ENERGY-AWARE ADAPTIVE LOW POWER LISTENING FOR SENSOR NETWORKS

Raja Jurdak, Pierre Baldi and Cristina Videira Lopes
School of Information and Computer Sciences
California Institute for Telecommunications and Information Technology Cal-(IT)^2
University of California, Irvine, CA 92697

ABSTRACT

Energy efficiency is a central issue for all wireless sensor network applications. Idle listening on the wireless channel constitutes a large portion of overall energy consumption in sensor networks. The need to reduce idle listening energy consumption has led recently to the design of BMAC [1], a MAC protocol for sensor networks which provides 8 low power listening modes and 8 corresponding transmit modes. Here, we propose a cross-layer mechanism called Energy Aware Adaptive Low Power Listening (EA-ALPL) that enables each individual sensor node running BMAC to set its own listening mode according to its duty cycle and its number of descendants in the routing tree. EA-ALPL also enables nodes to learn the listening modes of their neighbors and to choose the appropriate transmit mode in order to ensure correct packet reception. We show through deployment experiments that EA-ALPL yields overall power savings ranging between 16% and 55% depending on a node's logical topology position.

1. INTRODUCTION

Advances in processor, memory, communication and sensing technology have fueled increased interest in sensor networks. Sensor networks have a wide range of military, civilian, and environmental applications. Regardless of the application, sensor networks will require unattended network operation for months or years, typically with limited battery resources at each node. Consequently, energy efficiency is a central issue in all sensor network applications.

Many protocols have been designed to provide energy-efficient behavior at both the MAC layer [2] and the routing layer [3]. At the MAC layer, idle listening constitutes a large portion of power consumption because data is sent infrequently. This effect is even more pronounced in monitoring sensor networks [4]. Thus, energy-efficient MAC protocol proposals have focused on minimizing idle listening at sensor nodes [1,5].

The recent work by Pollastre et al. describes a new sensor network MAC protocol called BMAC [1], which aims at reducing idle listening at sensor nodes. BMAC proposes that each node wake up periodically to check for channel activity. The wake-up period is referred to as the check interval. BMAC defines 8 check intervals, and each check interval corresponds to one of BMAC's 8 listening modes. To ensure that all packets are heard by the nodes, packets are sent with a preamble whose reception time is longer than the check interval. BMAC therefore defines 8 different preamble lengths referred to as transmit modes. Additionally, Pollastre et al. analytically derive optimal listening modes based on the number of neighbors of a node. In their experiments, they determine the maximum neighborhood size in the network, and they set the optimal listening mode for that neighborhood size. The experimental results yield significant energy savings for BMAC over previous protocols.

The other major strengths of BMAC are its modularity and flexibility. BMAC provides interfaces that are accessible to higher layer protocols and applications to set listening and transmit modes on a per-packet basis if needed. Pollastre et al. also suggest that using these interfaces to set listening and transmit modes according to additional information on the application and operation of a sensor network could produce further power savings for BMAC.

Building on BMAC, our work proposes a cross-layer mechanism called Energy Aware Adaptive Low Power Listening (EA-ALPL) to better adapt to dynamic sensor network topologies and nonuniform energy consumption. In BMAC, setting a network-wide listening mode disregards the non-uniform and dynamic local states of individual nodes. EA-ALPL enables each sensor node to set its own listening mode according to its local state. Per node listening modes are more energy-efficient, but it is difficult to predict the state of each node prior to deployment. To address these challenges, EA-ALPL introduces the following novel contributions:

1. It enables each node to set its own listening mode based on its current state information.
2. It allows each node to dynamically learn the listening mode of its routing parent in order to locally set the appropriate transmit mode.
3. It proposes the dependence of listening mode on both topology-related information and duty cycle at a node.

The dependence of the listening mode on a node's topology position ensures adaptability to a dynamic sensor network topology. EA-ALPL reduces idle listening at each node by selecting the optimal listening mode for the node's current number of descendants in the routing tree.

