Power Save Mechanisms for Multi-Hop Wireless Networks*

(Invited Paper)

Matthew J. Miller
Dept. of Computer Science, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
mjmille2@uiuc.edu

Nitin H. Vaidya
Dept. of Electrical and Computer Engineering, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
nhv@uiuc.edu

Abstract

In this paper, we discuss power save mechanisms that allow hosts to go to sleep to conserve energy. When sleeping hosts need to receive packets from other hosts. Thus, these hosts must somehow wake up the sleeping hosts. The paper discusses a busy-tone mechanism for this purpose, and an approach to improve the mechanism by using multiple busy tones. We generalize this protocol to develop the notion of multi-level power save mechanisms, and present some examples to illustrate this notion.

1. Introduction

Sensor networks present many challenges. While the precise application of sensor networks is speculative, certain characteristics are typically assumed. First, the sensors are relatively static after initial deployment. Second, energy is scarce and it is inconvenient or impossible to replenish the energy source frequently.

Because energy should be conserved, power save protocols are needed. This problem can be addressed at each layer of the protocol stack. We restrict our focus to the Medium Access Control (MAC) and network layers in this paper. Radios typically have at least four power levels corresponding to the following states: transmitting, receiving, idle listening, and sleeping. Typically, the power required to idly listen is about the same as the power to transmit and receive. The sleep power is usually one to four orders of magnitude less. For Mica2 Mote sensors [7], these power levels are shown in Table 1. Thus, a sensor should sleep as much as possible when it is not engaged in communication.

* This work was partially funded by NSF grant ANI-0125859 and a NDSEG fellowship.

Radio State	Power Consumption (mW)
Transmit	81
Receive	30
Idle Listening	30
Sleep	0.003

Table 1. Characteristics of a sensor radio [7].

The fundamental question MAC layer power save mechanisms seek to answer is: *When should a device switch to a low power mode and for how long?* Power save protocols attempt to address each of the four major sources of energy waste at the MAC layer [23]: collisions, overhearing, control overhead, and idle listening.

At the network layer, one research direction has focused on determining routes based on the amount of energy remaining in nodes along the route [6, 16]. However, the network layer focus in this paper is on trading off latency and energy by selecting a subset of nodes on a data forwarding path to consume more energy. By consuming more energy, these selected nodes are able to reduce the latency of forwarded data packets. Such an approach has been used in various forms in previous work [1, 4, 24].

In Section 2, we review related work. In Section 3, we present a MAC protocol which uses busy tones and past traffic history to efficiently wake up desired neighbors. The disadvantage of this approach is that each busy tone must wake up a node's entire neighborhood since the intended receiver's identifier is not encoded on the wake-up channel. In Section 4, we explore the possibility of using multiple busy tones to wake up only a subset of a node's neighborhood. Section 4 also presents an algorithm to optimally assign two channels in a single-hop environment (i.e., all nodes are within range of each other) to minimize the amount of wake-ups in the network. We discuss the routing issue in Section 5, where we present the problem in the context of a multi-level scheme. In particular, we explore the design space where nodes on the active paths use a more energy consuming power save protocol to achieve a lower latency; nodes not on an active path consume less energy, but require a larger latency to engage in communication. We conclude the paper in Section 6.

2. Related Work

The IEEE 802.11 specification [5] is the WLAN standard currently in common use. It specifies a MAC protocol for wireless access in both ad hoc environments, called the Distributed Coordination Function (DCF), and centralized systems, called the Point Coordination Function (PCF). Additionally, a Power Save Mode (PSM) is also specified in the standard.

For 802.11's PSM, nodes are assumed to be synchronized and awake at the beginning of each *beacon interval*. After waking up, each node stays on for a period of time known as the *Ad hoc Traffic Indication Message (ATIM) window*. During the ATIM window, since all nodes are guaranteed to be on, packets are advertised that have been queued since the previous beacon interval. These advertisements take the form of ATIM packets. More formally, when a node has a packet to advertise, it sends an ATIM packet to the intended destination during the ATIM window (following the rules of IEEE 802.11's CSMA/CA mechanism). In response to receiving an ATIM packet, the destination will respond with an ATIM-ACK packet (unless the ATIM specified a broadcast or multicast destination address). When this ATIM handshake has occurred, both nodes will remain on after the ATIM window and attempt to send their advertised data packets before the next beacon interval, subject to CSMA/CA rules. If a node remains on after the ATIM window, it must keep its radio on until the next beacon interval. If a node does not receive an ATIM or ATIM-ACK (assuming unicast advertisements), it will enter sleep mode at the end of the ATIM window until the next beacon interval. This process is illustrated in Figure 1. The dotted arrows indicate events that cause other events to occur. Node **A** sends a data packet to **B**, while **C**, not receiving any ATIM packets, returns to sleep for the rest of the beacon interval.

