A Raspberry Pi Mesh Sensor Network for Portable Perimeter Security

Cullen Johnson
Dept. of EE and CS
U.S. Military Academy
West Point, USA
cullen.d.johnson.mil@mail.mil

Brian Curtin
Dept. of EE and CS
U.S. Military Academy
West Point, USA
brian.h.curtin.mil@mail.mil

Nikhil Shyamkumar
Dept. of EE and CS
U.S. Military Academy
West Point, USA
nikhil.shyamkumar.mil@mail.mil

Rachelle David
Dept. of EE and CS
U.S. Military Academy
West Point, USA
rachelle.h.david.mil@mail.mil

Emmet Dunham
Dept. of EE and CS
U.S. Military Academy
West Point, USA
emmet.d.dunham.mil@mail.mil

Preston C. Haney
Army Combat Capabilities Development
U.S. Army Futures Command
Picatinny Arsenal, NJ
preston.c.haney.civ@mail.mil

Harry L. Moore, Jr.
Army Combat Capabilities Development
U.S. Army Futures Command
Picatinny Arsenal, NJ
harry.l.moore.civ@mail.mil

Thomas A. Babbitt
Dept. of Electrical Engineering and Computer Science
U.S. Military Academy
West Point, USA
thomas.babbitt@westpoint.edu

Suzanne J. Matthews
Dept. of Electrical Engineering and Computer Science
U.S. Military Academy
West Point, USA
suzanne.matthews@westpoint.edu

***Abstract*—Wireless sensor networks play an important role for perimeter monitoring in remote environments. While commercial wireless sensor networks for providing audio-visual monitoring exist, they are often expensive to deploy. In this paper, we describe and implement a wireless mesh network consisting of inexpensive battery-operated Raspberry Pi nodes. The choice of the Raspberry Pi enables the construction of cost-effective sensor nodes that are extendable and expendable. We conduct a series of test to illustrate the efficacy of our network in a building monitoring use case. Our nodes can be built for as little as 49.00 per node and is capable of node-to-node transmission of up to 50 feet. Custom sleep states enable battery life to last 14 hours on 4 AA batteries. Our results support our thesis that an all-Pi mesh sensor is capable of providing portable perimeter security.**

Index Terms*—*wireless sensor networks, ad-hoc networking, mesh networking, Raspberry Pi, battery life, perimeter security

I. INTRODUCTION

Wireless sensor networks (WSNs) [1] consist of a series of lightweight nodes that are placed in an environment to collect data. This creates an ad-hoc wireless, or infrastructureless, network containing no set routers [2]. A number of use cases for wireless sensor networks exist for environmental monitoring [3], [4] and military applications [5]–[7], including for indoor, urban, and remote locations.

A popular application for WSNs is to monitor an area for activity. For example, scientists use WSNs to aid in conservation efforts. Researchers have used (or proposed using) wireless sensor networks for detecting elephant crossings at railroad tracks [8] and to detect the movements and social interaction of Badgers [9]. Many WSN applications focus on one or two types of passive sensing (e.g. temperature, noise, motion, humidity) to keep data transmission requirements low. Despite higher bandwidth requirements, audio-visual data offers a richer representation of the environment. In the context of wildlife monitoring, nodes that transmit audio or video enable scientists to directly and discretely monitor wildlife [10], [11].

Military personnel require the same capabilities to quickly secure a defensive perimeter or to secure a building [6], [7]. WSN sensor nodes that provide audio-visual data help eliminate "dead-space", or areas that cannot easily be observed and may allow an adversary to move unobserved. A sensor network capable of covering some of these vulnerable avenues of approach provides additional security without requiring manpower. To provide perimeter monitoring for a patrol base, which tends to rest for short periods of time, a WSN need only be in operation in 12 to 24 hour stretches.

Both of the aforementioned communities value the ability to quickly establish a robust network of sensor nodes capable of audio-visual capture. In remote environments, nodes must be battery-operated and cannot rely on cellular communication. While commercial solutions such as networked trail cameras exist, they are large and cost several hundreds of dollars per node. For applications where portability is a must, the size and weight of such nodes make them less desirable for use.

