Using Carrier Sensing to Improve the Energy Consumption of Sensor Network Wake-Up Protocols

Technical Report

August 2004

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

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

***Abstract*— In sensor networks, it is important for nodes to sleep a large portion of the time to conserve energy. However, nodes cannot communicate when sleeping and, therefore, need a mechanism to wake neighbors up when there is data to send. To address this problem, wake-up protocols are designed to determine when nodes should wake up. In this paper, we outline three categories to classify wake-up protocols and give examples of each. Based on this discussion, we choose to focus on protocols that use out-of-band mechanisms to wake other nodes up. After describing the trade-offs associated with this type of wake-up protocol, we present new analysis for two protocols from previous work, STEM [1], [2] and STEM-BT [2]. Based on this analysis, we identify areas where both of the protocols could be more energy efficient and present two new protocols, STEM-H and STEM-BT2, to address this. Extensive analysis and simulation testing shows that the new protocols outperform the old protocols in virtually every scenario tested.**

I. Introduction

As sensor network applications emerge, the need for aggressive power save protocols is evident. Because these small sensors are intended to operate unattended over long periods of time (possibly in areas that are difficult to access), the energy source is limited and it may be difficult or impossible to replenish frequently. However, because nodes cannot communicate when they are sleeping, wake-up protocols are needed to coordinate when nodes should (1) send data and (2) idly listen for data from neighbors. Thus, the goal of a wake-up protocol is to save as much energy as possible while transmitting the desired data across the network and maintaining a latency and throughput that is "good enough" for the given task.

Radio hardware for sensor networks typically provides four different states: *transmitting*, *receiving*, *idly listening*, and *sleeping*. Typically, the power consumption for the first three states is comparable whereas the sleeping state uses significantly less power. For example, the Mica2 Mote [3] sensor radio has the following power levels: 81 mW to transmit, 30 mW to receive or idly listen, 0.003 mW to sleep. Thus, the sleeping state consumes less power than the idle listening state by *four orders of magnitude* when this hardware is used.

This work was funded in part by NSF grant ANI-0125859 and a NDSEG fellowship.

There are three basic approaches taken by wake-up protocols. These can be categorized as *synchronous*, *asynchronous*, and *out-of-band*. We will now provide a brief description and examples of each of these wake-up methods.

Synchronous Nodes schedule a time in the future to wake up. The scheduled time can be absolute (e.g., clocks are synchronized and nodes will wake up at a specified time) or relative to some epoch (e.g., a node will wake up T seconds after the last packet reception). One example [4] is IEEE 802.11's Power Save Mode (PSM) where all nodes wake up and remain on for a fixed time at the beginning of each beacon interval. Another example [5], [6] is where two communicating nodes wake up T seconds after the last packet reception and T is adjusted dynamically based on past traffic patterns.

Asynchronous Nodes wake up independently according to their local schedule and attempt to discover other nodes that are currently awake. Thus, when the wake-ups of two nodes overlap, they can communicate. An example [7], [8] is when deterministic schedules are chosen to guarantee overlap within a bounded latency. Another example [9], [10] is when nodes wake up non-deterministically such that overlap is guaranteed in a bounded time with high probability.

Out-of-Band Nodes keep their data radio in the sleep state until an out-of-band channel alerts them to wake up. An example [11]–[13] is where a low-power radio idly listens on a separate, wake-up channel. Another example [1], [2] is a when wake-up radio (with power characteristics similar to the data radio) periodically idly listens to the channel. In both examples, when a wake-up signal is detected, the data radio will turn on.

These techniques are orthogonal. For example, a node could use an asynchronous protocol to discover neighbors and, after discovery, use a synchronous protocol to schedule subsequent wake-ups. Similarly, an out-of-band protocol could be used to wake up a neighbor to send the first data packet and synchronous wake ups could be scheduled for later packets (see [5], [6] for an example of this combination of techniques).

In this paper, we focus on out-of-band protocols. One advantage of this technique is, unlike synchronous protocols, no clock synchronization is needed. Unlike asynchronous protocols, nodes do not have to probe the channel whenever they wake up (i.e., less channel contention and control overhead). Also, the wake-up latency is bounded for out-of-band protocols, which is not true for asynchronous protocols that use non-deterministic schedules.

However, there are trade-offs in using out-of-band protocols. One disadvantage is the increased hardware complexity and cost to provide an extra wake-up channel. Also, the wake-up channel requires extra bandwidth to avoid interference with the data channel. Finally, the wake-up channel must be designed such that its monitoring does not consume much energy. Obviously, the wake-up channel is of little use, from an energy perspective, if it consumes a large amount of energy idly listening to the channel while the data radio is saving energy by sleeping.

STEM [1], [2] (Sparse Topology and Energy Management) and STEM-BT [2] (STEM with a Busy Tone) are simple out-of-band protocols which use periodic idle listening on the wake-up channel. In this paper, we identify a way to make each of these protocols more efficient. Based on this, we propose two new protocols, STEM-H (STEM Hybrid) and STEM-BT2, to address the shortcomings in STEM and STEM-BT, respectively. We develop analytical equations to characterize the energy consumption of all four protocols. We then provide extensive simulations results to show the performance of the four protocols. We note that our analysis is different from that in [1], [2] which compares STEM and STEM-BT's energy consumption relative to the no power save approach and uses a different traffic model. Also, the results in [1], [2] only consider the energy of nodes on the data path of a flow, rather than all nodes in the vicinity of the sender.

In Section II, we review related work. Section III describes all four protocols and presents analytical results. Section IV presents simulation results. Finally, we conclude and discuss future work in Section V.

II. Related Work

In this section, we present an overview of research in wake-up protocols for ad-hoc networks. We present the related work according the category of wake-up technique that best encompasses the protocol. In addition, we also present other power save protocols for completeness.

A. Synchronous Protocols

The IEEE 802.11 specification [4] is the standard currently used by commercial WLAN cards. It specifies a Power Save Mode (PSM) which we will now describe. Nodes awake at the beginning of each *beacon interval* and remain 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 that have been queued since the previous beacon interval are advertised by ATIM packets. The wake-up procedure takes place via a ATIM and ATIM-ACK handshake. All nodes that complete such a handshake during the ATIM window will remain on for the entire beacon interval to communicate. The rest of the nodes return to sleep until the next beacon interval.

S-MAC [14] is a protocol similar to 802.11 PSM which is designed specifically for sensor networks. In this protocol, nodes synchronize with their neighbors. By keeping track of neighbors' schedules, nodes send data when the specified receiver is scheduled to be awake.

In [5], [6], synchronous wake-ups are added on top of an out-of-band protocol. In particular, after a packet is sent to a receiver, the sender uses past traffic history to piggyback a future wake-up time, T. In response, the sender and receiver will wake up T seconds after the last packet reception in anticipation of sending packets that have arrived since the last wake-up occurred.

Another type of synchronous wake-up protocol is those that use TDMA. Nodes attempt to schedule wake-ups such that data is sent only when it will not interfere with other transmissions. TRAMA [15] is a TDMA protocol where nodes periodically advertise traffic for an upcoming interval and potential interfering nodes use hash functions to determine which slots they should use to avoid interference. TDMA techniques are also used in the context of "flows" of data packets [16]–[18] where periodic flows attempt to find slots to transmit data at regular intervals without interfering with existing flows. Thus, in all these protocols, the wake-up procedure is for the sender/receiver pair to awake at a slot in the future when they will have exclusive access to the medium.

