A Protocol for Wireless Sensor Networks based Routing Cost

Alpana P. Adsul and Shrikant K. Bodhe

Abstract—**As wireless sensor networks continue to attract more attention, new ideas for applications are continually being developed, many of which involve consistent coverage of a given surveillance area. Recently, several protocols and architectures have been proposed to maintain network connectivity and adequate coverage quality while minimizing the drain on the scarce energy resources of the sensor nodes. In this paper, we propose ACBR, an integrated protocol for routing and coverage preservation that is distinctly different from the previously proposed solutions. Guided by the intuition that certain sensors are more important to the sensing application than others because of limited neighborhood redundancy, we introduce a new routing metric — an "application cost" — that aims to avoid the use of sensors in areas of critically sparse sensor deployment as routers. We have implemented ACBR using MATLAB and present simulation results showing the effectiveness of the protocol in extending network lifetime.**

I. INTRODUCTION

A major research challenge in the design of wireless sensor networks is the unattended operation of networks for months or even years at a time. Impeding the realization of this goal is the fact that the energy supply of the sensor nodes, typically either small batteries or energy harvested from the environment, is very limited. However, as the cost of manufacturing sensor nodes continues to decrease, large-scale networks consisting of thousands of sensors will become realizable in the near future. The dense deployment of these networks creates redundancy in the information provided by sensors, allowing many sensors to be turned off for extended periods of time in order to conserve energy. An efficient way to manage the energy resources in the network is to develop a schedule (either predetermined or dynamic) that determines which sensors should send traffic at which times. The schedule should guarantee that the cumulative data from the activated sensors at any given time is sufficient to meet application quality of service (QoS) requirements (e.g., coverage of a certain fraction of the environment).

In addition to the sensing of the environment, another of the essential services provided by each sensor is the routing of other sensors' data to the sink(s). As sensor nodes are provided solely to support the sensor network application, each node should only be used for sensing or routing if this is the best role for that sensor to play to support the end goal. If global information about the network topology and sensing capabilities and initial energy of each node is available, it is possible to optimize the sensor schedule and data routing so that a maximum network lifetime is achieved for a given application quality requirement [1] [2]. Such methods require a large amount of processing capability at one of the nodes, which cannot always be assumed. Also, the optimized schedule is not robust to unexpected sensor failures and may need to be recalculated as new sensors die.

Finally, since the optimal solution is presumably calculated once during the initial stages of the network, it does not take advantage of the deployment of additional sensor nodes in the network.

For purposes of simplicity and scalability, solutions with distributed control and dynamic decision making are usually preferable over global optimizations. In this paper, we propose a distributed, integrated sensor management and routing protocol.

The protocol (ACBR- Application Cost Based Routing) allows sensors to become active as network coverage quality demands and to sleep whenever possible during the remainder of the time. The main contributions of ACBR are:

- nodes consider the cost to the entire network in their decision to become active,
- nodes use an application cost during route calculation, the first such routing protocol metric that attempts to avoid routing through critical sensors, and
- nodes that have activated themselves are given a chance to reverse their decision if other nodes in the network are subsequently activated so that the node is no longer necessary, ensuring that sensor sets are always chosen efficiently.

In this paper, we have chosen to focus on QoS requirements of coverage preservation. Coverage preserving protocols and algorithms have many potential applications, including intruder detection, biological/chemical agent detection, and fire detection. Also, these protocols and algorithms can be used in the initial stages of many target tracking architectures, where a more detailed description or location estimate of a phenomenon is required only when a tripwire threshold is crossed in the measurements of some of the active sensors.

The ACBR architecture could be used in networks with other models for QoS, although new "application cost" functions, different from the one proposed in this paper, would need to be developed for these applications.

The rest of this paper is organized as follows. Section II addresses related work. Section III presents ACBR and gives a discussion of various issues relating to the protocol that should be considered. Section IV provides simulation results and analysis of ACBR. Section V concludes the paper.

Mrs. Alpana P. Adsul, Lect., Dept. of Comp, Vishwakarma Institute of Technology, Bibwewadi, Pune.
Dr. Shrikant K. Bodhe, Prof. Dept of E & T, College of Engineering, Bharti Vidyapith University, Pune.

