An Address-Light, Integrated MAC and Routing Protocol for Wireless Sensor Networks

Sunil Kulkarni, Aravind Iyer, *Student Member, IEEE*, and Catherine Rosenberg, *Senior Member, IEEE*

***Abstract*—We propose an address-light, integrated MAC and routing protocol (abbreviated AIMRP) for wireless sensor networks (WSNs). Due to the broad spectrum of WSN applications, there is a need for protocol solutions optimized for specific application classes. AIMRP is proposed for WSNs deployed for detecting rare events which require prompt detection and response. AIMRP organizes the network into concentric tiers around the sink(s), and routes event reports by forwarding them from one tier to another, in the direction of (one of) the sink(s). AIMRP is address-light in that it does not employ unique per-node addressing, and integrated since the MAC control packets are also responsible for finding the next-hop node to relay the data, via an anycast query. For reducing the energy expenditure due to idle-listening, AIMRP provides a power-saving algorithm which requires absolutely no synchronization or information exchange. We evaluate AIMRP through analysis and simulations, and compare it with another MAC protocol proposed for WSNs, S-MAC. AIMRP outperforms S-MAC for event-detection applications, in terms of total average power consumption, while satisfying identical sensor-to-sink latency constraints.**

***Index Terms*—Addressing, anycast routing, cross-layer integration, MAC, power-saving mode, rare event detection, sensor networks.**

I. Introduction

RECENT advances in wireless communication technologies, and sophisticated techniques for miniaturization of electronic and sensor devices, have fueled a lot of research in the area of wireless sensor networks. Dense networks of wireless sensor devices are being deployed for sensing or monitoring various phenomena of interest. A wireless sensor device is a small battery-powered device, capable of sensing one or more physical quantities. In addition, it is equipped with a limited amount of storage, and computation capabilities. A wireless sensor network (WSN) consists of a large number of these devices, working collaboratively towards a certain common goal. These sensor nodes communicate with each other and with one or more sinks (or base-stations) over a wireless channel. The sink(s) is (are) responsible for collecting information from all sensor devices in the network and represents the interface of the WSN to the outside world.

The range of applications that WSNs are envisaged to support, is tremendous, encompassing military, civilian, environmental and commercial areas. Each application imposes a unique set of goals and requirements, and also produces a different type of traffic. For instance, an application to monitor the environmental conditions affecting crops and livestock [1], is a data-gathering application. The traffic it generates is expected to be more or less uniform, and the latency requirements on its data are expected to be loose. On the other hand, a sensor network deployed to detect forest fires [1], is likely to produce data in bursts, with severe latency constraints. Hence, a generic approach to design WSNs, will often be unable to take advantage of any application-specific features, and sometimes may even be unsuitable for certain applications. The danger in pursuing an application-specific approach though, is to end up developing a different protocol for each application. A *careful* examination of the tradeoffs involved, is necessary to avoid being too generic or too specific.

To this end, it is important to be able to classify WSN applications based on their data-delivery requirements and their traffic characteristics [14]. In particular, most of the current WSN applications fall into one of the following five broad classes: 1) event detection and reporting; 2) monitoring and periodic reporting; 3) sink-initiated reporting; 4) object detection and tracking; and 5) hybrid applications with more than one of the above four characteristics. Our work focuses on the first class of applications, namely, event detection and reporting. Applications which fall into this category include intruder detection and detection of fire and hazards. These applications exhibit prolonged periods of inactivity till the time an event of interest is detected. On detecting an event, a report of this event has to be *promptly* communicated to the sink. An event report is usually expected to carry some *location information* about the event. Hence, the network protocol should be designed to satisfy the requirements of latency and location, while consuming minimal energy.

For this, we examine the following salient features of the WSNs considered in this paper: the many-to-one communication paradigm, whereby all sensors intend to send their data to one (or few) sink(s); the large node density that begs for sensors that are cheap to manufacture and ready to deploy; and, the tight limitation in energy which calls for a highly optimized, lightweight protocol stack. This impacts the protocol design for WSNs, in the following way. In traditional communication networks, the need for modularity and interoperability, leads to a

Manuscript received December 16, 2003; revised October 6, 2004, and April 27, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Krunz. This work was supported in part by a grant from the Defense Advanced Research Projects Agency (Contract MDA 972-02-1-0032), and by a grant from the National Science Foundation (Contract 0087266).

S. Kulkarni was with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Google Inc., Mountain View, CA 94043 USA (e-mail: sunilkul@gmail.com).

A. Iyer is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: iyerav@ecn.purdue.edu).

C. Rosenberg is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1 (e-mail: cath@ece.uwaterloo.ca).

Digital Object Identifier 10.1109/TNET.2006.880163

layered protocol reference model. On the other hand, for WSNs, it is more important to satisfy application-specific requirements, and to be energy-efficient. Hence, cross-layer interaction or integration of protocol layers, is recommended when it can be effectively used to reduce the protocol overhead, and to make the protocol stack lightweight.

The above discussion leads us to consider the following issues in order to design our integrated protocol. The first is the issue of addressing. In general, there is a need for addressing or "identification" at three levels, for the purposes of: 1) MAC; 2) routing; and 3) location information about the data source. Strict per-node addressing is expensive in a dense network, because not only would the size of an address be large, but also these addresses would need to be allocated and exchanged at different layers of the protocol stack. Allocation of addresses in a dense network, is a real problem which is often underestimated. Our goal is to use (and even reuse) only as much addressing as is absolutely necessary. Now, the problem of location determination in a dense WSN, is an active area of research by itself, and is beyond the scope of this paper. In case the application requires some location information to be associated with each event report, we assume that the required granularity of location information is determined by some means, and is *embedded in the data payload* of each packet. Hence, we only seek to reduce the bit budget of addresses used for MAC and routing. It is clear that a further step would be to integrate all three levels of addressing.

The second issue concerns the routing protocol. Unlike in an ad hoc network, where any node can potentially communicate with any other node, a WSN exhibits the many-to-one communication paradigm. In addition, we assume that the sink is not required to communicate with a particular sensor.[1] These two points together imply that 1) the flow of data originates *only* at a sensor node, and 2) it is *always* destined for the sink node. Thus, the routing protocol overhead can be reduced in two ways: first, the routing protocol only needs to discover paths from each node to the sink; and second, since no communication is addressed to an individual node, routing can be performed at a coarser level of addressing than one address per node.

Thus, we can identify two major sources of wasteful energy expenditure. The first is the overhead required for the routing and MAC protocols. This can be minimized in two ways, namely, 1) by choosing a streamlined packet header structure, and reducing the size of each control field (e.g., the addressing fields) as much as possible, using integration, and 2) by minimizing the need for non-data related information exchange. The second is idle-listening in MAC protocols based on random access, especially in case of low traffic load. Indeed, for event detection applications, a medium-access mechanism based on random access is more suitable than one based on controlled access, due to the nature of the traffic generated. Hence, an effective MAC protocol, for this class of applications, would have to be coupled with a power-saving mechanism to minimize idle-listening. Besides, the power-saving mechanism itself, should not impose its own overhead by requiring a lot of information exchange.

This paper proposes an address-light, integrated MAC and routing protocol (AIMRP) which seeks to address all the issues raised above. AIMRP is an integrated MAC and routing mechanism designed specifically for WSNs which have to promptly detect and report relatively rare events. The contributions of our work are twofold. First, we design the AIMRP protocol with the following attractive features.

- **Integrated MAC and routing** to minimize the protocol overhead: AIMRP organizes the network into tiers around the sink, and routes packets by progressively forwarding them to tiers closer to the sink. This can be readily integrated into the MAC layer.
- **No per-node identification** for either MAC or routing: We use short random identifiers for MAC, on a per-transmission attempt basis, instead of physical MAC identifiers, and per-tier addresses for routing, instead of per-node addresses.
- **Power-saving mode** which requires **no coordination** between the nodes: Nodes repeatedly shut their radio modules off when not in use, independently of one another, while satisfying sensor-to-sink latency guarantees.

Second, we provide a detailed analysis for dimensioning the power-saving mode and to compute the average energy expenditure per event report for a given event frequency, while satisfying the latency constraints. We validate this analysis through simulations. In particular, we show that AIMRP outperforms S-MAC [16] in terms of total average power consumption, while satisfying identical end-to-end latency requirements.

The rest of this paper is organized as follows. In Section II, we review current work in the area of MAC and routing for WSNs. In Section III, we introduce the principles of our address-light, integrated MAC and routing protocol (AIMRP) for WSNs. In Section IV, we describe in detail the working of AIMRP. Section V provides guidelines for dimensioning AIMRP parameters, while Section VI evaluates the protocol through analysis and simulations, and compares its performance with that of a currently proposed protocol, S-MAC [16]. Finally, Section VII concludes the paper, and discusses possible extensions to this work.

II. Related Work

Network design has traditionally followed the principle of *layering*. Complex networking functionalities are broken down and decoupled into manageable and independent levels. This is done so as to allow interoperability, modularity, and to keep the protocols as general-purpose as possible. Following this principle, nearly all of the research in the area of WSNs considers the problem of medium access separate from the problem of routing, although the need for *integrated* and *application-specific* network solutions has been recognized [1], [14].