B. Asynchronous Protocols

In [8] and [19], nodes choose their awake times such that they are guaranteed to eventually overlap with each neighbor's awake time within a bounded time period. Multiple methods of achieving this are discussed. A similar approach is used in [7], which uses combinatorics theory to find schedules in which nodes wake up k out of every v slots. Using these schedules, every v slots, each node will have at least one overlapping slot with each of it neighbors. One key result of this paper is that the schedules still guarantee overlap even if the slot epochs of the nodes are not synchronized, which is commonly the case in ad-hoc networks.

In [10], a non-deterministic approach is used for neighbor discovery. Basically, each node chooses to probe, listen, or sleep during each slot according to a specified probability distribution. In a given slot, whenever one node chooses to probe and one of its neighbors chooses to listen, then a link is discovered. The paper then investigates what percentage of the links will be discovered within a bounded time period using this protocol. Though the protocol is presented as a neighbor discovery technique, it can easily be adapted to a wake-up protocol where nodes send packets in slots they discover the intended receiver to be awake.

The protocol from [9] uses continuum percolation theory as its basis. When a node has a packet to send, it continually broadcasts the packet until, with high probability, all of its neighbors have been awake at some point to receive the packet. Nodes without data choose the sleep time between awake periods according to an exponential distribution. The goal is then to determine how many hops a data packet will travel within a given amount of time. One disadvantage of the protocol is the large number of collisions that could be caused by such a scheme, which the authors fail to consider.

C. Out-Of-Band Protocols

The PicoRadio [11], [12] design uses a low-power wake-up channel. One of the main aspects of the project is hardware design. In particular, PicoRadio is trying to create a wake-up radio that can listen continuously and send wake-up signals while consuming very little power. This concept is also explored via electronic hardware simulation in [20].

In [13], a proof-of-concept implementation for low-powered wake-up channels is presented. The paper uses off-the-shelf hardware to create a paging interface for systems with centralized access points or proxies. If the proxy indicates their is data pending for the device, it will turn on its 802.11 WLAN card to engage in communication.

A theoretical approach is investigated in [21]. Here, nodes cycle through multiple sleep states. Each state uses less power, but requires more energy to transition back to the idle state. The access point tracks the current sleep state of each node. When it wants to wake up a certain node that is in sleep state s, it uses an RF wake-up channel to signal all nodes in sleep state s to wake up. The access point begins sending data to the intended node and all other nodes that woke up can return to sleep when they see the data is not for them. The protocol is designed such that it minimizes energy consumption while meeting QoS requirements.

STEM and STEM-BT [1], [2] are also out-of-band wake-up protocols. A second, wake-up radio periodically idly listens for wake-up signals according to a specified duty cycle. This a different approach from PicoRadio and similar protocols which idly listen constantly on their wake-up channel. STEM and STEM-BT will be described in detail in Section III.

D. Other Power Save Protocols

The PAMAS protocol [22] uses 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, it does not address the energy consumed due to idle listening.

Another popular approach in power save protocols is to keep a small subset of nodes in the idle listening state while most nodes enter the sleep state. The set of nodes in the idle listening state is chosen based on topology either proactively or on-demand.

In [23], energy is saved using the on-demand approach. Assuming an on-demand routing algorithm, nodes set a soft-timer to remain in the idle listening state when the detect that they are on an active data route (e.g., the receive a route reply or forward data). All nodes not on active routes use 802.11's PSM to save energy.

The proactive approaches attempt to approximate a minimum dominating set by having all nodes within one hop of an idly listening node. All other nodes run a power save protocol (e.g., 802.11 PSM) and when they have data to send, they turn on and send data to an idle listening node within one hop. In AFECA [24], nodes decide whether to idly listen based on the size of their neighborhood. With GAF [25], the decision is made based on geographic location. SPAN [26] uses information about a node's neighborhood and its remaining energy level to choose which nodes should idly listen.

III. Protocol Description and Analysis

In STEM [1], [2], a two-radio architecture achieves energy savings by letting the data radio sleep until communication is necessary while the wake-up radio periodically listens according to a 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. STEM-BT [2] is a variant of STEM that uses a busy tone, instead of encoded data, for the wake-up signal. Both protocols are orthogonal to the MAC layer transmission scheduling scheme.

In this section, we will describe and analyze the operation of STEM and STEM-BT. Based on this discussion, we make some observations about how the protocols could achieve better energy efficiency. Based on this, we present and analyze two new protocols, STEM-H and STEM-BT2, which reduce the energy consumption for STEM and STEM-BT, respectively. We note that our analysis for STEM and STEM-BT differs from what was done in [1], [2]. Whereas that work analyzed the energy relative to the "always on" (i.e., always idly listening) approach, our analysis computes the expected energy per bit using constant bitrate (CBR) traffic[1]. Our analysis would be useful if, for example, a system designer wanted to know how much energy consumption to expect for a given traffic rate. The notation for our analysis appears in Table I for easier reference. In our analysis, we assume that there are N nodes, all within range of each other. There is one sender (TX) and one receiver (RX) and the sender transmits CBR data at rate R. The other $N-2$ nodes are referred to as neighbors (nbr).

For each of the protocols, there are two sub-protocols to consider. First is the *transmitting* sub-protocol, which is performed when a node has data to send and attempts to wake up the intended receiver. The other sub-protocol is the *monitoring* part. This is the steady-state for nodes where they periodically listen to the wake-up channel to determine if a

[1] In our simulations, we relax this assumption and use Poisson traffic.

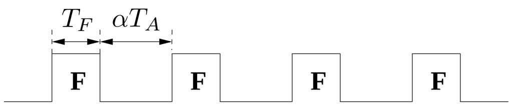


Fig. 1. Sending protocol for STEM and STEM-H
(**F** indicates a FILTER packet is being sent).

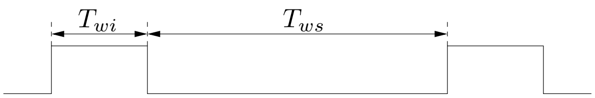


Fig. 2. Monitoring protocol for STEM.

signal is being sent and they need to wake up their data radio. In STEM and STEM-H, the wake-up radio must be capable of sending and receiving data packets. By contrast, in STEM-BT and STEM-BT2, the wake-up radio only needs to be capable of sending and detecting a busy tone (i.e., making a binary decision whether the channel is busy or not).

A. STEM

1) Sending Protocol: When a node has data to send it begins a continuous cycle of transmitting a *FILTER* packet followed by a idle listening period for the corresponding *FILTER-ACK* packet. This process is shown in Figure 1. The idle listening time for the FILTER-ACK has to be slightly longer than T_A due to factors such as propagation delay and hardware switching time. Thus, the length of the idle listening period is αT_A, where $\alpha > 1.0$.

When a sender receives the corresponding FILTER-ACK, it turns on its data radio and begins transmitting data packets according to the data radio's MAC protocol. Additionally, the sender will stop transmitting FILTERs at this point and return the wake-up radio to a monitoring state. If the sender transmits FILTERs for a sufficiently long time (discussed in Section III-A.3) and does not receive the corresponding FILTER-ACK, it assumes a collision occurred at the receiver and that the receiver turned its data radio on in response to the collision. Thus, the sender turns on its data radio and begins communication as if it had received the FILTER-ACK. When a node with its data radio on does not send or receive packets for T_{th} time, it will return the data radio to sleep.