In this paper, we describe, implement and test a wireless

Funding is provided by U.S. Army Futures Command, CCDC Armaments

sensor network made up entirely of Raspberry Pi single board computers. Unlike many WSNs that use microcontroller-based technology for their sensor nodes, our network uses Raspberry Pi Zero HWs (0HW) as the base of our sensor node. The memory, processing capabilities and low cost of the Raspberry Pi 0HW (512 MB of RAM, 1.0 GHz CPU, $14.00 cost) enable the integration of audiovisual and motion sensors into a single unit at very low cost. Prior work [12] discusses a prototype all-Pi wireless sensor network, but only supports two sensor nodes and direct node-to-base communication. In contrast, our system supports up to ten sensor nodes, uses a wireless mesh network for communication, and implements a number of sleep states to extend battery life by 320 percent. Sensor nodes are also smaller and more inexpensive (< $65.00 per node) than the previously implemented system.

Our results clearly demonstrate the feasibility of an all-Pi WSN for portable perimeter security. While additional work is needed to make the system ready for field use, we believe our work is an important step in demonstrating the utility of single board computers for such applications. Furthermore, we expect the system to only get more inexpensive, extendable, and powerful in the coming years, as the improvements in individual components enable our sensor nodes to be improved incrementally and with little cost.

The rest of the paper is organized as follows. Section II discusses related work. Section III describes the design of our mesh network. Section IV discusses our experimental results. Finally, we conclude in Section V.

II. RELATED WORK

Wireless sensor networks are used extensively for perimeter security applications. The capabilities of an individual WSN are dependent on many factors, including the design of the communication architecture and the quality of the sensors. Furthermore, sensor nodes need to be inexpensive, small, inconspicuous and have long battery lifetime [5].

A. *Sensor Node Design*

Trail cameras are perhaps the most well-known commercial solutions that transfer audio and video. Table I shows the top rated network trail cameras for 2019 from [13] with prices as listed through Amazon [14]. There are two classes of trail camera models currently available on the market. The first are cellular-enabled and requires access to a specific cellular network. The more expensive models can create a network between like model variants and require a separate base station, or use a camera as a base station. In Table I, the cameras capable of creating a mesh network between cameras are denoted with (+). Cellular-enabled cameras can be accessed individually through the cellular network. While each node can last months on battery supply, nodes either require a large battery or a designated power source.

TABLE I: Trail Camera Cost [14]

Vendor	Dimensions (in)	Weight	Battery	Cost
Cuddleback Dual+	7 x 3 x 3.5	≈2 lbs	4 D	$330
Spartan-HD GoCam*	6 x 5 x 3.25	1.3 lbs	12 AA	$300
Stealth Cam GXW**	4 x 2.5 x 5.5	1.8 lbs	12 AA	$524
Bigfoot 3G*	12 x 4.5 x 2.5	2 lbs	12 AA	$200
Average [13]	7.25 x 3.25 x 3.7	1.78 lbs	12 AA	$273

+ Mesh Enabled
* Cell Enabled
** Cell Enabled, Two Pack

The nodes themselves are $1-2$ pounds each, and are roughly the size of a small shoe box. The average cost of the listed trail cameras is $273.00. There are numerous models of web enabled trail cameras to choose from with a prices ranging from $175 to $350 per camera. The ones in the lower end of the spectrum are cell enabled; the variants that can create a mesh type network are generally on the upper end of the price range. However, cameras that rely on cellular communication will not work in cases of poor cellular service or if the system is taken outside of the purchase country, as is typically the case with use in remote environments.

To maximize battery life, much prior research relies on small microcontrollers such as the Mica platform [3] to act as sensor nodes. More contemporary systems use devices such as an Arduino Uno R3 and XBee [15], [16]. Most of these systems only transfer small amounts of data associated with one or two environmental conditions or variables from the sensor(s) to a base station.

A key limitation of microcontrollers is their limited memory and processing power, which limits the amount of data pre-processing that can be performed at-node. To mitigate this issue, several researchers have begun to use inexpensive single board computers like the Raspberry Pi [17] for data processing in wireless sensor networks [12], [18]–[21]. Researchers laud the Raspberry Pi's potential as a sensor node in a WSN [19], citing its better memory and processing capabilities over microcontrollers and ease of integration with existing hardware.