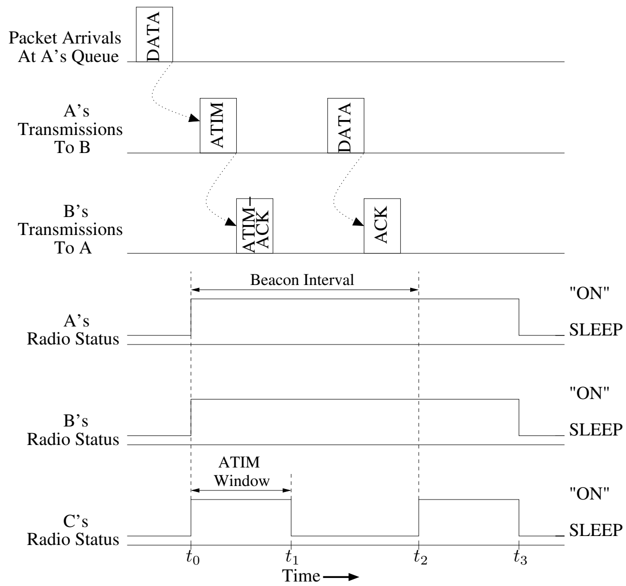


Figure 1. IEEE 802.11 power save mechanism [5].

LISP [4] adapts 802.11 PSM to predictively wake up nodes based on overheard ATIM-ACKs. The basic idea is when data is being sent on path $\mathbf{A} \rightarrow \mathbf{B} \rightarrow \mathbf{C}$, then $\mathbf{C}$ should remain on in any beacon interval in which it overhears $\mathbf{B}$ sending an ATIM-ACK to $\mathbf{A}$ during the ATIM window.

S-MAC [23] is a protocol developed specifically to address energy issues in sensor networks. It uses a simple scheduling scheme to allow neighbors to sleep for long periods and synchronize wake-ups. In S-MAC, nodes enter sleep mode when a neighbor is transmitting and fragment long packets to avoid costly retransmissions. T-MAC [19] extends S-MAC by adjusting the length of time sensors are awake between sleep intervals based on the communication of nearby neighbors. Thus, less energy is wasted due to idle listening when traffic is light. Other attempts at sensor network-specific protocols have tried to combine routing and MAC [20] to propagate data to a base station. This scheme uses beaconing to form a logical tree with the base station as the root and an adaptive rate control mechanism at the MAC layer.

The PAMAS protocol [12] adapts basic mechanisms of IEEE 802.11 [5] to a two-radio architecture. PAMAS allows a node to sleep to avoid overhearing a packet intended for a different destination or to avoid interfering with another node's reception by transmitting. However, unlike our work, it ignores the idle listening problem.

Other work has proposed using a second radio to wake up sleeping nodes. PicoRadio [3, 11] uses custom, low-power hardware to wake up specific neighbors. In [15], off-the-shelf hardware is used to wake up wireless devices in a centralized environment. Another centralized approach is presented in [2], where an RF wake-up channel is used to design a protocol which minimizes energy consumption while meeting QoS requirements

STEM [13,14] is a two-radio architecture which achieves energy savings by letting the primary radio sleep until communication is necessary while the wake-up radio periodically listens according to a specified duty cycle. When a node has data to send, it begins transmitting continuously on the wake-up channel long enough to guarantee that all neighbors will receive the wake-up signal. A variant of STEM [13] has been proposed that uses a busy tone, instead of encoded data, for the wake-up signal. Our protocol is similar to STEM, but achieves greater energy savings by periodically listening on the data channel and buffering packets.

In [24], energy is saved by integrating routing and MAC layer functionality. The protocol works with on-demand routing and uses 802.11's PSM when a node is not engaged in sending, receiving, or forwarding data. When a node is communicating, soft-timers are used to transition the node to an idle listening mode which reduces latency and preserves throughput better than using 802.11's PSM. However, the timers do not adjust to the traffic rate, so if traffic is not frequent enough to refresh the timers, the benefits of the protocol are lost. This technique is a special case of the general idea of multi-level wake-up discussed in Section 5.

Other work has also proposed designating a subset of nodes to idly listen to improve performance while most of the nodes remain in a sleep state to conserve energy. Most other work only considers geographical location, not traffic patterns. In AFECA [21], nodes sleep based on the size of their neighborhoods since a node with more neighbors is likely to be able to sleep without disconnecting the network. Similarly, SPAN uses neighborhood information to form a "virtual" backbone. GAF [22] uses GPS to form a grid such that adjacent boxes are guaranteed to be within communication range of each other. Then, selected nodes in each box remain on to keep the network connected.

3. Busy-Tone Power Save MAC Protocol

In this section, we will give described our previously proposed MAC protocol. We will only give a broad overview of the protocol and highlight some representative results. A detailed discussion and analysis of this protocol is available in previous publications [8–10].

We assume that a second radio is available to wake up neighbors [3, 12, 13]. We characterize the second radio as being able to transmit and detect a busy tone. The busy tone gives us a binary channel (i.e., it is either idle or busy), but no data is encoded on it. Busy tones allow for simpler, more energy efficient hardware designs. However, wake-ups must be broadcast instead of unicast to a particular receiver. Our protocol is designed to reduce the effects of these broadcast wake-ups on the energy consumption of the nodes. We accomplish this by selectively waking up the data radio of nodes that have previously engaged in communication via rate estimation. To differentiate between the two radios/channels, we use *primary* to refer to the radio used for sending data and control packets and *wake-up* to denote the radio which transmits and detects wake-up signals.

On the wake-up radio, we use the STEM-BT protocol [13] where nodes periodically listen to the channel according to a pre-determined *duty cycle*. Thus, a node will listen for a busy tone on the wake-up channel for τ_1 time, then sleep for τ_2 time, where $\tau_1 \ll \tau_2$. Given this notation, we define the duty cycle of the wake-up channel as: $\frac{\tau_1}{\tau_1+\tau_2}$. Thus, a lower duty cycle reduces idle listening energy, but increases the delay to wake up a node's neighborhood. A queue threshold, L, is specified as follows: when the queue holds L packets, a wake-up signal must be sent so the queue size can be reduced by transmitting packets to a receiver immediately. Thus, this threshold could be used to control delay or limit the storage usage on a sensor. We refer to this as a *full wake-up* because all sensors within one hop of the sender, after detecting the signal, must wake up their primary radio. After the full wake-up, nodes then listen on the primary channel for a *filter packet*, similar to an RTS (Request-To-Send), to inform the neighbors which nodes should stay on and which can return to sleep. To avoid the energy costs of full wake-ups, rate estimation is done and a *triggered wake-up* is scheduled by communicating nodes T seconds after the previous data transmission.

Figure 2 illustrates this concept for with a fixed T value (our protocol dynamically adjusts T in response to the sending rate) and $L = 2$. The dotted arrows represent a "causes" relationship between events. At t_0, a triggered wake-up occurs T time after the last transmission, even though the sender's queue contains less than L packets. A full wake-up begins at t_2 because the sender's queue reaches size L. At t_4, all neighbors are guaranteed to have their primary radios on, so a filter packet (shown as **F** in the figure) and L data packets (shown as **D**) are sent on the primary channel.

Selecting the T value presents a trade-off as shown in Figure 3. When T is too small for a given sending rate, *empty triggered wake-up* occur where nodes wake up when there is no data to send. If T is too large, then the L threshold will be reached with a higher probability and a full wake-up will occur. For Poisson traffic, we analytically determine the optimal value of T that minimizes energy consumption based on the sending rate and use that value in our simulations. See references [8–10] for more details.

3.1. Experimental Results

To observe our protocol's performance, we simulated it with *ns-2* [17]. In this section, we give some representative results. For brevity, we omit a detailed discussion of the simulation setup and analysis of results. This information can be found in [9]. For the results presented in this pa-

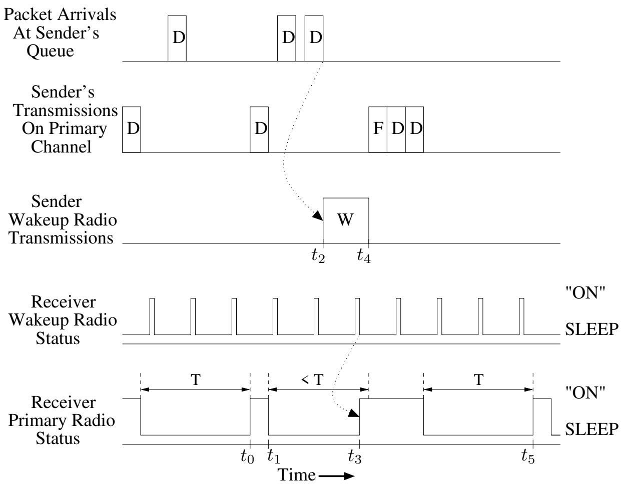


Figure 2. Static T and $L = 2$ (D = data packet, F = filter packet, W = wake-up signal).

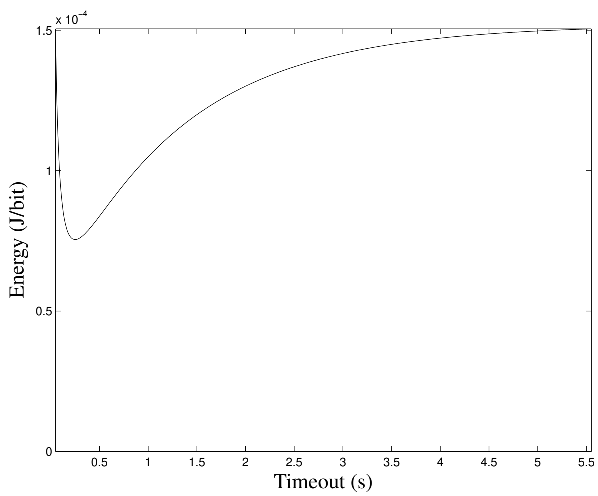


Figure 3. Energy versus timeout.

