

## Research Article

# An Experimental Study of WSN Power Efficiency: MICAz Networks with XMesh

Tyler W. Davis,<sup>1</sup> Xu Liang,<sup>1</sup> Miguel Navarro,<sup>2</sup> Diviyansh Bhatnagar,<sup>2</sup> and Yao Liang<sup>2</sup>

<sup>1</sup>Department of Civil and Environmental Engineering, University of Pittsburgh, 3700 O'Hara Street, Benedum Hall Room 949, Pittsburgh, PA 15261, USA

<sup>2</sup>Department of Computer and Information Science, Indiana University Purdue University, 723 West Michigan Street, SL 280, Indianapolis, IN 46202, USA

Correspondence should be addressed to Xu Liang, xuliang@pitt.edu and Yao Liang, yliang@cs.iupui.edu

Received 16 July 2011; Revised 13 October 2011; Accepted 20 October 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Tyler W. Davis et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This is an investigation of Wireless Sensor Networks (WSNs) using Memsic's XMesh routing protocol on MICAz wireless motes. It focuses on the study of the practical aspects of WSNs' power efficiency and network characteristics, which play a critical role in real-world WSN deployments for environmental monitoring. Based on an experimental study and following a quantitative approach, this work examines XMesh's high power and low power operation modes and the data transmission intervals, among other factors. Route utilization was identified as a major contributor of the mote's battery use. A field study was conducted as a point of comparison and the results obtained were comparable to the laboratory tests with regards to the battery life and the mote's route utilization. The network reliability was found to be considerably lower in the field study. In addition, it was found that the original WSN gateway, used during the study, presented severe practical limitations regarding the system's robustness and reliability. To address these problems, we present a solution based on our integrated network and data management system, which successfully facilitates the deployment of a new WSN gateway and significantly improves the operational robustness and reliability of the WSN system.

## 1. Introduction

Wireless sensor networks (WSNs) have demonstrated their potential and promising application in different fields of science and engineering [1, 2], such as geophysical studies for volcanic activities [3], environmental monitoring in glacier regions [4], structural monitoring [5], and healthcare applications [6]. However, the severe resource constraints of wireless sensor nodes (e.g., battery power, memory size, processor capacity, and network bandwidth) in WSNs raise new theoretical and practical challenges, drawing great attention in the research community. This work focuses on the practical aspects of WSN power efficiency, which is critical in real-world WSN deployments for environmental monitoring. It is essential that domain scientists and engineers who foresee the potential benefits of including WSNs in their studies and experiments have a fundamental and comprehensive understanding about the WSN power

efficiency and characteristics for different applications under dynamic operational environments.

Multiple research works have attempted to identify key features in WSN deployments [7–9]. While good qualitative results and guidelines are available, the lack of more quantitative results and descriptions is often a major obstacle in today's WSN design, implementations and deployments. In practice, WSN deployments can lead to various important tradeoffs among quality of service, network performance, power consumption, and operational cost. Understanding these tradeoffs in a quantitative way is fundamental for any successful and smart WSN practice. This experimental study *quantitatively* investigates the power efficiency and battery savings for WSNs with various network characteristics, using the popular MICAz wireless motes for environmental monitoring.

WSNs often use individual node power efficiency as a key performance metric, because the battery power of individual

nodes can lead to multiple failure types within the network. For example, [7] examined fourteen environmental WSN field deployments from year 2002 to 2008, ranging in scale from 3 to 98 nodes. Their examination showed that battery power can affect every level of wireless networking problems which they had classified into four categories: node, link, path, and global failures. Node failures can occur when the battery power drops below the level at which the mote can still operate. Link failures can occur when low battery power reduces the range that a mote can transmit its data, effectively removing it from the network's communication topology. Path failures can result when an important node or nodes that route transmitting data have node or link failures forcing the network to use less efficient paths for transmitting data to the base station. Global failures can occur when a critical or bottleneck node experiences node or link failure cutting off the transmission of data from the whole or part of the network. It is important to identify these problems and their causes such that they can be avoided in field deployments.

In this study, we adopt the commercially available Memsic's WSN platform as a vehicle to carry out our investigation. It is one of today's most widely used WSN platforms (previously developed by Crossbow Technology), in which motes are programmed in nesC language [10], linked together with specific data acquisition boards, to form a mesh WSN for various applications. The application code, compiled and loaded to wireless motes, runs under the motes' TinyOS operating system [11]. Memsic's XMesh routing protocol [12], which features TrueMesh technology, is utilized as the networking foundation for these applications. It provides a mesh network which is self-healing and self-organizing, where each mote acts as both a sensor node and a router for its neighbor's data. The ad hoc formation of the network is based on link estimates made between node neighbors and routes data down a path of lowest transmission cost to the base station where data is stored. Parameter assignment in the nesC code along with some runtime argument passing during compilation allows users to have partial control over the mote's power efficiency.

Using Memsic's WSN application platform, motes can be programmed in either a high power (HP), low power (LP), or extended low power (ELP) operation mode. Note that the ELP operation mode does not support routing. Since we consider multihop networks, the ELP operation mode is not included in this study. LP and HP operation modes implement different power efficiencies and thus have different battery savings associated with them. In addition to the two operation modes, the transmission frequency of data packets can also be manually adjusted. The transmission rate also plays an important role on the battery life of the motes, because data transmission is the most power consuming operation, as shown in [7, Table 3], [13, Table 2], and [14, Table 6-2]. Adjustments to the operation mode and transmission frequency can reduce the power consumption of the motes which increases their operating life. The power consumption of wireless motes is an important consideration when deploying networks as it may affect the sensor readings [15] and network connections [7].

We investigate the battery savings by implementing the two operation modes over various data collection intervals with Memsic's XMesh WSN. Battery life is measured over the transmission life of the motes and is used to compare the actual battery savings of each configuration. In addition, we share our experiences and lessons learned from our experimental study regarding the WSN gateway Stargate Netbridge's operational robustness. It is found that Stargate Netbridge, a Linksys NSLU2 device specially modified by Crossbow, has a severe robustness and reliability issues for practical WSN deployments. To address this problem, we present how the solution of our general integrated network and data management system can facilitate the new deployment of a WSN gateway to successfully replace the Stargate Netbridge and significantly improve the operational robustness of WSN deployments.

The remainder of this paper is organized as follows: Section 2 includes an in depth description of the laboratory experiments performed and their results. Section 3 examines a prototype testbed of eleven nodes and compares the network and mote operations to the laboratory experiments. Section 4 presents the solution of our integrated network and data management system for the deployment and robust operation of the XMesh-based WSN in order to overcome the original Stargate Netbridge gateway limitations and to facilitate a more complete study in the future. The concluding remarks based on the findings of this study are given in Section 5.

## 2. Laboratory Methods

To determine the effect of different sampling frequencies on battery life, a series of experiments was conducted using the MICAz wireless mote, equipped with a MDA300 data acquisition board, both manufactured by Memsic Corporation (previously Crossbow Technology). The mote's operation mode, sampling, and transmission intervals were investigated in these experiments. The mote's operation mode is set using the XMesh high power (HP) or low power (LP) configuration. The difference between these two settings is mainly in the bandwidth consumption and latency of the transmissions. The LP mode, which utilizes a sleep function that powers off all unnecessary electronics between operations, has a high latency (i.e., long transmission delay) and low bandwidth consumption (i.e., low data capacity) compared to the HP mode.

The mote's onboard radio power is adjustable within a range of 0 to  $-25$  dBm, the maximum and minimum power allocations, respectively. The MICAz CC2420 radio transceiver [16] uses IEEE 802.15.4 protocol [17] and transmits data in the 2.4 GHz frequency band with a maximum data rate of 250 kbits/s. Other investigations have already looked at the effects of the transmission power on mote connectivity and battery power [9]. The previous study was interested in understanding how the antenna power affected the transmission distance and how implementing various duty cycles increased the battery life. In contrast, this study is more focused on how the data collection rate

and multihop network functionality affects battery life. The default 0 dBm (1 mW) radio power setting in the 2.405–2.425 GHz frequency channel was used in this study. The data message interval (DMI) and the data transmission interval, which are designated to be the same value, can be set to any multiple of seconds. Intervals were chosen from 10 to 900 seconds.