In most prior work, while a Raspberry Pi serves as the master or gateway node in the WSN, the sensor nodes themselves are are microcontrollers such as the Arduino Uno R3. Due to power consumption, the Raspbery Pi is not extensively used as a sensor node itself. In many cases, the Pis are connected to a sustainable power source like the the grid [20] or use a rechargeable battery connected to a solar cell [21]. There is extremely limited prior work on WSNs based on battery-operated Raspberry Pi sensor nodes. For example, while researchers proposed [12] a wireless sensor network composed entirely of Raspberry Pis, the implemented prototype was very limited, consisting of only two sensor nodes and no mesh network. Our work improves on this prior work, supporting up to ten Raspberry Pi sensor nodes and employing a mesh network.

B. *WSN Routing*

While there are numerous protocols for WSNs [1], [22], we focus on those that are most appropriate for deployment in a remote environment. There are many ways to rank the quality of WSNs (e.g. energy conservation [23], security [24], ability to handle delay [25], use underwater [26], etc.); we limit our

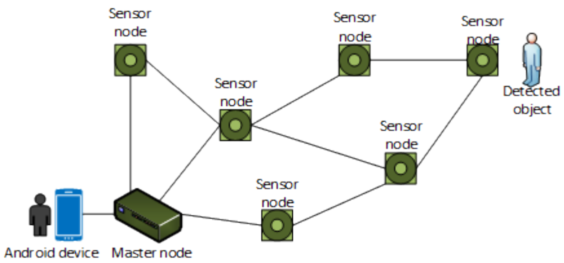


Fig. 1: Mesh network overview.

Fig. 2: Assembled sensor node.

discussion to those protocols that maintain forwarding tables and those that use some other means to ensure transmission.

One of the most popular choices for wireless sensor network architecture is the Delay Tolerant Network (DTN) [27], [28]. A DTN does not maintain routing tables and is capable of operation even when node status is unreliable and the topography of the network rapidly changes as nodes enter and drop out of the network [29]. DTNs even work when sections of the network become unreachable for a time. However, a key drawback to DTNs is a need to maintain redundant nodes to ensure coverage and maintain network connectivity. Most DTNs expect that nodes will fail (and therefore include extra sensors nodes), resulting in a larger weight, power consumption and cost.

An example of a protocol that does maintain routing information is high-speed multimedia mesh (HSMM) [30]. It was originally a design philosophy for use with ham radios. HSMM creates a mesh network that utilizes the frequency range and power available to licensed ham radio operators to create a data network capable of transmitting audio, video, and images. The hsmm-pi [31] mesh is a modification of the original design designated for use on a Raspberry Pi through the unlicensed frequency ranges associated with 802.11 Wi-Fi. We employ hsmm-pi since it enables us to minimize the number of required nodes.

C. Battery Conservation

Extending battery life of sensor nodes to maximize the "duration of usage" [5] is a major goal in WSN design for remote environments. Battery life conservation is especially critical to WSNs consisting of single board computers, due to their larger power requirements. Common approaches to conserve battery life include:

1) Saturating an area with enough sensor nodes such that a subset of the nodes can go into a near 100% sleep mode. In this mode, the node is completely shut down with the exception of a timer that indicates when the node is to turn back on. The timer can be a randomly selected time or pre-programmed [32], [33]. This process is often referred to as "duty cycling".
2) Using a DTN routing protocol that does not maintain routing tables and makes a best effort at delivery. These protocols are also resistant to temporary isolation of parts of the network [25]. This protocol is normally used in conjunction with "duty cycling".
3) Maintaining minimal routing function but put all other board functions into a minimal power sleep mode and only come to an active state when required. This technique is more of a data driven approach [34].

Each of the listed approaches has its advantages and disadvantages. While duty cycling or using a DTN routing protocol can extend battery life, the resulting networks require a significantly larger number of nodes to maintain adequate coverage, resulting in additional total power consumption, mass, and cost. While the data-driven approach prolongs battery life without requiring additional nodes, the battery can deplete more quickly. This can be problematic in environments where swapping out batteries is not always feasible. In our design, we choose to use a data driven approach to ensure there are no isolated sections of the network and to minimize the amount of equipment that is required to be carried to remote areas. We also develop a series of novel sleep states to help reduce the power consumption of our nodes.

