Sentry-Based Power Management in Wireless Sensor Networks

Jonathan Hui, Zhiyuan Ren, and Bruce H. Krogh

Carnegie Mellon University, Dept. of Electrical and Computer Engineering
5000 Forbes Avenue, Pittsburgh PA 15213-3890, USA
`{jhui@andrew|zren@andrew|krogh@ece}.cmu.edu`

Abstract. This paper presents a sentry-based approach to power management in wireless sensor networks for applications such as intruder detection and tracking. To minimize average power consumption while maintaining sufficient node density for coarse sensing, nodes are partitioned dynamically into two sets: sentries and non-sentries. Sentry nodes provide sufficient coverage for continuous monitoring and basic communication services. Non-sentry nodes sleep for designated periods of time to conserve power, and switch to full power only when needed to provide more refined sensing for tracking. Non-sentry nodes check for beacons from sentry nodes to determine when they should remain on. Experimental results are presented demonstrating trade-offs between power savings and tracking performance for a network of seventeen nodes using the first implementation of a basic sentry-based power management scheme. The paper concludes with a brief description of a full set of power-management services being implemented as middle-ware for general wireless sensor applications.

Keywords. Power management, wireless sensor networks, tracking

1 Introduction

This paper introduces a sentry-based approach to power management in wireless sensor networks. Nodes are classified dynamically into two categories, sentries and non-sentries. Sentries operate in full power mode, providing a backbone communication network and basic application functionality. Non-sentries operate in a low-power dormant state whenever they are not required to help with the sensing tasks. Sentries wake up the non-sentries when they are needed, as determined by the current operating context for the network.

We present the concepts of sentry-based power management (SBPM) in the context of intruder detection and tracking as a target application. Wireless sensor nodes form a large-scale, ad-hoc network. For object tracking, the network must respond quickly with respect to the object. This implies that the network must be optimized for low latency and high throughput communications. On the other hand, it is common for nodes to exhibit long periods of inactivity whenever there is no activity within the ranges of their sensors. Therefore, power can be saved by turning off nodes when they are not needed for the tracking task.

F. Zhao and L. Guibas (Eds.): IPSN 2003, LNCS 2634, pp. 458–472, 2003.

Power management should achieve a good trade-off between these objectives. It should allow as many nodes to turn off at any time, while leaving enough nodes on to maintain a multi-hop path between any two nodes. This implies that a coarse network of nodes must remain on to form a connected backbone. Also, the nodes that remain on need to be sufficient to perform the task, at least in a coarse mode. In particular, the nodes that are on need to be sufficient to sense the presence of an intruder in areas where most nodes are turned off. The SBPM approach presented in this paper is designed to fulfill the above requirements.

2 Related Work

The recent SPAN [1] scheme of Chen et al. has similar goals to those of SBPM. In SPAN, a sparse network of coordinators is created. This approach is similar to our sentry-based approach where the sentries can be considered coordinators. However, SPAN attempts to create a network of coordinators with minimal loss of network capacity throughout the entire network. When dealing with object tracking, it is often the case that network capacity is not an issue in areas far away from the object.

In AFECA [2], each node maintains a count of the all their neighboring nodes within radio range by listening to broadcast signals. A node transitions between power-down and power-on states by randomly sleeping for a specific amount of time proportional to the number of nearby nodes. The general effect is that the number of nodes remains roughly constant. As the density increases, the amount of power savings increases. However, AFECA's decisions on sleeping time are fairly conservative to ensure that there is a high probability of creating a fully connected graph among nodes to allow an ad hoc network to form. SBPM differs from AFECA in that a node is not left on unless it is absolutely necessary. Also in SBPM, always-on nodes are chosen so that a fully connected graph is virtually ensured.

In GAF [3], the nodes make use of their geographical location information to divide up all nodes into fixed square grids. The size of these squares stays constant regardless of node density. Nodes within a grid power-down and wake-up with the guarantee that at least one node per square is on at all times to maintain a backbone network amongst all nodes. SBPM differs from GAF in that groups are formed dynamically with nearby neighbors rather than specific geographic coordinates.