*2.1. Experiments and Analysis on Basic Battery Capacity.* In an effort to reduce uncertainties in the sampling interval study that may be caused by the variability of the individual motes or batteries, eight random motes were selected and tested four times each (tests A–D) under the HP mode using a 1-second DMI. To test the effect of the battery capacity, half of the motes tested were powered with two 2500 mAh AA batteries and the other half were powered with two 2450 mAh AA batteries. The batteries used in this experiment were all rechargeable nickel-metal hydride (NiMH) AA batteries. In this experiment, motes were placed on desktops in close proximity to the base station. A summary of the results of these tests for the eight motes is given in Table 1.

The starting and ending battery voltage (based on the two AA batteries) for the experiment are shown in Table 1. The starting voltage is based on the earliest stable battery level recorded in the mote's data packets. This is typically within the first 5–10 transmissions. The ending voltage is the last recorded battery level received by the mote before transmissions ceased. Also included in Table 1 is the number of data packets that were received by the base station from each mote during its operation life. Each data packet represents one sample transmitted and collected from a mote to the base station.

Results, as shown in Table 1, indicate that the batteries from both series had variances in their starting voltages. There is a large amount of variability in the starting voltages of each test. This is due to the rechargeable nature of the batteries. While there are varied differences in the voltage drop for each mote, individual motes have a specific lower-end power requirement for operation as shown by the similar ending voltages in each mote's series of tests. The average battery life for the four motes with 2500 mAh batteries compared to the four motes with the 2450 mAh batteries is 4625 minutes to 2979 minutes (a 55% longer battery life in the 2500 mAh batteries). Given these results, it can be concluded that variability in the batteries, for example, differences in charging time, age of the battery, and so forth, is the major component in the variability of the battery life in the motes. This is not to say that the variability in the battery life is caused by the batteries alone. There is evidence of both individual variability (as seen in the ending voltage for each individual mote) and intermote variability (as seen in the differences in the ending voltages between groups of motes), albeit smaller than the influence of the batteries.

*2.2. Experiments and Analysis on Data Message Intervals.* The battery life for various DMIs in the two power modes was analyzed on twelve motes. The twelve motes that were selected were separated into four groups of three motes.

The battery life for a given DMI and power mode (HP or LP) pairing was tested on one of the four groups of motes. The three motes in each group were used to determine statistical parameters of the results. Each DMI and power mode pairing was tested three separate times. To reduce the battery variability, the motes were powered by two Panasonic Industrial (AM3) AA batteries, rated at 2870 mAh. The motes were once again tested using the default radio power level, 0 dBm (1 mW).

In HP mode, the DMIs tested were 10 s, 30 s, 60 s, and 900 s. The same DMIs were tested in LP mode except for 10 s. Due to the battery saving nature of the LP mode's sleep functionality, the 10 s DMI was deemed inappropriate (personal communication with Crossbow technician) and was intentionally excluded from the LP tests.

A summary of the battery tests is given in Table 2. The recommended operating voltage for motes is 3.6–2.7 V [14, Table 6-1], however it has been shown that the mote can continue to collect data down to 2.2 V and transmit messages down to 2.1 V [18]. Included for each DMI and power mode pairing is the average number of packets received and the average battery life over the total range of battery voltages. Next to each of the average values is the standard deviation based on the nine results (i.e., three groups of three motes per test).

Table 2 shows that for both the HP and LP modes, there is a small difference between the total battery life of the various DMIs. In the HP mode results, the 10 s and 30 s DMIs have a slightly longer battery life than the 60 s and 900 s sampling intervals. The battery life for all four of the HP tests are similar to the results obtained in Table 1 for the 2500 mAh batteries. The variability in the battery life results in Table 2 is also similar to the variability in the battery life of a single mote tested in Table 1. Therefore, it can be concluded based on these results that the sampling interval has no significant influence on the battery life, which is contrary to what was expected. These results would suggest that the sensed data transmissions regardless of sampling DMIs may not be a major cause of the battery energy consumption in the XMesh network. Further investigation was then conducted, as described in the next section, to determine the possible major causes for the battery life results shown in Table 2.

*2.3. Experiments and Analysis on the Impacts of Health Messages.* One possible cause for the battery life results shown in Table 2 would be the XMesh network's health messages. The network's health messages are sent to the base station periodically for updates regarding each individual mote's neighbor list and its own physical health statistics. The default configuration is to send the health messages every 60 s and 600 s for motes in HP and LP modes, respectively. The health message interval (HMI) is adjustable in the nesC application code. The mote alternates sending the neighbor information and its own statistics health messages at each transmission interval.

The transmission of the health messages could effectively reduce the mote's transmission interval to that of the HMI if the HMI is smaller than the mote's DMI. These health

TABLE 1: Summary of the mote and battery experiment of four tests on eight motes in HP mode at 1-second transmissions.

2500 mAh Batteries																
Mote	1				2				3				4			
Test	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
Start (volt)	2.55	2.79	2.72	2.85	2.46	2.71	2.64	2.70	2.63	2.87	2.80	2.97	2.47	2.69	2.59	2.76
End (volt)	1.97	1.97	1.97	1.97	1.87	1.65	1.87	1.87	1.99	1.99	1.99	1.99	1.86	1.88	1.87	1.86
Pkts. (thou)	102.9	156.1	147.1	143.6	100.7	87.9	145.1	143.5	99.7	154.4	147.1	149.9	98.4	150.1	135.5	142.3
Life (min)	3481	5342	5094	5090	3468	3038	5094	5095	3488	5383	5230	5381	3458	5296	4872	5191
2450 mAh Batteries																
Mote	5				6				7				8			
Test	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
Start (volt)	2.72	2.77	2.75	2.74	2.70	2.75	2.72	2.80	2.75	2.79	2.76	2.85	2.77	2.73	2.71	2.80
End (volt)	1.95	1.91	1.93	1.94	1.93	1.93	1.93	1.92	1.96	1.96	2.11	1.95	1.91	1.91	1.91	1.88
Pkts. (thou)	43.7	78.0	89.6	53.0	56.5	112.9	71.7	103.3	84.2	85.6	63.8	103.9	118.1	78.5	63.9	106.4
Life (min)	1570	2773	3182	1910	2043	4056	2576	3744	3072	3093	2318	3810	4351	2881	2346	3942

TABLE 2: Average battery life and transmission packets from nine motes tested in HP and LP modes at various sampling intervals with the standard deviations.

Power mode	Samp. Int.	Life (min.)	Data packets
HP	10 s	5545 ( $\pm 341$ )	29991 ( $\pm 1909$ )
	30 s	5618 ( $\pm 345$ )	10889 ( $\pm 645$ )
	60 s	4610 ( $\pm 217$ )	4499 ( $\pm 213$ )
	900 s	4597 ( $\pm 139$ )	308 ( $\pm 10$ )
	30 s	20502 ( $\pm 446$ )	21157 ( $\pm 3213$ )
LP	60 s	20859 ( $\pm 790$ )	14035 ( $\pm 2736$ )
	900 s	19952 ( $\pm 1942$ )	1179 ( $\pm 112$ )

messages sent to the base station can influence the speed in which the mote's battery power is exhausted. Thus, an experiment was conducted in attempt to quantify the effect of different HMIs on the mote battery power.

The same four groups of three motes that were used in the previous battery life measurements were once again used to test the effect of the HMIs on mote battery life. All four groups of motes were programmed to collect data in the LP mode and sample at a 900 s interval. The motes were powered with two Panasonic industrial 1.5 V AA batteries. The HMI for each of the four groups of motes was set to 120 s, 300 s, 600 s, and 900 s, respectively. The experiment took place indoors in a laboratory environment. In the laboratory, two groups (i.e., six motes) were positioned approximately 45 cm away on either side of the centrally located base station. Motes within each group were spaced a few centimeters apart.

The initial results from the health message testing revealed little insight on the effect of these message transmissions on battery life. The average battery life, shown in minutes, over the total range of battery voltages is shown in Table 3. Similar to the results shown in Table 2, there is again little difference between each group's average battery life. There is, however, a large difference between the average LP battery life between Tables 2 and 3. The battery life

TABLE 3: Battery life and transmission packets for various health message intervals at 900-second sampling interval in LP mode over the total battery life.