One of the main approaches to MAC for WSNs, comes from its counterpart for ad hoc networks [1], *viz.*, the IEEE 802.11 standard. The IEEE 802.11 standard is a CSMA/CA based

[1] Clearly, if the sink *is* required to communicate with a particular sensor node, then there is a need for addressing each node. However, this is really the overhead we are trying to avoid by making this assumption. The assumption is not unreasonable in the context of event detection applications, since we feel that *the only reason* the sink would need to communicate to the sensor nodes would be for reprogramming or software updates.

protocol which is widely used in wireless LANs. Using plain 802.11 MAC for WSNs has many drawbacks, as discussed in [4], [12], [16]. In particular, [12] shows that energy consumption due to overhearing and idle-listening, is a major chunk of wasteful energy consumption. Hence, [12] suggests turning off the radio module of a node when it is "overhearing" (i.e., listening to the transmission of a packet not addressed to it).

[16] presents a specially modified 802.11 based medium access protocol (called S-MAC), for WSNs. In this protocol, the authors identify the following sources of energy wastage, *viz.*, *collision, overhearing, overheads,* and *idle-listening*. In order to reduce energy drainage due to idle-listening, nodes periodically sleep. Neighboring nodes form so-called *virtual clusters* to synchronize on their sleep schedules. The sleep schedules are completely synchronized within a cluster and are uncorrelated across clusters. The period of these sleep schedules is determined by the end-to-end delay constraint. S-MAC also uses in-channel signaling, to implement *overhearing avoidance* for nodes to avoid listening to long data packets not meant for them. Finally, S-MAC applies *message-passing* to reduce contention while transmitting relatively long data packets.

A drawback of this protocol is that synchronizing the sleep schedules by creating virtual clusters is a rather complex operation which produces its own overhead. Another drawback of this protocol is that it fails to exploit the many-to-few communication paradigm in WSNs, and does not consider the issue of addressing. In other words, S-MAC is a generic energy-aware MAC protocol which does not cater to specific WSN applications. Our protocol tries to improve upon these two issues for the event reporting class of applications.

MAC protocols based on controlled access rather than random access have also been proposed. For example, [2], [5], and [13] study MAC protocols based on TDMA, [4], [10], [13] on CDMA and/or FDMA. However, for the class of applications we consider, a MAC protocol based on random access would be more appropriate than one based on controlled access.

As in the case of MAC protocols, several routing protocols developed for ad hoc networks have been suggested for WSNs [1]. Specifically, distance vector protocols such as Ad hoc On-demand Distance Vector (AODV), Destination Sequenced Distance Vector (DSDV) and source routing protocols such as Dynamic Source Routing (DSR) have been adapted for WSNs, by optimizing for energy usage. An alternative strategy utilizing gradient-based routing, has been proposed in [6]. But, we believe that, owing to the many-to-one communication paradigm in WSNs, routing protocols can be further streamlined.

Ref. [5] proposes an application-specific protocol architecture for periodically routing reports from all nodes to a distant base station. A method which uses clustering and direct transmissions from cluster heads to the base station is proposed. For uniform energy consumption across all the nodes, the responsibility of being the cluster head is rotated among all the nodes periodically. Ref. [8] proposes a similar solution to the problem, but with two types of nodes, sensors and cluster heads. The authors evaluate the optimum node density for these two types of nodes, and their initial battery energies to guarantee a certain lifetime.

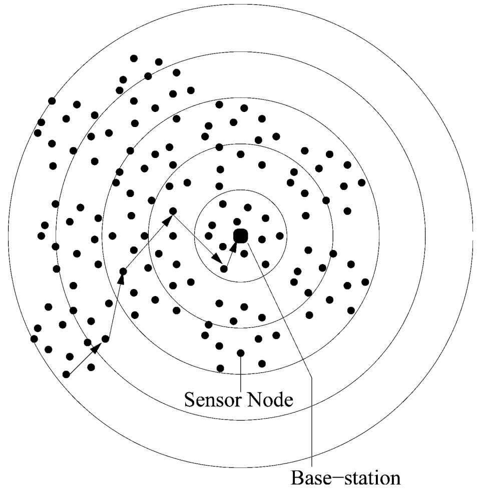


Fig. 1. Illustration of routing in AIMRP.

Refs. [3] and [15] notice that for reliable routing in dense ad-hoc networks not all of the nodes are required to be awake at the same time. In [3], nodes decide to go to sleep or to be awake and join the forwarding backbone, depending on the local information and available residual energy. In [15], the region is divided into virtual square grids, such that all nodes in neighboring grids are able to communicate with each other. Only a single node remains awake within each grid. It may be noted that putting nodes to sleeping when they cannot do anything useful at the routing level, is a kind of integration of MAC and routing.

III. AIMRP: Principles

In this section, we introduce AIMRP, and explain its principles. AIMRP is an address-light protocol which does not use or require the use of strict per-node identifiers or addresses. The routing mechanism employed in AIMRP is the following. For the sake of explaining the principle, let us assume that the WSN consists of several sensor nodes deployed in a circular region with a single sink at the center. By means of an initial configuration phase which will be explained later, the entire network is organized into tiers centered around the sink (refer to Fig. 1). The tiers are numbered $1, 2, 3, \ldots$ starting from the innermost tier, and are such that a node in the nth tier can relay a message to the sink in n hops. Now at the end of the configuration phase, the route discovery is complete, based on the rule that a node in a given tier n only relays messages from tiers farther away from the sink than itself, i.e., tiers $n+1, n+2, \ldots$. The routing is hop-by-hop, and at each hop the node which has the packet indicates its tier number in the packet so that another node with a lower tier number can receive the packet. In this way, routing can be done at the level of addressing of a tier, which has far less overhead than having one routing address per node. The overhead required for route discovery is also limited.

The mechanism for medium access is similar to that used in the distributed coordination function (DCF) in IEEE 802.11, except for two important differences. First, the nodes do not have preassigned MAC identifiers and do not use any unique addresses to communicate, instead choosing new short random

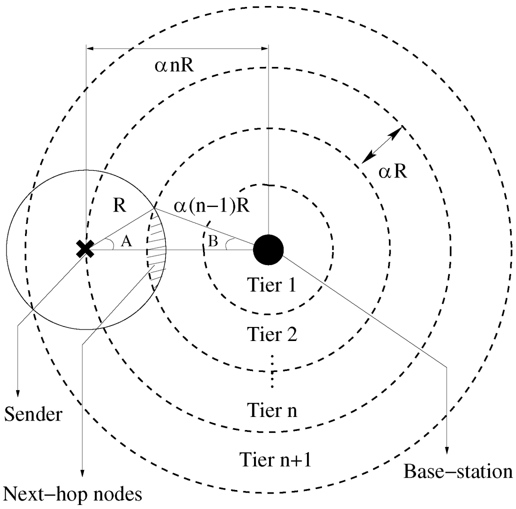
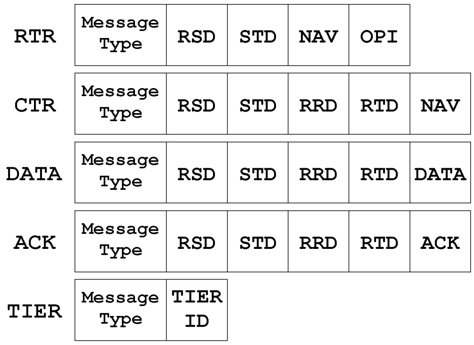


Fig. 2. Formation of tier structure.



RTR	Message Type	RSD	STD	NAV	OPI	
CTR	Message Type	RSD	STD	RRD	RTD	NAV
DATA	Message Type	RSD	STD	RRD	RTD	DATA
ACK	Message Type	RSD	STD	RRD	RTD	ACK
TIER	Message Type	TIER ID				

Fig. 3. Message formats for AIMRP.

identifiers for each communication attempt. The second difference can be explained as follows. In IEEE 802.11, the RTS message has two purposes: to initiate the communication between the source and the next-hop node; and to silence all nodes within the communication range of the source, except the next-hop node. The next-hop node is known because of the routing protocol, before the communication is initiated. In contrast, in AIMRP, the purpose of the analogous RTR (Request_to_Relay) message is to *seek* a receiver node which is closer to the sink and so the destination node is not known beforehand. In other words, the RTR message is an *anycast* message to which any node can reply if it can relay the data according to the routing algorithm outlined above (i.e., if its tier number is lower than the one indicated on the packet). Hence, before a node sends a CTR (Clear_to_Relay) message (analogous to the CTS message of IEEE 802.11), it chooses a random back-off to avoid systematically colliding with other nodes willing to receive and relay the data. The second difference is very important because that is how we integrate a hop-by-hop routing functionality into the MAC protocol. Note that the only control information necessary for this is the *one-hop* source MAC and tier identifier in the RTR message, and both the *one-hop* source and next-hop MAC and tier identifiers in the CTR message.

AIMRP is also equipped with a power-saving mode to curb the energy expenditure due to *idle-listening*. Owing to the nature of the application that AIMRP targets, it would be extremely wasteful to have all nodes keep their radio modules on for all time. But then if a particular node wishes to report an event to the sink, it should find a feasible path to relay the information to the sink, relatively quickly. In order to capture this application-specific characteristic, AIMRP employs a power-saving mode which is subject to a constraint on the maximum end-to-end delay that an event report can encounter. AIMRP relies on an uncorrelated sleep-and-wake pattern at each node, to meet the latency constraint with a pre-specified probability. Since the sleep-wake pattern at each node is independent of the other nodes, there is no need for any additional information exchange between nodes. This is in contrast to the sleep-and-wake algorithm proposed in S-MAC [16], but the important point to note is that S-MAC is a generic protocol which is not designed for this particular class of applications, i.e., event detection and reporting.