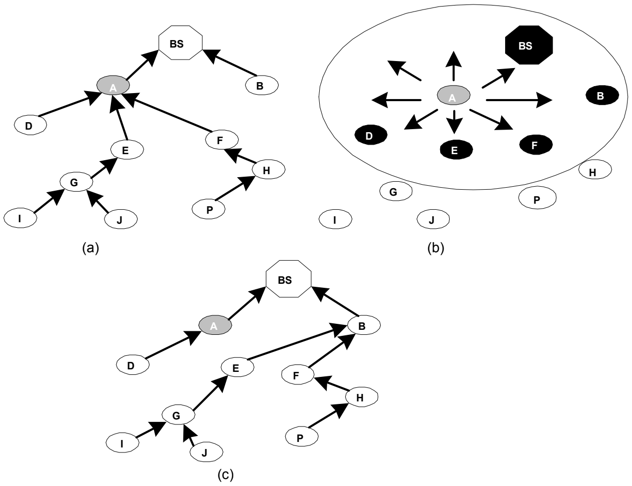

Figure 1: Node interaction in EA-ALPL

The dependence of the listening mode on duty cycle addresses the inherently non-uniform energy consumption in sensor networks [6-8]. Highly loaded nodes deplete their battery resources at a faster rate than other nodes in the network. Consequently, a highly loaded node informs its neighbors of its high duty cycle so that neighbors can increase the node's routing cost in their local routing table. The node’s higher routing cost causes its children to choose different parents, effectively reducing the node’s descendants. The reduced number of descendants enables the node to switch to a more energy efficient listening mode.

We show that EA-ALPL reduces overall power consumption at all nodes by using longer check intervals when possible, and it reduces power consumption for highly loaded nodes by promoting load balancing. We demonstrate the benefits of EA-ALPL both through analysis and experiments on a sensor network testbed.

The rest of the paper is organized as follows. Section 2 describes EA-ALPL and its design choices. Section 3 provides the analytical justification for EA-ALPL. Section 4 presents our experiments to validate the analytical model on mica 2 motes. Section 5 discusses the results and concludes the paper.

2. ENERGY AWARE ADAPTIVE LOW POWER LISTENING

In this section, we first describe the main steps involved in EA-ALPL that enable nodes to dynamically set their listening mode and to adaptively set their transmit mode. Then, we discuss the main design choices for EA-ALPL. Finally, we present the routing modifications required for EA-ALPL.

Description

EA-ALPL enables a node to set its own listening mode according to its current state, and to adapt its transmit mode to fit the listening mode of its routing parent. Figure 1 illustrates the node interaction to enable nodes to set their listening mode adaptively. We assume a proactive routing protocol in which nodes periodically send routing update messages to declare their state to their neighbors.

Initially, nodes are unaware of their neighborhood state, so all nodes listen at an initial listening mode L_{init} and use the corresponding transmit mode T_{init}, both of which are known a priori to all nodes. Each node begins sending periodic route update messages to declare its presence and state. Once nodes learn of their neighbors' presence, a routing graph is formed and data flows toward the base station (Figure 1(a)). As a result, each node learns how many descendants it has in the routing graph. Before sending the next route update message, node A first sets the optimal listening mode L_A for its current topology position. Then, node A sends a routing update message that includes its new listening mode and its power state along with other routing information (Figure 1(b)).

All of node A's neighbors hear the routing update message, and they learn A's current listening mode L_N and power state. Each neighbor of A records A's listening mode and power state in its local neighbor table. Consequently, each node in the network always has up-to-date information on the state of its neighbors. A node favors neighbors with a better power state in routing decisions. Whenever a node D chooses A as a routing parent, it simply checks its neighbor table for A's listening mode L_A. D then sends its data packets using the transmit mode T_A that matches L_A. Similarly, nodes learn of changes in their parent’s listening mode through route update messages and adapt their transmit mode accordingly.

The routing tree in Figure 1(a) puts most of the forwarding burden on node A. As a result, A depletes its battery resources quicker than node B. In order to reduce its forwarding load, A announces its high duty cycle to its neighbors, causing neighbors to increase A's routing cost. This in turn causes most of A's current children to choose another parent whenever possible (Figure 1(c)). Having diverted most of its forwarding load to node B, node A begins listening with a longer check interval to reduce its listening power consumption.