Health Int.	Life (min.)	Data packets
120 s	67304 ( $\pm 2174$ )	3655 ( $\pm 130$ )
300 s	67956 ( $\pm 641$ )	3672 ( $\pm 11$ )
600 s	68974 ( $\pm 1665$ )	3826 ( $\pm 63$ )
900 s	68112 ( $\pm 3697$ )	3818 ( $\pm 305$ )

results in Table 3 are over 200% longer than the LP results in Table 2. There was a single LP test completed, that was not included in the results in Table 2 because it was deemed an outlier but had an average battery life of 68000 minutes. It seems now that the outlying results from the DMI tests (as shown in Table 2) are in accordance with the results presented in Table 3 for the LP motes. This poses a new question concerning the reason for the low LP mote battery life presented in Table 2.

To see how the motes are performing over their battery life, the number of samples taken by each mote in the four groups, separated into voltage bins, is plotted in Figure 1. There is a distinct "double-hump" feature noticeable at around the 2.3 V and 2.65 V bins. The cumulative number of samples collected (as shown in Figure 2) shows that all the motes begin and end collecting approximately the same number of samples over their battery life. The variance present throughout the middle region may be explained by individual mote variability.

The similarities in battery life (e.g., Table 3) and samples collected (e.g., Figure 2) prompted further investigation on the mote's DMI over its battery life to understand these similarities. Figure 3 is an example of a typical mote's behavior over time. The plot of the battery voltage curve is based on the measurements from the data packets received by the base station. The DMIs are taken as the time difference, in minutes, between two successively received data packets. The number in parentheses indicates the count of data packets received at each given interval. It can be seen in Figure 3

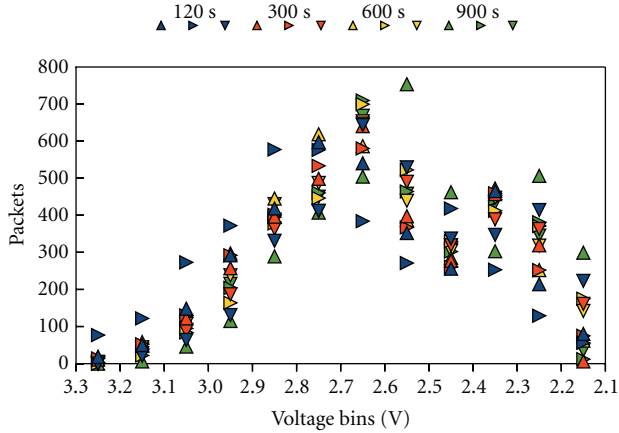


FIGURE 1: The number of data packets collected for motes in LP mode sampling at 900 seconds at various health message intervals. The four groups of three motes are plotted, each at their respective health update interval, that is, 120 s, 300 s, 600 s, and 900 s.

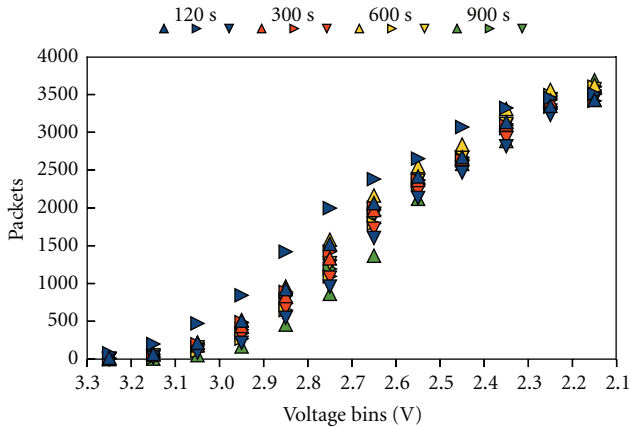


FIGURE 2: The cumulative number of packets collected over the battery life of motes in LP mode sampling at 900 seconds at various health message intervals. The four groups of three motes are plotted, each at their respective health update interval, that is, 120 s, 300 s, 600 s, and 900 s.

that the majority of the packets collected were made at the prescribed interval, that is, 15 min or 900 s. A large number of packets were also collected at twice the programmed interval, that is, 30 min. This indicates that the mote dropped one data packet between two successful transmissions to the base station. Fewer packets are shown to have been collected at intervals of 45, 60, 75, and 90 min, signifying that the mote dropped 2, 3, 4, and 5 data packets between successful transmissions to the base station. Using the time intervals between successfully received data packets, of the 3744 data packets received, there were an estimated 714 dropped packets (almost 20%). This is a considerably large number with respect to both the loss of data and the wasted power for transmissions. Therefore, the dropped packet rates of the wireless motes were more closely examined.

2.4. Experiments and Analysis on the Impacts of Dropped Packets. A dropped packet occurs when a mote’s data packet

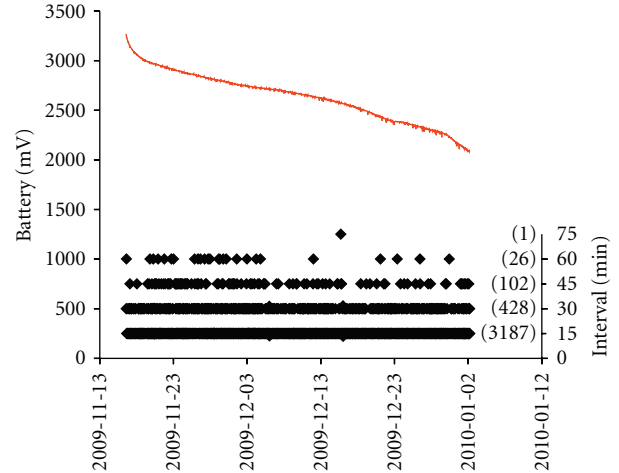


FIGURE 3: Mote battery voltage (in millivolts) over time (red line) in HP mode with a 15 minute DMI and the time interval (in minutes) between received data packets (black diamonds). The number of samples collected at each time interval is indicated in parentheses on the right side of the plot.

is unsuccessfully delivered through the network to the base station. This can occur at any individual node through the multihop network. The unsuccessful delivery of a data packet can be due to one of the four failure modes previously defined. In the transmission of a data packet over each consecutive hop, an acknowledgement message is returned to a mote by its parent to signify that the packet was received. If a mote does not receive an acknowledgement, it will attempt to resend the packet up to eight times until either an acknowledgement is received or the maximum number of retries is reached [19, Section 10.1.4]. If the maximum number of retries is reached without successful acknowledgment, the packet is “dropped” and results in lost data.

An analysis of the mote transmission performance is summarized in Table 4. The received packets in Table 4 are for the total transmission life of the motes as given in Table 3. Dropped packets were identified based on the time interval between successful receipts of data packets, as described above. Table 4 breaks down the number of data packets received into the number of packets received on time (no packet loss), the number of packets received with only one retransmission, the number of packets received after two retransmissions, the number of packets received after three or more retransmissions, and the number of asynchronous packets received.

Asynchronous packets occur when a mote does not receive an acknowledgement from its parent that its packet was received and therefore retransmits the same packet again. In some cases, the link quality from a mote to its parent is better than the link quality from the parent to the mote. This is also known as link quality asymmetry [9]. Under this circumstance, asynchronous packet delivery can occur and the base station will receive duplicate packets from a mote. The amount of this occurrence can be calculated by

TABLE 4: Transmission analysis of LP motes sampling at 900 seconds for various health message intervals.

Health Int.	120 s	300 s	600 s	900 s
Packets recv.	3655	3672	3826	3818
0 drop	84.3%	84.0%	84.9%	86.2%
1 drop	12.4%	12.8%	11.8%	11.2%
2 drop	2.4%	2.3%	2.4%	1.9%
2+ drop	0.6%	0.7%	0.7%	0.5%
Async.	0.3%	0.2%	0.2%	0.2%
Packets drop	713	728	735	638
Packets exp. <sup>a</sup>	4367	4399	4561	4456
Success rate	83.7%	83.5%	83.9%	85.7%

<sup>a</sup>Estimates based on summation of data packets received and data packets dropped.

identifying two or more duplicate data packets received by the same node within a few seconds of each other.

The number of packets expected shown in Table 4 is an estimated value based on the sum of received and dropped data packets. The success rate, shown in the last row of Table 4 as a percentage, is the ratio of received to expected data packets. In each case, only about 80% of the packets expected were received. Furthermore, each of the four sets of motes had a similar number of dropped packets. To better understand the nature of packet drops, we further conducted an investigation into the multihop structure of the network.

*2.5. Experiments and Analysis on Routing Usage.* This WSN's architecture uses XMesh which features TrueMesh technology. This means that the network is self-healing and self-organizing and each mote acts as both a sensor node and a router for its neighbor's data. The ad hoc formation of the network is based on link estimates made between node neighbors in order to send data down a path of lowest transmission cost to the base station. As a consequence, certain motes may be exploited as a relay due to their low path cost.