No carrier sensing is performed during this process; FILTERs are sent regardless of whether the channel is idle. In Section III-A.3, we discuss exactly how long this process is repeated to guarantee overlap with the receiver's monitoring process.

2) Monitoring Protocol: In the monitoring state, nodes periodically wake up for long enough to receive a FILTER and respond with a FILTER-ACK if they are the intended receiver. After idly listening for a sufficiently long period of time, the node's wake-up radio returns to sleep for a specified length of time. This process is shown in Figure 2. The T_{ws} parameter is chosen by the user. A longer T_{ws} saves more energy in the monitoring state, but increases the latency of the wake-up process. In Section III-A.3, we discuss how T_{wi} is chosen.

When a node receives a FILTER and it is the intended receiver, it sends a FILTER-ACK and turns its data radio on to idly listen for packets on the data channel. On the wake-up channel, the node continues the monitoring process. If a node detects a collision on the wake-up channel, it conservatively assumes that one of the packets was a FILTER for which it was the recipient. In this case, the node will *not* respond with a FILTER-ACK, but will turn on its data radio in anticipation of being the recipient of data packets. If no data packets are sent to the node for a sufficiently long time after the collision on the wake-up channel (i.e., $T_{wt} + T_{th}$), the node returns its data radio to sleep. If a node turned its data radio on and does receive data packets, it will return the data radio to sleep when no data packets have been sent or received for T_{th} time.

3) Analysis: The idle listening period, T_{wi}, must be long enough that a node will successfully receive a FILTER packet even if the node wakes up in the middle of a FILTER transmission (and hence cannot correctly decode the first FILTER packet). In the worst case, the node wakes up just after a FILTER transmission has begun. Thus, the node has to wait for this first, undecodable FILTER packet to finish, which takes about T_F time. It then must wait for the sender to idly listen for a FILTER-ACK, which takes αT_A time. Finally, it must stay on long enough to receive the *next* FILTER packet, which take T_F time. Thus, $T_{wi} = T_F + \alpha T_A + T_F = 2T_F + \alpha T_A$.

Now, we discuss, T_{wt}, the time the sending protocol must be done to ensure enough overlap that every neighbor doing the monitoring protocol will receive a FILTER and have time to respond with the FILTER-ACK, if necessary. In the worst case, a monitoring node's idle listening period ends just before the sender's first FILTER transmission ends, which takes T_F time. The sender must continue the process for T_{ws} time since the monitoring node will be asleep for that duration. When the monitoring node begins idly listening again, it may wake up just after a FILTER transmission began (T_F time). After the sending node idly listens for αT_A time, it sends another FILTER packet (T_F time) which will successfully be decoded by the receiver. Finally, the sending node must wait long enough to receive the corresponding FILTER-ACK, if necessary, which takes αT_A time. Thus, summing up the terms mentioned in this paragraph, we get: $T_{wt} = T_F + T_{ws} + T_F + \alpha T_A + T_F + \alpha T_A = 3T_F + T_{ws} + 2\alpha T_A$.

On average, the sender does the wake-up process for $T_{wt}/2$ time before the receiver replies[2]. First, we note that if $1/R < T_P + T_{th}$, then the nodes will remain "always on" since the rate is so high that the medium is never idle for at least T_{th} time. Next, we want to find p_w, the number of packets that are sent per wake-up (recall that the traffic is CBR for our analysis, so this value is deterministic). We observe that p_w will be the largest integer value that satisfies the following

[2]We note that $T_{wt}/2$ is approximate since, in reality, there are discrete times when the process can end (i.e., $k \times (T_F + \alpha T_A)$, where k is a positive integer). Thus, the process cannot does not end at any point on the interval $(0, T_{wt}]$ with equal probability as our analysis assumes.

TABLE I

NOTATION AND DEFINITIONS.

Notation	Definition
N	Number of nodes in the one hop neighborhood of the sender (including the sender)
B	Bitrate (bps)
R	Sending rate (packets/second)
b_P	Bits/packet, including any MAC layer overhead (e.g., RTS, CTS, ACK, headers)
T_P	Time to transmit data packet and associated overhead, b_P/B
b_D	Actual bits of data/packet (i.e., the payload)
b_F	Bits/FILTER packet
T_F	Time to transmit FILTER packet, b_F/B
b_A	Bits/FILTER-ACK packet (STEM, STEM-H)
T_A	Time to transmit FILTER-ACK packet, b_A/B (STEM, STEM-H)
α	Multiplier for how long idle listening period for FILTER-ACK should be relative to T_A (STEM, STEM-H)
T_{th}	Idle timeout threshold to return to sleep after sending or receiving a packet on the data channel
P_S	Sleep power
P_I	Idle power (assumes same power for RX and idle listening)
P_{TX}	Transmit power
F_I	Fraction of the wake-up signal transmission spent idly listening for ACKs (STEM, STEM-H)
F_{TX}	Fraction of the wake-up signal transmission spent sending FILTER packets (STEM, STEM-H)
T_I	Time spend idly listening to the data channel from when the wake-up signal is detected to when the FILTER packet is received (STEM-BT2)
w_r	Expected number of wake-ups per second
p_w	Packets sent per wake-up
T_{ws}	Sleep time in between idle listening periods on the wake-up channel
T_{wi}	Duration of idle listening periods on the wake-up channel
T_{wt}	Duration of the sender must transmit the busy tone (STEM-BT) or FILTER packets (STEM, STEM-H) on the wake-up channel to ensure it is heard by all neighbors
w_i	Wake-Ups per idle listening period (STEM-H)
T_{ws2}	Duration of sleep period between wake-ups during the idle listening period (STEM-H, STEM-BT2)
W_L	Lower limit on the amount of time to wake up a monitoring node (STEM-H)
W_U	Upper limit on the amount of time to wake up a monitoring node (STEM-H)
k	Number of discrete times a monitoring node can successfully receive and decode a FILTER packet (STEM-H)
pct_{nbr}	Percent which detect a FILTER packet transmission (STEM-H)
E_{w_TX}	Energy/second consumed by the wake-up radio of the data sender
E_{w_RX}	Energy/second consumed by the wake-up radio of the intended receiver
E_{w_nbr}	Energy/second consumed by the wake-up radio of nodes which are neither the intended receiver or data sender
E_{d_TX}	Energy/second consumed by the data radio of the data sender
E_{d_RX}	Energy/second consumed by the data radio of the intended receiver
E_{d_nbr}	Energy/second consumed by the data radio of nodes which are neither the intended receiver or data sender

equation:

$$\frac{p_w - 1}{R} < (p_w - 1)T_P + T_{th} + \frac{T_{wt}}{2} \tag{1}$$

where the left-hand side of the equation is the time is takes to generate $(p_w - 1)$ packets and the right-hand side of the equation is the time it takes to do the wake-up process ($T_{wt}/2$), send $(p_w - 1)$ packets, and idly listen for T_{th} time. Based on Equation 1, we compute p_w to be:

$$p_w = \left\lceil \frac{T_{th} + T_{wt}/2}{1/R - T_P} \right\rceil \tag{2}$$

The number of wake-ups per second, w_r, is computed as:

$$w_r = \begin{cases} \frac{R}{p_w} & \text{if } \frac{1}{R} \geq T_P + T_{th} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

On the wake-up radio, the energy consumed is the same for both the receiver and neighbor nodes:

$$E_{w\_RX} = \frac{T_{wi}P_I + T_{ws}P_S}{T_{wi} + T_{ws}} \tag{4}$$

$$E_{w\_nbr} = E_{w\_RX} \tag{5}$$

For the sender, we analyze the energy consumed per wake-up and multiply it by w_r. Every wake-up, the sender will consume P_I power for a duration of $\frac{T_{wt}}{2}$. In the remaining, $(\frac{1}{w_r} - \frac{T_{wt}}{2})$ time between wake-ups, $(T_{wi}P_I + T_{ws}P_S)$ will be consumed each period of the duty cycle. The number of periods that will occur in this remaining time between wake-ups is:

$$\frac{1}{T_{ws} + T_{wi}} \left(\frac{1}{w_r} - \frac{T_{wt}}{2} \right) = \frac{1}{T_{ws} + T_{wi}} \left(\frac{1 - w_r \frac{T_{wt}}{2}}{w_r} \right) \tag{6}$$

When the sender begins the wake-up process, it will transmit a FILTER packet and then idly listen for the FILTER-ACK.

