and open-software tools for development were sought. The hardware/software platform selected for the sensor nodes was the Arduino platform [6] primarily due to its cost, availability, lower power consumption and its small form factor.

The XBee (Zigbee 2.4GHz, Series 2) modules were selected for setting up radio links and they are integrated by the Arduino hardware together with one or more sensors to form a sensor node. The ground station hosts one radio configured as a coordinator, Raspberry Pi SBC running LINUX OS which enable it to execute programs simultaneously and finally, the GS incorporates an Iridium modem to set up satellite links with the Iridium gateway. The whole system includes four different sensor nodes which monitor wind speed, wind direction, air temperature, air humidity, soil temperature, soil moisture, rainfall and ambient light.
By utilising data from these sensors over time, important parameters such as soil absorption and drainage rates could be determined. This information is used in water resources mapping and rainfall run-off studies.
The architecture and the components selected for this project was based on the requirements and constraints of the system laid out in section III.

TABLE I. Comparison of radio power consumption

	Rx current (mA)	Tx current (mA)	Link budget (dBm)	Output power / (frequency Band) (dBm)
Xbee	40	40	110	3 / 2.4GHz
XBee Pro - S2	45	295	119	18 / 2.4GHz
AT86RF230	16.0	17.0	104	3 / 2.4HGz
AT86RF212	9.0	18.0	120	10 / 700/800/900 MHz
MC13192	42	35		
CC2420	18.8	17.4		4 / 2.4GHz

Table I displays some of the available RF modules which could be used. Despite a higher current consumption, we chose the Series 2 Digi XBee radio due to its superior capabilities. Some of them include its ease of use, greater range, its ability to automatically form self-healing mesh networks and the module operates in the free unlicensed spectrum band (2.4GHz).

We selected off the shelf hardware components, shown in Table II and customised them to suit our project. The table also shows the hardware selected for sensor nodes and the GS.

TABLE II. Hardware requirements

Component	Product	Description	Specifications
	Arduino Uno	Arduino MCU Board	
	LEA-4P-T-MRT	GPS Module	
	Davis 6470	Soil Temp sensor, resistive	-40^0 – 65^{0}C
Sensor Node	Davis 6440	Soil moist sensor, resistive	0 – 200 cb
	Davis 6382	Air Temp/humid,SH1x sensor	1 – 100% RH
	Davis 7852	Rain gauge, digital	0.2mm
	Davis 6410	wind sensor, digital	1 – 322 km/h
Ground Station	XB24-Z7WIT-004	Xbee Radio	2.4GHz, 120m (LOS)
	Raspberry Pi	SBC	Linux OS
	XB24-Z7WIT-004	Xbee Radio	2.4GHz, 120m (LOS)
	IR	5.5 VDC Iridium modem	-50^0 – 85^{0}C
Master Station		Display sensor data	

The selected sensors shown in table II above cover wide ranges of the sense parameters and are therefore suitable for our application.

V. Implementation

The system is divided into two main parts which are data acquisition and data retrieval. The data acquisition covers the sensor node and the sink node whereas data retrieval covers the Master station. The project layout is shown in fig 1 below.

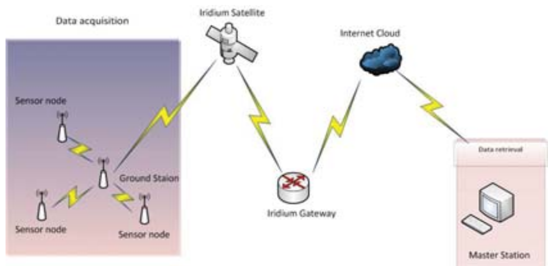

Fig. 1. Project layout.

As per Fig 1, the Zigbee network adopts a star topology. The coordinator forms the network and other routers are able to join. Data is sent to the coordinator and the coordinator forwards the data to the Iridium system. Data is then sent from the Iridium ground station to our client computer which provides a web user interface to the system.