Node-specific Listening Modes

In general, the topology of a sensor network is neither predictable nor static, and it is affected by factors such as terrain, deployment method, interference variations, and dynamic node membership. Thus, network designers have to make conservative assumptions in setting network-wide listening and transmit modes prior to network deployment. This causes unnecessary idle listening to occur in less active portions of the network. EA-ALPL's purpose is to reduce idle listening by allowing each node to set its own listening mode depending on its local state. The rationale is that in a dynamic sensor network, each node always has the most up-to-date view of its own local state [9]. Node states can be defined by the network designer or operator. In this work, we consider the logical topology position and duty cycle at each node to represent the local state.

Number of Descendants

We contend that for monitoring applications, using the number of descendants to choose the optimal listening modes is more energy-efficient than using the number of neighbors, as suggested in [1]. In monitoring applications, data flow is typically towards a single data sink. The basic requirement for correct data delivery is for each node to listen often enough to hear all the packets that it must forward toward the data sink. The number of packets that a node forwards depends on the number of its descendants in the routing tree. Setting the listening mode according to the neighborhood size is a more conservative option, which is partially related to the requirements of the development platform and the routing protocol. Simple modifications to the routing protocol enable nodes to use EA-ALPL without sacrificing data yield.

In EA-ALPL, nodes can use their logical topology position as a basis for setting their listening mode. Each node N learns how many descendants it has in the routing tree by counting the number of packets γ that it forwards during a route update interval. The number γ indicates how busy N was during the last interval. When it is time to send the next routing update message, N first sets its listening mode to the optimal listening mode L_N for a traffic load of γ packets and sends a routing update message declaring its new listening mode.

Load Balancing

The inherently nonuniform energy consumption in sensor networks motivates the load balancing aspect in EA-ALPL. Balancing energy consumption in EA-ALPL requires that individual nodes adapt their listening mode according to their power consumption rate. While the number of descendants at a node reveals the current activity of a node, the node's duty cycle indicates the node's past activity, in particular, the node's energy consumption so far. A high duty cycle indicates that the node has been highly active up till the present point in time and vice versa. Therefore, EA-ALPL uses each node's duty cycle as its power state.

Each node can piggyback its duty cycle value within its routing update messages in order to declare its power state to neighbors. As a result, nodes learn the power states of all their one-hop neighbors and store this information in their local neighbor table.

The main idea is to divert some multihop traffic from nodes with high duty cycle to less active nodes in order to allow highly active nodes to listen less often and to balance the load. EA-ALPL implicitly uses a node's duty cycle in setting the node's listening mode. A high duty cycle at a particular node N causes its neighbors to increase its routing cost. Consequently, fewer neighbors choose N as their routing parent, thereby reducing the number of N's descendants. As N observes that it has fewer descendants, N may select a listening mode that saves more power than its current listening mode. The next subsection discusses the details of incorporating energy-awareness into the routing protocol.

Routing Modifications

Our primary platform for sensor network development is TinyOs [10], developed at UC Berkeley. Within TinyOs, the standard routing protocol is called MintRoute. MintRoute is a proactive routing protocol in which nodes send periodic routing messages to declare their local states. The original cost metric in MintRoute combines hop count and link quality. To compute link quality, a node snoops [11] on the packets sent by each neighbor, and checks the sequence number of the packets. A node determines the link quality to a neighbor by monitoring the ratio of packets received from that neighbor to the number of packets sent by that neighbor. The rest of this section discusses routing modifications to ensure that EA-ALPL is compatible with the routing protocol for the purposes of: (a) data delivery; and (b) energy awareness.

The concept of adaptive listening modes raises the possibility that some nodes may not hear the packets sent by their neighbors because of mismatched preamble lengths and check intervals. For example, if a node A sends a packet with a short preamble to its parent B, one of node A's neighbors D that is listening infrequently may miss node A's packet. This situation does not affect data delivery, since it is only necessary for A's parent B to hear the packet. Missing a routing update packet is more detrimental, since routing packets hold important information on neighborhood routing state changes.