III. SYSTEM DESIGN

Figure 1 provides an overview of the Raspberry Pi high-speed multimedia mesh (HSMM) network. We adapt the hsmm-pi [31] software to the Raspberry Pi 0HW [35], enabling our Raspberry Pi 0HW sensor nodes to also be mesh nodes. The master node is a Raspberry Pi 3B that communicates wirelessly to an attached Android smartphone. All the sensor nodes connect to the master node via the internal, private HSMM wireless network. Each sensor node consists of a Raspberry Pi 0HW, a Pi camera, microphone and passive infrared (PIR) sensor. To facilitate rapid assembly, a PCB hardware attached on top (HAT) is constructed that contains all sensors. An assembled sensor node is shown in Figure 2. The use of a mesh network enables a sensor node deployed on the periphery of the network to, upon detecting movement, transfer data to the master node through intermediary sensor nodes, despite itself being out of immediate range of the master. Custom software enables sensor nodes to capture audio-visual data when motion is detected, and automatically transfer data to the user.

A key advantage of our sensor nodes over prior versions is the presence of "sleep states" that significantly conserve

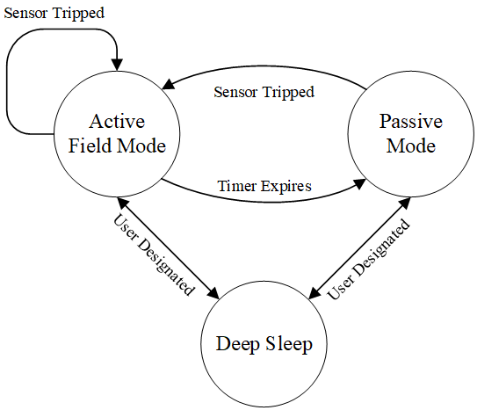


Fig. 3: Sleep state transitions.

battery life. Unlike other mesh networks that will cause an entire node to shut off, our sleep states selectively turn off individual ports on a Raspberry Pi until they are "woken up" by movement detected by the Pi's PIR sensor. Wireless functionality is left on regardless of sleep state, enabling "sleeping" nodes to transmit information to each other.

Figure 3 depicts the sleep state transitions implemented (excluding the test state). Raspberry Pi settings and transition conditions are as follows:

- **Active Field Mode:** This is the default state. Only those ports necessary to run connected devices are enabled. The microphone, camera, and PIR sensor are on. Upon entering this state, the camera takes a picture and a timer is initiated or reset. If no motion is detected in the next 20 minutes the Raspberry Pi transitions to the *Passive Mode* state.
- **Passive Mode:** Upon moving to this state, all sensors (except the PIR sensor) are disabled, and the clock speed is decreased. When the PIR sensor is tripped, the Raspberry Pi transitions to the *Active Field Mode* state.
- **Deep Sleep Mode**: In this state, all sensors and ports are turned off and the clock speed is reduced. This state is set by the user. The user also determines when to exit this state.
- **Test Mode:** All functions and ports on the Raspberry Pi are turned on. This is the default Raspberry Pi operation. This state is set and exited by the user, and exists primarily for testing and debugging.

The sensor nodes cost roughly $65.00 to construct with widely available commercial parts (see Table II), is roughly half the size of a credit card, and weighs roughly 220 grams. This is significantly smaller and cheaper than available battery-operated commercial sensor nodes (see Table I). If only visual data is needed, the sensor node can be built for $49.00.

IV. EXPERIMENTAL RESULTS

We perform a series of experiments to strength-test the sensor nodes and sensor network. We were chiefly interested in observing the level of battery life and range of nodes obtainable with the current design. Tests include assessing the range of the PIR sensor, the life of the battery, and the communication distance of nodes in the sensor network. The current set of tests focus on uses cases in an indoor environment, due to the constraints provided by walls. We note that our nodes and network have a much wider range in an outdoor environment with no obstacles between the nodes.