A. Data acquisition

1) Sensor node: In total, four sensor nodes were built and each of them monitored a different environmental parameter. Each sensor node is equipped with 5 individual units as shown in fig 2. They are the sensing unit, processing unit, transceiver unit, power unit and the GPS unit [5]. A sensor node can communicate directly with the GS if it lies within each other's transmission range, or they can communicate though intermediate nodes.

The sensor samples the environment and generates an analog signal which is converted to a digital signal by the ADC. The micro-controller or processing unit processes the data, performs calculations and sends the data to the transceiver for transmission over a wireless link.

 ISBN: 978-0-7972-1533-7

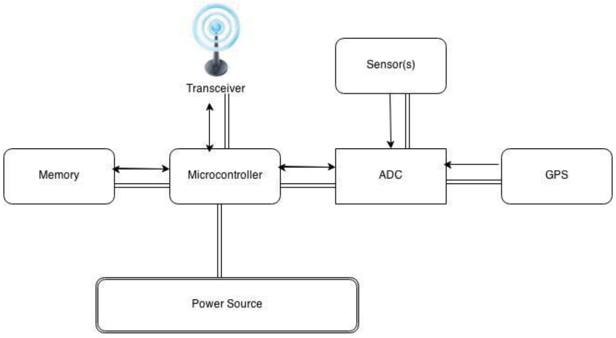

Fig. 2. Sensor node architecture

2) Arduino : Arduino [6] refers to a family of hardware which uses free, open source and cross platform software. An Arduino Uno, Fig 3 is based on the Atmega 328p [7], 8 bit microcontroller which clocks at 16MHz. It contains 128kB of program memory and 4kB of SRAM. Is has 28 pins of which 14 can be used for digital applications and 6 can be used as PWM outputs and 6 ADCs can be used for analog applications. It features a variety of interfaces, among them 1 USB port, 1 UART TTL (5V), I2C, power jack, ICSP header.

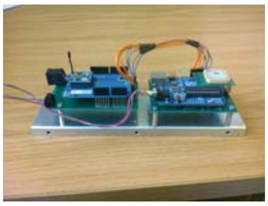

Fig. 3. Sensor node.

Fig. 4. Arduino IDE.

The setup of the sensor node is shown in Fig 3 above and it allows us to add custom hardware and to easily control power to the radio and sensors. The Arduino IDE, shown in Fig 4, allows the developer to program the Arduino hardware.

B. Ground Station

The GS (sink node) is composed of a Raspberry Pi SBC, XBee coordinator radio and an IR modem. The SBC has a GPIO which provides UART port and a USB port for connectivity with the modem.

1) Raspberry Pi: The Raspberry Pi [8], shown in Fig 5, is a small sized single board computer running Linux OS.

C. Data retrieval

The MS includes a PHP web page for data viewing and analysis. Data can be manually uploaded to the server or automatically by setting the mail settings. Fig 6 shows the architecture of the database. The diagram displays information about the system user, sensor information and ground station

Fig. 5. Raspberry Pi.

information

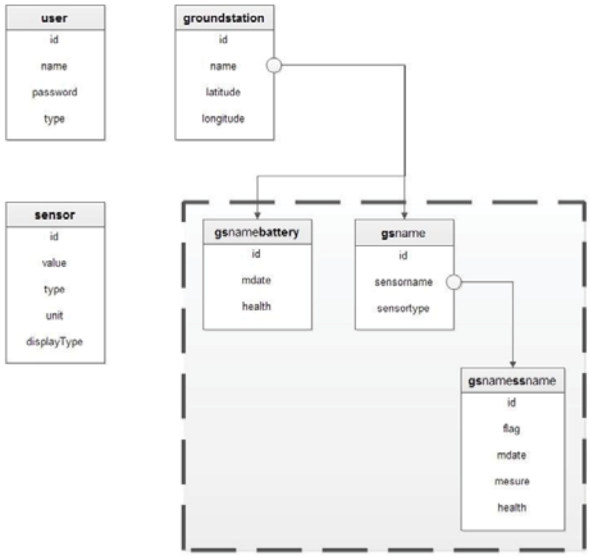

Fig. 6. Master station database

D. Software

The software for the sensor nodes is developed with the Arduino platform [6] and is shown in Fig 4. The code is divided in three parts, and shown in Fig 7.