Other papers on power management focus on taking advantage of multiple levels of power consumption that are available in wireless sensor nodes. Minimum-energy routing [4] minimizes energy consumption by varying the transmission power. Chang and Tassiulas [5] enhanced this method to allow for more uniform power dissipation across all nodes by varying the transmission power across all nodes fairly. In this method, nodes adjust their transmission power by choosing routes that minimize energy consumption. DPM [6] saves power by taking advantage of multiple power states. Individual components are shut down when not needed and are woken up when necessary. The net effect is a set of power states to which a node can transition. The sleep states differ by the amount of power consumed, the time to transition in and out of the states, and what services they provide. DPM does not make use of sentries to coarsely monitor and maintain the network. Another approach is taken by LEACH [7]. This protocol selects multiple coordinators to aggregate data and send the data to a base station. These techniques for exploiting multiple power states, varying transmission power, and data aggregation complement the issues addressed by SBPM. They can be applied to any system where powered on nodes form a fully connected network. Thus, these methods could potentially be combined with SBPM.

3 SBPM Design

There are several characteristics of wireless sensor nodes that limit their functionality. Wireless sensors often have limited energy due to the use of batteries. They also use unsophisticated wireless links that limit the range, reliability, and throughput of communication between nodes. This implies that total network capacity is limited. SBPM attempts to minimize the total power usage over all nodes, maximize the network capacity utilization, and maintain full connectivity among all nodes. For this approach, we must assume that there exists a multi-hop path between any two nodes when all nodes are powered on. We also assume that the nodes have at least three power states. In the power-on state, all of the node's components required for the application are powered on. In the power-off state, the microprocessor is placed in a low-power state while the rest of the components are turned off. In the checking state, a node is only providing power to the microprocessor and radio transceiver.

3.1 Sentries and Non-sentries

To initialize SBPM, a subset of nodes in the network are selected as *sentries*. In our current implementation, the set of sentries forms a spanning tree, that is, each sentry is able to communicate with at least one other sentry. The remaining *non-sentry* nodes are partitioned into groups assigned to each sentry so that every non-sentry is able to communicate with any other node though one or more sentries. Thus, the sentries provide a backbone network that allows communication between any two nodes in the network.

Sentries must also be chosen so that the sensing task can still be performed in at least a preliminary or degraded manner. For object tracking, each sentry provides coarse tracking information by detecting the presence of an intruder in its sensor range. Thus, the second criterion for the selection of sentries is that they provide complete coverage with respect to sensing. With coarse sensing, the intruder can be detected and surrounding non-sentries can be then be turned on to enable higher-resolution tracking.

Sentries are also responsible for waking up nodes that are in the power-off state. The non-sentries notify their sentries when they transition to the power-off state, and declare when they will check for a beacon to see if they need to return to the power-on state.

Since sentries persist in the power-on state, their power consumption is significantly higher than the non-sentry power consumption. While the sensor network is active, the sentry assignments can be rotated to share this power burden. This rotation is achieved by weighting the costs of assigning sentries according the the available remaining energy at each node. We are currently working on efficient procedures to solve the problem of optimal dynamic sentry assignment.

Figure 1 shows the state transition diagram of a node. The main purpose of the non-sentries is to provide more detailed information about the environment when it is needed. In the power-on state, they continuously monitor the area using onboard sensors. Non-sentries differ from sentries in that they are allowed to enter the power-off state when their contribution is not necessary for accurate object tracking. Non-sentries stay in the power-off state for a specific amount of time. When that time expires, the non-sentry enters the checking state to check if it is necessary to completely power on. If the node is not needed, it returns to the power-off state.

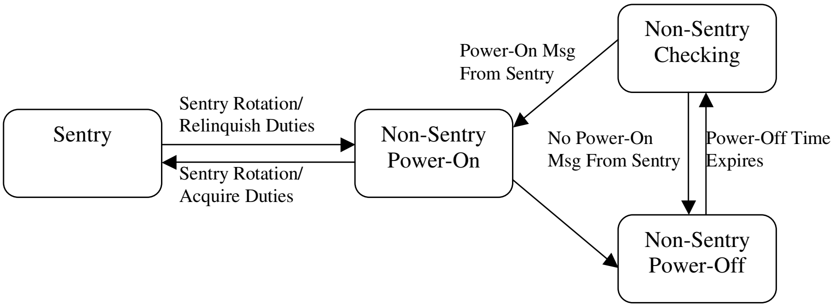

Fig. 1. Node state transition diagram.