Therefore, during the wake-up process, the sender will use P_{TX} for $F_{TX} = \frac{T_F}{T_F + \alpha T_A}$ fraction of the time and idly listen to the channel for $F_I = 1 - F_{TX}$ fraction of the time. Thus, we have:

$$\begin{aligned} E_{w\_TX} &= w_r \left(\frac{T_{wt}}{2} (F_{TX} P_{TX} + F_I P_I) \right) \\ &\quad + w_r \left(\frac{(1 - w_r \frac{T_{wt}}{2})(T_{wi} P_I + T_{ws} P_S)}{w_r (T_{ws} + T_{wi})} \right) \\ &= w_r \frac{T_{wt}}{2} (F_{TX} P_{TX} + F_I P_I) \\ &\quad + \frac{(1 - w_r \frac{T_{wt}}{2})(T_{wi} P_I + T_{ws} P_S)}{T_{ws} + T_{wi}} \end{aligned} \tag{7}$$

Combining Equation 4 and Equation 7, we get:

$$E_{STEM\_w} = E_{w\_TX} + E_{w\_RX} + (N-2) E_{w\_nbr} \tag{8}$$

For the data radio, we calculate the energy required per wake-up and multiply by w_r to get E_{STEM_d}, the energy per second consumed by the data radio.

With STEM, the nodes which are neither the sender nor the intended receiver always keep their data radio in sleep state. Thus,

$$\begin{aligned} E_{d\_nbr} &= w_r \left(P_S \frac{1}{w_r} \right) \\ &= P_S \end{aligned} \tag{9}$$

For the sender and intended receiver, we have:

$$\begin{aligned} E_{d\_TX} &= w_r (P_{TX} p_w T_P + P_I T_{th}) \\ &\quad + w_r \left(P_S \left(\frac{1}{w_r} - (p_w T_P + T_{th}) \right) \right) \end{aligned} \tag{10}$$

$$\begin{aligned} E_{d\_RX} &= w_r \left(P_I (p_w T_P + T_{th}) \right) \\ &\quad + w_r \left(P_S \left(\frac{1}{w_r} - (p_w T_P + T_{th}) \right) \right) \end{aligned} \tag{11}$$

Thus, we have:

$$E_{STEM\_d} = E_{d\_TX} + E_{d\_RX} + (N-2) E_{d\_nbr} \tag{12}$$

and the overall energy consumption per data bit for STEM is:

$$E_{STEM} = \frac{E_{STEM\_w} + E_{STEM\_d}}{b_D R} \tag{13}$$

4) Discussion: Based on the description and analysis of STEM, we make two observations. First, the wake-up process is relatively inexpensive (in terms of energy) for all nodes other than the sender when compared to the steady-state monitoring process. In particular, the neighbor nodes use almost the same amount of energy whether they are monitoring the channel or receiving the FILTER packet[3]. The receiving node uses slightly more energy because it has to respond with a FILTER-ACK packet. While the sender uses more energy due to its periodic FILTER transmissions, it only transmits for $T_{wt}/2$ time on average.

[3]Most radios use slightly more energy to receive a packet compared to idly listening, but the difference for many radios is negligible.

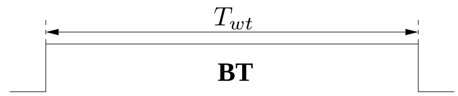


Fig. 3. Sending protocol for STEM-BT and STEM-BT2 (**BT** indicates a busy tone is being transmitted).

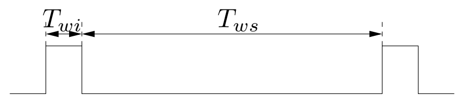


Fig. 4. Monitoring protocol for STEM-BT and STEM-BT2.

The second observation is that the steady-state monitoring process is relatively expensive when compared to STEM-BT, described in Section III-B. Since T_{wi} is proportional to T_F and T_A, it can be rather large for sensor networks which tend to have a relatively low bitrate (e.g., 19.2 kbps for Mica2 Motes [3]). Thus, all nodes in the network must spend $\frac{T_{wi}}{T_{ws}+T_{wi}}$ percentage of the time idly listening even if there is no traffic in the network. As we will see with STEM-BT, it is preferable to have the wake-up radio's idle listening period to be a small constant independent of T_F and T_A.

B. STEM-BT

1) Sending Protocol: When a node has data to send in STEM-BT, it begins transmitting a busy tone on the wake-up channel. No carrier sensing is done prior to beginning the busy tone transmission. The busy tone is sent for T_{wt} time, where T_{wt} is chosen such that it is long enough to guarantee overlap with every neighbor's idle listening period on the wake-up channel (the exact value is derived in Section III-B.3). STEM-BT's sending protocol is shown in Figure 3.

After the sender has transmitted a busy tone for T_{wt} time, it turns on its data radio. Once the data radio is on, a FILTER packet is sent on the data channel indicating which receiver will receive more data. The sender then begins its transmitting the data for the receiver on the data channel. As in STEM, when a node with its data radio on does not send or receive packets for T_{th} time, it will return the data radio to sleep.

2) Monitoring Protocol: For nodes that are monitoring the wake-up channel, the procedure is similar to that of STEM. This is shown in Figure 4. A major difference between the monitoring protocol for STEM and STEM-BT is then length of T_{wi}, the idle listening time. In STEM-BT, T_{wi} is much shorter because a monitoring node only has to detect a busy tone rather than decode a packet and wait for FILTER-ACKs to be sent.

When a node does detect a busy tone, it turns on its data radio and idly listens for a FILTER packet on the data channel. When the FILTER packet is received, the node remains on if it is the intended receiver. Otherwise, its data radio returns to sleep. If a node keeps its data radio on to receive data packets, it will return the data radio to sleep when no packets have been sent or received for T_{th} time. One key point to note about STEM-BT's monitoring protocol is that *all* nodes in the one-hop neighborhood of the sender must turn their data radio on and idly listen until the FILTER packet is received.