- The main program initialises and establishes a connection with the GPS.
- The other part sets PIN values, PIN directions and calculates readings from sensors.
- The XBee library: the functions defined here enable the Arduino hardware to communicate with the radio.

The same follows for the GS. Software to control GS hardware was written in C++ and runs on a LINUX platform. These executables are in-turn called and run by shell scripts. Fig 8 presents the diagram of the GS software. After start up, the Raspberry Pi opens the UART port and listens for any data from the XBee radio, if there is data, the program reads the data and stores it on a file. The program which monitors system time enters an infinite loop to check the time. If 24 hours have elapsed, the power control script switches ON power to the Iridium modem and the program will commence to transfer data to the Iridium gateway.

 ISBN: 978-0-7972-1533-7

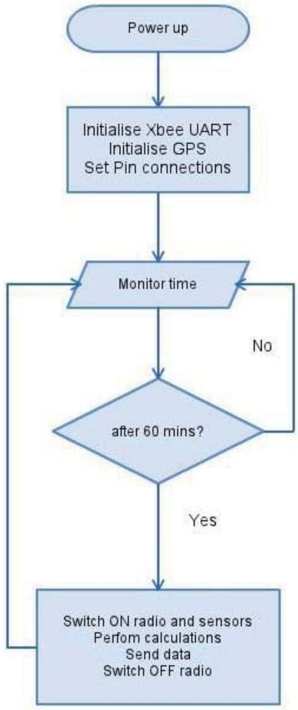

Fig. 7. Sensor node software flow diagram

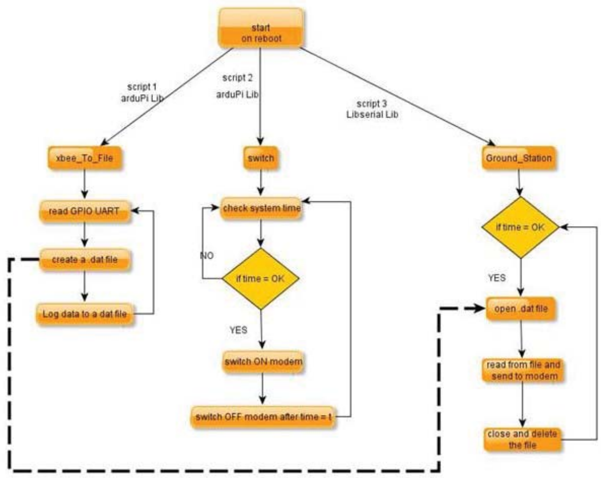

Fig. 8. Ground station software flow diagram

To enable the sensor node(s) to communicate with the sink node, our approach will utilise the star network topology since all the sensor nodes are within the communication range with the sink node. In this topology, all the sensor nodes direct their data to the coordinator which is connected to the GS. In some cases where one or more source nodes are out of range with the coordinator, the nodes use the Ad-hoc on demand Distance Vector (AODV) routing protocol [12] to perform route discoveries to create links thereby extending network coverage and range. If this mode is selected, the network will have to employ accurate time synchronisation algorithms.

VI. Results

We conducted field tests for each of the parameters being monitored. The sensor nodes collect data every 60 minutes and transmit the data to the GS. The GS in turn forwards the data to the MS every 24 hours. Sample results from an actual experiment of wind direction are shown in fig 9 below. The data was sent from the wind vane in a remote location using the developed system.