A mote's parent ID is included in the data packet sent to the base station. The parent node is defined as a mote's neighbor with the lowest transmission cost [20, Section 4.2]. Thus, by analyzing the parent data, it can be determined if there are any motes being exploited and therefore having their batteries drained at a higher rate. Table 5 shows the results of this analysis. The number of health packets that each group sent is estimated based on the average battery life of each group and the HMI. The packets forwarded represent the additional transmissions motes make relaying neighbor data (i.e., data and health packets) through the network. To determine the multihop forwarding through the network, the distribution of each mote's connection with their neighbors was calculated based on the parenting information collected in the data packets. The same distribution was used to determine the routing of the data and health packets through the network. The number of packets generated is the summation of the

TABLE 5: Route-utilization analysis of the LP mote health message interval experiment (from Tables 3 and 4).

Health Msg. Int.	120 s	300 s	600 s	900 s
Data Pkts. Gen. <sup>a</sup>	4367	4399	4561	4456
Health Pkts. Gen. <sup>b</sup>	33652	13591	6897	4541
Data Pkts. Fwd.	1123	2790	2454	12714
Health Pkts. Fwd.	4575	11745	10191	52682
Route-util. <sup>c</sup>	15.0%	80.0%	110%	727%
Life (min.) <sup>d</sup>	77390	122859	145088	563209

<sup>a</sup>Expected data packets based on the received and estimated dropped packets (see Table 4).

<sup>b</sup>Estimates based on the battery life and health message interval.

<sup>c</sup>Route-utilization is the ratio of forwarded data and health packets to generated data and health packets.

<sup>d</sup>Total battery life, see Table 3, scaled based on the route-utilization percentage.

estimated number of data packets expected (see Table 4) and the estimated health packets (based on the battery life and HMI). The route-utilization is the percentage of additional forwarding transmissions compared to those generated by the mote (i.e., data and health packets) and is defined as the ratio of packets forwarded to packets generated as shown in the following equation:

$$\text{Route-utilization} = \frac{\text{Fwd}_{\text{data}} + \text{Fwd}_{\text{health}}}{\text{Gen}_{\text{data}} + \text{Gen}_{\text{health}}}, \quad (1)$$

where  $\text{Gen}_{\text{data}}$  and  $\text{Gen}_{\text{health}}$  are the estimated total number of data and health packets generated by a mote and  $\text{Fwd}_{\text{data}}$  and  $\text{Fwd}_{\text{health}}$  are the estimated number of data and health packets forwarded by a mote.

It can be seen that the route-utilization increases as the HMI increases. The motes with the 900 s HMI were utilized considerably more than those with the smaller HMIs. Table 5 shows the scaled battery life of each set of motes, as given in Table 3, based on the route-utilization percentage. The results of scaling the mote battery life show the expected trend in increasing battery life with decreasing the number of samples taken.

The MICAz mote is expected to have a battery life up to one year [12, Table 3-1]. This corresponds to the estimated battery life of the 900 s HMI results. Given the results in Tables 2 and 3 for LP motes, the scaling may be an exaggerated battery life adjustment. It is more likely that the increase in battery life compared to the default 600 s would be closer to 14% (logarithmic trend) or 29% (linear trend). Regardless, this test shows that decreasing the HMI from the default 600 s to either 300 s or 120 s will affect approximately 15–46% of a mote's battery life, respectively.

The analysis of scaling the battery life according to the mote's route-utilization can be applied to the results in Table 2. The adjusted HP results in Table 6 show no change in the battery life trend from the original results in Table 2. Given that the HMI for motes in HP mode is 60 s, it is expected that the data sampling rates of 60 s and 900 s HP results would be similar, that is, motes sending data every

900 s are also sending health packets every 60 s. The HMI, however, does not explain why with 10 s and 30 s sampling rates the HP motes' battery lives are about 25% longer than those with the 60 s and 900 s sampling rates. A more detailed analysis of the power consumption of HP motes of XMesh-based WSNs is necessary to understand this difference which is beyond the scope of this work. The adjusted battery life for the LP results in Table 6 show the expected trend in battery life with increasing transmission interval.

The route-utilization shown in Table 6 indicates that the LP motes with the largest data transmission interval, that is, 900 s, suffer the highest routing relays. This is similar to what was seen in the results of Table 5. For the HP motes, however, it can be seen that the smaller transmission interval (10 s) leads to the highest routing relays. From both tests, the HP motes have a consistently higher rate of successful transmissions (above 90%) while the LP motes have generally lower rate of successful transmissions over a varied range (78–92%).

While increasing the HMI can save some battery life, it reduces the amount of information collected regarding node link quality and path cost through the network. For purposes of monitoring the network mesh, the HMI may be more important than the battery savings. However, if network monitoring is not being considered, the health messages can be disabled completely to maximize battery savings.

### 3. Field Study

In addition to the laboratory tests, an experimental testbed consisting of eleven nodes was deployed in a residential backyard site (see Figure 4) located in western Pennsylvania (40.5436° N, 80.0638° W). The nodes range 6–60 m away from one another and are located mainly along the perimeter of the yard which is an approximately 60 m by 30 m rectangle. The wireless motes, data acquisitions boards, and battery packs were all housed inside polycarbonate high-impact enclosures (Bud Industries Inc.; part no. PN-1337). An external antenna (Pulse Electronics; part no. W1038) was attached to the outside of each enclosure. The enclosures were mounted on wooden stakes, placing the antenna approximately 0.3 m above the ground.

The network collected data starting from the summer of 2009 until the summer of 2010. All nodes were programmed in LP mode with the default radio transmission power (0 dBm) and health message interval (600 s). Motes were powered by two sets of two AA rechargeable batteries connected in parallel. The LED indicator lights on the mote were turned off due to their power consumption (6–8 mA) on the mote's batteries [12]. From 07/22–10/20 in 2009, all nodes sampled data at a 15 min interval. An analysis of the network's battery life and behavior was completed during this time. Figure 5 shows the received packets from each of the eleven nodes over the three month period.

Table 7 shows the battery life and packet analysis for each of the eleven nodes. The start and end dates correspond to the time that the transmission of data packets began and ended (see Figure 5). It should be noted that mote 5140 was

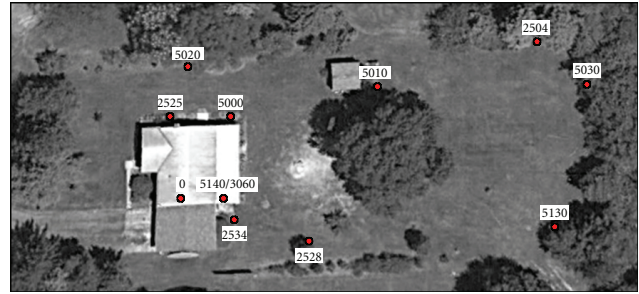


FIGURE 4: Residential backyard wireless sensor network testbed node locations during the late Summer and Autumn of 2009.

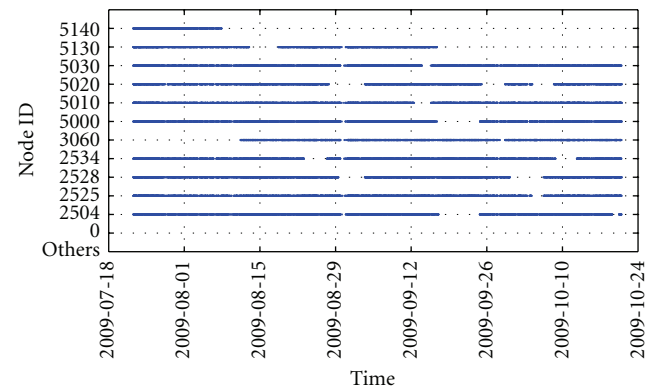


FIGURE 5: Received data packets by the base station for each node over the three-month monitoring period.

replaced by mote 3060 due to technical difficulties. Data was excluded from the analysis during periods where the motes did not complete a full battery cycle, that is, motes whose batteries were replaced before 10/20/2009 but were not fully depleted until after this time.