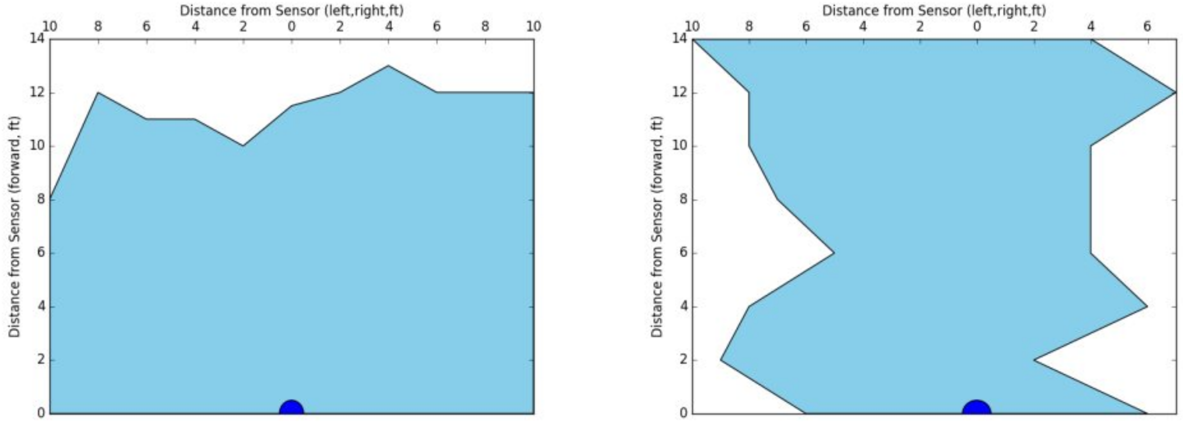
TABLE II: Raspberry Pi Sensor Cost

Part	Cost
Raspberry Pi Zero HW	$14.00
Raspberry Pi Camera	$9.95
Real-Time Clock	$4.95
Passive Infrared Sensor	$12.95
Microphone	$8.00
Audio Jack	$0.95
SD card	$5.00
4 AA batteries	$2.00
PCB board	$5.00
TOTAL	$62.80

A. *PIR sensor tests*

Our first set of experiments involved testing the range of the wide-angle PIR sensor. We were primarily interested in testing how well the PIR sensor picks up motion when approached from the front and from the sides. To accurately measure distances, the PIR sensor is hooked up to an Arduino that reads the input from the PIR Sensor. For the tests, we demarcate a 20 x 14 foot boxed area on the floor, and place the PIR sensor at the bottom, with 10 feet to the left and right. In the frontal motion experiments, a test subject walks either toward or parallel to the sensors field of view at two foot intervals, starting 10 feet to the right and 14 feet away from the sensor, and ending 10 feet to the left of the PIR sensor. The lateral motion tests are similar, with the test subject starting 14 feet away and 10 feet on the left (right), and moving toward the sensor (laterally) in two foot intervals. In both sets of experiments, we record the point where the PIR sensor first detects motion. Each test is run three times to yield an average first motion-detected distance.

Figure 4 illustrates the results of the PIR sensor tests. In both sub-figures, the shaded area represents the area of detected motion. Our experiments with the frontal motion tests (Figure 4(a)) show that the PIR sensor detects frontal motion consistently at distances 10-12 feet away from the sensor. Our lateral motion experiments (Figure 4(b)) show that the PIR sensor (despite being wide-angled) struggles at detecting lateral motion. Surprisingly, the sensor is able to detect motion at further distances from its left than from its right. While the sensor was able to detect lateral motion to the left at 10 feet to the left and 14 feet forward, it was only able to detect motion laterally up to 6 feet to the right and 12 feet forward. We are unsure why this discrepancy occurs; however, we note that the PIR sensor is one of many possible sensors capable of waking a sensor node into "field mode". For example, ultrasonic sensors may provide the same capability.



(a) Frontal motion range.

(b) Lateral motion range.

Fig. 4: PIR sensor motion-detection results.

TABLE III: Battery lifetime tests

Name	Battery Life (h)	% Active	% Passive
Anker Powercore Battery	80	18	82
4 AA Batteries	14	25	75
2 123 Batteries	10.5	20	80

TABLE IV: Node Communication Latency (ms) in Building

Number of Walls	Distance (ft)	Latency (ms)
1	15.41	10.73
2	27.50	33.01
3	37.41	53.74
4	50.08	39.12
5	66.91	Network unreachable

B. Battery tests

The second set of tests measured the longevity of our sensor nodes against a variety of commercially available batteries. The Anker Powercore Battery ($22,000$ mAh capacity) was used in the previous design [12]. The four AA alkaline batteries were organized in serial ($2,500$ mAh each, for a total of $10,000$ mAh). The last test was against two 123 lithium batteries, which have a capacity of $1,500$ mAh each ($3,000$ mAh total). For all tests, we designed a test suite that recorded the current over time until the battery was too depleted to power the system. The test suite switched between the active fielding and passive sensing modes while a sensor node was connected to a battery. Current was sampled every microsecond, with a target of field mode being active for 20 percent, and passive for 80 percent, which is the estimated level of usage when there is a lot of foot traffic in an area ("high activity"). The BenchVue software, an Agillent Digital Multimeter, and custom breakout cord were used to monitor the system over the power draw and life time. We note that while a $20/80$ percent "field"/"passive" split was the goal for all tests, variations with equipment prevented us from maintaining a perfectly even split across all experiments.

The results of the battery experiments are shown in Table III. Prior work [12] shows that a Raspberry Pi-based sensor node with no sleep modes drains the $22,000$ mAh Anker battery in approximately 25 hours. The implemented sleep modes improve the battery life to 80 hours, an improvement of 320 percent. For short-term monitoring applications, our sensor nodes can be powered for 14 hours using 4 AA batteries, or 10.5 hours using 2 123 batteries. Given the inexpensiveness of AA and 123 batteries, our results suggest that our battery-operated sensor node is appropriate for short-term perimeter security goals, in which a network is needed for part of a day [5].

C. Network Range tests

Our last set of experiments measured the range of our Raspberry Pi sensor nodes in an urban environment. Specifically, we were interested in seeing how well the sensor nodes were able to communicate with the master node in a building, where sensors may be separated by one or more walls. In the first set of tests, we separated the master node from a single sensor node through a series of walls (6" drywall) by placing the node in different rooms at different distances from the master node. In the second set of experiments, several nodes in the mesh were placed in separate rooms (with different numbers of 6" walls separating them). In both cases, we measured the latency of sending ping messages from the furthest node to the master node. Latency measurements shown represent an average of five pings.

Table IV depicts the results of our first set of tests. We separated the master from the sensor node at various distances and numbers of walls, and attempted to ping the sensor node. Our results show that the master is able to ping a sensor up to 50 feet away with four walls in between, with an average latency of 39.12 milliseconds. However, once five walls are

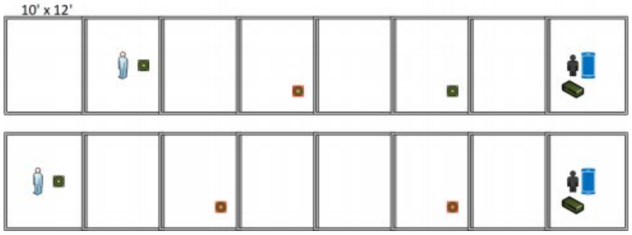


Fig. 5: Mesh network barrier tests.

between the sensor and the master, the network is no longer reachable. We note that in an open-area environment (where no barriers are present), a node can ping a master node that is nearly 100 feet (30 meters) away.

Our second set of tests were conducted in a hallway environment spanning eight identical rooms. Each room measured 10 feet by 12 feet. Using a subset of 3 sensor nodes and the master, we measured the distance that a sensor node can transmit through a series of walls. In all experiments, the user is in the room to the far right with the master node and Android app. The activated sensor node is shown in a room to the far left. Sensor nodes are placed in multiple rooms between the master and active sensor node.

Our tests (Figure 5) show that while the sensor in the room with movement could not directly communicate with the master node, it was able to send data to the master through intermediate sensor nodes (highlighted in red). Our tests show that that data was transferred through the mesh at distance of approximately 80 feet. While the range of our nodes and network is largely dependent on the composition of barriers such as walls, our results confirm that our system is able to successfully transmit images through the network via a series of intermediary sensor nodes in near real time.

V. CONCLUSION & FUTURE WORK

In this paper we discuss the design, implementation and tests on a wireless sensor network utilizing Raspberry Pis that provides an economical solution for short term perimeter security. Our results show that the Raspberry 0HWs have a very low power draw and function well as sensor nodes. Our system cost is as little as $\$49.00$ per node and up to $\$63.00$ for full audio-visual capabilities. Each of the nodes can transmit data through the mesh network to the master node with access via an Android smartphone app. A PCB board makes it possible to add and remove sensors easily to the system. Our system currently supports ten sensor nodes.

Our system has three key advantages over prior implementations. First, we use hsmm-pi to create a mesh network. A wireless mesh network is more appropriate than a cellular mesh due to the lack of availability of cellular networks in remote environments. Our system supports direct wireless transmission between nodes of up to 30 meters. When walls (6” drywall) are present, our tests show that direct node-to-node transmission is possible up to 50 feet (15.2 meters). Our testing in an 80-foot hallway with multiple rooms show that as little as three sensor nodes can be daisy-chained to transfer data from the furthest room to the room with the master node.

Second, we implement multiple sleep states that are capable of turning on and off Raspberry Pi ports and features to reduce battery consumption, making it possible to create a temporary perimeter monitoring system that operates on AA batteries for 14 hours. The small size and low weight of our sensor nodes make them very portable, making it easy for users to transport the entire system. The addition of our sleep states extended the average use of $22,000$ mAh battery from 25 hours to 80 hours, an improvement of 320 percent over prior work. We note that even when our sleep states are activated, network connectivity is still maintained, making it possible for nodes to passively transfer data while “sleeping”.

Lastly, our sensor nodes are modular and easily extendable, enabling users to modify and update individual nodes at low cost. We anticipate that over time, improvements in technology will cause various parts and sensors to get even more inexpensive, making sensor nodes more cost-efficient to build and incrementally improve. We note that this is not possible with commercial solutions, as they represent standalone products. We anticipate that individuals with limited budgets (or grant funding) will find our system especially attractive.

We note that our system is not an appropriate replacement for commercial systems in all circumstances. The majority of the bulk in commercial systems is due to their enormous battery packs and antennae. We note that extending our system with larger battery packs or antennae will duly increase the size and weight of our sensor nodes along with the communication range and battery life. However, the beauty of the Pi-sensor design is that users have a *choice* of how big their sensor nodes need to be and can easily add components as needed. Our results clearly demonstrate that for short-term perimeter monitoring scenarios, the AA battery-operated Raspberry Pi sensor network is a viable and portable solution.

There are a number of avenues for future work. To improve motion sensing range, we plan to explore other low power sensor alternatives such as seismic sensors and ultrasonic sensors. To further improve battery life, we plan exploring how to extend and improve our sleep states, including the “deep sleep” state. We hypothesize that this state will be useful to designate “repeater-only” nodes that will serve as range extenders. We also plan to extend testing with other types of batteries and begin exploring antennae. Lastly, we currently use a Raspberry Pi Model 3B for the master node. Ideally, all nodes would be Raspberry Pi 0HWs, enabling any node to transition to be the master node. While further work is needed to test and expand our mesh network and our capabilities, we believe our results strongly support the feasibility of all-Pi mesh networks for perimeter monitoring activities.

ACKNOWLEDGMENT

This paper summarizes the work of an undergraduate capstone project of which the first five authors were members. The next two authors were external faculty advisors, while the last two authors were the primary faculty advisors on the project.

This paper is authored by employees of the U.S. Government. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Army, Department of Defense or the U.S. Government.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[2] E. M. Royer and Chai-Keong Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, April 1999.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA '02. New York, NY, USA: ACM, 2002, pp. 88–97.

[4] M. F. Othman and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204 – 1210, 2012, international Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

[5] M. Winkler, K. Tuchs, K. Hughes, and G. Barclay, "Theoretical and practical aspects of military wireless sensor networks," *Journal of Telecommunications and Information Technology*, pp. 37–45, 2008.

[6] M. P. Duriŝić, Z. Tafa, G. Dimić, and V. Milutinović, "A survey of military applications of wireless sensor networks," in *2012 Mediterranean Conference on Embedded Computing (MECO)*, June 2012, pp. 196–199.

[7] T. Azzabi, H. Farhat, and N. Sahli, "A survey on wireless sensor networks security issues and military specificities," in *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, Jan 2017, pp. 66–72.

[8] P. Mathur, R. H. Nielsen, N. R. Prasad, and R. Prasad, "Wildlife conservation and rail track monitoring using wireless sensor networks," in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)*, May 2014, pp. 1–4.

[9] V. Dyo, S. A. Ellwood, D. W. Macdonald, A. Markham, C. Mascolo, B. Pásztor, S. Scellato, N. Trigoni, R. Wohlers, and K. Yousef, "Evolution and sustainability of a wildlife monitoring sensor network," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 127–140.

[10] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang, "Sensor network for the monitoring of ecosystem: Bird species recognition," in *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, Dec 2007, pp. 293–298.

[11] A. Tovar, T. Friesen, K. Ferens, and B. McLeod, "A dtn wireless sensor network for wildlife habitat monitoring," in *CCECE 2010*, May 2010, pp. 1–5.

[12] A. Alejos, M. Ball, C. Eckert, M. Ma, H. Ward, P. Hanlon, and S. J. Matthews, "Exploring the raspberry pi for data summarization in wireless sensor networks: Poster," in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security*, ser. HoTSoS '18. New York, NY, USA: ACM, 2018, pp. 18:1–18:1.

[13] Besttrailcamerareviews.org, "Best wireless trail cameras reviews and guide 2019," https://www.besttrailcamerareviews.org/wireless-trail-camera/ (accessed Jun. 11, 2019).

[14] Amazon.com, "Amazon," https://www.amazon.com (accessed Jun. 11, 2019).

[15] S. Ferdoush and X. Li, "Wireless sensor network system design using raspberry pi and arduino for environmental monitoring applications," *Procedia Computer Science*, vol. 34, pp. 103 – 110, 2014.

[16] J. Thyagarajan, S. Sundararajan, and A. Zungeru, "Active / passive mobility driven routing design and implementation for animal monitoring sensor network," *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 1, pp. 70–78, 2019, export Date: 4 February 2019.

[17] Raspberry Pi Foundation, "Raspberry pi - products," https://www.raspberrypi.org/products/ (accessed Jun. 11, 2019).

[18] S. J. Matthews, "Harnessing single board computers for military data analytics," in *Military Applications of Data Analytics*, K. Huggins, Ed. CRC Press, Taylor & Francis, 2018, ch. 4, pp. 63–77.

[19] V. Vujović and M. Maksimović, "Raspberry pi as a wireless sensor node: Performances and constraints," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2014, pp. 1013–1018.

[20] V. Vujović and M. Maksimović, "Raspberry pi as a sensor web node for home automation," *Computers & Electrical Engineering*, vol. 44, pp. 153 – 171, 2015.

[21] F. Leccese, M. Cagnetti, and D. Trinca, "A smart city application: A fully controlled street lighting isle based on raspberry-pi card, a zigbee sensor network and wimax," *Sensors*, vol. 14, no. 12, pp. 24 408–24 424, 2014.

[22] Xiang Yang, Dengteng Deng, and Meifeng Liu, "An overview of routing protocols on wireless sensor network," in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 01, Dec 2015, pp. 1000–1003.

[23] J. Yan, M. Zhou, and Z. Ding, "Recent advances in energy-efficient routing protocols for wireless sensor networks: A review," *IEEE Access*, vol. 4, pp. 5673–5686, 2016.

[24] I. Tomić and J. A. McCann, "A survey of potential security issues in existing wireless sensor network protocols," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1910–1923, Dec 2017.

[25] T. Babbitt, C. Morrell, and B. Szymanski, "Self-selecting reliable path routing in diverse wireless sensor network environments," in *2009 IEEE Symposium on Computers and Communications*, July 2009, pp. 1–7.

[26] G. Han, J. Jiang, N. Bao, L. Wan, and M. Guizani, "Routing protocols for underwater wireless sensor networks," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 72–78, November 2015.

[27] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 27–34.

[28] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R.Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," Internet Research Task Force (IRTF), RFC 4838, April 2007.

[29] T. A. Babbitt, C. Morrell, B. K. Szymanski, and J. W. Branch, "Self-selecting reliable paths for wireless sensor network routing," *Computer Communications*, vol. 31, no. 16, pp. 3799 – 3809, 2008, performance Evaluation of Communication Networks (SPECTS 2007).

[30] D. Rivenburg, "Hsmm-mesh design philosophy," http://www.broadband-hamnet.org/documentation/67-hsmm-mesh-design-philosophy.html (accessed Jun. 11, 2019).

[31] S. Kidder, "Hsmm-pi: A set of tools to easily configure the raspberry pi to function as a high-speed multimedia (hsmm) wireless node," https://github.com/urlgrey/hsmm-pi (accessed Jun. 11, 2019).

[32] E. Bulut and I. Korpeoglu, "Sleep scheduling with expected common coverage in wireless sensor networks," *Wirel. Netw.*, vol. 17, no. 1, pp. 19–40, Jan. 2011.

[33] D. Ye and M. Zhang, "A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 979–992, March 2018.

[34] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: A survey," *Computers & Electrical Engineering*, vol. 41, pp. 159 – 176, 2015.

[35] Raspberry Pi Foundation, "Raspberry pi zero w," https://www.raspberrypi.org/products/ raspberry-pi-zero-w/ (accessed Jun. 11, 2019).