Application of Smart City Technology in Aiken, South Carolina

Adam Prey and Jiannan Zhai

Institute for Sensing and Embedded Network Systems Engineering
Florida Atlantic University
Boca Raton, United States
e-mail: aprey2016@fau.edu, jzhai@fau.edu

***Abstract*— This paper examines how the city of Aiken, South Carolina implemented Internet of Thing (IoT) devices throughout the city to monitor storm water runoff that had been causing severe erosion in the nearby Hitchcock Woods. The role of the monitoring system is to provide feedback for city planners so that they can better understand the outcome of implemented Green Infrastructure to reduce water runoff. Previously, no monitoring system for water flow had been in place, leaving the results of past efforts to reduce runoff questionable, as data collection had been limited. Using a configurable IoT sensing platform, more data about the water runoff can now be collected. Our paper examines this IoT sensing platform and how it is implanted to make Aiken a smart city.**

Keywords- Internet of Things; IoT; Smart City; Erosion; MoteStack; Storm water management.

I. INTRODUCTION

Over the last 20 years, the cost and size of electronic devices have decreased significantly. During this same time, the capabilities of wireless networks dramatically improved as well. While the Internet of Things as a concept was thought of many years ago, having better wireless networks with smaller and cheaper electronics has finally made it possible. In the early generations of IoT, it was limited to city scale deployments due to the high cost of the technology and infrastructure to run properly. These devices started to become 'smart' by collecting data about their usage they began to automate themselves. With the decrease in the cost, IoT devices entered consumer markets and gained widespread use. Many uses have been found for these technologies, both on consumer and city-wide scales. By integrating smart devices into a city's infrastructure, important city data can be collected, enabling city planners and policy makers to better manage, plan, and protect a city's infrastructure, environment, and economy.

Taking advantage of smart city developments is the city of Aiken, South Carolina. Aiken is home to the Hitchcock Woods, one of the largest urban forests in the United States. Within the Hitchcock Woods there is an ephemeral stream called the Sand River, and as Aiken has grown so has the Sand River. The stream, once small enough that a person could easily step over, has grown to be over 15 feet deep and 70 feet wide over the past 40 years [5]. This rapid growth has been caused by erosion from Aiken's storm water runoff. With the city growing larger, more storm water runoff is sent through a 10-foot pipe at the head of the Sand River causing an increasing flow of water down the Sand River, worsening the erosion of the surrounding forest [10]. This erosion has resulted in extreme damage to the Hitchcock Woods and costs the city of millions [6] of dollars in attempts to control the damage.

An attempt to better manage the runoff was a Green Infrastructure project that took advantage of a sensing infrastructure previously proven as an effective tool for monitoring water conditions at the nearby Savannah River [7, 8]. This configurable end-to-end sensing infrastructure was adopted for the needs of Aiken to measure the effectiveness of their Green Infrastructure developments. It provided a reliable and scalable solution to monitor the environmental conditions of Aiken with real-time data collection, transmission and processing of the data sent directly to environmental scientists, policy makers and engineers of Aiken [12]. The sensor network was configured to collect data on the soil, and water flow throughout the city and watershed [9]. In this paper, we present an analysis of the technology used in Aiken's development as a smart city. In Section 2 is the background detailing other smart city implementations. In Section 3, we review the system architecture's sensing, networking and middleware layers. In Section 4, we conclude our work.

II. BACKGROUND

Over the past few years, numerous cities have begun to integrate sensing networks into their infrastructure. The city of Aarhus, Denmark began a smart city initiative in 2010 with the goal of developing new sensing and networking technologies to better manage the city's water, waste removal and heating services [1]. Aarhus worked with private and public sectors while encouraging volunteers to participate by having the data collected available to the public. With the data being public, several competitions focused on the sensing, networking, and data processing technologies became popular and helped grow the smart city initiative. Through these initiatives and engaging the public with the development of its smart technologies, Aarhus found effective and meaningful ways to improve upon the environmental impact of a large city while enhancing the quality of life of the people living within the city.

The city Kayseri, located in Turkey, served as a testing ground for a smart waste management system [4]. By attaching sensor systems to dumpsters and other waste collection boxes researchers could collect data about waste production. Using this data, the researchers developed a software system that planned optimized routes for garbage collectors and sent these routes directly to the garbage collectors via a cellular network. The implementation of this system reduced the amount of pollution Kayseri produced as garbage collection ran more efficiently and less frequently. The new system also provided the city with 30% cost savings of running garbage collection by reducing the amount of upkeep for the garbage trucks [4].