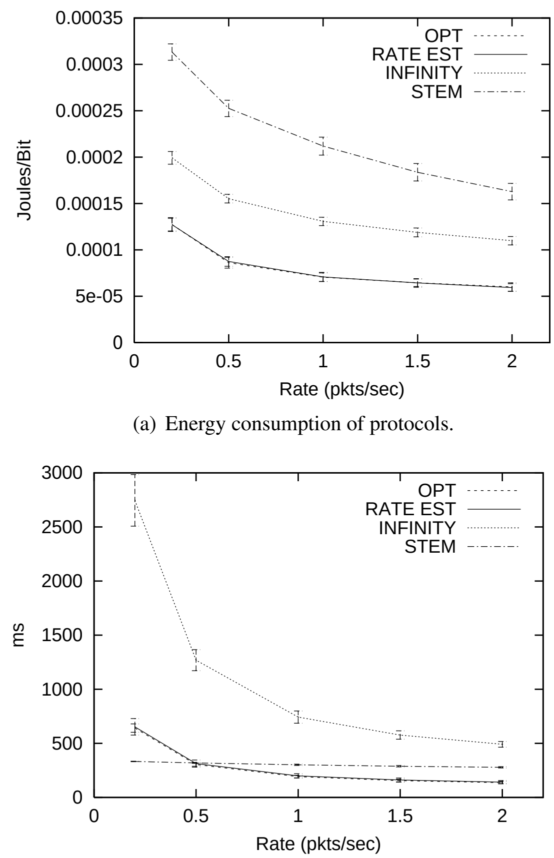
per, $L = 2$ and the sending rate remains constant throughout the simulation. The protocols we evaluated are:

Rate Estimation (RATE EST) Our proposed protocol with T dynamically adjusted by maintaining a weighted average of the packet sending rate.

Static Optimal (OPT) Our proposed protocol with a static T value that is determined analytically.

STEM This is a version of STEM-BT [13] with $L = 1$ and *no* triggered wake-ups (i.e., all wake-ups are full wake-ups and $T = \infty$).

$T = \infty$ (INFINITY) Similar to STEM except $L = 2$ to determine the effects of buffering.



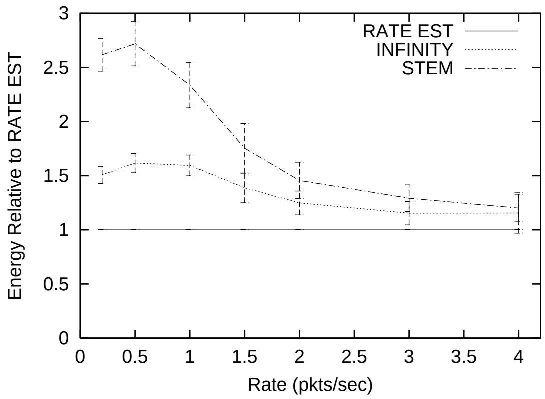
(a) Energy consumption of protocols.

(b) Latency of protocols.

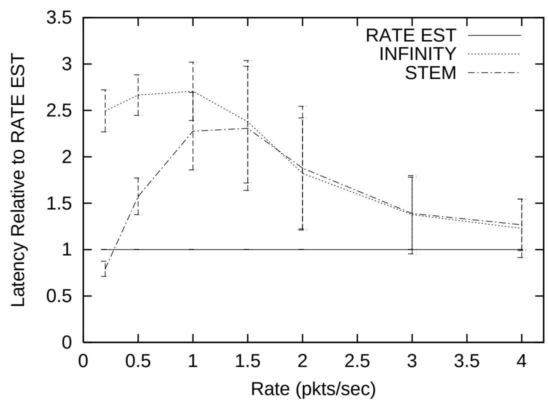
Figure 4. Comparison of protocols in single-hop environment.

Figure 4(a) plots the energy consumption of the protocols in a single-hop setting with one sender and one receiver (rate is on the horizontal axis). We see that regardless of rate, our protocol and the static optimal result in comparable energy consumption (the two curves almost overlap), which is significantly lower than the other protocols. Figure 4(b) shows that our protocol does well in terms of average packet latency as well. Only at low loads does STEM do better when the full wake-up latency is less than our protocol's chosen T value.

To determine the performance of the protocols in more realistic settings, we tested randomly generated, multi-hop topologies with multiple flows. The results are shown in Figure 5. Due to the inherent randomness in generating multiple topologies for each data point, we present the values relative to our protocol for each topology. The horizontal axis specifies the sending rate for *each* individual flow in a given scenario. Figure 5(a) and Figure 5(b) show that our protocol performs better than the other protocols in nearly



(a) Relative energy consumption for random network topologies.



(b) Relative latency for random network topologies.

Figure 5. Random multi-hop, multi-flow network topologies (relative values).

all scenarios in terms of energy and latency, respectively (OPT is omitted since we do not have an analytically optimal T for multi-hop scenarios).

4. Multiple Wake-up Channels

In previous work [8–10], we assume all nodes must share the same wake-up channel. However, we now consider the case in which the wake-up channel can be partitioned in some manner. This could be done by using non-interfering frequency bands. The advantage of using multiple wake-up channels is that full wake-ups will only affect a subset of the sender's neighbors, thus reducing their cost in terms of energy.

Having multiple wake-up channels also introduces new problems. First, there is wake-up channel assignment. Ideally, we would like to assign the channels in a way that minimizes the energy consumption of the network. This is difficult because the traffic patterns may not be known when the channels are assigned. Also, the algorithm for assignment must be performed in a distributed manner where neighbors may negotiate to determine which channel a node should use. This is particularly challenging in multi-hop networks where a node's neighbors may suggest different channels. Next, there is the problem of discovering on which of the wake-up channels each of a node's neighbors are listening. This is very difficult if there is no synchronization in the network since the nodes may not know when others will have their radios on. Also, if the channels are dynamically assigned in response to changes in traffic, the problem may become more difficult. Each of these problems is a direction for future research.

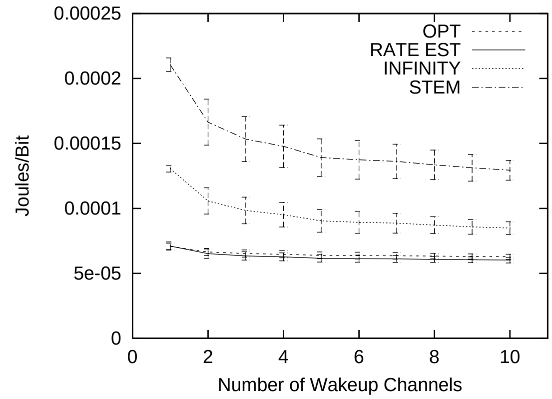


Figure 6. Energy consumption with multiple wake-up channels.

For our initial investigation of multiple wake-up channels, we consider a very simple assignment algorithm where each node chooses a channel randomly (from a uniform distribution). That is, if there are k channels, a node will select a given channel with probability $\frac{1}{k}$. We currently ignore the wake-up channel discovery issue and assume a node knows its neighbors' chosen channels. We tested a scenario where eight nodes are within range of each other, and there is one sender-receiver pair which communicates with rate of one data packet per second. Figure 6 shows the energy consumption of the protocols as a function of the number of wake-up channels used.

It is easy to see that STEM, the protocol with the most full wake-ups, benefits most from using multiple wake-up channels. However, the rate estimation protocol, which is designed to significantly reduce the number of full wake-ups, only shows marginal improvement. The improvement of the $T = \infty$ scheme is in between STEM and rate estimation since it causes more full wake-ups than rate estimation, but less than STEM.

4.1. Optimal Channel Assignment in a Single-Hop Environment

While random channel assignment is simple to implement and requires no knowledge of topology and traffic patterns, it is not optimal in the sense that it does not necessarily minimize the number of full wake-ups in a network. When the wake-up rate per sender-receiver pair is known, we have designed a channel assignment algorithm that minimizes the number of nodes waking up in response to full wake-up signals in a single-hop network (i.e., all nodes can communicate with each other directly). Our algorithm makes the assumption that two wake-up channels are available.

Initially, assume there is one sender with an outgoing link to each of the N receivers. We have (n, r_n) tuples, where r_n is the full wake-up rate to node n. Note that this rate may be different than the data sending rate if, for example, the sender if buffering some packets or performing triggered wake-ups in addition to the full wake-ups. We denote the two wake-up channels available for partitioning as C_1 and C_2. To state the problem more formally, we want to partition the N wake-up rates so as to minimize the function $f(C_1, C_2)$:

$$f(C_1, C_2) = |C_1| \times \sum_{n \in C_1} r_n + |C_2| \times \sum_{n \in C_2} r_n \qquad (1)$$

Our algorithm is as follows. First, the nodes are sorted in order of r_n. This takes $O(N \log N)$ time. Next, we begin by placing the node with the largest r_n value, without loss of generality, in partition C_1 and compute $f(C_1, C_2)$. We continue placing the node with the next largest r_n value in C_1 until such a move no longer causes $f(C_1, C_2)$ to decrease from its previous value. Thus, one channel will have the $|C_1|$ nodes with the highest r_n values and the second channel will have the $|C_2|$ nodes with the lowest r_n values. The first time $f(C_1, C_2)$ is computed, it takes $O(N)$ time since a summation is done over N elements. However, each subsequent computation of $f(C_1, C_2)$ only takes $O(1)$ time since the largest value in C_2 is subtracted out, added to the sum for C_1, and then one multiplication is then done for each set. Thus, the overall running time to find the partition point for C_1 and C_2 is $O(N)$ to do the linear scan with an $O(1)$ operation at each step. Thus, the overall execution time is $O(N \log N)$ due to the sorting operation.