IV. AIMRP: Description

In this section, we propose and describe the working of AIMRP. We consider a simple network geometry in which nodes are distributed in a circular region of radius L, centered at the sink. Each node has a communication radius R.[2] We assume that an event is equally likely to occur at any point in the region, and that only one node detects and reports this event. Under this setting, let us define AIMRP. AIMRP involves a configuration phase and an active phase. The configuration phase which has to be completed just after the deployment, works as described in the following subsection.

A. Configuration Phase and Path Discovery

The purpose of this phase is to organize the network into tiers around the sink (see Fig. 2). The sink sends a TIER message (see Fig. 3) with a power level corresponding to a communication range of αR, where the value of α needs to be chosen appropriately (see Section V). All nodes which can successfully receive this message recognize that they belong to TIER 1. Then the sink successively sends messages with communication radii of $n\alpha R$, with TIER_ID $= n$, for $n = 2, 3, \ldots$ All nodes which can receive a TIER n message successfully, recognize that they belong to TIER n, unless they have already "joined" a tier of lower rank.

Alternatively, instead of the sink sending communication messages of varying power, the sensor nodes themselves can form a tier structure by relaying TIER messages, with a power corresponding to a communication range of αR. Thus, a node receiving a TIER message with TIER_ID $= n$ "joins" TIER n, unless it already belongs to a tier of lower rank. Each node also increments the TIER_ID field before forwarding the TIER message it has received. An idea similar to this scheme has been discussed in [9]. Assuming that the radio propagation is identical in all directions, the configuration phase will result in the formation of annular tiers of thickness αR centered at the

[2] In practice, this sort of a "binary" model of a fixed communication and interference range R is often unrealistic. It is clear that a more accurate design would have to employ a more realistic channel model.

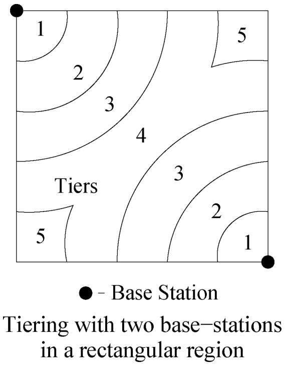


Fig. 4. Tiering with two sinks.

sink. It is possible that owing to some obstacle or due to the terrain, certain nodes may not find each other directly reachable via radio, even though they are physically quite close to one another. In such cases, the shape of the tiers formed will be dictated by the radio reachability of the nodes. In such cases, the second scheme for the configuration phase is more robust.

The configuration phase in case there are multiple sinks, is a simple extension of what was explained above. Consider Fig. 4 which depicts a rectangular region with two sinks situated diagonally opposite each other. For an event-reporting application, we expect the sinks to all be connected to the outside world. Hence, it is reasonable to assume that the sinks are *indistinguishable*, therefore it is irrelevant which exact sink receives the event report. Thus, the tier rank of a node represents its distance in number of hops, from its closest sink (see Fig. 4). Thus, either of the two schemes discussed above can be used for configuration.

B. Active Phase

A node in the active phase of AIMRP is always listening to the radio channel, unless it is transmitting. It is in the so-called *Listener* state. We discuss the *power-saving* feature of AIMRP in the next subsection, where nodes need not always listen to the radio channel. A node remains in the *Listener* state either till it detects an event or has to relay information from some other node, and therefore has `outstanding data` to send to the sink, or till it hears a transmission on the radio channel.

Whenever it has `outstanding data` to transmit, the node attempts to *find* a next-hop node, closer to the sink, which can relay its data. This is in contrast with IEEE 802.11 where a node attempts to transmit to a *particular* node as decided by the routing algorithm. The node waits for a guard time t_g before attempting to transmit anything. After the guard time expires or when the channel becomes free (whichever is later), the node waits for a random listening time t_l before transmitting. The guard time t_g is to ensure that nodes reliably estimate the channel as either busy or idle. The additional random listening time t_l is to prevent nodes attempting to transmit at about the same time, from colliding. Then the node transmits a Request_To_Relay (RTR) message (refer to Fig. 3), which contains a randomly chosen RSD (random source identifier), the source tier identifier (STD) i.e., its TIER id, a NAV entry which represents the length of the packet,[3] and some optional packet information (OPI). The RSD field is limited to a few bits in size, and hence is much smaller than what would be required to maintain per-node fixed MAC identifiers. It can be seen as a temporary (i.e., just for the sending of this message) physical node identifier. Now the node is waiting for a Clear_To_Relay (CTR) message, and is in a *Requesting* state.

If this RTR message is received successfully by another node with a lower tier number (which we call the next-hop node), then that node replies to the source node. The source node waits for a time t_w in the *Requesting* state before attempting to rebroadcast its RTR message. For each rebroadcast, the source node uses a freshly chosen RSD. This is to reduce the possibility of two source nodes choosing the same RSD. The next-hop node, in order to avoid contention with other potential next-hop nodes, chooses a random back-off time t_b and listens to the channel, before it replies. This again is in contrast with 802.11 where there is no contention between potential receiver nodes, since there is only one fixed receiver node, as determined by the routing protocol. If during this waiting period, the next-hop node hears either a CTR, with the correct RSD and STD, from another next-hop node or data from the source node, it goes back into the *Listener* state. Otherwise, it replies with a CTR message which consists of RSD, STD, as well as a randomly chosen receiver identifier (RRD), and the receiver tier identifier (RTD), in addition to the NAV (see Fig. 3). Now it is waiting for data, and is in the *Receiver* state.

Once this CTR message is correctly received by the source node, a DATA and an ACK message are quickly exchanged between the source node and the next-hop node, using the source and the next-hop node identifiers, for unambiguous identification. Detection of the loss of a DATA or an ACK message is inferred through time-outs of duration t_d at the receiver, and t_a at the sender, respectively. On receiving the data completely, the next-hop node which belongs to a tier closer to the sink, becomes the new source node. Thus, data is forwarded across tiers progressively moving closer and closer to the sink. In this way, AIMRP handles the twin problems of routing and medium access in an integrated fashion (see Fig. 1).

C. Resolution of Protocol Deadlocks

Since AIMRP is based on random-access, there are situations when the protocol could potentially deadlock, unless there are provisions to prevent it. AIMRP is based closely on IEEE 802.11, so it adopts some deadlock resolution mechanisms from 802.11. In particular, AIMRP uses the guard time t_g and the random listening time t_l, in a way similar to 802.11. It also uses the time-outs t_w, t_d and t_a which determine failure of an attempt at transmitting an RTR, a DATA and an ACK message respectively. Finally, it uses a NAV based virtual carrier

[3] Note that it is possible to eliminate the use of the NAV field. For several event detection applications, the event report is expected to be of a fixed length, containing the time, the location and a fixed length code-word describing the event. In such cases, assuming all packets to be of equal length, any transmission could be taken to reserve the channel for the fixed duration of the data transmission, thereby removing the need for the NAV.

TABLE I
AIMRP: Protocol Parameters and Their Functions

t_g	Guard time to reliably estimate channel state (busy or idle)
t_l	Listening time to prevent collision of messages from nodes attempting to transmit at the same time
t_b	Back-off time to avoid collision of CTR messages with other potential next-hop nodes
T_l, T_b	Upper bounds on t_l and t_b respectively
t_w	Waiting time to infer either unavailability of a next-hop node or erroneous transmission of the RTR message
t_d	Waiting time to infer incorrect transmission of DATA message
t_a	Waiting time to infer a lost ACK message
t_p	Total transmission time of all protocol messages (RTR, CTR, DATA and ACK)
t_{on}	On-period in power-saving mode
t_σ	Random sleep duration in power-saving mode (exponentially distributed with parameter σ)
t_r	Listening time corresponding to the highest value of the NAV to avoid collisions in power-saving mode

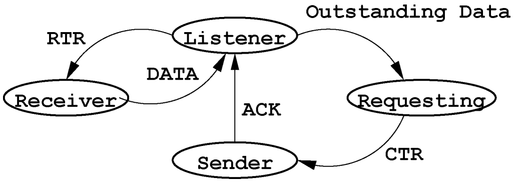


Fig. 5. States and state transitions for AIMRP.

sensing strategy to reserve the channel for the duration of the data communication. In contrast with 802.11 though, the receiver node in AIMRP is not pre-determined. Hence, there is an additional random backoff time t_b in order to prevent potential receiver nodes from colliding with their CTR messages. Also, since random identifiers, as opposed to fixed MAC addresses, are used, these identifiers are chosen afresh for each attempted RTR or a CTR. This reduces the possibility of nodes in the vicinity choosing identical identifiers and then systematically colliding. Refer to Table I for a summary.

The state transitions based on the various protocol messages for AIMRP are illustrated in Fig. 5. Note that Fig. 5 does not include transitions due to various time-outs or due to lost messages. Although the protocol states are analogous to the states for 802.11, the important difference is that the RTR/CTR mechanism based on randomly chosen node identifiers is used to perform a one-hop routing as well, in addition to, organizing communication between the two nodes, as in 802.11.