The amount of time a non-sentry stays in the power-off state before entering the checking state can be determined right before the non-sentry enters the power-off state. Picking this time correctly is not trivial. A power-off time that is too short does not optimize the amount of energy savings. However, choosing a time that is too long may harm the performance of the sensor network. A node that transitions to a power-off state for a long time may not check to enter the power-on state in time to provide adequate timely information for the task at hand. Section 5 presents results of experiments to study the trade-offs among these considerations for our current implementation.

3.2 Implementation Details

Figure 2 shows where SBPM fits as a service within the overall communication architecture for each node. SBPM communicates directly with both the routing layer and the MAC/physical layer. In our current implementation of SBPM, the routing layer is not notified when nodes transition between the power-on and power-off states. Although optimizations can be made to the routing layer when SBPM provides such notifications, it adds complexity to the integration of the two modules.

Most routing implementations attempt to be fault-tolerant. If a node fails, all paths through that node will be rerouted through other nodes. Due to this feature, we leave the responsibilities of route reconfiguration to the routing layer. Thus, messages can be communicated across any nodes that are powered on. However, when all but a few non-sentries are in the power-down state, the sentries provide a minimal network that can be used to send messages between any two nodes in the power-on state. Although the routing layer plays a passive role, an implementation of SBPM that provides callbacks can be fairly trivial. Since the non-sentry entering the power-off state must send a message to its sentry, this message can be turned into a broadcast, which also notifies neighboring motes of its transition. Our current implementation does not include this callback feature.

SBPM is implemented as two modules as shown in Figure 2. One module provides the services for power management while the other provides the services for sentry management.

SentryGroupManagement. This module provides a generic interface that allows the use of sentries in an application. This module provides the necessary functionality for sentry selection and group management. A group is defined as a sentry and all of the non-sentries that report to it. As shown in Figure 3, the SentryGroupManagement module uses both the routing layer and the power management module. The routing layer is necessary to negotiate with surrounding nodes when selecting sentries and maintaining groups. Also, the PowerManagement module can be helpful during the sentry selection process. Sentries use significantly more power since they generally need to handle more tasks than non-sentries. As a result, selecting a node that has more remaining energy allows for more uniform energy dissipation across all nodes. If such a feature is not requested, the PowerManagement interface should be implemented such that it always returns the same value when sampling a nodes remaining energy. The commands in the SentryGroupMgmnt interface provide the functionality that an application might need when requiring the use of sentries.

PowerManagement. This module provides a generic interface for use of a power management scheme, including the necessary functionality for a node to transition between power states. As shown in Figure 4, the PowerManagement module makes use of both the routing layer and the SentryManagement module. The routing layer is required so that a sentry and its non-sentries can communicate with each other whenever a transition between power states occurs.