Proof of the Algorithm's Optimality: First, we will show that to minimize $f(C_1, C_2)$, $|C_1| \leq |C_2|$. Initially, without loss of generality, place all N nodes in C_1 (i.e., $|C_1| > |C_2|$). We want to continue to move the smallest element from C_1 to C_2 while such a move will decrease $f(C_1, C_2)$ from its previous value. For ease of notation, let $s_1 = \sum_{n \in C_1} r_n$ and $s_2 = \sum_{n \in C_2} r_n$. We use r_{min} to denote the minimum r_n value among of all nodes in C_1. Let $f_i(C_1, C_2)$ refer to the value of $f(C_1, C_2)$ after the i-th move. We have:

$$\begin{aligned}
f_i(C_1, C_2) &= |C_1|s_1 + |C_2|s_2 \\
f_{i+1}(C_1, C_2) &= (|C_1| - 1)(s_1 - r_{min}) + \\
&\quad (|C_2| + 1)(s_2 + r_{min}) \\
&= |C_1|s_1 + |C_2|s_2 + 2r_{min} + (s_2 - s_1) + \\
&\quad r_{min}(|C_2| - |C_1|) \\
&= f_i(C_1, C_2) + 2r_{min} + (s_2 - s_1) + \\
&\quad r_{min}(|C_2| - |C_1|)
\end{aligned}$$

Therefore, we continue moving elements as long as $f_{i+1}(C_1, C_2) \leq f_i(C_1, C_2)$, or equivalently:

$$2r_{min} + (s_2 - s_1) + r_{min}(|C_2| - |C_1|) \leq 0$$

This inequality holds because, when $|C_1| > |C_2|$,

$$(s_2 - s_1) \leq -r_{min}$$

and

$$r_{min}(|C_2| - |C_1|) \leq -r_{min}$$

Thus, as long as $|C_1| > |C_2|$ we need to continue moving elements from C_1 to C_2 to further decrease the value of $f(C_1, C_2)$.

Now, we consider the case where $|C_1| \leq |C_2|$ and show that C_1 must contain the $|C_1|$ largest values to minimize $f(C_1, C_2)$. We assume that C_1 does contain the $|C_1|$ largest values and try to swap one element in C_1 with one element in C_2 to further decrease the value of $f(C_1, C_2)$. We let n_1 be an arbitrary node in C_1 before the swap and n_2 be an arbitrary element in C_2 before the swap. Therefore, $r_{n_1} \geq r_{n_2}$. Before the swap, we have:

$$f_{orig}(C_1, C_2) = |C_1|s_1 + |C_2|s_2$$

After the swap, we have:

$$\begin{aligned}
f_{swap}(C_1, C_2) &= |C_1|(s_1 - r_{n_1} + r_{n_2}) + \\
&\quad |C_2|(s_2 - r_{n_2} + r_{n_1}) \\
&= |C_1|s_1 + |C_2|s_2 + |C_1|(r_{n_2} - r_{n_1}) + \\
&\quad |C_2|(r_{n_1} - r_{n_2}) \\
&= f_{orig}(C_1, C_2) + \\
&\quad (|C_2| - |C_1|)(r_{n_1} - r_{n_2})
\end{aligned}$$

For $f_{orig}(C_1, C_2) \leq f_{swap}(C_1, C_2)$, we need:

$$(|C_2| - |C_1|)(r_{n_1} - r_{n_2}) \geq 0$$

This inequality is true because $r_{n_1} \geq r_{n_2}$ and $|C_2| \geq |C_1|$. Thus, the value of f will not decrease after the swap, indicating $f_{orig}(C_1, C_2)$ is minimal. $\square$

This result can easily be extended to a clique topology where there are multiple sender-receiver pairs. In this case, the channels should be partitioned by each receiver's cumulative full wake-up rate from all senders rather than from just one sender. However, nodes to be able to learn each other's cumulative full wake-up rate and choose a channel accordingly.

In the future, we hope to extend this result for the more general cases where there are k channels ($k > 2$) and the network is multi-hop. Also, we are developing more distributed versions of the protocols.

5. Multi-Level Power Save

In this section, we discuss an approach for cross layer enhancements for power save at the network and MAC layer. Section 3 and Section 4 presented protocols to address power save concerns at the MAC layer. However, by incorporating network layer in the power save process, we can improve performance. For example, consider the scenario where data packets are being sent from $\mathbf{A}$ to $\mathbf{C}$ via the route $\mathbf{A} \rightarrow \mathbf{B} \rightarrow \mathbf{C}$. Using network layer information, we can design power save protocols to consider the whole route. In pure MAC-based approaches, the power save protocol would run independently at links $\mathbf{A} \rightarrow \mathbf{B}$ and $\mathbf{B} \rightarrow \mathbf{C}$.

We propose viewing the problem as a multi-level design problem. The idea of using multi-level design to improve performance is prevalent in computer science (see [18] and references therein). For example, in computer architecture, accessing cache is much faster than main memory. However, main memory is cheaper in terms of cost per byte and is capable of storing much more data.