D. Path Failure and Path Repair

AIMRP is a protocol optimized for event detection and reporting. It could possibly be deployed in hostile surroundings. Nodes following AIMRP could fail either permanently or intermittently. Node failures could either be more or less uniform throughout the network, or they could be concentrated in a particular area in the network. In all these scenarios, it is important for AIMRP to maintain connectivity and continue functioning, in the best manner possible. Now, if a given node (or set of nodes) becomes completely disconnected from the rest of the network, then no routing algorithm will be able to find a path from the node(s) to the sink. However, since AIMRP uses a tier-based routing algorithm, it is possible that although a node is not disconnected, it still finds the sink unreachable, if there is no node with a lower tier-id, in its neighborhood. This could happen if all the neighbours of a node have higher tier-ids. In such a case, we say the TIER_ID of the node is misconfigured.

In order to combat with path failures arising out of misconfigured TIER_IDs, we suggest the following path repair strategy. Note that the TIER_ID of a node, if configured correctly, represents in some sense its distance to the sink, in number of hops. In the configuration phase, nodes set their TIER_IDs as one greater than the lowest ranked TIER message they receive. The rationale behind this is that they are one hop away from a node which knows its distance to the sink. Based on this observation, we suggest the following. Let MAX_TIER_ID denote an upper bound on all TIER_IDs. If a node is unable to send an event report for more than a certain number of tries, PATH_REPAIR_THRESH, then it reattempts the transmission with the STD field (see Fig. 3) of its RTR message set to MAX_TIER_ID. Now unless the node is completely disconnected from the rest of the network, it is bound to receive a CTR reply. On receiving this reply, the node sets its TIER_ID as RTD+1, where RTD (see Fig. 3) is the TIER_ID of the replying node.

This local path repair strategy represents a good approximation to the repair strategies used in several routing protocols such as AODV, DSDV or DSR. We note that a more fool-proof, but expensive technique for route repair, is to re-run the configuration phase periodically. This will enable all the nodes to maintain correctly configured TIER_IDs. In practice, the designer can choose to deploy either of the two strategies (local repair versus periodic configuration) mentioned above, depending on the needs of the application.

E. Power-Saving Mode

There are two major sources of wasteful energy expenditure in a WSN running a random access MAC protocol, namely, *idle-listening* and *overhearing*. A node is said to be in idle mode, if its radio module is on when there is no transmission from any other node. A node is said to be overhearing, if its radio module is on during a DATA message transmission intended for another node. In order to reduce this energy wastage, we need a power-saving mode for AIMRP. Previous works on power-saving schemes include PAMAS [12], S-MAC [16] and the IEEE 802.11 power-saving mode [17]. PAMAS [12] is proposed for use in an ad hoc wireless network of nodes communicating with an any-to-any communication paradigm. PAMAS uses overhearing avoidance to save power. In other words, a node shuts off its radio module during the transmission of a DATA message intended for another node. S-MAC [16] and the IEEE 802.11 power-saving mode [17] use periodic duty-cycling to reduce idle-listening. Specifically, nodes follow a scheduled cycle of on-periods when their radio modules are on, and off-periods with the radio modules off.

In AIMRP, we take a different approach to design our power-saving mode. We propose a *completely asynchronous* and *random* duty-cycling scheme. The basic idea of the power-saving mode in AIMRP is the following. We introduce a

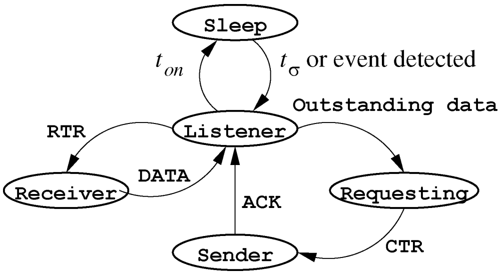


Fig. 6. Sleep state for AIMRP with power-saving.

new state called the *Sleep* state (see Fig. 6), in which nodes shut their radio modules off (i.e., sleep). Nodes in the *Listener* state sleep from time to time, with the length of their random sleep duration t_σ chosen according to an exponential distribution with parameter σ. If a sensor node detects an event when it is in the *Sleep* state it wakes up and moves to the *Listener* state immediately. Otherwise, the node wakes up on expiry of the sleep duration t_σ, remains awake for time t_{on}, and then goes back to sleep for a freshly chosen random sleep duration, except under certain scenarios discussed below. The nodes remain awake for a time t_{on} on waking up, in order to be available to other nodes looking to relay their data closer to the sink. The on-period t_{on} has to be dimensioned in such a way that it enables a node to listen to at least one RTR message from another node looking to relay some data. This can be achieved by requiring $t_{on} \geq t_g + T_l$, where t_g is the guard time, as defined earlier, and T_l is an upper bound on the random listening time t_l defined earlier (refer to Table I). The time t_{on} is smaller than a DATA packet transmission time, t_{DATA} (see Section IV-F).

In general, a node (say node A) following the power-saving mode of AIMRP needs to be awake under the one of the following two scenarios:

1) either node A has `outstanding data` and is attempting to find a next-hop node to relay the data closer to the sink,
2) or node A is merely awake as part of the random duty-cycling, to see if another node needs its help in relaying.

In order to explain the exact protocol behavior of a node in power-saving mode, it is useful to consider these two scenarios separately.

Let us consider scenario 2 first. In this scenario, the default behavior of node A is the simple duty-cycle rule stated earlier. Node A remains awake for a time t_{on}, and then goes to sleep again for a time t_σ. However, if node A receives any protocol messages which it can successfully decode, then it has to respond to them according to the rules of AIMRP. For instance, if node A receives an RTR message from another node (say node B), with a higher tier rank, and node A manages to be the first node to reply with a CTR to the RTR from node B, then node A is the next-hop node for node B. In this case, node A continues following the message exchanges according to AIMRP, and receives the data from node B.

However, if node A infers from the RTR from node B that node B has a lower tier rank than itself, or if node A hears another node reply with a CTR to node B, or if node A hears the preamble of a DATA message from node B intended to some other node, then it will be able to conclude that it is not the next-hop relay for node B. Then, node A has to remain silent until the DATA transmission from node B is concluded, as dictated by the NAV field. In this case, since the on-period t_{on} is anyway shorter than t_{DATA}, node A goes to sleep immediately for a random duration t_σ. We show later that the sleep durations t_σ are orders of magnitude longer than message transmission times. Hence, in this scenario overhearing avoidance is actually implicit since node A goes to sleep for t_{DATA} and much more.

Now let us consider scenario 1. The default behavior of node A in this case, is to attempt to find a next-hop relay by transmitting an RTR message. However, if it hears any activity in the radio channel, it has to respect the physical and virtual carrier sensing rules, and the other rules of AIMRP. For instance, if it receives an RTR from another node (say node B) with a higher tier rank than itself, it has to offer to relay the data from node B, by attempting to send a CTR after a random backoff. However, if node A is able to conclude that it is not the next-hop relay for node B, then it has to remain silent until the DATA transmission from node B is concluded. In this case, we propose to use overhearing avoidance. So node A shuts its radio module off until the conclusion of the DATA transmission from node B, and then wakes up again to attempt sending its own data.

In practice, the decision of whether or not to use overhearing avoidance in scenario 1, depends on whether it is more energy-efficient for nodes to put their radios off and bring them up again, or to just remain awake for the entire duration of the packet transmission (t_{DATA}). Using the notation introduced later in Sections V and VI (refer to Table III), if $E_{up} + E_{dw} \leq P_{on} t_{\mathrm{DATA}}$, then overhearing avoidance should be used in scenario 1. However, due to the infrequency of event reports, it is quite unlikely that two or more *neighboring* nodes *simultaneously* have `outstanding data`, and hence overhear one another. Hence, the difference in the power consumption between using and not using overhearing avoidance, would be negligible. Later in Section VI, for calculating the power consumption of AIMRP, we do not take into account the fact that nodes use overhearing avoidance in scenario 1.

In scenario 1, one of the reasons node A could have `outstanding data` is because it detected an event. Now node A could have been in the *Sleep* state when it detected the event. In this case, node A moves immediately to the *Listener* state and attempts to relay the event report one hop closer to the sink. However, since node A has just woken up, it might have missed the RTR or CTR of an on-going communication in its neighborhood. So it listens to the radio channel for a time t_r corresponding to the highest value of the NAV field. This is done to reduce the possibility of node A colliding with an on-going DATA transmission in the neighborhood.

Since the nodes use multi-hop relaying to send their data to the sink, they cannot put their radio modules off indefinitely. The choice of the parameter σ which governs their sleep-wakeup schedules is determined by the end-to-end latency required by the application. All nodes should wake-up often enough, so that for a given node trying to send a report to

TABLE II
SIZES OF AIMRP MESSAGES

Field	Width	Message	Size
`Message Type`	3 bits	`RTR`	2-3 bytes
MAC or Tier identifiers	4 bits	`CTR`	3-4 bytes
`NAV`	8 bits	`DATA`	≈ 128 bytes
`OPI`	0-4 bits	`ACK`	3-4 bytes

the sink, there will always be nodes available to relay the report within the specified latency period. The dimensioning of σ is discussed in detail in Section V.

F. Setting Back-Off Times and Timeouts

This subsection is intended as a guideline to set the widths (in number of bits) of various subfields in each packet, and to set the values of various back-off times and timeout intervals. In order to distinguish between different message types, we use a three bit message type field. For the random MAC identifiers, we will need an address width of about 4 bits, while for representing the tier number of the nodes, we will again need an address about 4 bits wide. The NAV field is taken to be 8 bits wide, and we assume that the data payload is 1000 bits in size. A summary of bit widths and message sizes follows in Table II.