It can be seen from Table 7 that the outlier nodes (e.g., 2525, 2504, and 5030) have less forwarded packets and longer battery lives. While the outlier node 5130 did not forward many packets, its battery life was not as long as the other outlier nodes. This could be due to its isolated position not having a good connection to its neighbors. Nodes with the closest contact with the base station (e.g., 3060, 2534, and 2528) forwarded the most packets with battery lives on average less than the outlier nodes.

The success rate is approximately the same, around 40%, for all the nodes tested in the field. The high dropped packet rate may be attributable to the motes' close proximity to the ground. From the laboratory experiments the dropped packet rate for LP motes was found to be 10–20% compared to the 60% dropped packet rate in the field. The network's stability can be analyzed by examining the number of times a mote's parent changes. The high number of parent changes for the motes suggests that the link quality between the motes and their neighbors was low and fluctuated around the threshold for new parent selection. A large number of parent changes is representative of low network stability. This may have also attributed to the high dropped packet rate.

TABLE 6: Route-utilization analysis of the HP and LP mote sampling interval experiment (from Table 2).

Sampling Int.	HP				LP		
	10 s	30 s	60 s	900 s	30 s	60 s	900 s
Health Pkt. <sup>a</sup>	5545	5618	4610	4597	2050	2086	1995
Packets drop	156 ( $\pm 77$ )	12 ( $\pm 3$ )	2 ( $\pm 2$ )	0 ( $\pm 0$ )	5655 ( $\pm 4672$ )	1229 ( $\pm 472$ )	112 ( $\pm 41$ )
Packets Fwd. <sup>b</sup>	1855 ( $\pm 4099$ )	64 ( $\pm 157$ )	1 ( $\pm 1$ )	0 ( $\pm 0$ )	11525 ( $\pm 3962$ )	12294 ( $\pm 12498$ )	9531 ( $\pm 18069$ )
Packets Gen. <sup>c</sup>	35692	16519	9111	4905	28862	17350	3286
Route-util.	4.6%	0.5%	0.0%	0.0%	42.2%	66.5%	289.6%
Success rate	99.5%	99.9%	99.9%	99.9%	78.9%	91.9%	91.3%
Life (min.) <sup>d</sup>	5800	5646	4610	4597	29154	34730	77732

<sup>a</sup> Estimates based on battery life and the default health message intervals, that is, 60 s for HP motes and 600 s for LP motes.

<sup>b</sup> Forwarded packets are estimates of both data and health packets, taking into account the multi-hop network functionality, based on the statistical distribution of mote routing.

<sup>c</sup> Packets generated is equal to the sum of a mote's total data and health packets.

<sup>d</sup> Total battery life, see Table 2, scaled based on the route-utilization percentage.

TABLE 7: Battery life and transmission statistics of the eleven node prototype testbed network.

Node ID	Start date	End date	Life (min)	Packets rcvd.	Packets dropped	Packets Fwd.	Parent changes	Success
2504	07/22/2009	09/17/2009	81294.9	2152	3136	1080	1149	40.7%
	09/24/2009	10/19/2009	35322.6	895	1399	321	370	39.0%
2525	07/22/2009	10/04/2009	106103.0	2889	4018	371	495	41.8%
	07/22/2009	08/29/2009	54618.2	1432	2113	2029	686	40.4%
2528	09/03/2009	09/30/2009	38652.5	1121	1382	1946	477	44.8%
	10/06/2009	10/20/2009	20038.7	529	768	835	235	40.8%
2534	07/22/2009	08/23/2009	45425.7	1136	1802	1931	518	38.7%
	08/27/2009	10/08/2009	60846.7	1646	2287	2753	634	41.9%
3060	08/11/2009	09/28/2009	69075.5	1934	2567	3492	741	43.0%
5000	07/22/2009	09/16/2009	80909.1	2194	3116	2636	783	41.3%
	09/24/2009	10/20/2009	37022.7	945	1491	601	318	38.8%
5010	07/22/2009	09/12/2009	74773.6	1944	2947	1517	1007	39.7%
5020	07/22/2009	08/27/2009	52074.3	1312	2110	689	570	38.3%
	09/03/2009	09/25/2009	31147.3	923	1122	625	277	45.1%
5030	07/22/2009	09/13/2009	76867.8	2014	3013	663	938	40.1%
5130	07/22/2009	08/12/2009	30834.1	790	1281	363	410	38.1%
	08/18/2009	09/16/2009	42237.0	1214	1611	380	581	43.0%
5140	07/22/2009	08/07/2009	23470.4	752	807	674	213	48.2%

The field test results are comparable to the laboratory experiments for motes programmed in LP mode. In Tables 3 and 5, the results for motes with the smallest health message interval (120 s) are close to the field experiment results for nodes with higher traffic (e.g., more forwarded packets) such as motes 2504 and 5010. The comparable battery life (approximately 70000 min) and forwarded packets (approximately 1400 packets) show that the laboratory experiments were completed in a relatively high traffic condition compared to other motes in the field. It should be noted that under the same HMI, the scaled battery life from the laboratory experiment (Table 5) is about double the results in the field. The original battery life (Table 3) is similar to what was seen from the motes in the testbed. This would suggest that route utilization affects approximately 50% of the battery life. While there does not appear to be any correlation between mote location and the transmission

success rate, battery life does show some dependence on the network's topology.

An analysis on the route utilization in this field test was then performed. Based on the motes' parent IDs included in the data packets (see Figure 6), the link selection probability distribution for a packet to be forwarded by a mote through any of the mote's neighbors was calculated. Based on the conditional probabilities, a graph of the network topology was created where the vertices represent motes and each edge corresponds to a communication link associated with its selection probability. Considering the possibility of asymmetric links, the topology graph is a directed graph and its adjacency matrix is presented in Table 8. In this graph, the data for mote 5140 was included in mote 3060.

For this graph, each individual mote's link selection probability distribution is calculated based on the parent IDs included in its generated data packets, which represents



TABLE 8: Field test adjacency matrix for the network topology graph.

	0	2504	2525	2528	2534	3060	5000	5010	5020	5030	5130
0	0	0	0	0	0	0	0	0	0	0	0
2504	0	0	0	0.162	0.242	0.303	0	0.117	0.013	0.104	0.059
2525	0	0	0	0	0	0	0.631	0.002	0.367	0	0
2528	0.274	0.030	0	0	0.261	0.253	0.046	0.076	0.007	0.029	0.025
2534	0.478	0.025	0	0.17	0	0.221	0.008	0.046	0.019	0.022	0.011
3060	0.528	0.032	0	0.13	0.222	0	0.026	0.042	0.007	0.007	0.006
5000	0	0	0.038	0.348	0.134	0.188	0	0.191	0.094	0.001	0.007
5010	0	0.074	0.001	0.214	0.271	0.318	0.038	0	0.012	0.045	0.028
5020	0	0.001	0.107	0.129	0.168	0.310	0.230	0.041	0	0.003	0.011
5030	0	0.150	0	0.210	0.329	0.079	0.001	0.129	0.017	0	0.084
5130	0.005	0.185	0	0.228	0.168	0.135	0.021	0.119	0.015	0.124	0

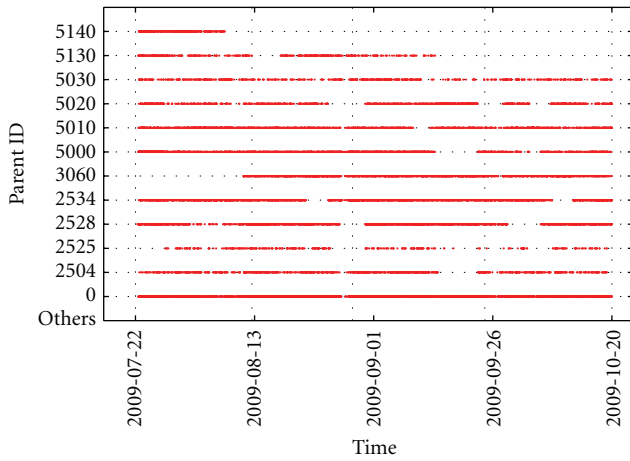


FIGURE 6: Parent ID for the received data packets for each node over the three-month monitoring period.

the behavior of the mote's first hop of its route towards the base station. In order to understand the overall routing behavior of the multihop network, it is reasonable to assume that the forwarded packets at each relay mote will follow the same link selection distribution of the generated packets at this mote for there is no parent ID information for relayed packets. Thus, in the following analysis, the link selection in routing for both originally generated packets and relayed packets at each individual mote follows the same link selection distribution.