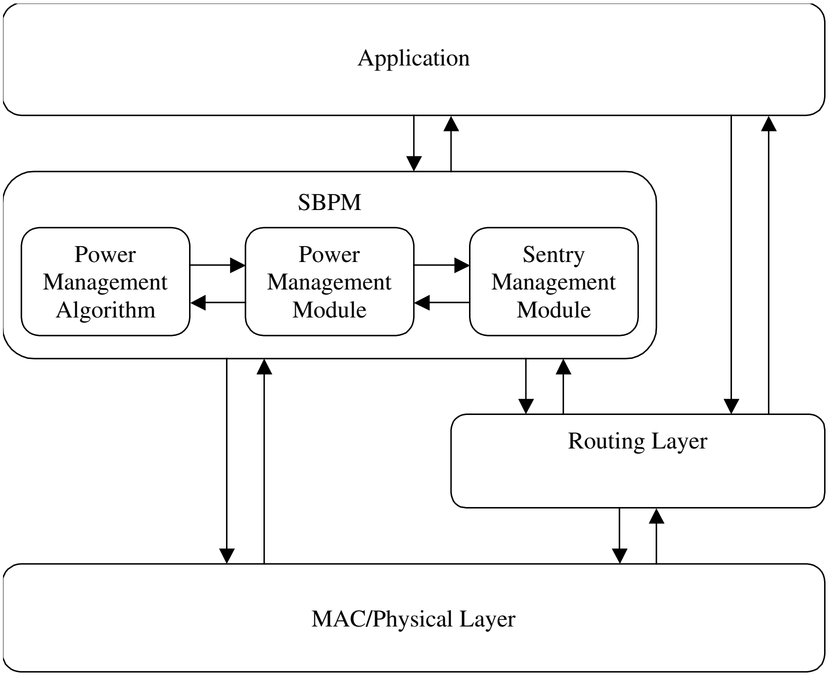

Fig. 2. SBPM is a protocol that uses both the MAC/Physical Layer and the Routing Layer. SBPM is split up into three different modules.

Since SBPM requires the use of sentries, the SentryGroupManagement module is required. This module is needed so that non-sentries can obtain the address of their sentries. The module is also useful when the PowerManagement module requests for a sentry to pass its duties along to another node. The PowerManagement module does not provide the decision making process of when nodes should transition between power states. The PowerMgmntAlgorithm module makes this decision. Having the decision making process as a separate module makes it possible to quickly change between different decision algorithms.

4 Experiment: Intruder Detection and Tracking

Out test bed to implement and study the performance of sentry-based power management is based on a network of *motes*, wireless sensor nodes designed at UC Berkeley and produced at Crossbow Technologies [8]. Each mote has a transceiver, a micro-controller, and a sensor board that includes a photo sensor, temperature sensor, magnetometer, accelerometer, speaker, and microphone. In our experiment, a photo sensor is used to demonstrate intruder detection and tracking. In a real system, sound, sonar, or some other sensor would be used.

```
module SentryGroupManagement {
    provides {
        interface SentryGroupMgmnt;
    }
    uses {
        interface ReceiveMsg as RxMsg;
        interface SendMsg as TxMsg;
        interface PowerMangement;
    }
}

interface SentryGroupMgmnt {
    command result_t startSentryGroupMgmnt(void);
    command result_t stopSentryGroupMgmnt(void);
    command result_t isOn(void);
    command result_t changeSentries(void);
    command addr_t getMySentry(void);
    command result_t getNumNonsentries(void);
    command result_t isSentry(void);
    command result_t isNonsentry(void);
    command result_t isBase(void);
    command result_t declareBase(void);
    command result_t undeclareBase(void);
}
```

Fig. 3. SentryGroupMgmnt module interface and commands.

SBPM is implemented on top of TinyOS [8], a minimal operating system developed for the motes.

Figure 5 shows our experimental setup, which was inspired by the experimental setup described in [9]. Eighteen motes are mounted on a vertical board in a skewed array, as shown in Figure 5. One of the motes is tagged as a base mote and is directly connected to a computer though a serial connection. The base mote does not participate in sensing. In our system, a computer is directly connected to a projector that shines an image on the sensor array. In this experiment, as shown in Figure 5, the "intruder" is a dark circle with a radial gradient. The radial gradient allows a mote to measure its distance from the intruder by measuring the light intensity. The intruder has a radius equal to the distance between any two motes in the system. This forces the intruder to cover at least three motes at any time while inside the sensor array. The motes can then use triangulation to an estimate the position of the intruder.

In our initial implementation we pre-select four motes from the sensor array to act as sentries. The rest of the motes act as non-sentries while the base station serves as a link between the sensor array and the computer. In order to execute the experiment, a simple static routing layer was developed. Due to the static nature of the routes, each non-sentry is assigned a sentry and sends all messages

```
module PowerManagement {
    provides {
        interface PowerMgmnt;
    }
    uses {
        interface ReceiveMsg as RxMsg;
        interface SendMsg as TxMsg;
        interface SentryGroupMgmnt;
        interface PowerMgmntAlgorithm;
    }
}

interface PowerMgmnt {
    command result_t startPowerMgmnt(void);
    command result_t stopPowerMgmnt(boolean
    stop_sentry_mgmnt);
    command result_t pausePowerMgmnt(void);
    command result_t isOn(void);
    command result_t updateIdletime(uint16_t idle_time);
    command result_t currentEnergyLevel(void);
}
```

Fig. 4. PowerMgmnt module interface and commands.