3) Analysis: In STEM-BT, T_{wi} is a fixed value based on how long the radio must listen to detect a busy tone with a specified level of confidence (see [27] for a discussion on this). The busy tone transmission time, T_{wt}, must be sufficiently long to ensure enough overlap such that every neighbor doing the monitoring protocol receives the busy tone. In the worst case, a monitoring node's idle listening period begins just before the sender starts transmitting the busy tone. In this situation, the busy tone is not detected at the specified level of confidence. Thus, the sending node must transmit the busy tone long enough that the monitoring node's *next* idle listening period will completely overlap with the busy tone. Thus, $T_{wt} = T_{wi} + T_{ws} + T_{wi} = 2T_{wi} + T_{ws}$. On average, each neighboring node wakes up halfway through the busy tone transmission and, therefore, idly listens with its data radio for $T_{wt}/2$ time before receiving the FILTER packet. For STEM-BT, the value of p_w is similar to that of Equation 2 except that now the wake-up process takes T_{wt} instead of $T_{wt}/2$ (on average) and packets may arrive while the FILTER is being sent on the data channel. Thus,

$$p_w = \left\lceil \frac{T_{th} + T_{wt} + T_F}{1/R - T_P} \right\rceil \tag{14}$$

The number of wake-ups per second, w_r, is the same as given in Equation 3 except that the p_w variable refers to that of Equation 14, rather than Equation 2.

The derivation for E_{STEMBT_w} is nearly identical to the derivation for E_{STEM_w}, except that the sender's busy tone transmission will always last T_{wt} time unlike STEM. STEM's wake-up process only last $T_{wt}/2$ on average. Therefore, we have:

$$E_{w\_RX} = \frac{T_{wi}P_I + T_{ws}P_S}{T_{wi} + T_{ws}} \tag{15}$$

$$E_{w\_nbr} = E_{w\_RX} \tag{16}$$

$$E_{w\_TX} = w_r T_{wt} P_{TX} + \frac{(1 - w_r T_{wt})(T_{wi}P_I + T_{ws}P_S)}{T_{ws} + T_{wi}} \tag{17}$$

$$E_{STEMBT\_w} = E_{w\_TX} + E_{w\_RX} + (N-2)E_{w\_nbr} \tag{18}$$

For the data radio, we will calculate the energy required per wake-up and multiply by w_r to get E_{STEMBT_d}, the energy per second consumed by the data radios. In STEM-BT, the sender will only have to turn its radio on long enough to transmit the FILTER packet (on the data channel) and data packets. Therefore,

$$E_{d\_TX} = w_r \left(P_{TX}(T_F + p_w T_P)\right) + w_r P_I T_{th} + w_r P_S \left(\frac{1}{w_r} - (T_F + p_w T_P + T_{th})\right) \tag{19}$$

In addition to keeping its data radio on as long as the sender, the receiver has its data radio on after it has received the busy tone but before the FILTER packet arrives. On average, this will take $T_{wt}/2$. Thus,

$$E_{d\_RX} = w_r P_I \left(\frac{T_{wt}}{2} + T_F + p_w T_P + T_{th}\right) + w_r P_S \left(\frac{1}{w_r} - \left(\frac{T_{wt}}{2} + T_F + p_w T_P + T_{th}\right)\right) \tag{20}$$

All other nodes will have to turn their data radios on as long as the receiver, except that they can return to sleep after receiving the FILTER packet. This gives us:

$$E_{d\_nbr} = w_r P_I \left(\frac{T_{wt}}{2} + T_F\right) + w_r P_S \left(\frac{1}{w_r} - \left(\frac{T_{wt}}{2} + T_F\right)\right) \tag{21}$$

Thus, we have:

$$E_{STEMBT\_d} = E_{d\_TX} + E_{d\_RX} + (N-2)E_{d\_nbr} \tag{22}$$

and the overall energy consumption per data bit for STEM is:

$$E_{STEMBT} = \frac{E_{STEMBT\_w} + E_{STEMBT\_d}}{b_D R} \tag{23}$$

4) Discussion: From the description of STEM-BT, we can compare it to STEM. First, the monitoring process in STEM-BT uses much less energy than STEM. This is because T_{wi} is significantly smaller for STEM-BT since it only has to be long enough to detect whether there is a busy tone being emitted. Thus, unlike STEM, T_{wi} is independent of the time it takes to send a FILTER or FILTER-ACK packet. For example, as we will see in Section IV, T_{wi} is about 80 times larger for STEM than for STEM-BT for our experimental parameters. Thus, when the traffic load is low in a network, STEM-BT is more energy efficient than STEM because of its low monitoring costs.

The next comparison is the wake-up procedure for STEM and STEM-BT. For STEM-BT, this is relatively expensive compared to STEM. Recall that in STEM, only the sender used significantly more energy to do a wake-up compared to monitoring. However, in STEM-BT, every neighboring node that detects the busy tone must turn its data radio on to listen for the FILTER packet on the data channel. Thus, on average, each neighboring node idly listens to the data channel for half of the time that the busy tone is emitted. Based on this, we can conclude that STEM-BT's performance degrades when (1) there are a large number of neighboring nodes in the vicinity of the sender[4] and (2) wake-ups become more frequent (e.g., due to a higher traffic load). This contrasts with STEM's wake-up procedure which is relatively inexpensive and does not greatly increase energy consumption as the size of the sender's neighborhood increases.

Thus, we expect STEM-BT to perform best in environments where the traffic load is relatively low and a significant portion of the of the time is spent monitoring the wake-up channel.

[4]Many applications assume that sensor networks will be rather dense for reasons such as increased reliability, connectivity, and adequate sensing coverage.

Also, STEM-BT will do better in less dense networks. STEM, however, should do better, relative to STEM-BT, as the traffic load increases and more time is spent doing wake-ups rather than monitoring the wake-up channel.

C. STEM-H

Based on the description of STEM in Section III-A and Section III-B.4, we see that STEM's energy consumption can be significantly improved by reducing the monitoring costs while retaining its relatively low wake-up cost. Thus, we propose STEM-H (STEM-Hybrid) which combines aspects of STEM and STEM-BT to create a protocol more energy efficient than STEM.

The basic idea of STEM-H is to only idly listen long enough to detect whether or not a packet is being transmitted on the wake-up channel during the monitoring phase. This detection time is relatively small; it is similar to that of STEM-BT. When the wake-up channel is detected as busy, then the monitoring node will turn its wake-up radio on long enough to receive and decode a FILTER packet.

1) Sending Protocol: The sending protocol is identical to that of STEM, described in Section III-A.1 and shown in Figure 1. The only difference is the length of T_{wt}, the amount of time FILTER packets are transmitted. The value of T_{wt} for STEM-H is derived in Section III-C.3.

2) Monitoring Protocol: For STEM-H's monitoring state, nodes only wake up long enough to detect whether the wake-up channel is busy or idle. This contrasts with STEM's monitoring protocol, where nodes wake up long enough to receiver FILTER packets and send FILTER-ACKs. In STEM-H, whenever the wake-up channel is detected as busy, a node stays on long enough to receive the next FILTER and send a FILTER-ACK, if necessary. Like STEM, nodes reply with a FILTER-ACK if the FILTER is for them. Otherwise, the node returns to its regular monitoring state. Once, the FILTER/FILTER-ACK handshake takes place, nodes follow the same procedure for turning on their data radios as described in Section III-C.2. They also follow the same protocol for returning their data radios to sleep.

It is important to note that in STEM-H, T_{wi}, the idle listening time on the wake-up channel, is comparable to that of STEM-BT since only a binary decision on the channel status is necessary. STEM, on the other hand, requires a much longer T_{wi} because it must completely decode packets during its idle listening period on the wake-up channel.