The results of route utilization for each mote are presented in Table 9. The average route utilization for the field in this test is 105%. Given that the network is configured for a DMI of 900 s with a default HMI of 600 s, these results are comparable to the corresponding lab test with the same DMI and HMI (see Table 5). These results can be justified based on the higher traffic loads and route utilization in some specific motes. The analysis for the route utilization in each individual mote shows that motes 2528, 2534, and 3060 have the highest route utilization in the network followed by motes 5000 and 5020. The group with the smaller route-utilization consists of motes 5010, 2504, 5130, 5030, and

TABLE 9: Route-utilization percentage for each mote and network average.

Mote ID	Generated packets	Forwarded packets	Route utilization
2504	1524	742	48.7%
2525	2889	181	6.3%
2528	1027	2417	235%
2534	1391	2689	193%
3060	1343	2584	192%
5000	1570	2270	144%
5010	1944	1106	56.8%
5020	1118	1343	120%
5030	2014	444	22%
5130	1002	385	38.4%
Network avg.	1582	1416	105%

2525. These three groups of motes are highlighted in Figure 7 according to their route-utilization status.

Lastly, aiming to provide a major insight in the network dynamics, the highest probability routing links between nodes were selected based on the network topology graph (see Table 8). Based on these probability links, the most possible paths through the network for each node are presented in Table 10. From these results, it can be confirmed how motes with higher route utilization are used to forward packets from other motes. Since their main route is directly connected to the base station (mote 0), these motes are associated with a much higher route selection probability.

While the results from the field study show comparable results with experiments done in the laboratory setting, we note that there is still a considerable amount of missing information with respect to the effect of transmissions on the battery life of motes. The information transmitted via the health messages, which include a large portion of the missing information (e.g., retransmissions, dropped packets, forwarded packets, link quality, neighbor nodes, etc.), is not stored by default in the gateway for further investigation.

TABLE 10: Routes with highest probability for the different motes in the field test.

Mote ID	Route 1	Route 2
2504	3060–0 (16%)	2534–0 (11.5%)
2525	5000–2528–0 (6%)	5000–2528–2534–0 (2.7%)
2528	0 (27.4%)	2534–0 (12.4%)
2534	0 (47.8%)	—
3060	0 (52.8%)	—
5000	2528–0 (9.5%)	2528–2534–0 (4.3%)
5010	3060–0 (16.7%)	2534–0 (12.9%)
5020	3060–0 (16.3%)	5000–2528–0 (2.1%)
5030	2534–0 (15.6%)	—
5130	2528–0 (6.2%)	2528–2534–0 (2.8%)

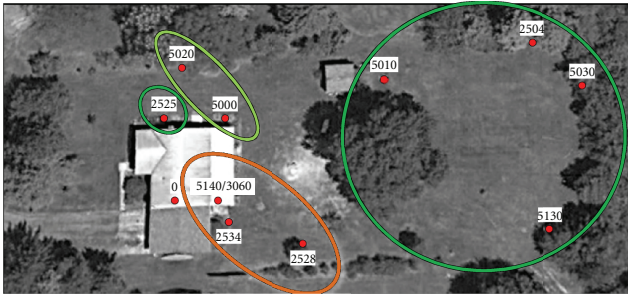


FIGURE 7: Locations of three mote groups in the prototype testbed according to their route utilization status. Groups highlighted in green correspond to the smaller route utilization, and orange corresponds to the highest.

One lesson learned in our experimental study is that the gateway in a WSN plays a critical role in the WSN reliability and robustness which could be easily overlooked at the beginning of WSN deployment. The gateway that was used in both the laboratory and field studies presented here was the Stargate Netbridge, a Linksys NSLU2 device specially modified by Crossbow. One attractive feature of the Stargate Netbridge is that it has a local web-based monitoring and management interface called MoteExplorer. MoteExplorer allows easy visualization of the network status, the current mesh topology, and a live stream of mote data collected by the base station, which made the Stargate Netbridge a favorable choice of the WSN gateway in deployment and operation. On the other hand, the Stargate Netbridge runs a Debian Linux operating system from an attached 4 GB flash drive. The deprecated version of Debian Linux came with limited and mostly outdated software. Due to the limited resources, in computational speed and memory storage, and an unconventional architecture (ARM), updates and upgrades to the device were not possible. With no direct access to the computer (monitor, keyboard, and mouse are not supported), all operations had to be performed through an SSH connection (for more information regarding the limitations on the Stargate Netbridge, see [21]). It has been discovered in our experimental study that the

Stargate Netbridge has severe reliability issues, and its strong limitations in hardware and extensibility make it an inconvenient solution for a practical WSN deployment. Following several unrecoverable crashes to the Stargate Netbridge, it was decided to move the gateway platform to a Linux x86 computer running Ubuntu Linux 10.04 operating system. While the new gateway platform would allow for faster, easier, and more reliable network operations, it raises an important challenge due to the lack of the convenient WSN management tool, MoteExplorer. In order to facilitate the change in the gateway platform to fundamentally improve the WSN system's robustness and reliability, we apply our general integrated network and data management solution [22] to the WSN testbed in this study. The following section describes the network and data management system and its application to the deployed outdoor WSN testbed. An example of its operation is also shown for the field study testbed described above.

#### 4. Network and Data Management System

A web-based integrated network and data management system called INDAMS, presented in [22], has been applied to the WSN testbed to facilitate the management and monitoring needs of real-world WSN deployments. This management system employs the following fundamental features: (1) separates the WSN management functions from WSN applications, (2) utilizes an accessible web-based user interface for management functionalities, and (3) systematically supports multiple WSNs from multiple platforms and technologies. Multiple users cannot only independently in real-time remotely access the WSN testbed operations via this management system, but also retrieve and monitor sensing data and network management information with unified web-based management tools that may not be available at local gateway system(s). This eliminates the complexity involved with users dealing with the complex commands and configurations of the WSN gateway (e.g., Memsic's XServe). Instead, this management system enables users to access the important and critical WSN management data for conducting network analyses. Integrated with the new gateway platform in this study, INDAMS not only successfully addresses the issue of regular network management needs in the operations of the WSN testbed, but also collects and saves the network health statistics that were previously discarded, most likely due to storage requirements of the Stargate Netbridge, such that comprehensive power and routing analyses can be conducted. In this section, we highlight some key aspects of the web-based WSN management system INDAMS, with the focus on its interaction with Memsic's XServe gateway used in our study, and the partial deployment in the WSN testbed (for the detailed description of the general INDAMS development, see [22]).

*4.1. Management System Architecture.* The overall architecture of the management system is illustrated in Figure 8. For the management server to communicate with the WSN gateway (in this case XServe), a representative agent is

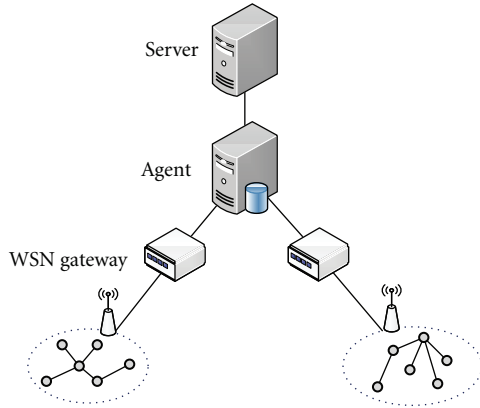


FIGURE 8: An illustration of the management system's overall architecture. The main server of the management system communicates with an on-site agent which interacts with the WSN gateway and network.

introduced which works directly with the WSN gateway. A management protocol, that is, the agent-server protocol, is designed to formally communicate between the management server and the agent. In this respect, the system's architecture is generic such that all management functions developed on the management server are independent to the WSN gateway platform. For example, in the event that the XServe gateway is replaced by another gateway platform only the agent's gateway interface needs to be modified. Thus, the agent-server protocol is the key component of the design and implementation of this management system.

The agent-server protocol is an application-layer protocol that is carried by TCP for reliable transmissions. In general, the functionalities of the management system are classified into two categories: control request/response functions and data functions. The agent-server protocol defines a specific control connection and a specific data connection for the control functions and data functions categories, respectively. The message exchange sequence of the agent-server protocol depends on each specific function request. An example of one of the most complex scenarios which involves message exchanges over both control and data connections between the management agent and the server is shown in Figure 9.

In Figure 9, the sequence of messages starts with messages  $m_1$  and  $m_2$  for the registration process. After  $m_2$  is received, the agent waits for requests. A client request (e.g., a remote user's request through the internet) to start data collection is translated and forwarded by the server, that is,  $m_3$ , which triggers a response by the agent, that is,  $m_4$  and  $m_5$ . After  $m_5$  is transmitted by the agent and received and processed by the server, the agent and server resume waiting for requests.

The data function represented in Figure 9 begins with a monitoring request, that is,  $m_6$ , from the server to the agent which responds with message  $m_7$ . The difference between the data collection and the monitoring request is the continuous use of the data connection for sending and collecting data from the agent by the server. When  $m_6$  is processed, the agent

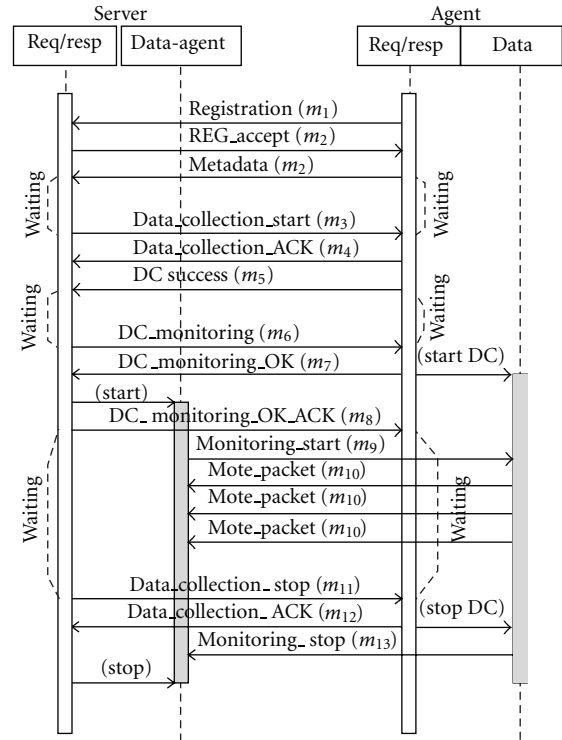


FIGURE 9: A complex data collection sequence which involves message exchanges over both control and data connections between the management agent and the server. The direction of each message's transmission is indicated by the message arrow, with the message's name and number shown above the arrow.

waits for requests on the data connection. After the server receives  $m_7$ , the server starts the data connection for the agent and sends an acknowledgement message back to the agent, that is,  $m_8$ . Following the transmission of  $m_8$ , the server waits for client requests. When  $m_8$  is received, the agent waits for requests to be forwarded by the server, while the message sequence continues on the data connection.

At this point, both the control and data connections are active. The server sends a monitoring data request, that is,  $m_9$ , to the agent which then begins sending the data packets back to the server, that is,  $m_{10}$ . Finally, the sequence to stop the monitoring function is given, that is,  $m_{11}$ ,  $m_{12}$ , and  $m_{13}$ . The data connection is finished with the closing of both sides of the protocol. Note that many acknowledgement messages are used to synchronize the execution at both sides of the protocol and to detect any possible errors at the other side.

In the management system, client requests are processed by the control and data handler component. The data handler subscribes clients as event listeners and assigns them parameters according to their request. After receiving data from the protocol server, the data handler decides which clients are going to receive the data and does so in the appropriate format. The data handler is capable of differentiating data sent from multiple agents and the data type. Moreover, it is a simple task to add new clients and new types of parameters. In this way, the management system

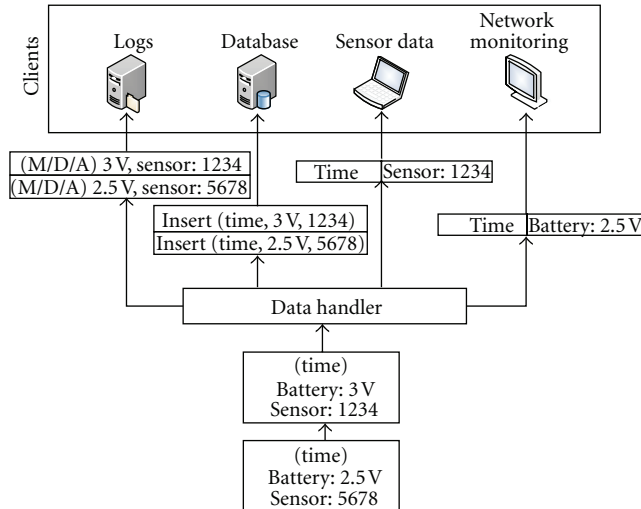


FIGURE 10: An illustration of operations of the data handler component when two packets are received. The data handler sends the notification information to each of the corresponding subscribed listeners separately.

provides flexible functionality to ensure the proper delivery of data to the appropriate clients. The process performed by the data handler in the control and data handler component is shown in Figure 10.

The agent for XServe WSN gateways was implemented in Java as a standalone application. The XServe agent implements the agent-server protocol with a communication interface to the management system. For this study, XServe's functions were classified into different types, such as data collection and network monitoring.

Xserve runs on a variety of platforms including Linux x86, which is the platform of the new gateway, and also uses a set of parameters to activate different functions of the protocol. A technology-specific section of the agent's metadata was assigned to store the parameters and values required for the agent to start XServe's application. XServe also provides an interface, that is, XServeTerm, to allow external applications to send commands, that is, XCommands, to the WSN or to the gateway itself.

The XCommands required by the agent are included in the agent's metadata for the purpose of mapping the function requests. Each request received by the agent can be mapped to a combination of parameters for XServe and XServeTerm. Therefore, when the agent receives a request, it checks the metadata for the appropriate mapping and syntax and communicates with XServe and XServeTerm applications via Java interprocess communication mechanisms. An example of the command mapping between the agent and the server for the data collection function, shown in Table 11, considers only the basic function parameters, that is, start and stop, which leaves all remaining values to default. In this case, the mapping for the request to start the data collection corresponds to a single command that executes XServe with a set of parameters. The mapping to stop the data collection corresponds to a single command that executes

TABLE 11: An example of agent/server command mapping.

Protocol server command	Agent command	
	XServe	XServeTerm
Data collection: Start	<code>Xserve-db-c-s=&lt;COM/USB port&gt;-xmlport=&lt;xml port&gt;</code>	(not required)
Data collection: Stop	(running)	<code>xservevterm xserve.shutdown</code>

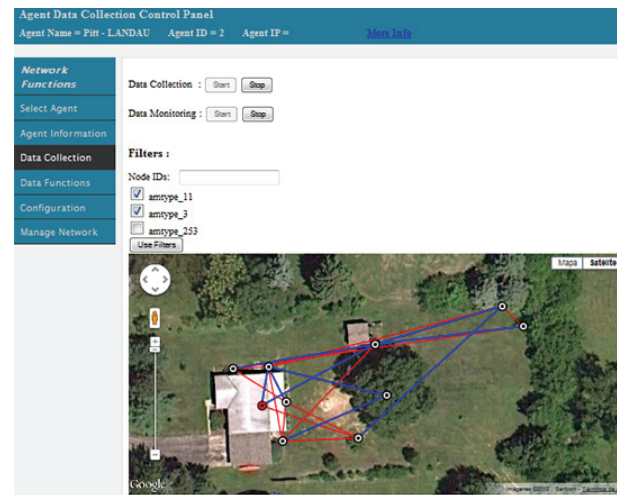


FIGURE 11: A recent screenshot from the web interface of the management system showing the topology monitoring feature for the prototype testbed. Blue links represent the link chosen by a mote to send a packet and red links represent the neighbors of each mote. Some adjustments have been made to the node locations since the network was originally deployed.

XServeTerm with the XCommand, that is, `xserve.shutdown`. More complex functions may require a combination of XServe and XServeTerm with different parameters and different sequences of execution.

**4.2. Field Deployment.** The developed management system is deployed and tested in the experimental testbed described in Section 3. The management system offers a geographical and topology monitoring feature as shown in Figure 11. This allows users to see a map of the mote locations in the WSN. Each node on the map shows the connections to each of mote's neighbors and highlights the mote's parent. The neighbor and parent information is received from the data and health packets transmitted by each mote. The map is updated continuously as the information is being received by the server. This provides a useful tool for a fast and updated view of the network's state and routing for each of the motes.