We implement modifications to MintRoute to address missed routing update packets. A central issue in designing EA-ALPL is to ensure that asymmetric listening modes do not affect maintaining an up-to-date neighborhood view at each node. Achieving this goal requires that nodes always hear the routing update packets of their neighbors. Thus, EA-ALPL specifies that nodes always send their routing update packets with the longest preamble, so that a neighbor in any listening mode can hear the update packets. Secondly, MintRoute defines quality as the percentage of data packets correctly received from a neighbor. The example above on missed data packets causes the quality metric in MintRoute to drop. Consequently, we modify MintRoute so that nodes only snoop on periodic routing updates instead of data packets to determine the link quality to their neighbors. Monitoring route update packets for determining link quality ensures that asymmetric listening modes at neighboring nodes have no detrimental effect on link quality, because all routing update packets are sent with the longest preamble.

In all proactive routing protocols, each node periodically selects its routing parent. In MintRoute, a node N first selects from its neighbor table the best contender M with the least routing cost. Next, N compares the cost of M with the cost of the current routing parent RP. N chooses M as its new parent only if:

$$C(M) + \varepsilon < C(RP) \tag{1}$$

where C(N_j) is the routing cost of node N_j, and ε is the switching threshold, which ensures that a node switches its routing parent only when there is an appreciable benefit in doing so.

In order to integrate energy-awareness into MintRoute, the choice of a routing parent must also depend on how busy a node has been relative to its neighbors. Highly loaded nodes should have a higher routing cost. We introduce the cost metric *C(power)* for a neighbor N_i:

$$C(power) = \varepsilon \frac{\delta_i - \sum_{j=0}^{k} \delta_j / k}{\sqrt{\sum_{j=0}^{k} \delta_j^2 / k - \sum_{j=0}^{k} \delta_j / k}} \tag{2}$$

where k is the number of neighbors in the node's local neighbor table, and δ_j is the duty cycle of neighbor N_j. Equation 2 compares the duty cycle of neighbor N_i to the average duty cycle in the neighborhood and normalizes the difference. The normalized difference determines the extent of statistical deviation of the power state of N_i among all the nodes in the neighborhood.

The overall routing cost should include *C(power)* only if the node M is at the same level or at a lower level than the current routing parent RP. To clarify the need for this condition, consider again Figure 1(a) as an example. Node I has 2 neighbors: its parent G and node J. I should not consider J as a better parent than G even if J has a smaller duty cycle than G. Choosing J as a parent would not lower G's forwarding load or number of descendants, and it would also increase node J's forwarding load.

Therefore, the new overall cost of a neighbor includes the original MintRoute cost of quality and hops, as well as the power cost according to the following equation:

$$C_{new}(M) = \begin{cases} C(M) + \alpha C(power) \ldots L(M) \leq L(RP) \\ C(M) \ldots\ldots\ldots\ldots\ldots\ldots\ldots L(M) > L(RP) \end{cases} \tag{3}$$

where α is a constant representing the weight of *C(power)*, and $L(N_j)$ represents the number of hops of node N_j from the base station. With the new cost definition, neighbors with a higher duty cycle have a higher routing cost, and they are less likely to be chosen as forwarders.

3. ANALYSIS

This section presents the analytical basis for EA-ALPL. We assume that the sensor nodes collect sensor data and transmit the data in a packet once in every period T. The following equation governs power consumption E at a sensor node [1]:

$$E = E_t + E_r + E_d + E_{listen} + E_{sleep} \tag{4}$$

where E_t is the power spent on transmissions during time T, E_r is the power for packet reception during time T, E_d is the power required to collect sensor values, E_{listen} is the power consumed for checking the channel for activity, and E_{sleep} is the power consumed while the node is asleep. The quantitative expressions for each energy component are given in [1]. We limit the discussion here to the qualitative aspects that are relevant to EA-ALPL.

In monitoring applications, the sampling period T is typically in the order of minutes. Therefore, each node collects sensor data, transmits packets, and receives packets once every few minutes. On the other hand, nodes wake up to monitor the channel for activity much more frequently, for instance once every several milliseconds. Thus, idle listening on the channel has a profound effect on the overall power consumption, so reducing idle listening yields significant power savings.

Topology

Our ultimate goal is to minimize E for every node by enabling each node to locally select its own optimal listening mode while maintaining correct and timely data delivery. Minimizing E on a per-node basis builds on the following observations about Equation 4:

1. E_d and E_{sleep} are not significant factors in determining optimal listening mode. E_d is equal for all nodes throughout the network. E_{sleep} is at least an order of magnitude smaller than the other terms in equation 4, so it has a negligible effect on E.
2. E_t and E_r depend on the node's position in the logical topology. If a node is a leaf in the routing tree, it has fewer packets to forward.
3. The listening mode of N determines E_{listen}. It also determines the preamble length for packets that are received at N. Consequently, the listening mode at a node N also affects E_r at N and E_t at N's children.

For example, a more frequent listening mode at N increases E_{listen} and decreases E_r at N. E_{listen} increases because N wakes up more frequently to check for channel activity, and E_r decreases because the frequent listening enables N's neighbors to send their packets to N with shorter preambles, thereby reducing packet reception time at N. Similarly, the listening mode of N affects E_t at the children c_i of N. A more frequent listening mode at N enables c_i to send its packets with shorter preambles, thus reducing E_{ti}. Less frequent listening at N forces c_i to send packets with long preamble and consume more power for packet transmissions.

These dependencies further support the need for setting per-node listening modes. In practice, each node locally computes E_t, E_d, and E_{sleep} and then selects the listening mode that provides the combination of E_{listen} and E_r that yields the lowest power consumption E.

Power State

The energy consumption in sensor networks is nonuniform among the nodes. First, data always flows to one or a few

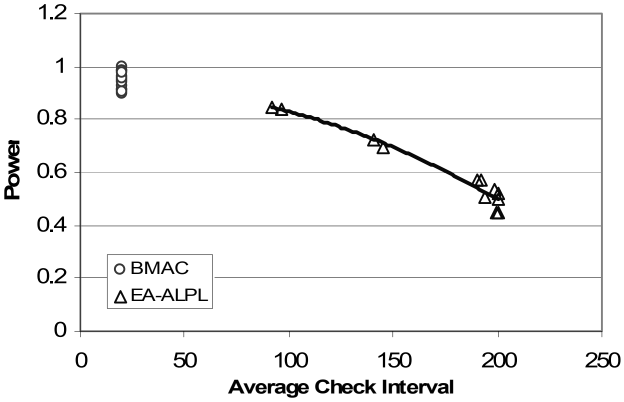

Figure 2: Power consumption vs. average check interval

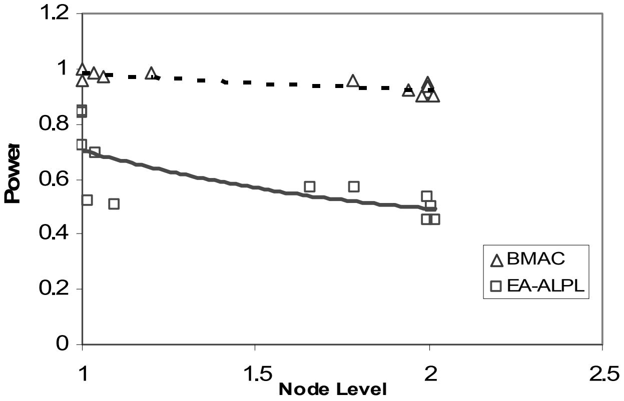

Figure 3: Power consumption vs. node level

sinks. The nodes that are one hop away from a data sink are called critical nodes [8]. Critical nodes have a larger forwarding burden and consume more energy than nodes further away from the sink [7]. EA-ALPL contributes further to nonuniform energy consumption because nodes may have different listening modes. Therefore, E_{listen} may also be different among nodes. All these factors indicate that nodes can use up battery resources at different rates. One aim of EA-ALPL is to balance the power consumption rates among network nodes by manipulating data forwarding patterns and listening modes. Of course, the degree of achievable load balancing depends on the node density.

In terms of equation 4, highly loaded critical nodes have larger E_t and E_r. Because EA-ALPL sets the listening mode according topology information, a loaded critical node will also choose to listen frequently to the channel, so it has a high E_{listen} As a result, the energy consumption E and the duty cycle at a loaded critical node are higher than that of its neighbors. The loaded critical node informs its children to choose a new routing parent, thereby reducing its E_r and E_t. The loaded critical node also switches to a listening mode with a longer check interval to reduce E_{listen}. The reductions in E_r, E_t and E_{listen} cause the overall power consumption E at the critical node to drop.

4. DEPLOYMENT RESULTS

In this section, we explore the performance of EA-ALPL for a testbed of sensor nodes deployed in our laboratory. The sensor nodes in our experiments consist of 14 mica 2 motes from Crossbow [12]. Our implementation of EA-ALPL is in NesC, a component-oriented variant of C customized for networked embedded systems and built into TinyOs.

The nodes are placed at random positions in the laboratory and the base station is placed near one of the walls of the room. We reduce the transmit power of nodes to limit their radio range, enabling multihop communication. The aim of the experiments is to compare the energy savings of using EA-ALPL over the case of network-wide listening modes. Thus, we conduct 2 experiments for the same physical network topology. In the first experiment, we initially determine the listening mode for the busiest node in the network, and we assign that listening mode to all the nodes. In the second experiment, the nodes use EA-ALPL that adapts listening modes on the basis of both the number of descendants and duty cycle. Each experiment lasts for 43 hours. In both experiments, nodes run the Surge application that is available with the standard distribution of TinyOs. In the EA-ALPL experiment, the routing update period is 90 seconds. The routing update period for the network-wide listening mode experiment is 120 seconds.

The data yield for both EA-ALPL and BMAC is 98.5%. Figure 2 shows that the overall power consumption is closely correlated with the average check interval at each node in EA-ALPL. The average check interval for EA-ALPL ranges between 20 ms and 200 ms. In plain BMAC, all nodes use a check interval of 20 ms, so most of them consume almost the same power as the busiest node because all nodes use the same listening mode. EA-ALPL reduces power consumption at the most active nodes by about 16% and by up to 55% at other nodes. Note that the most active node in EA-ALPL consumes less energy than the least active node in BMAC.

Figure 3 compares the normalized power consumption according to node level. Note that some nodes have average levels between level 1 and level 2, as a result of the dynamic topology which causes nodes to choose new parents and change levels during the deployment. Nodes at level 2 achieve more than 50% savings in EA-ALPL over BMAC. The extent of power savings at level 1 nodes varies. For highly loaded level 1 nodes, the power consumption is about 16% lower for EA-ALPL than the case of BMAC. For level 1 nodes with fewer or no descendants, power savings of EA-ALPL over BMAC range between 31% and 50%.

Figure 4 plots the normalized power consumption based on the average number of descendants throughout the deployment. EA-ALPL incorporates energy cost into routing decisions to balance the forwarding load, so the most loaded node in EA-ALPL has an average of 2 descendants in comparison with an average of 2.5 in BMAC.

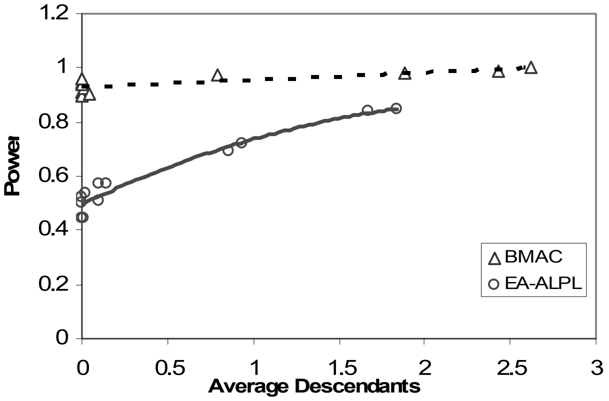

Figure 4: Power vs. average descendants

Power consumption for the BMAC case is correlated with the number of descendants, but the variation is limited. The overall trend for EA-ALPL is that it yields more energy savings for nodes with fewer descendants, because these nodes can use longer check intervals. The most loaded node in ALPL still has achieves 16% reduction in power consumption than the most loaded node in BMAC through load balancing.

The results of the experiments in this section confirm the energy benefits of using EA-ALPL. The main trend in the experimental results is that nodes can easily adapt their listening modes to their forwarding load. The nodes with a smaller forwarding load achieve the most significant power savings, whereas busier nodes have smaller power savings since they typically have to listen more often. The other advantage of EA-ALPL is that it shifts some of the forwarding burden from the most loaded nodes to other nodes. The degree of power savings in EA-ALPL depends on the topology, interference conditions, and available redundancy in a particular network.

5. DISCUSSION

We have proposed the concept of Energy Aware Adaptive Low Power Listening for sensor networks to allow nodes to adapt their listening modes in BMAC to their current forwarding load. EA-ALPL optimizes listening modes on basis of descendants and duty cycle.

EA-ALPL enables nodes to adapt to new network conditions. For example, if we need to shorten the sensor nodes' sampling period, we can send a short command message into the network indicating the change. As nodes begin sampling and sending data more frequently, each node uses its local traffic load information to choose its listening mode and adapt to the new network conditions. Network nodes can also adapt autonomously the introduction or disappearance of nodes.

The adaptive nature of EA-ALPL supports the dynamic nature of sensor networks and can exploit information about the present, the past, and the predicted future state of individual sensors to reduce power consumption. In this work, we have studied how adapting listening modes to the node's current logical topology position (which represents the node's present state) and duty cycle (which represents the node's past state) can reduce the power consumption. The choice of listening mode in EA-ALPL can also be based on information about the expected future behavior of the node, such as the node's dynamic role in the network application.

The discussion in this work has focused on time-driven monitoring sensor networks. However, EA-ALPL also applies to event-driven and demand-driven sensor networks. For example, nodes that detect a certain event can listen more frequently for the duration of the event. In the meantime, nodes that are further away from the event can keep listening infrequently.

In sum, we have proposed EA-ALPL for sensor networks to allow nodes to adapt their listening modes in BMAC to their current forwarding load and duty cycle. EA-ALPL provides flexible and power efficient behavior in monitoring sensor networks. Its power efficient behavior stems from using listening modes that are optimal for a particular node's logical topology position.

REFERENCES

[1] J. Pollastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In Proc. of *ACM SenSys*, 2004.
[2] W. Ye and J. Heidemann. Medium Access Control in Wireless Sensor Networks. *Wireless Sensor Networks*, Kluwer Academic Publishers. 2003.
[3] Q. Jiang and D. Manivannan. Routing Protocols for Sensor Networks. In Proc. *IEEE (CCNC)*, 2004.
[4] R. Szewczyk et al. Habitat Monitoring with Sensor Networks. Comm. of the ACM, 476):34-40, 2004.
[5] W. Ye, J. Heidemann and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In Proc. *IEEE Infocom*, 2002.
[6] M. Haenggi. Energy-Balancing Strategies for Wireless Sensor Networks. In Proc. *(ISCAS'03)*, May 2003.
[7] R. Jurdak, C. V. Lopes and P. Baldi. Battery Lifetime Estimation and Optimization for Underwater Sensor Networks. *Sensor Network Operations*, IEEE Press, 2005.
[8] V. Mhatre, C. Rosenberg, D. Kofman et al. A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint. IEEE Transaction on Mobile Computing, January 2004.
[9] R. Szewczyk, J. Pollastre, A. Mainwaring, and D. Culler. Lessons from a Sensor Network Expedition. In Proc *European Workshop on SensorNetworks*, 2004.
[10] Tiny Operating System. UC Berkeley. http://tinyos.net.
[11] R. Sivakumar, B. Das, and V. Bharghavan. The clade vertebrata: spines and routing in ad hoc networks. In Proc. *IEEE (ISCC '98)*, 1998.
[12] Mica 2 Motes. Crossbow Technology Inc. available: http://www.xbow.com/Products/productdetails.aspx?sid=72