The following list explains how the different time-out intervals and back-off times are set. These values are based on the physical characteristics of the μamps sensor nodes [11], with a data rate of 500 kbps. Thus, the total time for the transmission of all the messages (RTR, CTR, DATA, and ACK), $t_p = 2.2$ ms. The individual message transmission times are given by $t_{\mathrm{RTR}} = 48\ \mu$s, $t_{\mathrm{CTR}} = 64\ \mu$s, $t_{\mathrm{DATA}} = 2.0$ ms, and $t_{\mathrm{ACK}} = 64\ \mu$s. Also, the value of the time duration t_r, defined in the previous subsection (refer to Table I) can be calculated to be around $t_r = 2.0$ ms.

- t_g: The value of t_g is selected such that a sensor node is able to reliably estimate the busy/idle state of the medium. This should be as small as possible and we choose $t_g = 50\ \mu$s.
- t_l: We take t_l to be uniformly distributed in $[0, T_l]$. The value of T_l should be chosen such that collisions between two senders are avoided as much as possible. We choose $T_l = 500\ \mu$s.
- t_b: Again, we take t_b to be uniformly distributed in $[0, T_b]$. The value of T_b should be chosen to limit the probability of collision between two active receivers which reply to the CTR message. We choose $T_b = 500\ \mu$s.
- t_w: The timeout period, t_w, is used to infer either unavailability of a next-hop node or erroneous transmission of the RTR message. Therefore, t_w must be greater than the maximum back-off time T_b for which a receiver might remain silent. We choose $t_w = 600\ \mu$s.
- t_a, t_d: We choose $t_a = t_d = 50\ \mu$s for inferring lost DATA or ACK messages. This is chosen to be the same as t_g since that is the time it takes a node to reliably estimate a channel.
- t_{on}: In the power-saving mode of AIMRP, the receiver should be awake for long enough to be able to receive an RTR message from some node within its transmission range, at least once, i.e., $t_{on} \geq t_g + T_l$. We choose this value to be 1100 μs to allow, in the worst case, the reception of two RTR messages within one active period.

V. DIMENSIONING OF AIMRP PARAMETERS

There are two protocol parameters in AIMRP that need to be dimensioned for the protocol to work "best", namely, α and σ. The first parameter α which is a measure of the width of each tier, impacts both the connectivity of the network, as well as the average power dissipation. In this section, we investigate how α affects the connectivity, in terms of the number of next-hop nodes available for relaying the data, according to the tier-based routing algorithm of AIMRP. Later in Section VI, we show that the minimum average power dissipation is achieved at $\alpha = 0.45$. The second parameter σ has to be chosen in order to guarantee an end-to-end constraint on the latency of an event report, as specified by the application. For the remainder of the paper, we make the assumption that the nodes are distributed randomly and uniformly over the region with spatial density λ nodes/m^2.

A. Impact of α on Connectivity

In this subsection, we study the effect of α on connectivity, in terms of the number of next-hop nodes that could potentially relay data from a given node. Consider Fig. 2 and suppose that the node indicated by a cross wants to send some data to the sink. Based on the routing algorithm used in AIMRP, the only nodes that could relay the data from this node, would be the ones lying in the hatched region in Fig. 2. The hatched region is the region of overlap of two circles: the first one being a circle centered at the node with radius R which is its communication range; and the other being a TIER circle centered at the sink, with id $n-1$ and radius $\alpha(n-1)R$. Note that, we require α to be less than unity to ensure that any node in the nth tier is able to communicate with the $(n-1)$th tier. Now, irrespective of the power-saving mechanism used, the number of nodes in this region of overlap, is a measure of the connectivity, since eventually only a node from this region will relay the data from the sender node.

The region of overlap shown in Fig. 2 has the minimum area for all nodes in TIER n, since the sender node is at the edge of the tier. It is easy to see that this area will be minimized with respect to n when n is made as small as possible, i.e., at $n_0 = \lfloor 1/\alpha \rfloor + 1$. This is because all nodes in tier $\lfloor 1/\alpha \rfloor$ or lower, are within a distance R from the sink and hence can communicate directly with the sink. Now by the *cosine* rule for triangles, we have $\cos A = ((2n_0 - 1)\alpha^2 + 1)/2n_0\alpha$ and $\cos B = ((n_0^2 + (n_0-1)^2)\alpha^2 - 1)/2n_0(n_0-1)\alpha^2$. Hence, the area of the shaded region is given by

$$\mathcal{A}(\alpha) = R^2(A + (n_0-1)^2\alpha^2 B - n_0\alpha \sin A). \qquad (1)$$

Thus, on an average the number of nodes potentially available to any sender node for relaying its data is at least $\lambda\mathcal{A}(\alpha)$.

Based on this calculation, the appropriate value of λ, for a given value (or range of values) of α, can be dimensioned. However, in this paper, we are not trying to dimension the node density, λ, since it would be a design issue, as opposed to a protocol parameter setting. In what follows, we simply assume that the value of λ chosen, is large enough to provide good connectivity irrespective of the value of α. Later in Section VI, we find that the average power consumed in the entire network, is minimized at $\alpha = 0.45$, independent of λ.

B. Dimensioning σ

The parameter σ is chosen based on an end-to-end latency requirement on a data message. We consider the constraint on latency to be probabilistic. In particular, we assume that the worst-case latency constraint is specified as a probabilistic tolerance of the form:

$$\mathbf{P}\left(\sum_{k=1}^{H} \tau_k \leq \tau\right) \geq 1 - \Phi \tag{2}$$

where τ_k denotes the delay encountered in the kth hop out of H hops in total, τ denotes the specified event report latency objective, and Φ denotes a tolerance on the probability of achieving this latency.

In the equation above, τ_k denotes the delay encountered in the kth hop. In the context of AIMRP, this delay includes the following: A sender node has to wait for a time $t_l + t_g$, and possibly t_r if it has just woken up from the *Sleep* state before sending an RTR message, a time t_b before receiving a CTR message, and a time t_p for the actual transmission of RTR, CTR, DATA and ACK messages, in addition to a time t_{sleep} which represents the delay caused due to the *Sleep* state of the next-hop node. Thus, we have

$$\tau_k = t_r + t_g + t_l + t_b + t_p + t_{sleep}. \tag{3}$$

Since in general, t_{sleep} is expected to be much greater than t_r, t_g, t_l, t_b, and t_p, we can ignore them in (3). Thus, we have

$$\mathbf{P}\left(\sum_{k=1}^{H^A} t_{sleep}^{(k)} \leq \tau\right) \geq 1 - \Phi \tag{4}$$

where H^A denotes the maximum number of hops under an AIMRP setting.

Let us now calculate t_{sleep}. We know from the calculations above that the expected minimum number of nodes available to relay data for any sender node is given by $\lambda\mathcal{A}(\alpha)$. Since nodes repeatedly go to sleep independently, following an exponential distribution, the sender node needs to wait for the first node which wakes up to relay its message. It is possible that some next-hop nodes might already be awake, but in the worst case, all of them could be sleeping when the sender node attempts to transmit. Since the sleep times of all the $\lambda\mathcal{A}(\alpha)$ nodes are exponentially distributed with parameter σ and are independent, the sender node needs to wait (in the worst case) for a random time t_{sleep} which is exponentially distributed with parameter $\sigma\lambda\mathcal{A}(\alpha)$. Thus, we have that $\tau_{sleep} = \sum_{k=1}^{H^A} t_{sleep}^{(k)}$ is an Erlang distributed random variable with parameters $(H^A, \sigma\lambda\mathcal{A}(\alpha))$ which we write as (H^A, σ^A) for ease of notation. Equation (4) then rearranges to

$$\mathbf{P}(\tau_{sleep} > \tau) = \frac{\Gamma(H^A, \sigma^A \tau)}{\Gamma(H^A)} \leq \Phi \tag{5}$$

where $\Gamma(.,.)$ is the upper incomplete Gamma function, and $\Gamma(.)$ is the "complete" Gamma function. From the geometry of the network, H^A is given by $H^A = \lceil L/\alpha R \rceil - n_0 + 1$, since all nodes within tier $n_0 - 1$ can directly communicate with the sink. Thus, given a latency constraint τ, a tolerance Φ and α, the value of σ required to ensure the probabilistic latency guarantee defined as in (2), can be calculated from (5), by substituting for H^A and for $\mathcal{A}(\alpha)$.

Although (5) can be solved numerically, let us obtain an approximate closed form expresssion for σ. Note that τ_{sleep} is the sum over H^A hops of all the one hop delays t_{sleep} which are independent, exponentially distributed random variables with parameter σ^A. Now, if H^A is large, then we can apply the *central limit theorem* and approximate τ_{sleep} by a Gaussian random variable with mean, $m = H^A/\sigma^A$, and standard deviation, $s = (H^A)^{1/2}/\sigma^A = o(H^A)$. Hence, as H^A grows larger the standard deviation, $s = o(H^A)$, can be neglected in comparison to the mean, $m = \Theta(H^A)$. Thus, we can approximate τ_{sleep} to be nearly equal to a constant, $m = H^A/\sigma^A$. We require τ_{sleep} to be less than or equal to the latency τ, and thus we have

$$\sigma \leq \frac{H^A}{\lambda\tau\mathcal{A}(\alpha)}. \tag{6}$$

Note that this is only an *engineering approximation*. However, as we observe later in Section VI, (6) still gives reasonably accurate values of σ even for $H^A \approx 10$ which successfully meet the end-to-end requirement on the latency of the event reports. For a discussion, see Section VI.