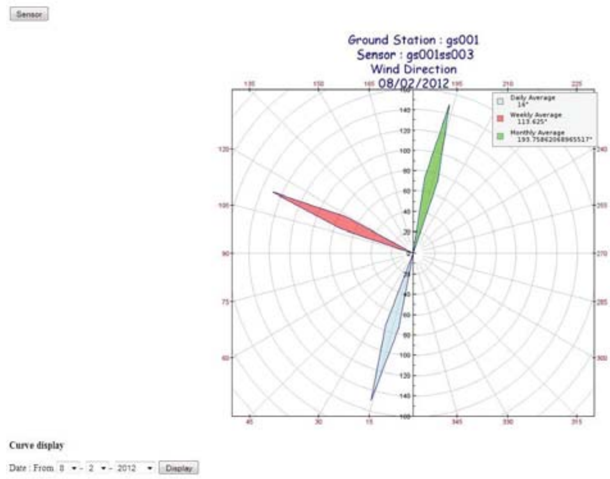

Fig. 9. Polar clock display

Fig 10 below shows the results of an experimental investigation of the power discharge characteristics of the batteries of the sensor nodes and the GS. All the sensor nodes follow an almost similar discharge curve due to the fact that the hardware configuration in the sensor nodes is almost the same. The GS, however, has a unique discharge curve due to a different hardware configuration. The rainfall sensor node achieves roughly 70 hrs of lifetime which satisfies the three day autonomy requirement.

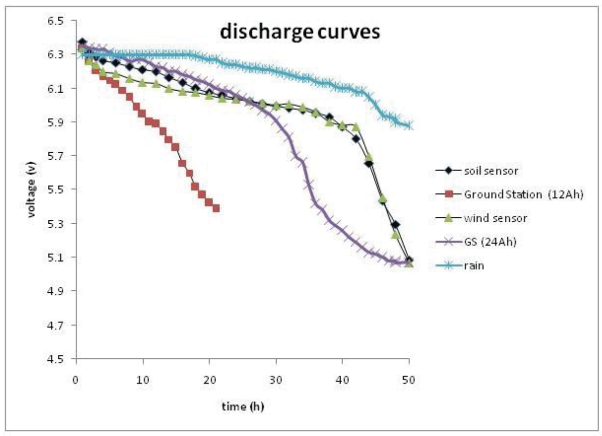

Fig. 10. Battery discharge curves

Fig 11 shows the results of the signal strength of the received packet as a function of the distance between the coordinator and the router. The results from this experiment enable us to characterise the link quality. This can be part

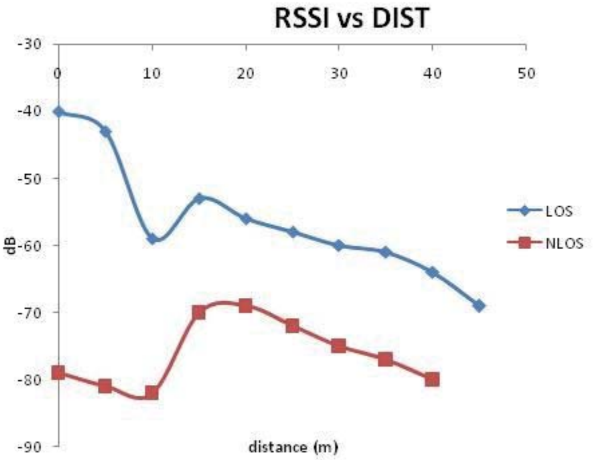

Fig. 11. RSSI vs distance

of future work. The experiment was conducted for Line-of Sight conditions and Non-Line of Sight conditions with only two nodes (one coordinator and one router). The RSSI value for both LOS and NLOS decreases linearly as the distance is increased.

