Comparison of Wireless Ad-Hoc Sensor Networks

David Spinden, Jeffrey Jasper and Kurt Kosbar
Telemetry Learning Center
Department of Electrical and Computer Engineering
University of Missouri – Rolla
Rolla, MO 65409-0040

ABSTRACT

There are a number of telemetry applications where it would be helpful to have networks of sensors that could autonomously discover their connectivity, and dynamically reconfigure themselves during use. A number of research groups have developed wireless ad-hoc sensor network systems. This paper reviews the state-of-the-art in wireless ad-hoc networks, examining the features, assumptions, limitations and unique attributes of some of the more popular solutions to this problem.

KEYWORDS

Ad-Hoc Networking, Sensor Networks, Network Protocols

INTRODUCTION

A wireless ad-hoc sensor network must be able to take in sensor data, perform signal conditioning and data reduction, and propagate the measurements through a multi-hop wireless network until it reaches its final destination. There are many different ways to use wireless sensor networks. They allow interdisciplinary exploration of wildlife and natural environments that were never achievable without intrusion before. Because they are portable, they can be used to monitor tests such as seismic activity in a building without a large expense. They also can be used to monitor changing conditions in tight areas such as planes. All of these are possible because the sensor devices, the nodes (commonly referred as motes), are small and they do not require routing power and data lines long distances or through objects.

With the advent of less expensive hardware, wireless ad-hoc sensor networks have grown in popularity. Unfortunately, the technology is still in the early stages of development as a whole. There is not a fully complete wireless network solution that meets all requirements of different uses. Instead, there are many groups working on implementing network designs based off of many different hardware and software systems. Some are designed for a specific need; others are working towards a general implementation. Some of these groups have a mature amount of research and progress towards their set goals, while others have yet to reach even a small portion of their goals.

This paper compares wireless sensor research of several of the leading research groups. It examines: what each group has done and where they are in developing their final product, what features the have been implemented into the sensors, what the limitations of the project include, and some unique features that the group added.

TinyOS

TinyOS (TOS) is an open source operating system for wireless sensor networks designed at the University of California, Berkeley [1]. There is a comprehensive website [1] with links to the latest TOS release, tutorials, support, FAQs, related work, and hardware designs. Of all of the wireless ad-hoc sensor networks currently being developed, TOS appears to be the most mature. At the time this paper was written, the latest release (1.1.6), dated 5/17/2004, was available for download on the website. TOS has many features, some unique, that contribute to its popularity and continued growth. It also has some design and functionality limitations that need to be considered when choosing a sensor network to develop and/or implement. The mature developmental state of TOS is another aspect that is surely adding to its growing popularity.

The TOS system is a wireless ad-hoc network operating system rich in features and unique attributes. It was written in a C-based, event driven, language designed specifically for embedded systems called NesC [2]. Although NesC is modeled on the C programming language, it is not identical to C, and has a bit of a learning curve. All NesC applications are built around components and their interfaces. These components are bidirectional and give TOS its modular design of reusable components. These systems can handle real-time tasks by using two types of functions, high priority event handlers and low priority task execution functions. The lower priority tasks are posted to a queue and executed until the queue is empty or a hardware interrupt is received. When a hardware interrupt occurs, it preempts the tasks currently in the queue. After the hardware interrupt is processed the tasks are resumed and processed until the queue is empty. Once the queue is empty TOS goes into a sleep state until the next hardware event or task execution is received.

Power management and conservation capabilities are an integral part of TOS. Power management is achieved through the interaction of three key elements [3]: stopping services and placing hardware controlled by components into a low power state, a power management component placing the processor in the lowest power mode compatible with the current hardware state, and the ability of the TOS timer to function even in the lowest power consumption mode. Using these three elements with the help of coordination of database software called TinyDB [4], TOS is able to support deployments of months at a time by coordinating dynamically scheduled periods of time in which the mote “wakes up”, samples sensor data, transmits the data to other motes, and then goes back into a low power consumption mode.

TOS is rooted in a single hop communication protocol called active messaging [5] which enables the mote to send and receive single-hop broadcast messages and unicast messages. As the TOS hardware platform has evolved, so has the protocol stack containing active messaging. Another evolution, thanks to the contributions from the large community of TOS users, has been the inclusion of multi-hop routing. There are fully developed applications for TOS that contain various implementations of multi-hop routing algorithms.

Many different hardware platforms currently support TOS. Table 1 lists some of the different platforms and their release date. Data transmission in the older platforms (WeC, Rene, and dot) is limited to 10 kbps but changes in the protocol stack and radio transceiver has increased the transmission rate the approximately 40 kbps. Most of the TOS platforms have hardware schematics available for download on the website but fully functional commercial products are currently available for purchasing. Crossbow Inc. [6] has released the MICA, MICA2, and MICA2DOT platforms for TOS and also offers TOS on a platform developed by Intel called Stargate, which is compatible with the MICA2 platform. Recently, a new IEEE 802.15.4 compliant radio called TELOS was released by the Moteiv corp. TELOS has increased the data transmission rate to 250 kbps and also employs a USB PC to mote interface as opposed to a serial interface. On the WeC, Rene, dot, MICA, MICA2, and MICA2DOT platforms, TOS applications are downloaded from a PC onto a mote via a serial link.

Platform	WeC	Rene	Dot	MICA	MICA2	MICA2DOT
Release Date	1999	2000	2001	2002	2003	2003

Table 1. Hardware Platforms which support TOS

A large community of users has adopted TOS, leading to a large pool of projects which demonstrates its capabilities. These projects have necessitated advancements in the features of TOS, such as power consumption and routing. The Great Duck Island project [7] was a habitat monitoring study in which motes were deployed on an uninhabited island off the coast of Maine, and environmental conditions monitored over periods of months at a time. A shooter localization project [8] performed at Vanderbilt University used a network of TOS motes to measure the muzzle blast and acoustical shockwave of a firearm to successfully determine the location of the firearm and direction of the projectile being fired. Many more projects and research involving TOS can be found at the TOS website.

Working with, and developing, TOS can be enhanced with the use of other software and applications. Interaction with the motes can be accomplished using Matlab, a tutorial of which is located on the TOS website. TOSSIM [9] is a simulation application included in the standard release of TOS that allows networks of thousands of nodes to be simulated. TOSSIM can also be run with TinyViz, a customizable graphical interface allowing interaction with running simulations. Data collection and organization can also be enhanced using TinyDB[4]. TinyDB has an SQL like interface aiding in the collection, processing, and routing of collected data without having to write embedded C code.

TOS appears to be the most complete package when looking at the current state of wireless sensor network development. The website provides developers a large source of resources from development to debugging. The option of purchasing commercial hardware to implement TOS with makes it a “plug and play” wireless sensor network that can be operational in a short period of time.

MANTIS

While TinyOS has the largest development, there are many other groups working towards the same goals. One of those groups is the University of Colorado-Boulder Computer Science department. They have developed MultimodAl system for NeTwork of In-situ wireless Sensors (MANTIS). MANTIS [10] tries to cater to both beginners and experts in wireless networking by having a small learning curve and providing many advanced features.

MANTIS has many features that making it an excellent project for working with wireless sensors. The software uses the standard C language to program the notes. This makes the learning curve very small for designing new applications. The operating system, MANTIS Operating System (MOS) is multithreaded which allows applications to be much more complex. It also supports three different hardware platforms: the MICA2, Linux, and the groups own custom hardware, the MANTIS Nymph. By supporting Linux, emulation of nodes is simple and even can be networked to physical nodes. To aid in developing, MOS has a remote shell that can log into the nodes and debug them. The major advantage of using the Nymph nodes over the Mica2s is the ability to have a Global Positioning System (GPS) receiver on the node.

Even without the large number of developers that like TinyOS has, the project has progressed rapidly in the year since the first release of MOS. Colorado-Boulder group members have developed four versions of the software and working on adding many new features. Currently, they are working on the ability to dynamically reprogram the nodes over the wireless network. While there are reliability and security issues with reprogramming over the network, it is a large step forward in sensor networks. Members are also looking into ways to reduce power usage. As of yet, there is no low power operation, a large disadvantage. MANTIS uses custom protocol called Very Lightweight Mobile Multicast (VLM2) as the say foundation for communication. It has advanced features like a security layer that can block intrusions.

SCADDS

The Scalable Coordination Architectures for Deeply Distributed Systems (SCADDS) Project at the University of Southern California was setup to research wireless sensor networks [11]. Their research includes such topics as: mote radio communication stacks, time synchronization, and medium access control for wireless sensor networks (S-MAC). While the SCADDS group designed their own test bed hardware, they have focused on the TinyOS hardware platform.

The S-MAC project is of particular interest to wireless sensor networks. S-MAC is designed to save power by: implementing node sleeping at set times, making the radio sleeping when other nodes transmit, and message passing for applications that require data to be moved through the network. S-MAC also handles long messages efficiently. These techniques result in as much as 1/6 the power requirements of an 802.11 like MAC. Unfortunately, there is higher latency and every node may not have a fair chance to send information. S-MAC has been implemented into TinyOS as a separate communications stack starting with TinyOS version 1.1.0.

Based off of their website, it appears that the SCADDS group is no longer actively pursuing any research researching SCADDS technology. The projects have been left in various stages of development. Some papers are available describing the work and several projects have code available to the public.

µAMPS

One of the goals in wireless sensor networks is to make the sensor devises as little obtrusive as possible. To make sensors smaller the battery size must be reduced. The Massachusetts Institute of Technology µAMPS project group [12] is working on developing sensors that can make their own power using the sensor's environment, thus eliminating the need for batteries altogether and making the sensor much smaller.

MIT µAMPS has a working commercial off-the-shelf (COTS) node called the µAMPS-1. The design was designed with using as little power as possible. Most of the hardware components can be turned off or put in a low power state. The software was also designed to be energy conscience. The use of a multihop routing algorithm helps reduce reduces the radio transmission power.

The team is currently working on developing a second generation µAMPS that would take the digital part of µAMPS-1 and place it into an FPGA. This would be a midway step to the final goal of putting the µAMPS into two ASIC chips. The chips would use MEMS technology so that it could generate its own power and contain all the digital and radio components in a very small package.

The µAMPS project has a few drawbacks. There is no available hardware or software for others to use, as the MIT group is working alone. Placing everything on one chip has some design issues. Fabricating a new design is expensive, so any change in the sensor would be expensive. If the current design is not the final tested version, it is unclear from the µAMPS website how they plan on dealing with upgrades and changes.

Sensor Web

NASA's Jet Propulsion Laboratory (JPL) has also done research into wireless sensor networks. Their implantation is called Sensor Webs [13]. Since the start of the project in 1997, JPL has tested their Sensor Webs in seven deployments including at Kennedy Space Center and Antarctica, all with great success. The sensors, called sensor pods, can also be linked to an online real-time streaming graph that anyone can view from an Internet browser. This end-to-end solution allows for sensor data to be easily viewed and compared.

JPL is planning on sending Sensor Webs to other planets. Because the Sensor Webs have been tested in some of the harshest environments on earth, it seems likely that this goal is a good possibility. They are currently focusing developing more applications for the project, as their current design is stable and robust [14].

Little technical information is given about the project. What is known is the sensors support multihop routing, use the same 900 MHz ISM band that the MICA2 platform uses, and can handle many different types of sensors such as humidity, gas, and light levels [15]. None of the hardware or software is available for others and nothing technical is given about the software in the sensor pods or the computer software that generates the streaming graphs.

ZebraNet

Researchers at Princeton University have built wireless sensor nodes that help study the behavior of wild zebras in Kenyan wildlife reserve [16]. Their custom design uses a GPS on the wireless sensor to record each zebra's location. This data is then logged and shared with other nodes when they are in range. To retrieve data from many of the zebras, a person takes a mobile base station in range of one of the zebras. This project is an excellent example of how ad-hoc networking can be used.

There are many challenges that the group had to solve before having a working prototype. First, power consumption was a major concern. Princeton's custom operating system, Impala, was created with power needs in mind. When battery life is low, it adapts so that primary functions can continue, but at a slower rate. Instead of listen to the radio at all times, Impala uses periodic node discovery so that power can be saved.

While the project support advanced features such as dynamic software updates, and large flash memory storage, Princeton is the only developer. They have described their hardware and software well, but unfortunately have not made them available [17, 18]. In the future they plan on adapting the design to monitor other animal's behavior such as the Hyena. In order to do this successfully, the nodes must be reduced in size from its current 2.5 pounds. They would also like to add more sensors that would allow researchers to determine when animals are eating and drinking [19].

Other Projects

PicoRadio [20] is a technology researched at the Berkeley Wireless Research Center at UC Berkeley. The goal of PicoRadio was to create low cost (less than $0.05), low energy, and low power consumption transceivers by making the networks self-configurable, by using a system-on-chip (SOC) design, and by balancing trade offs between communication and computation. According to the website, as of June 2002 two of the three phases of work had been completed. The first phase was designing a PicoRadio test bed in order to test and analyze factors involved (protocol design, wireless communication, …) in designing a PicoNode. The second phase consisted of integrating a PicoNode onto two chips, designing a tool chain for project development, and exploration of alternative designs of wireless sensor nodes.

The Smart-Its Project [21], collaboration of Lancaster University, ETH Zurich, University of Karlsruhe, Interactive Institute in Sweden and VTT Electronics in Finland, intended to place sensors in common household items and have them interact using wireless communication. They were successfully able to build two types of prototypes: a Bluetooth device and a RFM radio transceiver device. They also performed research with RFID tags. Research on the project appears to have stopped as their website has not been updated in the past three years.

The Center for Embedded Network Sensing (CENS) at UCLA [22] is currently working on a large variety of implementations of sensor related projects although not all of them are wireless ad-hoc related. Some of the areas of research include habitat sensing, seismic monitoring and structural response, and contaminant transport monitoring. An interesting project to note is one involving acquisition of seismic data in a building at UCLA damaged during a 1994 earthquake. The sensor network consists of 10 Crossbow MICA2 motes. According to the CENS website they were successfully able to acquire, with 100 percent reliability, 20 seconds of forced vibration data in 5 minutes. More information on the various research projects underway can be found at the CENS website.

The Center for Multimedia Communications at Rice University has developed sensors called Generalized Network of Miniature Environmental Sensors (GNOMES) [23]. The some of the notable goals are deployment for collecting environmental and meteorological data. They have prototype hardware and source code available for download off of the Internet, but offer no information about GNOMES networking protocols or operating system.

The Energy Efficient Sensor Networks (EYES) [24] project is a 3 year European project beginning in 2002 and ending in 2005. The focus of the project is the design, implementation, and testing of a whole new communications technology and networking and service guidelines. These guidelines are to be tailored to the unique requirements of wireless sensor networks. The EYES website contains documentation of the project plans and theory behind sensor networks, but has no implementation as of yet.

There are also commercial organizations creating wireless ad-hoc sensor networks. Crossbow, as mentioned earlier, bundles motes with the TinyOS operating system. Companies such as Dust Networks [25], who also has roots based out of UC Berkeley, and Sensoria [26] also offer an end product available for purchase. As new applications for wireless ad-hoc sensor are discovered, undoubtedly the list of companies offering commercial wireless sensors will grow.

CONCLUSION

As wireless ad-hoc sensor networks become more popular, the need for practical solutions becomes more important. While the popularity of this topic is evident from the numbers of projects that exist, there exist only a few groups that have successfully demonstrated functioning networks. TinyOS appears to be leading the way with its large user base of both individuals and commercial entities pushing development forward with new hardware and software tools. Projects such as MANTIS are contributing to the development of the technology by making a much more user friendly C based implementation. All of the projects encountered generally focus on the similar goals of efficiency in terms of network function, power consumption, and small size. As these and other project mature, new ideas and applications for ad-hoc wireless sensor networks materialize so will the number of successful implementations of functioning networks. These networks promise to help us learn more about our world in ways that were impossible just a few years ago.

REFERENCES

[1] TinyOS website maintained by the University of California, Berkeley, http://www.tinyos.net .

[2] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler ”The nesC Language: A Holistic Approach to Embedded Systems”, *In Proceedings of Programming Language Design and Implementation (PLDI 2003)*, San Diego CA, June 2003.

[3] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer, and D. Culler, ”The Emergence of Networking Abstractions and Techniques in TinyOS”, *In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004),* San Francisco CA, March 2004

[4] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong ”The Design of an Acquisitional Query Processor for Sensor Networks”, In Proceedings of *Special Interest Group on Management of Data(SIGMOD 2003)*, San Diego CA, June 2003.

[5] J. Hill, P. Bounadonna, and D. Culler. ”Active Message Communication for Tiny Network Sensors”. http://webs.cs.berkeley.edu/tos/papers/ammote.pdf .

[6] Commercial website for Crossbow Inc., http://www.xbow.com

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. ”Wireless sensor networks for habitat monitoring”, *In Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, Atlanta GA, USA, Sept. 2002.

[8] Janos Sallai, Gyorgy Balogh, Miklos Maroti, and Akos Ledeczi. ”Acoustic Ranging in Resource Constrained Sensor Networks”, Vanderbilt University Institute for Software Integrated Systems (ISIS) Technical Report, http://www.isis.vanderbilt.edu/projects/nest/documentation/Vanderbilt_NEST_Acoustic_Ranging.pdf, 2004

[9] P. Levis, N. Lee, M. Welsh, and D. Culler, ”TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)* Los Angeles CA, Nov. 2003.

[10] MANTIS website maintained by the University of Colorado, Boulder Computer Science Department http://mantis.cs.colorado.edu/index.php

[11] SCADDS website at the ISI/University of Southern California Viterbi School of Engineering , http://www.isi.edu/scadds/

[12] Microamps website at the Massachusets Institute of Technology, http://www-mtl.mit.edu/research/icsystems/uamps/

[13] Sensorwebs website maintained by NASA/JPL, http://sensorwebs.jpl.nasa.gov/

[14] K.A. Delin, ”A Distributed, Wireless Monitoring System” Sensors Online, April 2004, http://www.sensorsmag.com/articles/0404/20/ .

[15] K. A. Delin and S. P. Jackson. “Sensor Web for *In Situ* Exploration of Gaseous Biosignatures”. *Presented at the IEEE Aerospace Conference 2000*, Big Sky Montana

[16] Zebranet Project Website at Princeton University, http://www.princeton.edu/~mrm/zebranet.html

[17] Ting Liu and Margaret Martonosi. ”Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems” *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP 2003)*, San Diego CA, June 2003
[18] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li-Shiuan Peh, and Daniel Rubenstein. "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet". *the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems.* October, 2002
[19] Geoff Watts, “Zebranet”, British Broadcastng Corporation, http://www.bbc.co.uk/radio4/science/leadingedge_20021205.shtml
[20] Website for PicoRadio maintained by the University of California, Berkeley, http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/
[21] Smart-Its Project website, http://www.smart-its.org/
[22] CENS project website maintained by the University of California, Los Angeles, http://www.cens.ucla.edu/
[23] GNOMES project website at Rice University, http://cmclab.rice.edu/sensors/index.cfm
[24] EYES Project website, http://eyes.eu.org/
[25] Commercial website for Dust Networks Inc., http://www.dust-inc.com/
[26] Commercial website for Sensoria, http://www.sensoria.com/