VI. Performance Analysis and Simulation Results

In this section, we evaluate the performance of AIMRP through analysis and simulations. In what follows, we calculate the average power consumption of a network running AIMRP with power-saving, and compare this with the power consumption of S-MAC [16]. Then, we provide simulation results to validate our analysis and make some observations. In comparing AIMRP with S-MAC, we couple S-MAC with a zero-cost, optimal routing protocol. To be precise, we assume that S-MAC is coupled with a routing protocol that imposes no additional protocol overhead, and routes packets to the sink in the least number of hops. Even under these favorable conditions for S-MAC, AIMRP outperforms S-MAC for event detection applications.

In a WSN, power is consumed due to three reasons, for sensing the phenomenon of interest, for communicating detected events to the sink via the communication protocols, and for exchanging control information necessary for the protocols. The first component is common to all protocols, and needs to be considered as a constant for dimensioning the initial battery energy of the sensor nodes. In what follows, we only compare the power consumed due to the protocol stack. Table III provides a summary of the important notation.

A. Average Power Consumption in AIMRP

The power consumed in a network running AIMRP can be broken up into two components. First, the network has to detect and report the events of interest. So assuming that the *apriori* frequency of these events is $1/T$, the average power consumed

TABLE III
SUMMARY OF IMPORTANT NOTATION

N	Mean number of sensor nodes
R	Communication range
L	Radius of the region
T	Mean duration between sensor events
τ	Maximum permissible end-to-end latency on event reports
E_{up}, E_{dw}	Energy required to power the radio of a node on and off, respectively
P_{on}, P_{tr}	Power consumption with the radio on, and with the radio transmitting, respectively
σ	$\frac{1}{\sigma}$ is the mean sleeping interval in AIMRP
T_{sw}	Mean sleeping interval in S-MAC
$P_{avg}, E_{report}, E_{hop}$	Average power consumption,energy consumed per event report, and energy per hop (AIMRP)
$P^S_{avg}, E^S_{report}, E^S_{hop}$	Average power consumption, energy consumed per event report, and energy per hop (S-MAC)
H^A_{avg}, H^A	Average and maximum hops (AIMRP)
H^S_{avg}, H^S	Average and maximum hops (S-MAC)

for reporting these events is given by E_{report}/T, where E_{report} is the average energy required per report. Second, each node is running the power-saving mode whereby the node sleeps, wakes up, remains awake for a certain time and sleeps again, and so forth. Since the time with which a given node sleeps is exponentially distributed with mean $1/\sigma$, the total average power consumed due to this process is given by $N(E_{up}+E_{dw}+P_{on}t_{on})\sigma$, where N is the number of nodes, E_{up} and E_{dw} respectively represent the energy required to power a node up and down, and P_{on} is the power consumption when the radio module is on. As discussed in Section IV-E, there are some scenarios when a node may terminate its on-period without staying awake for a time t_{on}. So our analysis actually overestimates the energy consumption. Thus, we have

$$P_{avg} = N(E_{up} + E_{dw} + P_{on}t_{on})\sigma + \frac{E_{report}}{T}. \qquad (7)$$

Now the average energy consumed per event report is given by

$$E_{report} = E_{hop}H^A_{avg} \qquad (8)$$

where E_{hop} is the energy consumed per hop, and H^A_{avg} is the expected number of hops that an event report has to travel. The energy consumed in each hop on an average is given by

$$E_{hop} = t_pP_{tr} + \left(t_r + t_g + \frac{T_l}{2} + \frac{T_b}{2} + t_p + \frac{1}{\sigma^A}\right)P_{on} + \left(\frac{1}{\sigma^A t_w}\right)t_{\mathrm{RTR}}P_{tr} + \left(t_p + \frac{T_b}{2}\right)P_{on}. \qquad (9)$$

The first term is the energy required to transmit the RTR, CTR, DATA, and ACK messages. As defined previously, t_p represents the time required for transmitting all of these messages. The second term is the energy consumed at the sender node due to the radio being on. The different time durations correspond to the average values of the various terms in (3) which defines the kth hop delay τ_k. The third term is the energy consumed by periodically sending RTR messages till a receiver node wakes up from its *Sleep* state. Finally, the fourth term is the energy spent at the receiver node due to the radio being on. Note that we consider a worst case scenario, in terms of power consumption by assuming that the nodes that are involved in the relaying of the event report, begin doing so just at the end of their on-cycle (t_{on}), in the power saving mode. Since the number of nodes N is large, this upper bound for the average power dissipation, is a good approximation.

Noting that the time the sender node waits for a receiver node to wake up, namely, $1/\sigma^A$, is much larger compared to the other terms, we have

$$E_{hop} = \frac{P_{on}}{\sigma^A} = \frac{P_{on}}{\sigma\lambda\mathcal{A}(\alpha)} \approx \frac{P_{on}\tau}{H^A} \qquad (10)$$

where the last equality follows from using the approximation in (6). Substituting from (10), (8) and (6), into (7), and recognizing that $N = \pi L^2\lambda$ we get the following expression for the average power dissipation:

$$P_{avg} = \frac{\pi L^2(E_{up} + E_{dw} + P_{on}t_{on})H^A}{\tau\mathcal{A}(\alpha)} + \frac{P_{on}\tau H^A_{avg}}{TH^A}. \qquad (11)$$

Now in order to calculate H^A_{avg}, consider the following. AIMRP routes messages from nodes based on their tier numbers. Thus, a message originating due to an event at a node in the nth tier, would go through $n - n_0 + 1$ hops before it reaches the sink. The nodes are uniformly distributed over the region of interest with a spatial density of λ nodes/m^2, and an event is equally likely to occur at any node in the region. The area of the nth tier is given by $\pi(2n-1)(\alpha R)^2$. Hence, the probability of an event occurring in tier n is given by $(2n-1)(\alpha R/L)^2$. Thus, we have

$$H^A_{avg} = \sum_{n=n_0}^{\lceil L/\alpha R\rceil - 1}\Big(n - n_0 + 1\Big)(2n-1)\Big(\frac{\alpha R}{L}\Big)^2 + H^A\left(1 - \left(\left\lceil\frac{L}{\alpha R}\right\rceil - 1\right)^2\left(\frac{\alpha R}{L}\right)^2\right) \qquad (12)$$

where the second term accounts for the last tier. Substituting for H^A, H^A_{avg} and $\mathcal{A}(\alpha)$, we can calculate the average power consumption in AIMRP from (11). It may be noted that the average power dissipation turns out to be independent of the density of nodes in the network, owing to the assumption of uniform distribution.

B. Average Power Consumption in S-MAC

First let us formulate and solve the latency constraint that needs to be satisfied when employing S-MAC. Nodes form virtual clusters to synchronize on sleep schedules, i.e., all nodes in a virtual cluster go to sleep and wake-up simultaneously (for details, refer to [16]). Let us assume that sleep-and-wake schedules are of length T_{sw}. Messages get routed through a higher layer routing protocol, which we assume to be *optimal* (i.e., it minimizes the number of hops). Again we assume the latency constraint to be of the form of (2). Due to the relatively long duration of the latency constraint τ, the only significant component of the per-hop delay will be due to the sleep-and-wake cycles of sensor nodes, which gives us an equation similar to (4):

$$\mathbf{P}\left(\sum_{k=1}^{H^S} t_{sleep}^{S-MAC} \leq \tau\right) \geq 1 - \Phi \tag{13}$$

where t_{sleep}^{S-MAC} is the delay caused by the sleep-and-wake cycle of the relaying sensor nodes, and H^S is the maximum number of hops required by the routing algorithm, on top of S-MAC.

Now in S-MAC, messages get routed from one virtual cluster to another. Within a virtual cluster, nodes sleep and wake-up simultaneously, whereas the sleep schedules of two virtual clusters are completely uncorrelated. Thus, the delay t_{sleep}^{S-MAC} is uniformly distributed between 0 and T_{sw}. We can evaluate H^S assuming that the routing algorithm running on top of S-MAC routes messages in the least number of hops. Thus, we have $H^S = \lceil L/R \rceil - 1$, since the message would not suffer any delay on the last hop as the sink is always awake. Then we can solve for T_{sw} as in [16]:

$$T_{sw} \geq \frac{2\tau}{H^S}. \tag{14}$$

The average energy consumed in each hop can be calculated to be the following:

$$E_{hop}^S = t_p P_{tr} + \left(\frac{T_{sw}}{2} + t_p\right) P_{on} + t_p P_{on} \tag{15}$$

where again the first term represents the energy spent in transmitting the messages across a distance R, the second term represents the energy spent in keeping the radio module on at the sender, and the third term represents the energy spent at the receiver due to the radio being on. Again, the component $T_{sw}/2$ is large compared to the other quantities, and so we have the following approximation for E_{hop}^S:

$$E_{hop}^S = \frac{P_{on} T_{sw}}{2}. \tag{16}$$

Hence, the expected total energy consumption per report becomes

$$E_{report}^S = \frac{P_{on} T_{sw}}{2} H_{avg}^S \tag{17}$$

where H_{avg}^S is the average number of hops that an event report may encounter. By following steps similar to those used for calculating H_{avg}^A, we can evaluate H_{avg}^S as follows:

$$H_{avg}^S = \frac{(L+R)(4L-R)}{6RL} - 1. \tag{18}$$

Considering the energy consumption in waking up the nodes every T_{sw} time units, the average power consumption using S-MAC can be given as

$$P_{avg}^S = \frac{N(E_{up} + E_{dw} + t_{on} P_{on})}{T_{sw}} + \frac{E_{report}^S}{T}. \tag{19}$$

Substituting for T_{sw}, E_{report}^S and H_{avg}^S from (14), (17) and (18), we can calculate from the expression above, the average power consumption in S-MAC, with an optimal routing protocol without accounting for the overhead due to the routing protocol.