II. RELATED WORK

A. Sensor Management

Several energy-efficient protocols have been proposed under the classification of topology control. These protocols select which nodes should remain active as routers at a given time, while allowing others to sleep [3] [4] [5]. Most of these protocols make decisions based on network routing considerations but do not consider any application requirements specific to sensor networks such as sufficient coverage of an area.

Other coverage preservation protocols have been developed to provide consistent environmental coverage and robustness to unexpected sensor failures. In PEAS [6], sleeping sensors periodically enter a probing state, querying all sensors within a probing range (based on communication and/or connectivity requirements), and become active if no active sensors exist within the desired probing range. In [7], the problem of sensor selection was modeled as a Gur game, where sensors operate as finite state machines and change states (sending traffic only in selected ones) based on feedback from the base station, which is based on the state of the network's resolution. [8] proposes a round scheduling scheme in which sensors exchange reference times and schedule themselves around their own reference time, guaranteeing that the environment is completely covered at all times. [9] proposes a distributed selection algorithm for coverage preservation in sensor networks, in which a sensor measures its neighborhood redundancy as the union of the sectors/central angles covered by neighboring sensors within the sensor's sensing range. In CCP (Coverage Configuration Protocol) [10], sensors consult an eligibility rule, in which each sensor finds all intersection points between the borders of its neighbors' sensing radii and considers itself eligible for deactivation if each of these intersection points is covered with the desired sensing degree.

The aforementioned protocols generally aim to provide consistent coverage while ignoring the impact that active sensors will have on other sensors in the network, specifically the additional sensors that are required to route data. The algorithm presented in [11] considers routing implications when activating sensors. The goal of this algorithm is to find a minimum set of sensors and additional routers necessary in order to cover a given geographical region. At each iteration of the algorithm, the sensor with the best combination of i) a short path to the active subset and ii) a large number of additional unique sections covered and the sensors along that sensor's path are selected for inclusion in the sensor set.

B. Routing Protocols:

The field of ad hoc routing has been explored extensively. Initially, protocol design focused on efficiently finding shortest path routes in the presence of node mobility [12]. Later research addressed the need for energy based metrics to be used in energy-efficient ad hoc routing protocols. Singh et al. proposed several routing costs based on the residual energy of individual nodes [13]. Chang et al. proposed a routing cost that was a combination of residual energy, normalized residual energy, and required transmission energy and found an optimal combination of these parameters [14]. In this work, we build on the work of [13] and [14] and develop a routing cost for use specifically in wireless sensor networks, where the property of node redundancy is important. Our proposed routing cost is based not only on a sensor node's residual energy, but also the residual energy of redundant neighboring sensors.

Li et al. have proposed distributed energy-efficient routing algorithms that limit the amount of overhead messages involved in route calculation [15]. In these algorithms, delays that are proportional to a node's cost are introduced before forwarding route request messages, reducing the number of packets sent while ensuring that the minimum cost paths are found. We follow this approach in ACBR's route discovery mechanism, discussed in Section III-B.

III. ACBR - APPLICATION COST BASED ROUTING

A. Application Cost:

Results from previous work have suggested that providing joint solutions for data routing and active sensor selection can be beneficial [2]. Specifically, sensors those are more important to the sensing application (i.e., those that are located in sparsely deployed areas) and those whose residual energy is least should not be chosen as routers over those who are less important to the application (i.e., those with more redundant neighbors) and those with more residual energy. Guided by this intuition, we propose the use of an "application cost," which considers not only the residual energy of the sensor to whom the cost is being assigned, but also that of its redundant neighboring sensors.

In this work, we have assumed an application model where the entirety or a portion of a region needs to be monitored by any one or multiple sensors that are within their sensing range of that location. We assume for simplicity that a sensor can detect the presence of a phenomenon within its sensing range with perfect reliability. Let $C(S_j)$ represent the area that sensor S_j is capable of monitoring. In the simplest case, $C(S_j)$ consists of a circular area around the sensor with radius equal to its sensing range. Each location is capable of being monitored by a number of sensors. We will call the total energy at one location $E_{total}(x, y)$, defined as

$$E_{total}(x, y) = \sum_{S_j : x, y \in C(S_j)} E(S_j) \quad (1)$$

where $E(S_j)$ represents the residual energy of sensor S_j .

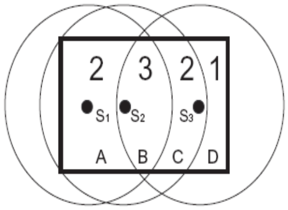


Figure 1: Proposed topology of Sensor Network.