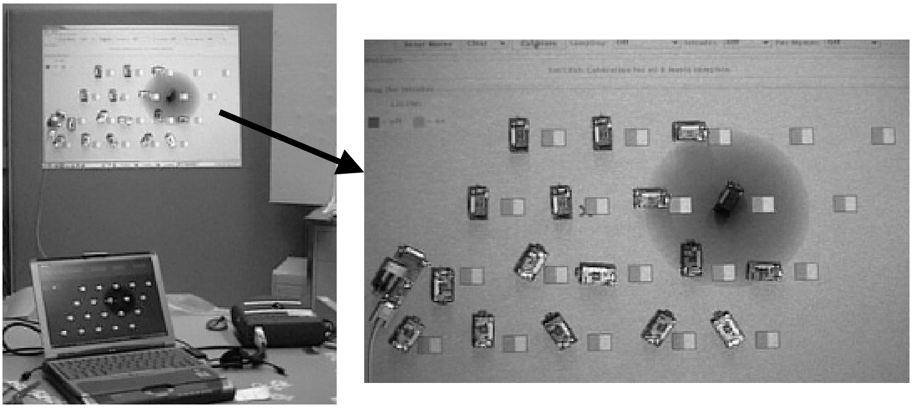

Fig. 5. Setup for light-based intruder detection and tracking experiments. The intruder is the dark circle with a radial gradient.

to that sentry. The sentries act as a backbone network forwarding all messages towards the base mote. This allows all non-sentries to power-down and maintain communication throughout the sensor array without changing any of the routing parameters.

Due to variance in ambient light depending on where the experiment is set up and the projector used for the experiment, the sensor array is calibrated whenever the sensor array is first turned on. To calibrate every mote efficiently, we make use of the computer to create an automated calibration process. For simplicity, we have divided the range of light intensity into a fixed number of levels. During calibration, the computer shines a specific light intensity level on every mote and sends a message to the sensor array informing them which light intensity level is currently being projected. Every mote replies upon successful calibration.

Our initial implementation takes a centralized approach to object tracking. Whenever a mote senses a change in light intensity, it sends a message to the computer through the sensor array. The computer takes the data points and triangulates an estimated position of the intruder. A data validation scheme has also been implemented to keep erroneous data from affecting the estimation. In this scheme the computer chooses data points in which the most motes agree. In situations where an equal number of motes agree on different data points, then the motes reporting a stronger presence of the object are chosen for triangulation.

The power management scheme is also centralized in our initial implementation. The computer turns a group of non-sentries off if the intruder is more than one hop away from any of the motes in the sensor array. We consider one hop as a movement from one triangle to another, where lines between neighboring motes define the triangles.

When a non-sentry receives a message commanding it to enter the power-off state, it goes into the power-off state for a set amount of time. This amount of time is constant throughout the operation of the sensor network. Part of the experiment includes the affect of this time on tracking performance and energy consumption. While in the sleep state, power to the transceiver and sensor board is turned off. Also, the micro-controller is placed into a low power mode that preserves the contents of registers and allows a wake-up interrupt. A non-sentry periodically enters the checking state and remains in that state long enough to determine whether or not its sentry is commanding it to enter the power-on state. If there is no beacon from the sentry, it returns to the power-off state.

5 Results

We present results from experiments performed to evaluate the effects of varying several of the design parameters. All experimental results in this section are single-run results. The path of the intruder through the sensor array was fixed to the path shown in Figure 6.

Figure 7 displays the amount of energy consumed with respect to the object movement speed. One curve shows the total energy consumed without power management and another curve shows the total energy consumed with power management. As can be seen in Figure 7, there is a fairly consistent energy savings of about 30% with power management on. This percentage is biased to the low end since the size of the object is large relative to the sensor array.

Expanding the sensor array would decrease the ratio of motes in the power-on state to those in the power-down state.