TABLE IV
PARAMETERS FOR COMPARING S-MAC AND AIMRP

P_{tr}	P_{on}	t_{up}	t_{dw}	t_{on}
100mW	150mW	500μs	500μs	1100μs
R	L	T	λ	τ
100m	500m	6s	1/ 200 m^{-2}	0.6s

C. Some Numerical Results

As a concrete example we consider the parameter values given in Table IV. The values of the physical parameters of the sensor nodes are taken from [11], which contains representative values for μamps sensor nodes. The μamps node is designed for transmitting data up to 1 Mbps at a range of up to 100 m. We choose 500 kbps as a typical data rate. Thus, we can evaluate t_p as defined earlier to be about 2.2 ms. Using (5) we get $\sigma \approx 0.59$ s^{-1} with $\Phi = 0.1$, while from (14) we get $T_{sw} \approx 0.30$ s. Thus, in AIMRP, nodes need to wake up about once in $\mathbf{E}[t_\sigma] = 1/\sigma = 1.7$ seconds. Now in S-MAC, nodes wake up once in 0.3 seconds (as opposed to once in 1.7 seconds in AIMRP) to guarantee the required delay constraint. This clearly demonstrates the efficiency of AIMRP. The efficiency comes through randomizing the sleep and wake cycles of all the nodes. As opposed to such randomized sleep and wake cycles for individual nodes, S-MAC uses fully synchronized sleep and wake cycles for nodes within a cluster and uncoordinated sleep and wake cycles for different clusters.

Now, let us compute the average power consumption, for the two protocols. For calculating E_{up} and E_{dw}, we take an approach similar to the one suggested in [11]. In particular, we take $E_{up} = P_{on} t_{up}$ and $E_{dw} = P_{on} t_{dw}$. Here t_{up} is the transceiver stabilization period. Within this startup period, the transceiver cannot operate because the phase-locked loop (PLL) circuitry is not locked to the carrier frequency yet. For the μamps node, $t_{up} \approx 500$ μs. We also take the time required for powering the transceiver down t_{dw} to be 500 μs. The power consumed by a node with its radio module on, P_{on}, depends on whether the node is receiving or transmitting. In the transmitting mode, $P_{on\text{-}tx} = 81$ mW while in receiving mode $P_{on\text{-}rx} = 180$ mW. According to [11], the receiving mode has a larger power consumption because the receiving circuitry is more complex than the transmitting circuitry. For keeping the analysis simple we take $P_{on\text{-}tx} = P_{on\text{-}rx} = P_{on} = 150$ mW. P_{tr} denotes the power consumption required for transmission. We take this value to be 100 mW for distances of up to 100 m.

For these values (see Table IV), we get the following results:

$$\left.\begin{array}{lll} E_{hop} = 12.64 \text{ mJ} & E_{report} = 65.22 \text{ mJ} & P_{avg} = 0.74 \text{ W} \\ E_{hop}^S = 23.37 \text{ mJ} & E_{report}^S = 65.44 \text{ mJ} & P_{avg}^S = 4.13 \text{ W} \end{array}\right\}. \tag{20}$$

Equation (20) shows that AIMRP outperforms S-MAC, in terms of average power consumption, for rare event applications.

There is one concern for using the average power consumption as a performance metric for comparing AIMRP with S-MAC. Namely that, in our model, the sink is located at the center of the region, and hence all the event reports are directed towards the sink. Hence, the relaying burden on the nodes closer to the sink, is expected to be higher than that on the nodes farther away. Thus, the nodes closer to the sink would run out of power earlier, thereby leaving the sink disconnected from the rest of the nodes. In such cases, the right metric for comparing two protocols, should be based on the useful lifetime of the sensor network, starting with the same initial battery energy (see for instance, [7]). While this observation is valid for any WSN, and affects all the currently proposed protocols for WSNs, the use of the average power consumption, as a performance metric is justified because of the following reason. For the application scenario we have considered (i.e., rare event detection), the problem of non-uniform energy drainage is not so severe. In fact, we show later through simulations that the energy consumption is nearly constant across the different tiers.

D. Simulations and Observations

In order to evaluate the performance of AIMRP and to validate the analysis above, we have simulated AIMRP for a range of values of the parameters, T, τ, α and λ.

First we describe the simulation setting. We consider a circular region with a radius $L = 500$ m. We take the communication range for each sensor node to be $R = 100$ m. The sensor nodes are uniformly and randomly distributed in the region, with an average node density $\lambda = 1$ node per 200 m^2. Sensors detect events which are equally likely to occur anywhere in the region. The inter-event times follow an exponential distribution, with an average value of $T = 6$ s. The maximum tolerable sensor-to-sink latency is set as $\tau = 0.6$ s. The values of all the other relevant parameters are as in Table IV and α is chosen to be 1/2 wherever necessary. The value of σ is calculated from the approximate formula in (6).

We simulate AIMRP, with the power-saving mode, according to the specifications of the protocol described in Section IV. However, we have not implemented the channel-sensing to determine idle/busy channel state (i.e., we assume all the transmissions are error and collision free). This assumption is justified because the events are rare for all the combinations of parameter values studied. For each setting of the parameters, the simulation is run for 10,000 simulation seconds. Each point on all the plots that follow represents the average of the plotted quantity over the entire simulation run. We have observed that the difference between the maximum and minimum values of the plotted quantity, is within 5% of its average value.

We observe that AIMRP performs as expected. The average delays (in seconds) encountered by event report messages versus the tier-id are shown in Fig. 7. Also shown are the maximum delay values encountered (in seconds). The scale for the average delays is on the right-hand side, while that for the maximum delays is on the left-hand side. From this we conclude that AIMRP successfully meets the specified latency guarantees (i.e., the delays are always below 0.6 s). Even though, the value of the power-saving parameter, σ, is calculated using the approximate formula in (6), we find that AIMRP is still able to meet the latency guarantees successfully. We feel that this is due to the fact that the parameter σ is dimensioned 1) using the smallest area of overlap $\mathcal{A}(\alpha)$, and 2) assuming that each event report may have to go through H^A hops before it reaches the sink.

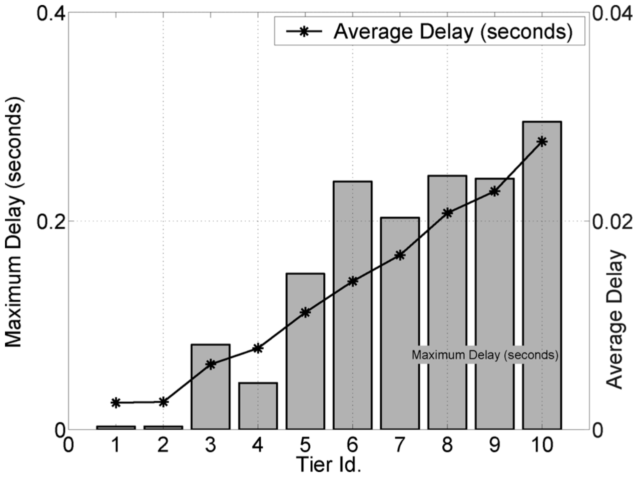


Fig. 7. Average message delay for all tiers.

TABLE V
AVERAGE POWER CONSUMPTION PER NODE VERSUS TIER-ID

Tier-Id	Power Per Node (μW)
1	185.887
2	187.914
3	186.813
4	186.041
5	185.833
6	185.810
7	185.577
8	185.542
9	185.305
10	185.262

Table V lists the average power consumption per node as a function of the tier-id. The power consumed by different nodes is observed to be nearly independent of their tier-ids. This is due to the fact that the energy consumption in repeatedly turning the radio on, keeping it on for t_{on}, and then powering it down, dominates the total energy consumption. Compared to this, the energy required to transmit and receive messages is negligible. Since the sleep-and-wake frequency, σ is the same for all the tiers, the energy consumed by a node is independent of its tier-id.

To understand the effect of the parameters α, λ, τ, and T, on the energy consumption of AIMRP and S-MAC we plot the average power consumption as a function of each of these parameters for the two protocols. In all of the following plots, the parameter on the horizontal axis is varied as specified, while the other parameters are set as given in Table IV. The continuous line in all the plots represents the curve obtained via analysis, using (11). The simulation points for AIMRP are indicated by a "+", and they are obtained, as explained earlier, by averaging over a simulation run of 10,000 simulation seconds. The average power consumption of the WSN as a function of α is depicted in Fig. 8. The value of α which achieves the minimum average power consumption, can be obtained numerically from (11), as $\alpha = 0.5$. However, as can be seen from Fig. 8, there is very little variation in the average power consumption for $\alpha \in [0.3, 0.5]$.