Smart systems are now used in agriculture as well. These systems commonly use soil sensors to provide farmers valuable data about soil conditions and crop health. In Jojoba, Israel, soil sensors are used to collect real-time data in orchards and other farmlands, allowing farmers to develop irrigation plans for their crops. Machine learning techniques are also applied to the data collected from the soil sensors to build better prediction models for farm irrigation [2]. Soil sensors in agriculture need to be buried deep enough to avoid damage from plowing machines while still having access to power and a means to communicate with the surface and other sensors.

For Aiken's application, a more robust networking solution was put in place by having cellular and Wi-Fi access points than using only one or the other. The Aiken system is also easy to configure allowing for rapid deployment and servicing of the system. The sensing layer is also more sophisticated in that a single node has access to multiple sensors and wirelessly transmits collected data.

III. SYSTEM ARCHITECTURE

As shown in Figure 1, the sensing infrastructure consists of four layers: (i) The sensing layer that uses a wireless sensing platform called MoteStack [3]. (ii) The networking layer, where information collected from the sensing layer is sent to a computing center for data processing. (iii) At the computing center the data is processed through middleware to automate the validation, storage and dissemination of the collected data. (iv) The fourth layer is the front-end used to present the information to end-users.

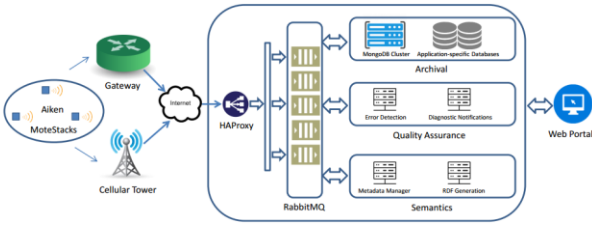


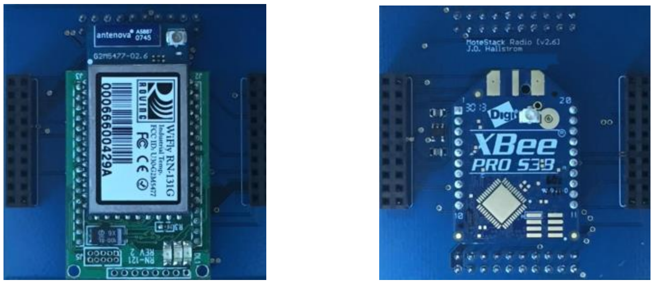
Figure 1. Layered Sensing Infrastructure

A. *Sensing Layer*

The stackable MoteStack sensing platform allows it to be configurable and scalable for many applications. MoteStacks are built from multiple layers of interchangeable circuit boards to perform different tasks. This interchangeability gives the macroscope a high degree of flexibility in its applications. To ensure proper functionality and ease of deployment, the MoteStack includes a Software Stack and purpose built MoteStack Compilation Tool.