There is another consideration in STEM-H's monitoring process. Because the wake-up idle listening period is much smaller than the time a sender waits for a FILTER-ACK to be sent (i.e., $T_{wi} \ll \alpha T_A$), it is possible that the idle listening period begins and ends during the time a sender is idly listening for the FILTER-ACK. In this case, the monitoring node never detects one of the sender's FILTER packets. To address this, we require multiple idle listening periods between the potentially long T_{ws} periods to guarantee at least one idle listening period overlaps with a FILTER transmission. This is shown in Figure 5. The value for T_{ws2}, as well as the number of idle listening periods between T_{ws} periods (w_i), is derived in Section III-C.3.

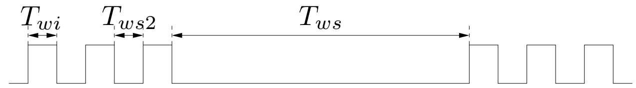


Fig. 5. Monitoring protocol for STEM-H.

3) Analysis: We begin by determining T_{ws2} and w_i, the sleeping time between idle listening periods and the number of idle listening period required to guarantee a FILTER is detected, respectively. To determine the frequency and number of times a nodes must wake-up during the monitoring periods, we observe the following constraint. If a node begins its idle listening period after the sender has started its FILTER packet transmissions, T_{ws2} and w_i must be chosen to guarantee at least one of the T_{wi} duration wake-ups will completely overlap with one of the sender's FILTER transmissions. Similar to STEM-BT, we assume that T_{wi} is the minimum amount of time required to classify the wake-up channel as busy (with sufficiently low error probability). Thus, if a FILTER packet only partially overlaps with a T_{wi} period, the channel may not be detected as busy.

In the worst case, a wake-up idle listening period begins just before a FILTER transmission begins. For example, the listening period begins at time t_0 and the FILTER transmission begins at $t_1 = t_0 + \epsilon$, where ϵ is a small positive number very close to zero. In this case, $t_0 + T_{wi} < t_1 + T_{wi}$, which means that the FILTER transmission will not be detected for T_{wi}, the minimum required time for correct detection. Thus, T_{ws2} needs to be chosen such that the next idle listening period will begin and end before the current FILTER transmission ends. The FILTER transmission will end at $t_1 + T_F$. Thus, the next idle listening period needs to begin by $t_1 + T_F - T_{wi} = t_0 + \epsilon + T_F - T_{wi}$ to allow the minimum detection time. The first idle listening period ended at $t_0 + T_{wi}$. Thus, subtracting the first idle listening period's end time from the second idle listening period's start time, we get: $(t_0 + \epsilon + T_F - T_{wi}) - (t_0 + T_{wi}) = T_F - 2T_{wi} + \epsilon$. Thus, we need: $T_{ws2} \leq T_F - 2T_{wi} + \epsilon$. Because $\epsilon \to 0$ and the T_{ws2} inequality must be valid for the smallest ϵ possible, we get: $T_{ws2} \leq T_F - 2T_{wi}$ to ensure that the second idle listening period completely overlaps with part of the FILTER packet transmission. To avoid unnecessary wake-ups, we set[5]:

$$T_{ws2} = T_F - 2T_{wi} \tag{24}$$

Next, we consider how many times these idle listening periods must occur on the wake-up radio to ensure that one overlaps with part of a FILTER packet transmission. This is necessary since the idle listening periods may occur during the αT_A time that the sender is idly listening for a FILTER-ACK. We assume that T_{ws2} is set according to Equation 24.

[5]If false negatives are a problem with detecting the wake-up channel busy, T_{ws2} and w_i could be adjusted to provide redundancy in the amount of times idle listening periods are guaranteed to overlap with FILTER packet transmission. The obvious trade-off is that more energy is consumed during the monitoring phase as T_{ws2} becomes smaller and w_i becomes larger.

In the worst case, the first wake-up idle listening period ends just after a FILTER transmission ends. For example, the FILTER packet transmission ends at t_1 and the first idle listening period ends at $t_2 = t_1 + \epsilon$. Thus, that idle listening period began at $t_0 = t_2 - T_{wi}$. In this case, $t_0 + T_{wi} > t_1$, which means that the FILTER transmission will not be detected for T_{wi}, the minimum time required for correct detection. After this most recent FILTER transmission, the sender will wait for αT_A time before beginning the next FILTER transmission (i.e., it begins at $t_1 + \alpha T_A$). We need to guarantee that enough idle listening periods with the T_{ws2} spacing will occur such that the last one begins after $t_1 + \alpha T_A$ (and hence is detects the next FILTER transmission). The next (second) idle listening period begins at $t_2 + T_{ws2}$. If there is another one (the third), it will begin at $t_2 + T_{ws2} + T_{wi} + T_{ws2}$. If we have w_i such idle listening periods, the last one will begin at $t_2 + (w_i - 1)T_{ws2} + (w_i - 2)T_{wi}$ (trivially, $w_i \geq 2$). Using Equation 24, the last idle listening period begins at: $t_2 + (w_i - 1)(T_F - 2T_{wi}) + (w_i - 2)T_{wi} = t_2 + (w_i - 1)T_F - w_i T_{wi}$. Thus, we need:

$$\begin{aligned} t_1 + \alpha T_A &\leq t_2 + (w_i - 1)T_F - w_i T_{wi} \\ t_1 + \alpha T_A &\leq t_1 + \epsilon + (w_i - 1)T_F - w_i T_{wi} \\ \alpha T_A &\leq \epsilon + (w_i - 1)T_F - w_i T_{wi} \end{aligned} \tag{25}$$

Because $\epsilon \to 0$ and the inequality in Equation 25 must be valid for the smallest ϵ possible, we get:

$$\alpha T_A \leq (w_i - 1)T_F - w_i T_{wi} \tag{26}$$

Therefore, we need w_i to be the smallest integer which satisfies the inequality in Equation 26. This gives us:

$$w_i = \left\lceil \frac{\alpha T_A + T_F}{T_F - T_{wi}} \right\rceil \tag{27}$$

To determine T_{wt}, we consider the worst case where the first FILTER transmission starts just after the last idle listening period begins for a monitoring node. In this case, the sender has to do the wake-up procedure for the length of that idle listening period (i.e., T_{wi}) plus the subsequent sleeping time (i.e., T_{ws}). Additionally, it must transmit for the time it takes the monitoring node to do w_i idle listening periods (i.e., $w_i T_{wi} + (w_i - 1)T_{ws2}$. If the beginning of the FILTER transmission is detected during the monitoring node's last idle listening period, it will keep its wake-up radio on to receive the next FILTER. This occurs after the sender idly listens for a FILTER-ACK (i.e., $2T_F + \alpha T_A$). Finally, the sender must continue the wake-up process long enough to receive the FILTER-ACK which follows the last FILTER transmission (i.e., αT_A). Combining all this time, we get:

$$\begin{aligned} T_{wt} &= T_{wi} + T_{ws} + w_i T_{wi} \\ &\quad + (w_i - 1)T_{ws2} + 2\alpha T_A + 2T_F \\ &= (w_i + 1)T_{wi} + T_{ws} + (w_i - 1)T_{ws2} \\ &\quad + 2\alpha T_A + 2T_F \end{aligned} \tag{28}$$

In STEM-H, there are significant effects from the fact that monitoring nodes are only awakened at discrete times (i.e., after detecting a FILTER transmission, waiting for the corresponding αT_A time, and the receiving the subsequent FILTER packet). In particular, we have discovered that our analysis and simulation trends for STEM-H do not match if we assume that the wake-up process lasts $T_{wt}/2$ time (i.e., a uniform distribution for the time the wake-up process will end). Thus, we extend the analysis for STEM-H as follows. First, the earliest time a FILTER can be completely received and decoded (as opposed to just being detected), W_L, is after a detection of the first FILTER, followed by the sender's idle listening period. Thus,

$$W_L = 2T_F + \alpha T_A \tag{29}$$

Next, we determine the maximum number of discrete times that a node may receive and decode the FILTER packet, which we denote as k. Obviously, the first FILTER packet sent cannot be decoded correctly since all the monitoring nodes are only trying to detect the wake-up channel as busy or not. Each successive possible FILTER reception time occurs $\alpha T_A + T_F$ after the end of the current FILTER transmission. We have:

$$k = \left\lfloor \frac{T_{wt} - T_F}{\alpha T_A + T_F} \right\rfloor \tag{30}$$

From Equation 30, if follows that the latest time a FILTER can be completely received and decoded, W_U, is:

$$W_U = T_F + k(\alpha T_A + T_F) \tag{31}$$

since the first FILTER cannot be received and decoded as discussed earlier.

Based on this, we derive $\overline{T_{wt}}$, the average time it takes a wake-up procedure in STEM-H.

$$\overline{T_{wt}} = \frac{W_U - W_L}{2} + W_L \tag{32}$$

From Equation 32, we can also compute $\overline{k}$, which is the average number of discrete times until a FILTER packet is correctly received.

$$\overline{k} = \left\lfloor \frac{\overline{T_{wt}} - T_F}{\alpha T_A + T_F} \right\rfloor \tag{33}$$

In STEM-H, p_w is calculated similar to STEM and STEM-BT:

$$p_w = \left\lceil \frac{T_{th} + \overline{T_{wt}} + \alpha T_A}{1/R - T_P} \right\rceil \tag{34}$$

The value of w_r is calculated as in STEM using Equation 3.

The energy used by the sender's wake-up radio is calculated similar to that of STEM in Equation 7. The only difference is now there are w_i idle listening periods per cycle with a sleep time of length T_{ws2} between periods. As in STEM, $F_{TX} = \frac{T_F}{T_F + \alpha T_A}$ and $F_I = 1 - F_{TX}$. Therefore,

$$\begin{aligned} E_{w\_TX} &= w_r \overline{T_{wt}} \left(F_{TX} P_{TX} + F_I P_I\right) \\ &\quad + \frac{(1 - w_r \overline{T_{wt}})(P_I w_i T_{wi})}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \\ &\quad + \frac{(1 - w_r \overline{T_{wt}})[P_S(T_{ws} + (w_i - 1)T_{ws2})]}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \end{aligned} \tag{35}$$

For the wake-up radio energy consumption of the receiving node, we need to compute the amount of time the node is in idle listening/receiving mode when a FILTER packet is detected. The first FILTER packet, the receiving node will detect, but not decode, halfway through the transmission (on average). The receiving node then keeps its wake-up radio on during the sender's idle listening phase (αT_A), receives and decodes the next FILTER packet (T_F), and replies with a FILTER-ACK. Thus, overall the receiving node uses the idle power level for $\frac{3}{2}T_F + 2\alpha T_A$ per wake-up[6]. Therefore, the derivation for E_{w_RX} is similar to E_{w_TX} in Equation 35. For ease of notation, let $\beta = \frac{3}{2}T_F + 2\alpha T_A$ and we have:

$$\begin{aligned} E_{w\_RX} &= w_r P_I \beta \\ &\quad + \frac{(1-w_r\beta)(P_I w_i T_{wi})}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \\ &\quad + \frac{(1-w_r\beta)[P_S(T_{ws} + (w_i-1)T_{ws2})]}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \end{aligned} \tag{36}$$

To determine the wake-up energy consumption during the monitoring process for neighboring nodes other than the receiver, we first need to figure out what percentage of the neighbors, on average, will detect the channel busy and remain on the receive a FILTER packet. The average number of FILTER transmission that occur before the receiver replies with a FILTER-ACK is $\overline{k} + 1$. Overall, there is is possible to have $k + 1$ total FILTER transmission. However, T_{wt} is designed such that all nodes will detect one of the first k FILTER transmissions and the last FILTER is only intended to be decoded by a node. Therefore,

$$pct_{nbr} = \frac{\overline{k} + 1}{k} \tag{37}$$

When a neighbor node detects a FILTER transmission, it reacts similar to the receiving node, except it does not respond with a FILTER-ACK. Thus, the average amount of time a neighbor, which we denote as γ for ease of notation, is: $\frac{3}{2}T_F + \alpha T_A$. For the overall wake-up energy consumption for neighbors, we get[7]:

$$\begin{aligned} E_{w\_nbr} &= pct_{nbr} \times w_r P_I \gamma \\ &\quad + pct_{nbr} \times \frac{(1-w_r\gamma)(P_I w_i T_{wi})}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \\ &\quad + pct_{nbr} \times \frac{(1-w_r\gamma)[P_S(T_{ws} + (w_i-1)T_{ws2})]}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \\ &\quad + (1 - pct_{nbr}) \times \frac{P_I w_i T_{wi}}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \\ &\quad + (1 - pct_{nbr}) \times \frac{P_S(T_{ws} + (w_i-1)T_{ws2})}{T_{ws} + (w_i - 1)T_{ws2} + w_i T_{wi}} \end{aligned} \tag{38}$$

[6]We do not account for the time that the receiver uses P_{TX} to send the FILTER-ACK since this is not a significant part of the overall energy consumption. We assume that the receiving node uses P_I power while sending the FILTER-ACK for simplicity.

[7]We assume that there is no synchronization among the neighboring nodes.

Combining Equation 35, Equation 36, and Equation 38, we get the overall wake-up radio energy consumption for STEM-H:

$$E_{STEMH\_w} = E_{w\_TX} + E_{w\_RX} + (N-2)E_{w\_nbr} \tag{39}$$

The data radio energy consumption for STEM-H is identical to that of STEM from Equation 12:

$$E_{STEMH\_d} = E_{STEM\_d} \tag{40}$$

Which gives an overall energy consumption of:

$$E_{STEMH} = \frac{E_{STEMH\_w} + E_{STEMH\_d}}{b_D R} \tag{41}$$

4) Discussion: STEM-H improves the energy consumption of STEM by reducing the cost of the steady-state monitoring process while maintaining the benefits from the relatively inexpensive wake-up process discussed in Section III-A.4. As we will see in Section IV, STEM-H does no worse than STEM, in terms of energy consumption, and in most environments does significantly better. This is true even with significant degradation due to false positives being detected on the wake-up channel. Intuitively, even if every idle listening period results in a false positive for STEM-H, monitoring nodes will stay on for $2T_F + \alpha T_A$ once after every T_{ws} sleeping period. Thus, it essentially behaves identically to STEM.

D. STEM-BT2

Based on the description of STEM-BT in Section III-B.4, we see that STEM-BT's energy consumption can be significantly improved by reducing the wake-up cost while retaining its relatively low monitoring costs. Thus, we propose STEM-BT2 which combines STEM-BT's wake-up protocol with data channel probing to create a protocol more energy efficient than STEM-BT.

The basic idea of STEM-BT2 is to perform the same wake-up protocol while avoiding excessive idle listening on the data channel while waiting for the FILTER packet to be sent. Rather than turning on the data radio and doing continuous idle listening like STEM-BT, STEM-BT2 will periodically probe the data channel to detect whether it is busy or not. When the data channel is detected busy, then STEM-BT2 remains on to receive the FILTER packet like STEM-BT.