The relative value of a sensor to the task of continuously monitoring a given location is the fraction of energy in the region that it contributes. The overall value of a sensor $V(S_j)$ is the maximum of the sensor's value over the space that the sensor is capable of covering.

$$V\left(S_j\right) = \max \frac{E\left(S_j\right)}{E_{total}\left(x, y\right)} \quad x, y \in C\left(S_j\right) \quad (2)$$

For example, consider the network shown in Figure 1, where the rectangular area is the region to be monitored and sensors S_1, S_2, and S_3 are capable of monitoring the regions within the circles representing their respective sensing ranges. It can be seen that region A can be covered by 2 sensors (S_1, S_2), while B, C, and D can be covered by 3, 2, and 1 sensor(s), respectively. If, for simplicity, we consider that all sensors have equal initial energy, sensor S_1 contributes ½ of the energy toward monitoring region A and ⅓ of the energy toward monitoring region B. Since the duties of monitoring each region can be divided among the capable sensors, S_1 is most valuable for its contribution to region A and is thus assigned a value of $V(S_1) = ½$. Similarly, sensors S_2 and S_3 are assigned values of $V(S_2) = ½$ and $V(S_3) = 1$, respectively.

In some cases, the use of this value function by itself as a routing metric will perform poorly. For example, consider two sensors S_1 and S_2, each with sole responsibility for covering some portion of the desired coverage area ($V(S_1) = V(S_2) = 1$). If one sensor has much more residual energy than the other, clearly this sensor is preferable for use as a router, a fact that would not be accounted for with the use of the value function as the routing metric. Thus, we divide the value function by the sensor's residual energy to obtain the final application cost.

$$\begin{aligned} C_{app}\left(S_j\right) &= \frac{V\left(S_j\right)}{E\left(S_j\right)} \\ &= \max \frac{1}{E_{total}(x, y)} x, y \in C\left(S_j\right) \end{aligned} \quad (3)$$

Note that several sensors, whose least redundantly covered portions of the monitored region consist of overlapping portions, will have identical application costs, regardless of their individual residual energy. This follows the intuition of our design since these sensors are equally effective at monitoring this portion. The only requirement is that one of the sensors is alive as long as possible. However, since it is undesirable to choose a sensor that is expected to die in the middle of a round, sensors with extremely low energy should be considered undesirable candidates. The manner in which this issue is dealt with is discussed in the following subsection.

While this application cost can be used for the sensor network models presented in this paper, other methods for determining application costs may be used for different sensor network applications that do not conform to this coverage model. In developing an application cost, the general goal is to provide information about the importance of the individual sensors to the sensing application.

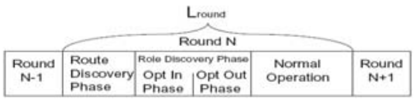


Figure 2: An ABCR round.

B. *The ACBR Protocol:*

In ACBR, time is divided into rounds, during which a sensor's roles (sensing and routing) typically remain unchanged. The beginning of a round consists of a Route Discovery Phase, followed by a Role Discovery Phase that is divided into an Opt In Phase and an Opt Out Phase. Upon completion of the Role Discovery Phase, sensors resume normal activity until the beginning of the next round, as shown in Figure 2.

The beginning of a new round is triggered by a "Round Start" message that is sent by the data sink. The data sink may be a central base station in the network or a (possibly rotating) cluster head. While these "Round Start" messages should typically be periodic, there is no explicit need for synchronization at the agent level, since these messages serve as a self-clocking mechanism. The "Round Start" messages are forwarded out by each sensor and also act as a mechanism for sensors to discover an optimal route to the base station or cluster head.

In ACBR, routes are calculated so that minimum cumulative cost paths are used. As we have mentioned, one of the major contributions of ACBR is the use of an application cost as a routing metric. Application costs are assigned to individual sensors and the cost of activating a sensor for a given route is a weighted sum of the "work" that each sensor must perform.