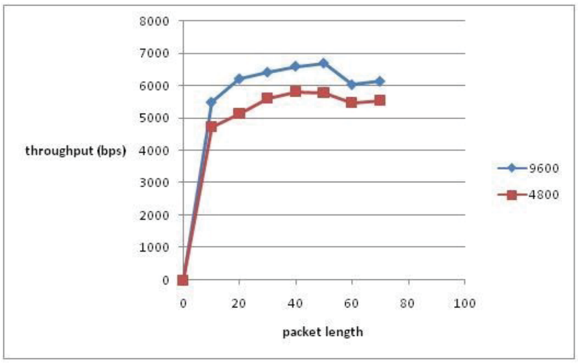

Fig. 12. Throughput for 4800bps and 9600bps

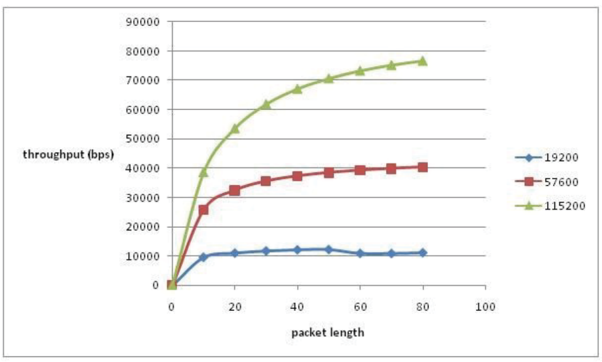

Fig. 13. Throughput for 19200bps, 57600bps and 115200bps

Figs 12 and 13 above present the throughput of the system for 4800bps, 9600bps, 19200bps, 57600bps and 115200bps as a function of packet size. The experiment was conducted with two nodes directly communicating with each other. The time taken to transmit a packet of n bytes is measured and the throughput is calculated as the ratio between number of bits received and the total transmission time. The experiment is repeated 3 times and the results are displayed in fig 12 and 13. It is noticed that as the speed of transmission is increased, the throughput also increases. We achieved a throughput of 12kbps at a speed of 19200bps and 85kbps was achieved at 115200bps. This could be attributed to a clean channel which was selected by the coordinator and this ensures that data is sent reliably and it is received without the loss of any packets, thereby satisfying the data reliability requirement set out in section III.

VII. Conclusion and Future Work

In this paper, we have developed a satellite based water resources monitoring system which can be deployed in any remote area and its sole purpose is to provide data in near real time. The project was tested and is functioning well. Overall, the completed system performed satisfactorily as intended and proved its suitability for the particular application.

In future, we hope to develop a system which uploads data automatically to the MS. In the current solution, a user downloads data from the email, sorts the data and uploads to the MS.

The power consumption of the system should also be able to benefit from additional development by reducing the solar panel size.

References

[1] Christine Colvin, David Le Maitre, Daleen lotter, 1. "Water Resources and Climate Change Case Study, South African Risk and Vulnerability Atas", 20:317–330.

[2] M. A. N2. "Fresh water real-time monitoring system based on wireless sensor network and GSM", *IEEE Conference on Open Systems (ICOS2011)*, pp 354-357, ISBN 978-1-61284-931-7

[3] S. B. Shamsuddin, M. D. Baba, D. K. Ghodgaonkar, "Design of a Virtual Instrument for Radio Telemetry Station", *Student Conference on Research and development*, pp 414-417, ISBN 0-7803-7565-3.

[4] Figueredo, A. J. and Wolf, P. S. A. (2009). "Development of an Oceanographic Drifter with Iridium Bi-Directional Communication Capability" *IEEE*, 2012.

[5] I. F. Akyildiz, E. Cayirci, S. Weilian, "A survey on sensor networks" *IEEE communications magazine, Vol 40, Issue 8*, pp 102-114, ISBN 0163-6804

[6] Enrique Ramos, "Arduino Basics" *Apress*, pp 1-22, ISBN: 978-1-4302-4168-3

[7] Atmel Coorporation, "Atmega 328P datasheet 2011"

[8] M. Richardson, S. Wallace, "Getting Started with Raspberry Pi", 2013, ISBN-13: 978-1449344214

[9] P. T. V Bhuvaneswari, R. Balakumar, V. Vaidehi, "Solar energy harvesting for wireless sensor networks"

[10] Thiemo Voigt, Hartmut Ritter, Jochen Schiller, "Utilizing solar power in WSN", 2013

[11] Vongsagon Boonsawat, Jurarat Ekchamanonta, Kulwadee Bumrunghet and Somsak Kittipiyakul, "XBee wireless sensor networks for temperature monitoring", 2005

[12] A. A Pirzada, M. Portmann, "High Performance AODV Routing Protocol for Hybrid Wireless Mesh Networks", *Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Vol 4*, pp 102-114, ISBN 978-1-4244-1024-8, 2007