The management system also includes a live feed of the information received from the WSN gateway displayed in a rolling table as shown in Figure 12. All packets received from

Time Stamp	Packet Type	Node ID	Data			
2012-01-31 13:09:55.172	3	5020	neighborID2	3039	rsnl5	0
			neighborID1	2534	lquality3	12
			neighborID3	3060	lquality1	12
			node_count	3	lquality2	15
			old_type	15	rquality2	15
			pathcost2	4	rquality1	13
			pathcost3	9	rquality3	14
			pathcost1	9	socketid	3
			rsnl2	0	type	2
			rsnl1	0	version	1
			2012-01-31 13:09:53.379	253		
2012-01-31 13:09:48.496	253					
2012-01-31 13:09:45.614	253					
2012-01-31 13:09:28.55	11	2525	adc5	0.610352	board_id	129
			adc4	0.610352	group	130
			adc6	0.610352	humid	47.833076
			adc1	59.204102	humtemp	13.780000
			adc0	823.974609	parent	5020
			adc3	0.610352	packet_id	137
			adc2	656.127930	socketid	51
					voltage	3772

FIGURE 12: A screenshot from the web interface of the management system presenting the live feed information from the prototype network.

the motes, that is, data and health, and base station, that is, heartbeats, are displayed. Filters were implemented to specify the types of packets, for example, data or health, or packet fields, for example, specific node IDs, to be displayed.

Figure 12 shows the three types of data packets that are collected at the testbed site. The first is the health packet, type 3, which is giving the current neighbor statistics for node 5020. The second packet type shown is the base station heartbeat, type 253. This shows the user that the base station is on and working. The third type is the data packet, type 11, which shows the mote's sensor data, current configuration and parent ID used to transmit the data. All of the data that is collected, with the exception of the base station heartbeats, is stored in a database on the gateway and is recorded with their received time stamp by the gateway.

## 5. Conclusions

This paper thoroughly examines the power efficiency and battery savings, as well as some key characteristics of wireless mote transmissions in both laboratory and field settings, based on experimental study and following a quantitative approach. The major contributions of this work are summarized as follows.

(1) This study shows that motes in LP mode drop between 10–20% of their sampled measurements (see Tables 4 and 6). This is one of the drawbacks of operating the Memsic MICAz motes in LP mode, as opposed to the HP mode. In contrast, in the HP mode, mote measurements at 15 min intervals have less than 1% packet loss occurrence. While the LP mode decreases the success rate of data transmission compared to using the HP mode, it supplies around four times the battery life in comparison with that in the HP mode (see Table 2).

(2) We show that by adjusting the health message interval, an additional 15–46% in battery life can be gained (see Table 5). In addition, the multihop capability of the network can exploit certain motes which have relatively low path cost to the base station. This leads to motes serving as relays to have a shorter battery life than those that do not. This investigation shows that route-utilization can result in over 50% reduction in the battery life in motes in LP mode in the laboratory setting (see Table 6). Little to no route utilization was found in HP motes.

(3) The field study shows that the battery life of the LP motes was comparable to those tested in the lab. Route utilization was identified in motes with good link quality with the base station. The battery life of these motes was variable but generally less than motes located on the outer edge of the network which forwarded less packets. The dropped packet rate was significantly higher in the field than that in the laboratory setting, about 60% compared to 10–20%, respectively. The higher rate of dropped packets may be due to the low clearance of the motes above the ground and the stability of the network which was found to be questionable due to the high occurrence of parent changes.

(4) Our study reveals the practical vulnerability of the original Stargate Netbridge gateway in WSN testbed deployment, when the deployed WSN moderately scales up. To address this critical issue, a new gateway platform integrated with our general web-based WSN management solution is presented. Our solution not only succeeds in replacing the Stargate Netbridge gateway for reliable WSN deployment and operations, but also enables the network management data collection of all the important network health and neighbor statistics information which was not possible before due to the limitations of the Stargate Netbridge gateway. Indeed, this will allow for more comprehensive studies of the power consumption of WSN deployments in the future. Our network management system also further improves upon the functionality of the original web-based monitor provided by Crossbow on their Stargate Netbridge. This testbed study demonstrates the merits of the general integrated WSN network and data management system INDAMS presented in [22].

In conclusion, our work provides new quantitative insights into the power characteristics of practical WSNs using the currently most popular and commercially available wireless sensor networking technology, which would be useful toward energy-efficient WSN developments for environmental monitoring in real-world practice.

## Acknowledgment

This work was supported by NSF under CNS-0721474 and CNS-0758372 to the University of Pittsburgh and to IUPUI, respectively. Special thanks go to Mr. and Mrs. Lucas Landau for allowing us the use of their property for this field study. Thanks also go to Mr. Thomas Hare for his programming and analysis of the network data. Lastly, we thank all of the people whose time making comments and suggestions has helped to improve this work.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] Y. Liu, Y. He, M. Li et al., "Does wireless sensor network scale? A measurement study on GreenOrbs," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '12)*, pp. 873–881, April 2011.
- [3] G. Werner-Allen, K. Lorincz, M. Ruiz et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.
- [4] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: application-specific sensor network for environmental monitoring," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, article 17, 2010.
- [5] N. Xu, S. Rangwala, K. K. Chintalapudi et al., "A wireless sensor network for structural monitoring," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 13–24, ACM Press, New York, NY, USA, November 2004.
- [6] Z. Chaczko, A. Kale, and C. Chiu, "Intelligent health care—a motion analysis system for health practitioners," in *Proceedings of the 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 303–308, Brisbane, Australia, December 2010.
- [7] J. Beutel, K. Römer, M. Ringwald, and M. Woehrl, "Deployment techniques for sensor networks," in *Sensor Networks: Where Theory Meets Practice*, G. Ferrari, Ed., pp. 219–248, Springer, Berlin, Germany, 2010.
- [8] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The hitchhiker's guide to successful wireless sensor network deployments," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 43–56, ACM Press, New York, NY, USA, 2008.
- [9] A. Teo, G. Singh, and J. C. McEachen, "Evaluation of the XMesh routing protocol in wireless sensor networks," in *Proceedings of the 49th Midwest Symposium on Circuits and Systems (MWSCAS '06)*, pp. 113–117, IEEE, San Juan, Puerto Rico, August 2007.
- [10] D. Gay, P. Levis, R. V. Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: a holistic approach to networked embedded systems," *ACM SIGPLAN Notices*, vol. 38, no. 5, pp. 1–11, 2003.
- [11] "TinyOS," 2010, <http://tinysos.net/>.
- [12] Crossbow Technology, Inc., *XMesh Users Manual (Doc.#7430-0108-01) Rev. D.*, San Jose, Calif, USA, 2007.
- [13] 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*, pp. 88–97, New York, NY, USA, September 2002.
- [14] Crossbow Technology, Inc., *MPR-MIB Wireless Module Users Manual (Doc.#7430-0021-08) Rev. A*, San Jose, Calif, USA, 2007.
- [15] T. Davis, X. Liang, C.-M. Kuo, and Y. Liang, "Analysis of power characteristics for sap flow, soil moisture and soil water potential sensors in wireless sensor networking systems," *IEEE Sensors Journal*. In press.
- [16] A. S. Chipcom, "SmartRF CC2420 Preliminary Datasheet," 2004, <http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf>.
- [17] IEEE 802.15.4 Standard-2003, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE-SA Standards Board," 2003.
- [18] E. R. Musaloiu, A. Terzis, K. Szlavec, A. Szalay, J. Cogan, and J. Gray, "Life under your feet: a wireless soil ecology sensor network," in *Proceedings of the 3rd Workshop on Embedded Networked Sensors*, pp. 51–55, Cambridge, Mass, USA, 2006.
- [19] Crossbow Technology, Inc., *MoteView Users Manual (Doc.# 7430-0008-05) Rev. A.*, San Jose, Calif, USA, 2007.
- [20] Crossbow Technology, Inc., *XServe Users Manual (Doc.# 7430-0111-01) Rev. E.*, San Jose, Calif, USA, 2007.
- [21] F. Stanjo, D. Cvrcek, and M. Lewis, "Steel, cast iron and concrete: security engineering for real world wireless sensor networks," in *Proceedings of the 6th Applied Cryptography and Network Security Conference*, vol. 5037 of *Series Lecture Notes in Computer Science*, pp. 460–478, Springer, June 2008.
- [22] M. Navarro, D. Bhatnagar, and Y. Liang, "An integrated network and data management system for heterogeneous WSNs," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '11)*, Valencia, Spain, October 2011.