Specifically, the cost of a link is calculated using the sensors' application cost as

$$C_{link}\left(S_i, S_j\right) = C_{app}\left(S_i\right) * E_t + C_{app}\left(S_j\right) * E_r \quad (4)$$

where E_t represents the energy required to transmit a packet and E_r represents the energy required to receive a packet. The cumulative cost of a sensor's activation is

$$\begin{aligned} C_{act}\left(S_{src}\right) &= \sum_{\left(S_i, S_j\right) \in \rho\left(S_{src}, S_{dst}\right)} C_{link}\left(S_i, S_j\right) \\ &= \sum_{\substack{S_i : \left(S_i, S_j\right) \in \rho\left(S_{src}, S_{dst}\right) \\ S_i \neq S_{src}}} C_{app}\left(S_i\right) * \left(E_t + E_r\right) + \end{aligned}$$

$$C_{app}\left(S_{src}\right) * E_t + C_{app}\left(S_{dst}\right) * E_r \quad (5)$$

where p(Ssrc, Sdst) represents the set of links along the chosen optimal path from Ssrc to Sdst. As a sensor's application cost is dependent on the residual energy among itself and neighboring sensors rather than solely its own residual energy, it may happen that a sensor expected to die very soon has a low application cost and is chosen as a router. While this is not counterintuitive to the general design of our protocol, as explained in Section III-A, the situation can have costly effects on the network. Specifically, if a sensor is chosen as a router and dies during the round for which it is chosen, costly route repair mechanisms must be invoked. To avoid this situation, when a sensor's residual energy falls below a predetermined threshold, its application cost is calculated assuming no neighborhood redundancy, so that it is inversely proportional to the sensor's individual residual energy.

Delays are introduced before forwarding the "Round Start" messages, as in the protocols described in [15], so that the number of transmitted packets in minimized. These delays are proportional to the routing costs so that ideally, the "Round Start" packets always arrive at a sensor along the desired shortest cost path first and thus, there is no need to forward multiple copies of a packet to announce more optimal routes. After the Route Discovery Phase, sensors attempt to activate themselves if necessary. A sensor will assign itself an activation delay, proportional to the cost of its route, after which it may send an Opt In beacon, informing its neighbors that it intends to become active for the coming round. If, after the activation delay, enough Opt In beacons from neighboring sensors have been received so that the sensor may conclude that its neighborhood is entirely covered, the sensor withholds its beacon and remains inactive for the coming round.

It is common that after a sensor decides to become active, enough neighbors subsequently send Opt In beacons that the sensor's neighborhood is covered with a higher degree than the application requires. For this reason, ACBR reserves an Opt Out phase for these sensors to reverse their decision. The sensors with the highest route costs are given highest priority to opt out and so the order in which sensors send the Opt Out beacons is reversed from that in which they send the Opt In beacons. These Opt In and Opt Out beacons may be sent over a single hop if it is assumed that the transmission range is significantly higher than the sensing range (at least twice as great). If we cannot make this assumption, the beacons must be forwarded (through controlled flooding) until they reach all sensors that redundantly cover at least some portion of the sending sensor's coverage region (i.e., those within twice the sensing range). After the Opt Out phase, the subset of sensors to be used during the current round has been established and normal network operation resumes.

C. *Considerations for Clustering Networks:*

Because of the many-to-one nature of traffic patterns in sensor networks, deploying a static data sink or choosing one from among the previously deployed sensors causes a critical hot spot in the network around the sink. Unless sensors are deployed with much higher density near the sink, the lifetime of the sensors surrounding the sink should be expected to be much shorter than those far from the sink. Thus, rotating the sink location among the deployed sensors is critical in wireless sensor network applications such as those that we are focusing on. This is similar to a clustering approach where the sink resides at the cluster head. In fact, the ACBR architecture can be used for operation within a cluster for many proposed clustering architectures.

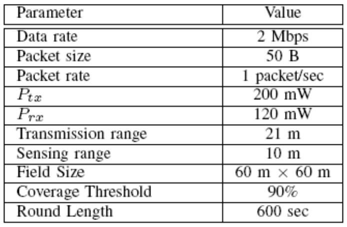
The selection of a cluster head may be performed separately at a higher layer or else within the ACBR protocol, requiring little overhead. Toward the end of a round, the current cluster head may send a unicast packet to the desired cluster head for the next round, informing it of its upcoming role. The decision of which sensor to choose may be random or, more appropriately, based on one of a number of criteria (e.g., highest energy, smallest application cost, or highest connectivity).