It is important to note that a number of different power save strategies could be employed at each of the levels. An example of a protocol which uses this technique is found in [24]. In this paper, there are two levels with nodes on the active route remaining in an "always on" state while all other nodes enter 802.11 PSM. The set of nodes on active routes is determined by which nodes have received route-reply (RREP) messages or forwarded data recently. When a node has done neither of these activities recently, it returns to the inactive state. However, the work in [24] does not generalize to the notion of multi-level wake-up, instead concentrating on one specific case of this idea. Our proposition is to investigate the range of protocols which can apply this multi-level technique.

5.1. Two-Level Power Save Examples

In this section, we give some examples of how a two-level technique could be applied using some current power save protocols. We use two-level examples as an illustration, but note that our proposal is for more general multi-level power save. We consider the following two levels:

Communicating Nodes These are nodes that are on "active routes." Nodes in this state choose a power save protocol that uses more energy than that of inactive nodes, but also requires a lower latency to wake up neighbors. Because data communication is a relatively frequent event for these nodes, their aim is to reduce latency for performance while sacrificing some energy conservation.

Inactive Nodes These are nodes that are not on an "active route." Inactive nodes sacrifice performance, in terms of latency, to conserve energy. When communication with an inactive node is desired, there is a relatively large latency before beginning the data transmission.

For each of the proposed protocols, we describe the protocol used by active nodes and the protocol used by inactive nodes. In addition, we provide a brief overview of some design issues with each of the protocols.

I. **Communicating Protocol:** *Always on*
Inactive Protocol: *Power Save*
This is the protocol presented in [24] where communicating nodes are always on and inactive nodes use 802.11 PSM. Other power save protocols could be used among inactive nodes, such as STEM [14].

II. **Communicating Protocol:** *STEM with a Large Duty Cycle*
Inactive Protocol: *STEM with a Small Duty Cycle*
In this scenario, the τ_2 parameter (i.e., sleep time between idle listening periods on the wake-up radio) discussed in Section 3 is much smaller for communicating nodes. Thus, they have a much smaller wake-up delay, but consume more energy on the wake-up radio. If wake-up rates for the two levels are known in advance, then the duty cycles could be chosen optimally to minimize the energy consumed by nodes. If the duty cycle is too large for a given wake-up rate, energy will be wasted because of frequent, unnecessary idle listening. However, if the duty cycle is too small, the wake-up procedure will be costly in terms of energy since the wake-up signal must be transmitted for a long time.

III. **Communicating Protocol:** *802.11 PSM with a Small Beacon Interval*
Inactive Protocol: *802.11 PSM with a Large Beacon Interval*
In this scenario, the beacon interval length is adjusted for the two levels. For communicating nodes, ATIM windows occur frequently. Thus, nodes can send their ATIM packets with a small latency, but consume more energy due to the frequent ATIM windows. For inactive nodes, the beacon interval is chosen to be an integer multiple of the smaller beacon interval length.

Thus, the inactive nodes will have one ATIM window for every M ATIM windows that communicating nodes have. The M parameter can be adjusted to achieve a balance between energy and latency.

IV. **Communicating Protocol:** *STEM with a Large Duty Cycle*
Inactive Protocol: *802.11 PSM with a Large Beacon Interval*
This is a variant combining Example II and Example III. One disadvantage of this approach when compared with Example II is time synchronization is now required for the 802.11 PSM part of the protocol. In the approach that only uses STEM, all wake-ups are asynchronous.

V. **Communicating Protocol:** *802.11 PSM with a Small Beacon Interval*
Inactive Protocol: *STEM with a Small Duty Cycle*
This approach combines Example II and Example III in the opposite way when compared to Example IV.

5.2. Future Research Directions

Introducing multi-level schemes presents many opportunities for future research. First, there is the issue of keeping track of which state a neighbor is currently in. When a node wants to communicate, it needs to know which protocol and what parameters to use (e.g., beacon interval length, duty cycle) to wake up the desired neighbor. Next, we need to provide a precise definition of an "active route."

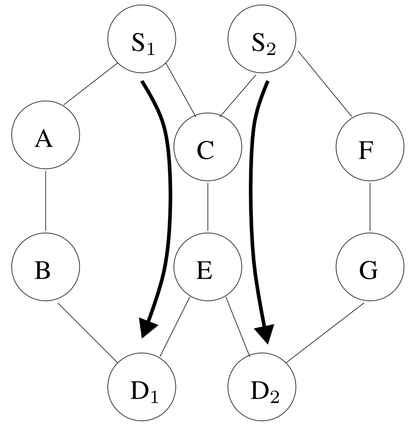
Another area of interest with multi-level power save is attempting to minimize the number of high energy nodes at the network layer. Consider the routing strategies shown in Figure 7. Most routing strategies that attempt to maximize throughput prefer the routing scheme in Figure 7(b) to take advantage of spatial reuse. However, with multi-level power save, the routing strategy shown in Figure 7(a) may be preferable since only six nodes are on active routes as opposed to eight nodes in Figure 7(b). In the future, we plan to explore routing strategies which work well with the multi-level power save design.