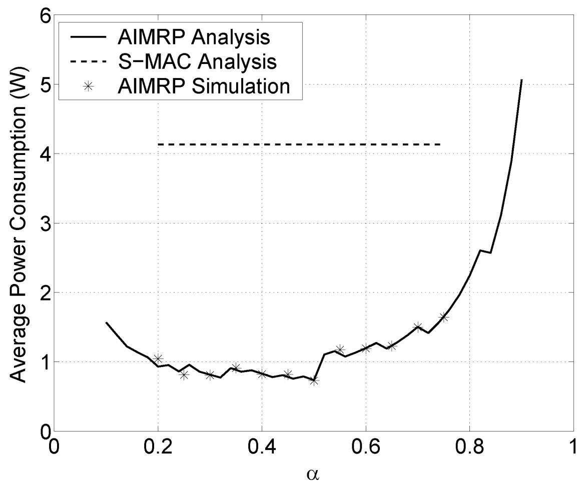


Fig. 8. Power consumption versus α.

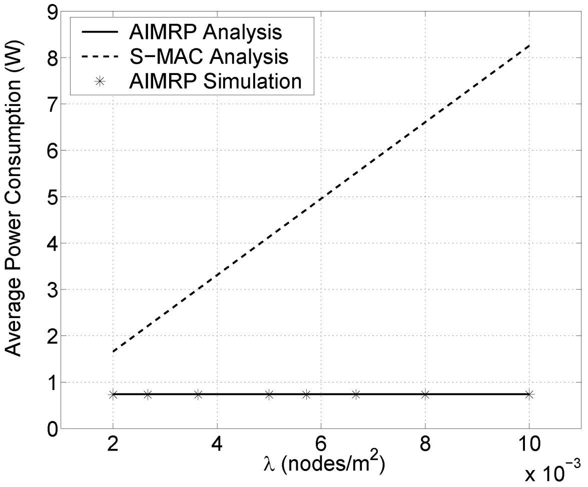


Fig. 9. Power consumption versus λ.

From Fig. 9, we observe that the average power consumption for S-MAC increases with λ, while it does not change for AIMRP. This is also as expected [see (11) and (19)] because of the following reason: as the number of nodes increases, S-MAC consumes more and more energy due to the periodic wakeup of all clusters; while the power consumption of AIMRP is constant since with more nodes they have to sleep and wake up less often.

We see from Fig. 10 that for large values of T the average power consumption of AIMRP and S-MAC remains nearly unchanged. But as T decreases (i.e., the events become more frequent) the power consumption due to the reporting of events, starts dominating giving a sharp increase in the average power consumption. Although AIMRP has a lower power consumption than S-MAC for all values of T, we note that the power consumption of both the protocols may increase rapidly due to collisions as T decreases. As the latency constraint τ, becomes more strict (see Fig. 11) we observe that AIMRP performs better than S-MAC in terms of the average power consumption. Thus, in all scenarios considered, AIMRP is a more energy-efficient protocol than S-MAC.

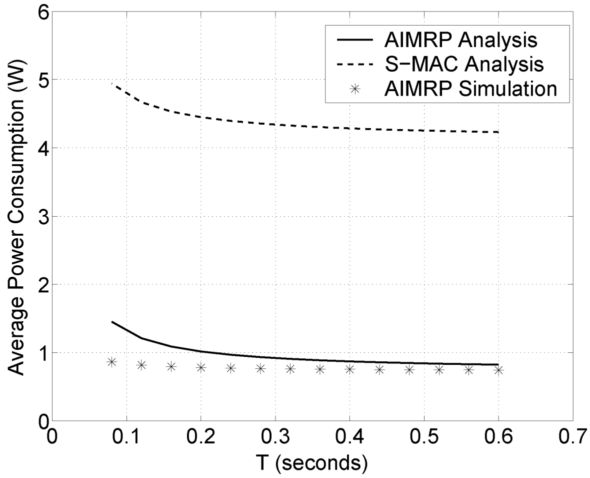


Fig. 10. Power consumption versus T.

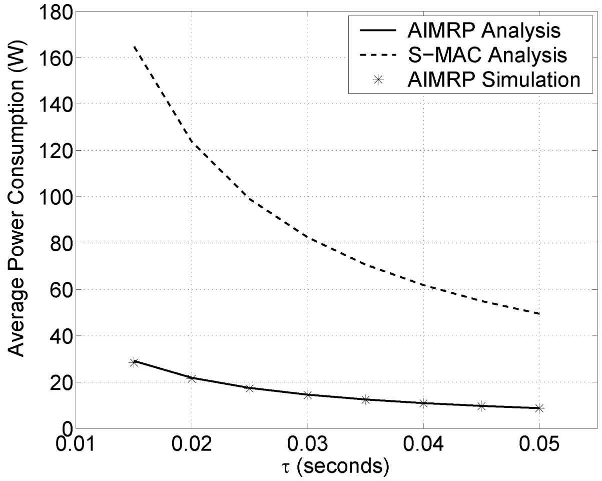


Fig. 11. Energy consumption versus τ.

VII. Conclusions and Extensions

Unlike a traditional network, where individual nodes communicate with each other independently, a WSN is deployed for a certain common objective which all nodes collaborate to achieve. Hence, the design of a WSN should be dictated by the end objectives that it seeks to achieve, rather than any other design considerations like layering or interoperability. In this paper, we consider a class of applications which can be described as *event detection and reporting*. We develop a simple model for this application class: namely that the events have a certain *apriori* frequency of occurence, and that they can occur with equal likelihood in the region of interest. Based on this model, we propose a protocol design (AIMRP) that is both address-light, employing non-unique node identifiers, and integrated, by combining the medium access and routing functionalities. With a given latency constraint on the event report, we design a power-saving mode to reduce energy drainage due to idle-listening. AIMRP outperforms S-MAC in terms of average power dissipation. This is, in fact, due to the fact that S-MAC is generic and unoptimized for this application class. There are some possible extensions to this work. The model is simplistic in the sense that it assumes that only one node detects an event. Also there are no special provisions in the protocol to handle a burst in the traffic. The protocol, as such, would survive although the performance could degrade considerably.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality of the paper.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw. J.*, pp. 393–422, Mar. 2002.

[2] K. Arisha, M. Youssef, and M. Younis, "Energy-aware TDMA-based MAC for sensor networks," presented at the IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), New York, May 2002.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proc. MOBICOM'01*, Rome, Italy, Jul. 2001, pp. 85–96.

[4] C. Guo, L. C. Zhong, and J. M. Rabaey, "Low power distributed MAC for ad hoc sensor radio networks," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 2001, pp. 2944–2948.

[5] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. MobiCOM'00*, Boston, MA, Aug. 2000, pp. 56–67.

[7] V. Mhatre and C. Rosenberg, "Design guidelines for wireless sensor networks: Communication, clustering and aggregation," *Ad Hoc Networks J.*, vol. 2, no. 1, pp. 45–63, 2004.

[8] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogenous sensor network with a lifetime constraint," *IEEE Trans. Mobile Comput.*, vol. 4, no. 1, pp. 4–15, Jan. 2005.

[9] R. Min and A. Chandrakasan, "Energy-efficient communication for ad-hoc wireless sensor networks," in *Proc. 35th Asilomar Conf. Signals, Systems, and Computers*, Nov. 2001, vol. 1, pp. 139–143.

[10] A. Muqattash and M. Krunz, "Power controlled dual channel (PCDC) medium access protocol for wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Apr. 2003, pp. 470–480.

[11] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks," in *Proc. MOBICOM*, Rome, Italy, 2001, pp. 272–287.

[12] S. Singh and C. S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad-hoc networks," in *Proc. MOBICOM'98*, Oct. 1998, pp. 181–190.

[13] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Pers. Commun. Mag.*, vol. 7, no. 5, pp. 16–27, Oct. 2000.

[14] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless microsensor network models," *ACM Mobile Comput. Commun. Rev.*, vol. 6, no. 2, Apr. 2002.

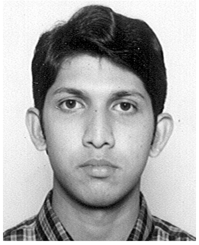
[15] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. MOBICOM'01*, Jul. 2001, pp. 70–84.

[16] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, 2002, pp. 1567–1576.

[17] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11-1999, IEEE 802 LAN/MAN Standards Committee, 1999 [Online]. Available: http://standards.ieee.org/getieee802/802.11.html

Sunil Kulkarni received the B.E. degree in electronics engineering from Walchand College of Engineering, Sangli, India, in 1997, the M.E. degree in electrical communication from the Indian Institute of Science, Bangalore, in 1999, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2004 and 2005, respectively.

He is currently working at Google Inc., Mountain View, CA. His research interests include wireless cellular networks, ad hoc networks and sensor networks, and performance modeling of communication networks.



Aravind Iyer (S'03) received the B.Tech. and M.Tech. degrees in electrical engineering from the Indian Institute of Technology, Bombay, in August, 2002. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

His research interests include the design, modeling and optimization of wireless sensor networks and ad hoc networks.

Catherine Rosenberg (SM'95) received the M.Sc. degree from the University of California at Los Angeles (UCLA) and the Doctorat en Sciences from the University of Paris, Orsay, France.

She is currently the Chair of the Department of Electrical and Computer Engineering at the University of Waterloo, Canada, where she also holds a University Research Chair. She has authored over 70 papers and has been awarded six patents in the USA.

Dr. Rosenberg is currently the Series Editor for the Series on Adhoc and Sensor Networks for the *IEEE Communications Magazine* and the Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING.