Energy Consumption of Wireless Sensor Nodes for Monitoring Forest Ecosystems

By Emma Miller

ABSTRACT

This paper presents an experiment conducted to measure the energy consumption of wireless sensor network nodes currently being used as part of an ongoing forest monitoring project at Duke University. It describes the experiment setup and method, summarizes the results found, including an estimation of energy consumption over one hour, and provides suggestions for where future effort in regard to energy can be focused.

1. INTRODUCTION

Battery lifetime is one of the most critical concerns in wireless sensor networks. In-depth study of energy consumption can prevent the loss of data and reduces time spent by researchers going out to the forest to replace batteries. To save energy sufficient planning has to be done to guarantee that nodes' hardware is operating correctly, algorithms are optimized, and that nodes aren't doing unnecessary work.

The wireless sensor network project in development at Duke University involves deploying five sensor networks, each comprising of twenty nodes, into the Duke Forest. The nodes' sensors collect data on light, soil moisture, and temperature levels, and the data is transmitted by radio to a central base station. The information retrieved will be used to study biodiversity, carbon levels and its affect on forests, and to predict future environmental change.

The wireless sensor network nodes being used as part of the ongoing project at Duke University are called Wisards and were custom-built for the project [1, 2]. The goal of this project is to measure and study Wisards' power and energy consumption and the significant events they implement.

2. RELATED WORK

Energy has similarly been measured and studied in other sensor network projects, and many tools have been developed to profile, estimate, or simulate energy consumption.

PowerScope [3] is a tool for measuring energy consumption of mobile devices. By analyzing the structure of a program it can determine energy consumption of the various processes of a system. It has been used successfully to decrease energy consumption of various applications.

PowerTOSSIM [4] is a wireless sensor network simulator that can estimate power consumption of Mica2 motes running TinyOS. In developing PowerTOSSIM motes' CPU, memory, radio, etc, were tested individually using a comparable method as in this project. The simulator also uses "block profiling" to measure power consumed by a mote's CPU. As blocks of code are executed they are mapped to the corresponding assembly language instructions, and this data can be used to count the total number of CPU cycles. CPU energy was measured in various states such as sleep, idle, and active. The real measurements found are used in the simulator, and have been found to be sufficiently accurate.

3. EXPERIMENT SETUP AND METHOD

The equipment used in this experiment included two Wisard nodes, two laptops and an Agilent 34402A multimeter. Of the two nodes, one played the role of the gateway, referred to as "master". It stores data gathered from other nodes. The other node played the role of the collection nodes, referred to as "slave", which take readings and send them (via multi-hop) to the gateway node.

The node under test was connected via serial port to one laptop which runs a program that displays the schedule of node events. The other laptop recorded energy levels to an Excel file at an interval of about 2 tenths of a second. The experiment setup is illustrated in figure 1.

Nine events are currently implemented on Wisards: Light, temperature, and soil moisture readings are gathered via probes connected to nodes that are deployed in the forest. "Send data" (SD) and "receive data" (RD) are part of discovery mode and occur when nodes are searching for other nodes to communicate with. "Send message" (SM) and "receive message" (RM) events deliver the data collected from the sensors, and are part of the process of sending readings to the gateway node. The battery (BAT) event records the current state of a node's battery, and the schedule event (SCH) calculates the schedule for the next frame. A list of events is provided in table 1.

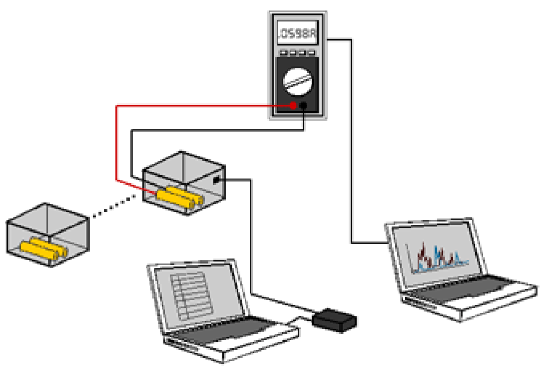

Figure 1: Experiment setup showing Wisard nodes (left and center), multimeter (top), and two laptops (bottom and right).

	Master	Slave
soil moisture	x	x
light	x	x
temperature	x	
battery	x	x
send message		x
receive message	x	
send data	x	x
receive data		x
schedule	x	x

Table 1: Wisard node events.

4. RESULTS

Graphs were created for the significant events that occur on the nodes. For light, temperature, soil moisture, and battery readings average energy was calculated over a 4 second slot. For the send and receive message events the energy consumed depends on the number of packets sent or received. Therefore, peaks in energy, rather than averages, were noted. Example graphs are shown in figures 2 and 3.

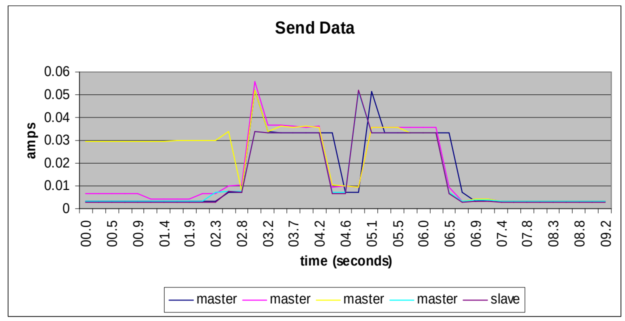

Figure 2: Current recorded during the send data event.

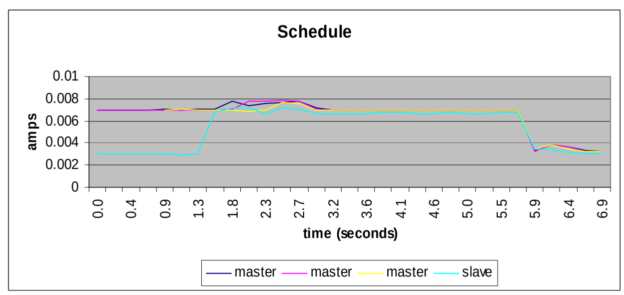

Figure 3: Current recorded during calculation of the schedule.

Relative energy was estimated over one hour based on the frequency of events and the averages calculated. On both the master and slave nodes it was found that idle time consumes the majority of total energy. On the master node send data consumed the second most amount of energy, followed by receive message and schedule. (This is the case partly because send data and schedule are the most frequent events, occurring about 14 times per hour). On the slave node send data consumed the second most amount of energy, followed by the schedule. The estimations are illustrated in figures 4 and 5.

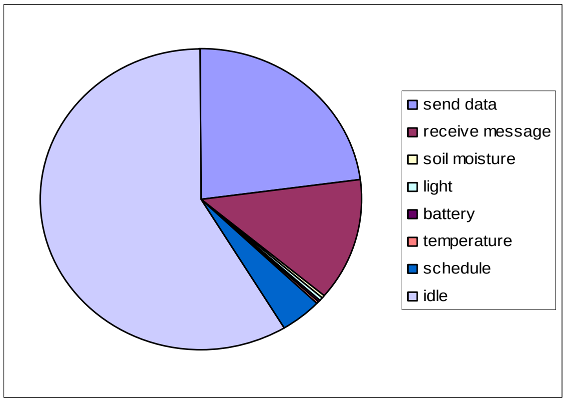

Figure 4: Master node – Estimation of relative energy over one hour.

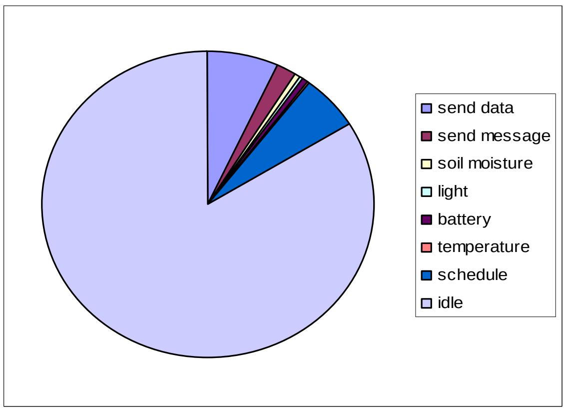

Figure 5: Slave node – Estimation of relative energy over one hour.

5. CONCLUSIONS

Judging by the results of the energy experiments two significant improvements can be made to reduce energy consumption.

First, energy consumed by the nodes while in an idle state could be reduced. One possible improvement that can be made is to switch the nodes to an ultra-low power state when they are not doing anything for extended periods of time. This is a known bug and is currently being fixed.

Second, rather than having the nodes wait for the completion of a 4-second slot they ought to immediately drop to low power after an event completes a reading, calculation, or transmission task. As can been seen in the schedule event (figure 3) the actual calculation finishes in about 1 second, but the energy does not fall until the entire 4-second time allotted has passed.

6. REFERENCES

[1] Z. Yang, B. Ruggeri, P.G. Flikkema, D. Johnson, M. Wright, K. Xia. “WiSARDNet: A System Solution for High Performance In Situ Environmental Monitoring,” in *2nd International Workshop on Networked Sensor Systems (INSS)*, 2005.

[2] P.G. Flikkema, P.K. Agarwal, J.S. Clark, C. Ellis, A. Gelfand, K. Munagala, J. Yang. "Model-Driven Dynamic Control of Embedded Wireless Sensor Networks," in *Workshop on Dynamic Data Driven Application Systems, International Conference on Computational Science (ICCS)*, 2006.

[3] J. Flinn and M. Satyanarayanan. “PowerScope: a tool for profiling the energy usage of mobile applications,” in *2nd IEEE Workshop on Mobile Computing Systems and Application (WMCSA)*, 1999.

[4] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, M. Welsh. "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.