6. Conclusions

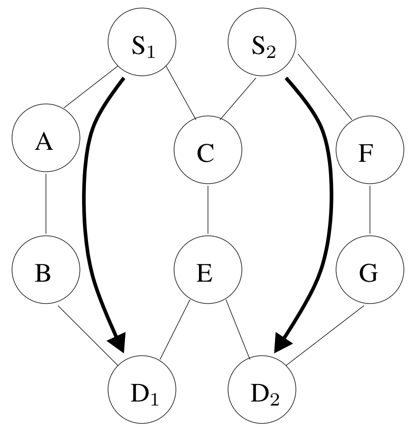
In this paper, we have addressed power save issues for sensor networks at both the MAC and network layer. At the MAC layer, we have presented our previous work [8–10] which uses a wake-up radio for on-demand wake-ups in addition to scheduling triggered wake-ups based on past traffic patterns. Because our scheme uses busy tones for on-demand wake-ups, all nodes in a sender's one-hop neighborhood must wake up.

Next, we extended the protocol to make on-demand wake-ups less costly by using multiple, non-interfering wake-up channels. Initial results indicate multiple wake-up channels can effectively reduce energy costs by using a simple, random assignment of wake-up channels. We also present an algorithm which is optimal in minimizing the number of wake-ups in single-hop networks when there are two channels available and the wake-up rates are known in advance.

Finally, we presented the idea of generalizing power save protocols to multiple levels based on how active a node is in communicating. The basic idea is the more active a node is in communicating, the more energy it will consume to achieve a lower latency. We described some scenarios where current power save protocols could be used with two levels as well as some future research directions. In particular, it opens many questions about interactions between the MAC and network layer, such as which routing strategy to use to minimize the number of nodes in high energy states.



(a) Favoring shared paths.



(b) Favoring separate paths.

Figure 7. Routing with two-level power save.

Acknowledgments

We would like to thank Pradeep Kyasanur for helping with the proof for optimal wakeup channel assignment in Section 4.1.

References

[1] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *ACM MobiCom 2001*, July 2001.

[2] C. F. Chiasserini and R. R. Rao. Combining Paging with Dynamic Power Management. In *IEEE Infocom 2001*, April 2001.

[3] C. Guo, L. C. Zhong, and J. M. Rabaey. Low Power Distributed MAC for Ad Hoc Sensor Radio Networks. In *IEEE GlobeCom 2001*, November 2001.

[4] C. Hu and J. Hou. LISP: A Link-Indexed Statistical Traffic Prediction Approach to Improving IEEE 802.11 PSM. In *IEEE International Conference on Distributed Systems (ICDCS) 2004*, March 2004.

[5] IEEE 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[6] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *ACM Wireless Networks*, July 2001.

[7] MICA2 Mote Datasheet. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-01_A_MICA2.pdf`.

[8] M. J. Miller. Minimizing Energy Consumption in Sensor Networks Using a Wakeup Radio. Master's thesis, University of Illinois at Urbana-Champaign, November 2003.

[9] M. J. Miller and N. H. Vaidya. A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio. *IEEE Transactions on Mobile Computing*, 2004. accepted for publication.

[10] M. J. Miller and N. H. Vaidya. Minimizing Energy Consumption in Sensor Networks Using a Wakeup Radio. In *IEEE WCNC 2004*, March 2004.

[11] J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, and S. Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, July 2000.

[12] C. S. Raghavendra and S. Singh. PAMAS – Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks. *ACM Computer Communications Review*, July 1998.

[13] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.

[14] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Topology Management for Sensor Networks: Exploiting Latency and Density. In *ACM MobiHoc 2002*, June 2002.

[15] E. Shih, P. Bahl, and M. J. Sinclair. Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *ACM MobiCom 2002*, September 2002.

[16] S. Singh, M. Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *ACM MobiCom 1998*, October 1998.

[17] `ns-2` – The Network Simulator. `http://www.isi.edu/nsnam/ns`.

[18] N. H. Vaidya. A Case for Two-Level Distributed Recovery Schemes. In *ACM SIGMETRICS 1995*, May 1995.

[19] T. van Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *ACM SenSys 2003*, November 2003.

[20] A. Woo and D. E. Culler. A Transmission Control Scheme for Media Access in Sensor Networks. In *ACM MobiCom 2001*, July 2001.

[21] Y. Xu, J. Heidemann, and D. Estrin. Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks. Technical Report 527, USC/Information Sciences Institute, 2000.

[22] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed Energy Conservation for Ad Hoc Routing. In *ACM MobiCom 2001*, July 2001.

[23] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *IEEE Infocom 2002*, June 2002.

[24] R. Zheng and R. Kravets. On-demand Power Management for Ad Hoc Networks. In *IEEE Infocom 2003*, April 2003.