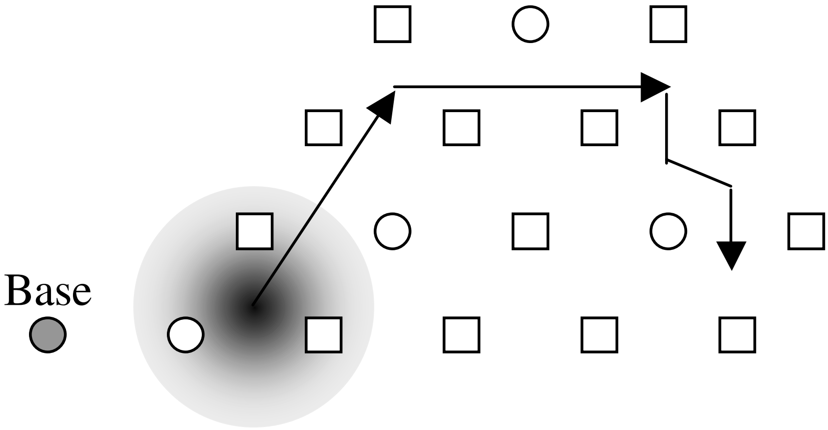

Fig. 6. The path of the intruder through the sensor array used to gather results.

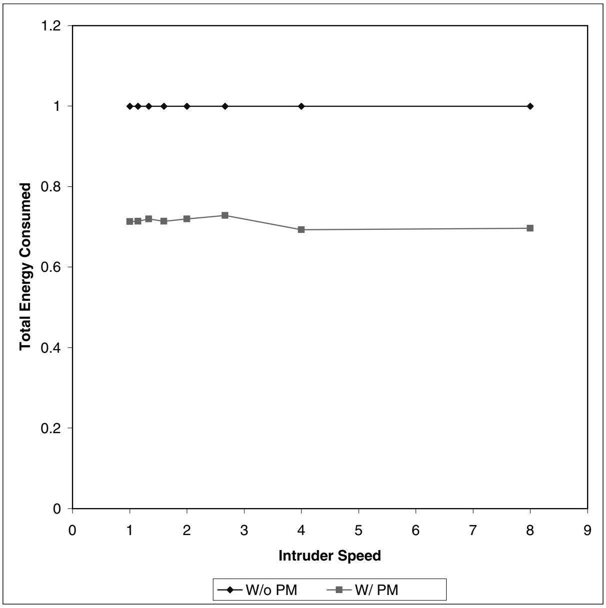

Fig. 7. Energy consumption vs. object movement speed.

For the sentry assignment scheme in our current experimental setup, the ratio of sentries to non-sentries would approach 1/4 as the number of nodes increases. Also, for our current implementation, motes in the power-down state are actually on about 20% of the time (4 seconds sleeping, 1 second checking for power-on message). Thus with a large sensor array, there is a theoretical energy reduction of 60% in our current setup.

We varied the object movement speed while keeping the power-down duration constant. To evaluate the effects of objected movement speed on tracking error, we repeated the experiment for several intruder speeds with power management on and once with power management off. As can be seen in Figure 8, the average tracking error at slow object speeds is roughly equivalent to the quantization error due to the limited light intensity levels. However, as object movement speed increases the amount of tracking error increases in both cases of power management on and off. Due to the limited size of the sensor array, the tracking error is limited to the size of the sensor array. The increase in tracking error can be attributed to the limited network capacity and centralized approach of processing data. As a result of the centralized data processing, this rate increases with the speed of the object for two of reasons. When the object moves more quickly, the motes send messages at a quicker rate because they sense changes in light intensity more often. The increase in number of messages increases the number of collisions, thus increasing the number of retries. The number of retries delays a mote's ability to send a message. As the mote is trying to send a message successfully, the object moves on, which in turn induces even more motes to begin sending messages.

The difference in tracking error when power management is on and power management is off is negligible at slow object movement speeds. When the sensor array can react to the object's motion, all sensors around the object are in the power-on state and provide enough information for accurate estimations. However, at high object movement speeds the tracking error with power management is significantly larger than without power management. Two factors cause this characteristic. When the object is moving quickly, it is more difficult to make correct power management decisions due to the increase in tracking error. Also, if the object moves faster than the sensor array can respond, motes around the object may not be in the power-on state and cannot contribute data that would provide a more accurate estimate.

We also varied the power-down duration while keeping the object movement speed constant. Figure 9 illustrates the relationship between tracking error and power-down duration while keeping the object movement speed constant. The power-down duration directly affects the sensor array's ability to react to object movements. As can be seen by Figure 9, at small values of the predetermined power-down duration, the tracking error remains somewhat constant. When the power-down duration is small enough, the sensor array can react quickly enough to ensure that all nodes around the object are in the power-on state. However, as the predetermined power-down duration increases, there is an increase in

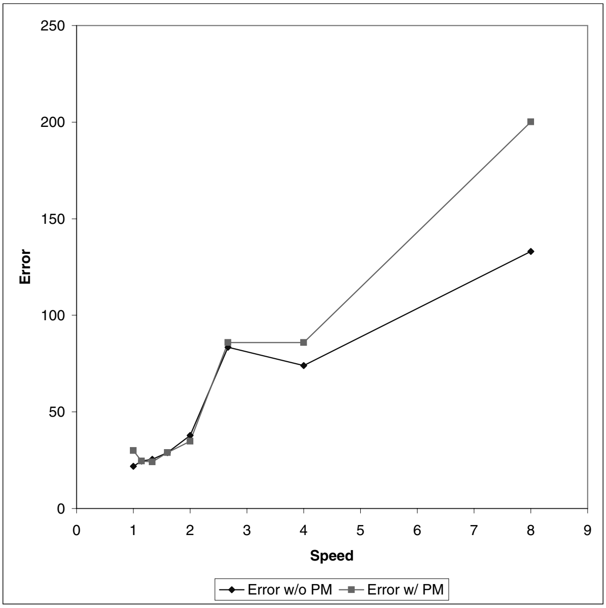

Fig. 8. Tracking error vs. object movement speed with power management on and power management off.

tracking error. This is caused by the sensor array's inability to react quickly enough to ensure that all motes around the object are in the power-on state.

Figure 10 illustrates the relationship between energy consumption and power-down duration while keeping the object movement speed constant. Increasing the power-down duration decreases the frequency at which a mote needs to power on and check whether it needs to switch into the power-on state. As can be seen by Figure 10, there is a small but noticeable decrease in total energy consumption as the power-down duration increases. The small magnitude in difference is due to the fact that the initial power savings is relatively small as explained above.

6 Conclusions

This paper presents SBPM, a sentry-based power management scheme in wireless sensor networks. SBPM makes use of sentries to define a coarse network of nodes that are necessary to perform the task and turn on non-sentry motes when necessary. Our initial implementation demonstrates the feasibility of such a sentry-based approach. Significant issues remain to be explored.

A dynamic sentry selection algorithm would allow for sentry rotation and fault tolerance. In our initial implementation, sentries were pre-selected and fixed. Fixed sentries are not amenable to optimal power savings. Sentries must remain in the power-on state at all times and are burdened with the need to

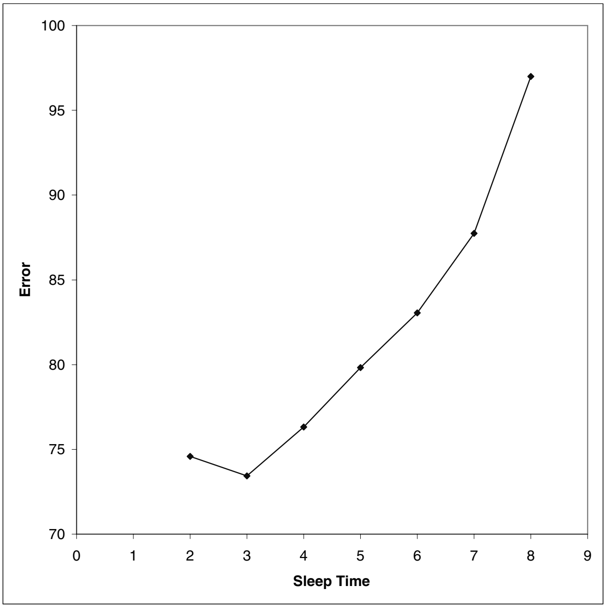

Fig. 9. Tracking error vs. power-down duration.

send messages when non-sentries are in the power-down state. Thus, sentries consume energy much faster. For optimal performance, we would ideally like to have an even rate of energy consumption across all nodes. Fixed sentries also cause problems when dealing with fault tolerance. In our setup, sentries are critical to maintaining both a coarse network and waking up non-sentry motes. Without the ability to dynamically choose sentries, we are assuming that the chosen sentries will never fail.

A distributed object tracking algorithm could minimize messages across the network and ultimately minimizes tracking error. In our current initial implementation, all data is sent to the computer for processing. However, if the processing can be done on the motes themselves, then only one message needs to be sent to the computer. Also, the data can instantly be broadcasted to neighboring motes for other purposes, rather than having the results being sent back out by the computer.

A distributed stochastic-based power management decision algorithm would allow for far less total energy consumption across all nodes. By making power management decisions directly on the motes, the sensor array can react much quicker and require less messages being sent. Our initial implementation also turns whole groups on and off. The nodes in the power-down state also sleep for a fixed amount of time before checking whether to power-on. Varying the amount of time to sleep will allow us to find the optimal point between tracking error and energy savings.

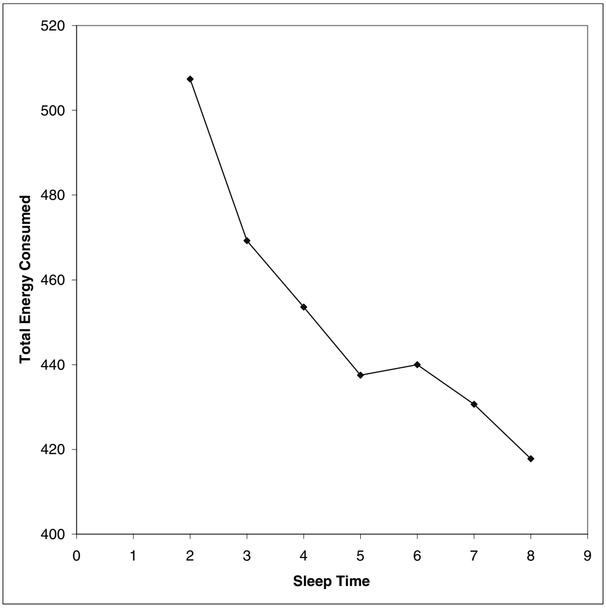

Fig. 10. Energy consumption vs. power-down duration.

As shown by our experiments, the first steps have been taken successfully in implementing a sentry-based object tracking and power management approach for wireless networks. Even in our initial implementation, we have experimentally shown that significant energy can be saved.

Acknowledgment. This research has been supported in part by the DARPA NEST (Network-Enabled Software Technology) program.

References

1. Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In: Proceedings of Mobicom 2001. (2001) 85–96
2. Xu, Y., Heidemann, J., Estrin, D.: Adaptive energy-conserving routing for multihop ad hoc networks. Technical Report Tech. Rep. 527, USC/ISI (2000)
3. Xu, Y., Heidemann, J., Estrin, D.: Geographically-informed energy conservation for ad hoc routing. In: Proc. Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Rome (2001)
4. Shepard, T.: A channel access scheme for large dense packet radio networks. In: Proc. ACM SIGCOMM. (1996) 219–230
5. Chang, J., Tassiulas, L.: Energy conserving routing in wireless ad hoc networks. In: Proceedings of IEEE INFOCOM, Tel Aviv, Israel (2000)

6. Sinha, A., Chandrakasan, A.: Dynamic power management in wireless sensor networks. IEEE Design and Test of Computers (2001) 62–75
7. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocols for wireless microsensor networks. In: Proceedings of the Hawaaian International Confference on Systems Science. (2000)
8. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for network sensors. In: Proc. Ninth International Conf. on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA (2000)
9. Liu, J., Cheung, P., Guibas, L., Zhao, F.: A dual-space approach to tracking and sensor management in wireless sensor networks. In: First ACM International Workshop On Wireless Sensor Networks and Applications in conjuction with ACM MobiCom 2002. (2002) 131–139