Of course, the current cluster head must gather this information from the sensors in the network in order to make this decision. Toward the end of a round, data packets from active sensors may piggyback the sending sensor's value of the metric used for cluster head selection. Of course, this leaves only the actively sensors as potential cluster head candidates for the next round. Alternatively, routing sensors may snoop the packets and update the relevant packet fields if their metric value makes them more desirable as a cluster head. Under this approach, all active sensors and routers are potential candidates for the cluster head role in the next round.

IV. SIMULATIONS AND ANALYSIS

A. *Methodology*

We simulated ACBR by writing a customized simulator in MATLAB. In our simulations, active sensors sent constant bit rate traffic to a cluster head chosen from among the deployed sensors. The cluster head rotated throughout the course of each simulation in order to avoid high energy drain in the sensors surrounding a static data sink. In the simulations, whenever shortest path routing and shortest energy cost routing were used, the cluster head for the next round was chosen from among the potential candidates as the sensor with the highest residual energy. When shortest application cost routing was used, the candidate with the smallest application cost was chosen as the next cluster head. The default parameters that we used in our simulations are summarized in Table I.



Parameter	Value
Data rate	2 Mbps
Packet size	50 B
Packet rate	1 packet/sec
P_{tx}	200 mW
P_{rx}	120 mW
Transmission range	21 m
Sensing range	10 m
Field Size	60 m × 60 m
Coverage Threshold	90%
Round Length	600 sec

TABLE-I: SIMULATION PARAMETERS.

The most important metric to quantify the effectiveness of our algorithm is total network lifetime. In this paper we define network lifetime as the last time at which the percentage of area covered remains above a predetermined threshold (90%). Coverage degrades as a result of the death of all sensors in a region or the lack of a route from any sensor in a region.

In radios where idle power consumption is comparable to the transmit and receive power consumption, protocols such as ACBR may not have such a large effect on overall power consumption or network lifetime. While this can be assumed with such MAC protocols as IEEE 802.11's DFWMAC, it is not necessarily the case with more energy-efficient MAC protocols, such as some TDMA-based solutions, since they allow sensors to sleep during periods of inactivity. In our simulations, the underlying MAC layer that we used was an "ideal" reliable TDMA MAC that falls under this classification. This MAC is considered "ideal" in the sense of power consumption because there is no overhead in establishing transmission schedules, and idle power consumption was assumed to be zero.

B. *Results*

1) Effectiveness of the Application Cost:

In order to verify the effectiveness of the application cost in ACBR, we compared versions of the protocol in which a sensor's cost was 1) constant (fewest hops), 2) the energy cost ($1/E$), and 3) the application cost (as presented in Equation 3).

In the first simulations, 150 sensors with equal initial energy were randomly deployed in a field with a region of the field purposely left more sparsely populated than others. The network lifetimes of ACBR with the different cost assignments are shown in Figure 3. The results show that while application cost assignment performs best, energy cost assignment also performs well. Energy cost assignment performs well because in the early stages of network operation, the sensors in the sparsely deployed areas are used most frequently, leading to an immediate drop in energy and rise in energy cost. This causes them to be avoided as routers very early.

If the initial energy of the sensors is not equally distributed, the benefit from using application cost increases, as shown in Figure 4. In these simulations, the energy of some sensors in the sparsely deployed areas is initially high, giving them a low energy cost and causing other sensors to believe that they are attractive candidates as routers. Not until their battery levels decrease to the levels of the other sensors do they begin to be avoided. Meanwhile, when using the application cost, these sensors are avoided, even in the early stages of the network.

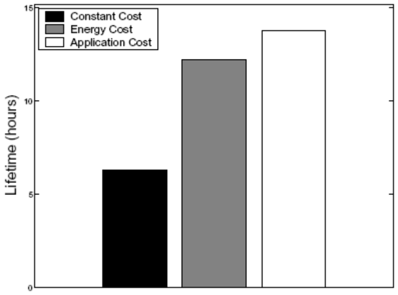


Figure 3: Protocol performance under different cost assignments for a network with equal energy distribution.

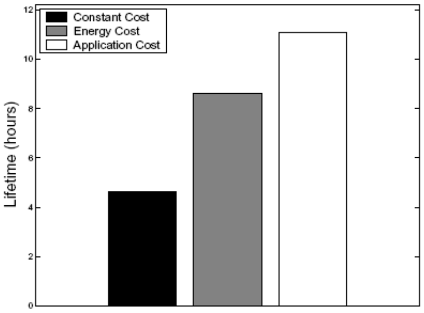


Figure 3: Protocol performance under different cost assignments for a network with random energy distribution

2) Effect of network density:

Higher sensor density should obviously extend network lifetime when using sensor management techniques such as ACBR since more energy is distributed throughout the network. In order to observe just how much, we varied the number of sensors (randomly deployed) in the network to observe the effect on the performance of ACBR. Figure 5 shows the network lifetime using ACBR (with application cost assignment) and using no sensor management (i.e., all sensors always remain active, sending traffic) with shortest energy cost routing, averaged over 10 trials. To be fair, packet sizes and most semantics of the ACBR protocol architecture (besides the Opt In and Opt Out beacons) were the same as those in ACBR. Without sensor management, the effect of an increase of data generated on the network cancels the effect of the increase in the number of sensors available to route the data, and so network lifetime is not affected much. With ACBR, however, the fraction of deployed sensors that are used as sensors decreases as the number of potential routers increases and lifetime is extended significantly. For large networks, the use of ACBR can result in network lifetime improvement by more than a factor of 10. The purpose of these simulations is not to show that ACBR optimally selects which sensors should remain active, and in fact, we do not claim this to always be true. Here, we simply demonstrate the general gains in terms of network lifetime that can be achieved through sensor management.

V. CONCLUSIONS AND FUTURE WORK

We have proposed a distributed, integrated protocol for sensor management and routing to be used in large-scale wireless sensor networks. Our simulation results show that there can be a significant benefit when utilizing an application-based routing cost that considers the residual energy of neighboring sensors.

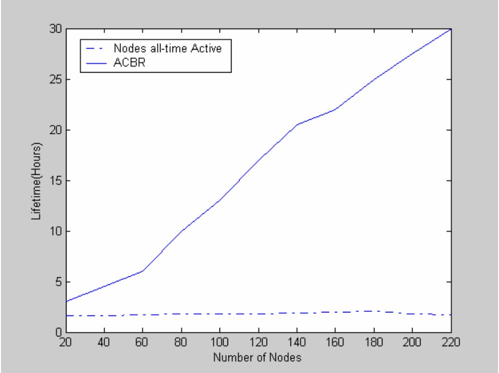


Figure 5: Performance of ACBR for node density.

VI. REFERENCES

[1] M. Bhardwaj and A. Chandrakasan, “Bounding the Lifetime of Sensor Networks Via Optimal Role Assignments,” in Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2002.

[2] M. Perillo and W. Heinzelman, “Simple Approaches for Providing Application QoS Trhough Intelligent Sensor Management,” Elsevier Ad Hoc Networks Journal, vol. 1, no. 2-3, pp. 235–246, 2003.

[3] A. Cerpa and D. Estrin, “ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies,” in Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2002.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, 2000.

[5] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed Energy Conservation for Ad Hoc Routing,” in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, July 2001.

[6] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, “PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks,” in Proceedings of the Twenty Third International Conference on Distributed Computing Systems, 2003.

[7] R. Iyer and L. Kleinrock, “QoS Control For Sensor Networks,” in Proceedings of the IEEE International Conference on Communications, 2003.

[8] T. Yan, T. He, and J. A. Stankovic, “Differentiated Surveillance for Sensor Networks,” in Proceedings of Sensys, 2003.

[9] D. Tian and N. Georganas, “A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks,” Wireless Communications and Mobile Computing Journal, vol. 3, no. 2, pp. 271–290, March 2003.

[10] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, , and C. Gill, “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks,” in Proceedings of Sensys, 2003.

[11] H. Gupta, S. Das, and Q. Gu, “Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution,” in Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2003.

[12] E. Royer and C. Toh, “A review of current routing protocols for adhoc mobile wireless networks,” IEEE Personal Communications, vol. 6, no. 2, pp. 46–55, April 1999.

[13] S. Singh, M. Woo, and S. Raghavendra, “Power Aware Routing in Mobile Ad Hoc Networks,” in Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking, 1998.

[14] J. Chang and L. Tassiulas, “Energy Conserving Routing in Wireless Ad Hoc Networks,” in Proceedings of the Nineteenth International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2000.

[15] Q. Li, J. Aslam, and D. Rus, “Distributed Energy-Conserving Routing Protocols for Sensor Networks,” in Proceedings of the Thirty-Sixth Hawaii International Conference on System Sciences, 2003.