(a) Control Board (b) I/O Board

(c) NVSRAM Board (d) Cellular Board



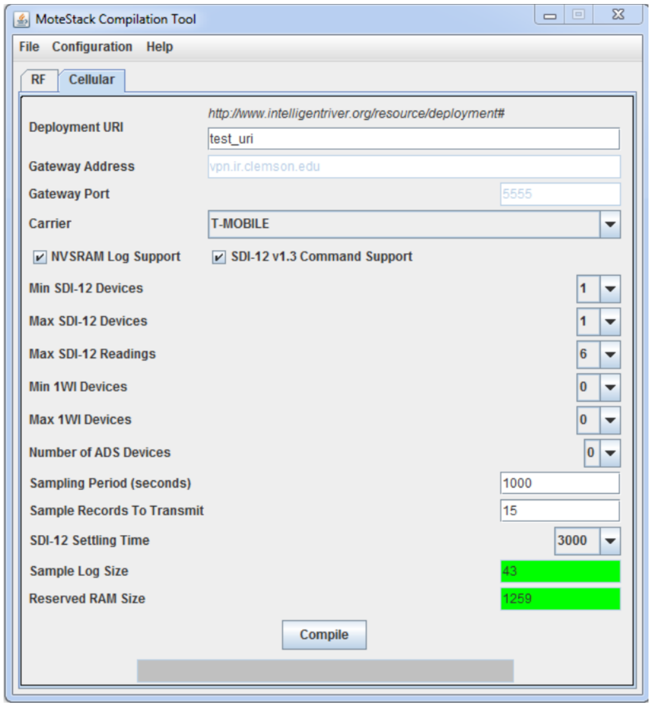
(e) Wi-Fi Board (f) Xbee Board

Figure 2. MoteStack Boards

Figure 2 (a) shows the Atmel ATmega644 microcontroller used in the MoteStack for sensor readings, radio control, and task scheduling. Figure 2(b) shows the I/O and SDI-12 board that handles communication to the sensors over various protocols with on board LED's to indicate system status. The MoteStack includes three RF boards, a XBee RF module Figure 2(f), a Wi-Fi board Figure 2(e), and a cellular board Figure 2 (d) to allow an internet connection at designated base stations. To support non-volatile data logging an NVSRAM board is included in the MoteStack shown in Figure 2(c).

A software driver is developed for each hardware module and can be included in the software as needed. Due to the nature of the embedded systems -- batteries powered and resource constrained, the software designed for the MoteStack focuses on efficiency and reliability. The core of the MoteStack software is an efficient task scheduler. At startup, the drivers for the installed board are initialized and the designed tasks are passed to the task scheduler with parameters used to specify the times when the tasks will be executed. To accurately schedule the tasks, an external crystal with corresponding driver is used to keep track of time.

In the early phase of the project, each MoteStack was configured and programmed by the engineering team and shipped to the field team for deployment. To avoid spending unnecessary time on MoteStack reprogramming and shipment, a tool called MoteStack Compile Tool (MSCT) was developed, allowing the field team to reconfigure and reprogram the MoteStack without the help from the engineering team. The MSCT consists of two components: (i) a server application used to generate executable images for the MoteStacks based on the configurations entered by the end user; and (ii) a desktop application, shown in Figure 3 used to modify MoteStack configurations, download executable images and reprogram the MoteStacks. Once the Compile button on the desktop application Graphical User Interface (GUI) is pressed, the new configurations are sent to the server application that generates a new executable image based on the received configurations. Then, the image is sent to the desktop application that reprograms the connected MoteStack with the downloaded image.



Deployment URI: http://www.intelligentriver.org/resource/deployment# test_uri
Gateway Address: vpn.ir.clemson.edu
Gateway Port: 5555
Carrier: T-MOBILE
☑ NVSRAM Log Support
☑ SDI-12 v1.3 Command Support
Min SDI-12 Devices: 1
Max SDI-12 Devices: 1
Max SDI-12 Readings: 6
Min 1WI Devices: 0
Max 1WI Devices: 0
Number of ADS Devices: 0
Sampling Period (seconds): 1000
Sample Records To Transmit: 15
SDI-12 Settling Time: 3000
Sample Log Size: 43
Reserved RAM Size: 1259

Figure 3. MoteStack Compilation Tool GUI

The MoteStacks are assigned a Unique Resource Identifier (URI) to be used by middleware before deployment. The middleware uses the URI assign the correct metadata for each MoteStack.

The MoteStacks deployed in the city of Aiken uses (i) soil sensors to collect soil temperature, conductivity, and water content at depths of 6, 12 and 18 inches, and (ii) doppler flow sensors to monitor water quality and flow as it moved through the Aiken watershed. These sensors were selected as they could provide city planners a clearer picture of where water off is flowing too and where it is coming from. A total of 20 sensor locations where deployed throughout the major areas of water runoff within Aiken and the Hitchcock Woods. With the developed software tools and configurable design of the MoteStack the field team was able to easily deploy MoteStacks in Aiken.

B. *Networking Layer*

To send messages from the MoteStacks to the back-end, already existing Wi-Fi and cellular infrastructures are used along with Xbee and Wi-Fi meshes for communications. Custom gateways are used to connect the MoteStacks to the Internet. The gateways service all the messages, network maintenance, and monitors system status. In the mesh network, the gateways provide the MoteStacks with base stations to the Internet. The messages collected from the MoteStacks at a gateway are delivered to the back-end middleware. Several gateways include a cellular connection to as a backup connection to the Internet in case of an Internet outage, improving the robustness of the networking layer. A Cellular Signal Tester, shown in Figure 4, was built to test the cellular signal strength of different carriers on each deployment site, based on the signal strength, a carrier is chosen for the deployment.

Figure 4. Cellular Signal Tester

ISBN: 978-1-61208-659-0

C. Middleware Processing

The middleware stack is a cloud-based virtual machine network hosted on a private multi-server computer infrastructure. This network allows the middleware to be configured, scaled and processes data through three modules, a messaging module, data storage module, and processing module.

A load balancer is used with RabbitMQ for the Advanced Message Queuing Protocol (AMQP) standard. Information from the MoteStacks pass through the load balancer to be distributed to a RabbitMQ cluster consisting of several RabbitMQ nodes. The messages are then sent to a diagnostic or observation queue to be processed. When a message is successfully received an acknowledgement is sent to the MoteStack, if no acknowledgement is received by the MoteStack, it buffers and sends the message again.

The data storage module is dispersed through several virtual machines on different physical to provide improved system scalability and reliability. The data is stored in MongoDB instances with a duplicate instance for storing metadata of the module. A single instance serves as an interface to the storage module.

The processing module consists of consumer programs with access to the diagnostics and observation queues for message processing. Messages are archived in the data storage module to create a log for all received data. Diagnostics messages are archived and sent to a notification program that checks timestamps of the diagnostic messages to verify that the MoteStacks are working normally.

Afterwards, messages from the observation queue are processed. The metadata from the MoteStacks is used to verify received data matches the parameters outlined in the specific MoteStack's metadata. Finally, the metadata is used for processing raw data into more human-readable data and sent back to the messaging module for storage in application specific databases.

IV. CONCLUSION

While the results of Aiken's Green Infrastructure plan show a consistent reduction in the calculated runoff, there was no statistically relevant reduction for the watershed [8, 11]. However, the MoteStacks and their sensing infrastructure allowed for consistent, meaningful data collection before and after the green infrastructure plan was completed. The infrastructure proved to be more than adequate for the demands placed on the system during deployment. The system was reliable and built in redundancies such as having both a Wi-Fi and cellular connection ensured that little data would be lost. This data provided key measurements that evaluated the effectiveness of the changes made by Aiken. Using these smart technologies, the city of Aiken, SC could effectively design and implement a smart city component for itself to aid in the management and control of the city and their water runoff.

REFERENCES

[1] C. Snow, D. Håkonsson, and B. A. Obel, "A Smart City is a Collaborative Community: Lessons From Smart Aarhus" California Management Review, vol. 59, pp. 92-108, Nov. 2016, doi: 10.1177/0008125616683954.

[2] A. Goldstein et al., "Applying machine learning on sensor data for irrigation recommendations: revealing the agronomist's tacit knowledge." Precision Agriculture, vol. 19, pp. 421-444, 2017, doi: 10.1007/s11119-017-9527-4.

[3] J. Zhai et al., "Harnessing the Flow of Ecological Data Across Networks, Middleware, and Applications" IEEE 3rd World Forum on Internet of Things(WF-IoT) pp. 129-134, 2016, doi: 10.1109/WF-IoT.2016.7845411

[4] Z. Oralhan, B. Oralhan, and Y. Yiğit, "Smart City Application: Internet of Things (IoT) Technologies Based Smart Waste Collection Using Data Mining Approach and Ant Colony Optimization." International Arab Journal Of Information Technology, Vol 14, pp. 423-427, Jan. 2017

[5] P. Kent, "Of Seeds and The River", Glimpse, Spring 2012, pp. 14-22Y.

[6] Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] "Sand River Headwaters Green Infrastructure Project" February 2013. Available from http://media.clemson.edu/public/restoration/sand%20river/agi_finalreport_022113-web.pdf retrieved: Jul. 2018

[8] G. W. Eidson, et al., "Sand River Headwaters Green Infrastructure Project, City of Aiken, South Carolina: A collaborative Team Approach to Implementing Green Infrastructure Practices." 2010 South Carolina Water Resources Conference, Oct. 13, 2010.

[9] C. Post et al., "Monitoring spatial and temporal variation of dissolved oxygen and water temperature in the Savannah River using a sensor network." Environmental Monitoring and Assesment May 2018, doi: 10.1007/s10661-018-6646-y.

[10] S. Esswein, J. O. Hallstrom, C. Post, G. Eidson, and D. White, "Augmenting Hydrologic Information Systems with Streaming Water Resource Data" South Carolina Water Resources Conference, Oct 2010.

[11] G. Eidson et al., "The South Carolina Digital Watershed." International Journal of Distrubted Sensor Networks, Jul. 2010, doi: 10.1155/2010/970868

[12] D. White et al., "The Intelligent River©: Implementation of Sensor Web Enablement technologies across three tiers of system architecture: Fabric, middleware, and application" Conference: Collaborative Technologies and Systems, Jun. 2010, doi: 10.1109/CTS.2010.5478493