1) Sending Protocol: The sending protocol is identical to that of STEM-BT, described in Section III-B.1 and shown in Figure 3. The only difference is that two FILTER packets are sent on the data channel rather than one. The first FILTER packet is a "dummy" packet that allows probing nodes to detect the channel as busy and the second packet is the one that actually gets decoded.

2) Monitoring Protocol: The monitoring cycle is the same as STEM-BT as shown in Figure 4. The difference in STEM-BT and STEM-BT2 is how they react after a monitoring node has detected a busy tone on the wake-up channel. The reaction of STEM-BT is described in Section III-B.2. STEM-BT2 reacts by turning on its data radio and idly listening for

T_{wi} time[8]. If the data channel is detected as busy, the data radio remains on in anticipation of a FILTER packet being received. If the data channel is detected as idle, the data radio returns to sleep for T_{ws2} time before attempting to idly listen again. This idle listening cycle is repeated until the channel is detected busy or a timeout occurs.

Once the FILTER packet has been received, nodes behave the same as in STEM-BT. The intended receiver, as specified by the FILTER packet, remains on while all other nodes return their data radios to sleep. The sender and receiver return their data radios to sleep when the data channel has been idle for T_{th} time.

3) Analysis: The value of T_{wt} in STEM-BT2 is computed exactly the same as for STEM-BT, as described in Section III-B.3. The value of p_w is slightly different since two FILTER packets are sent instead of one:

$$p_w = \left\lceil \frac{T_{th} + T_{wt} + 2T_F}{1/R - T_P} \right\rceil \tag{42}$$

The value of w_r is the same at STEM-BT except that the p_w value from Equation 42 is used. In fact, other than this new p_w value, the wake-up radio energy consumption is identical to that of STEM-BT.

$$E_{STEMBT2\_w} = E_{STEMBT\_w} \tag{43}$$

The data radio energy of STEM-BT2 is similar, but not identical, to STEM-BT. The difference is that the receiver and neighbors only idly listen a fraction of the time between the detection of the busy tone and the reception of the FILTER packet. For the sender, the data radio energy is slightly different because two FILTER packets are sent. Thus,

$$\begin{aligned} E_{d\_TX} &= w_r \left(P_{TX}(2T_F + p_w T_P)\right) + w_r P_I T_{th} \\ &\quad + w_r P_S \left(\frac{1}{w_r} - (2T_F + p_w T_P + T_{th})\right) \end{aligned} \tag{44}$$

For the receiver and neighbors, we set $T_{ws2} = T_F - 2T_{wi}$ for the same reasons discussed in Section III-C.3. Thus, once the data radio begins to probe the data channel, it idly listens for $\frac{2T_{wi}}{T_F}$ fraction of the time. Because each node detects the wake-up signal for $\frac{T_{wt}}{2}$ before receiving the FILTER packet (on average), the expected amount of time idly listening to the data channel after receiving the wake-up signal, T_I, is:

$$\begin{aligned} T_I &= \frac{2T_{wi}}{T_F} \times \frac{T_{wt}}{2} \\ &= \frac{T_{wi} T_{wt}}{T_F} \end{aligned} \tag{45}$$

Now, we derive the data radio energy consumption for the receiver and neighboring nodes. These values are similar to Equation 20 and Equation 21, respectively.

$$\begin{aligned} E_{d\_RX} &= w_r P_I \left(T_I + \frac{3}{2}T_F + p_w T_P + T_{th}\right) \\ &\quad + w_r P_S \left(\frac{1}{w_r} - \left(T_I + \frac{3}{2}T_F + p_w T_P + T_{th}\right)\right) \end{aligned} \tag{46}$$

$$\begin{aligned} E_{d\_nbr} &= w_r P_I \left(T_I + \frac{3}{2}T_F\right) \\ &\quad + w_r P_S \left(\frac{1}{w_r} - \left(T_I + \frac{3}{2}T_F\right)\right) \end{aligned} \tag{47}$$

Overall, we have:

$$E_{STEMBT2\_d} = E_{d\_TX} + E_{d\_RX} + (N-2)E_{d\_nbr} \tag{48}$$

$$E_{STEMBT2} = \frac{E_{STEMBT2\_w} + E_{STEMBT2\_d}}{b_D R} \tag{49}$$

4) Discussion: STEM-BT2 improves the energy consumption of STEM-BT by reducing the cost of the expensive wake-up process while maintaining the benefits from the relatively inexpensive steady-state monitoring process discussed in Section III-B.4. As we will see in Section IV, STEM-BT2 rarely ever does worse than STEM-BT, in terms of energy consumption, and in most environments does significantly better. This is true even with significant degradation due to false positives being detected on the data channel. Intuitively, in the worst case, where every idle listening period on the data radio detects the channel as busy, STEM-BT2 will behave identical to STEM-BT except that it sends two FILTER packets instead of one. Thus, in this case, STEM-BT2 uses extra energy to transmit the extra FILTER packet, but otherwise behaves the same as STEM-BT.

IV. Analytical and Experimental Results

To test the protocols described in Section III, we implemented all of them in *ns-2* [28] by modifying the 802.11 MAC and physical layer code in *ns-2*. We compare the simulation results with the analytical equations from Section III. In our tests, we use the values from Table II. These values are based on Mica2 Motes [3] and TinyOS [29]. For STEM-BT, STEM-H, and STEM-BT2, we set $T_{wi} = 1$ ms [27]. This is the time it takes to reliably detect that the wake-up and data channel are busy when a busy tone or packet is being transmitted. Each data point is averaged over 20 runs.

In Section IV-A and Section IV-B, 10 sensors nodes are placed within range of each other (i.e., $N = 10$) and a random sender and receiver are chosen to communicate with Poisson traffic at a specified rate. The sleep interval, T_{ws}, for the protocols is varied to determine its effects on energy and latency.

In Section IV-C, 50 sensor nodes are placed in a 1000 m × 1000 m area[9]. For each topology tested, a path exists between every node in the network. We vary the number of connections

[8]More generally, the idle listening time can be T_{wi2}, where $T_{wi2} \neq T_{wi}$.

[9]The transmission range of the nodes is 250 m. If a smaller transmission range is assumed for sensors, the area could be scaled proportionally to maintain the tested node density.

TABLE II

PROTOCOL PARAMETER VALUES.

Parameter	Value
Physical Layer Header	28 bytes
MAC Layer Header	6 bytes
Payload per Packet	30 bytes
Total Packet Size[a]	64 bytes
Bitrate	19.2 kbps
P_{TX}	81 mW
P_I	30 mW
P_S	3 μW
T_{th}	30 ms
α	1.1

[a] We assume that FILTER, FILTER-ACK, data, and ACK packets are all the same size.

per scenario while keeping T_{ws} and the Poisson traffic rate fixed. For each connection the sender and receiver of the flow is chosen uniformly at random.

In the simulations, we choose to use Poisson traffic despite the fact that the analysis considers CBR traffic. The reason for this is that CBR traffic demonstrated synchronization effects in the simulations that caused significant oscillations between individual runs for STEM and, particularly, STEM-H. By adding the non-determinism of Poisson traffic, these oscillations did not occur. The effects of using Poisson rather than CBR traffic leads to slightly lower energy/bit calculations, especially at longer T_{ws} values. The reason for this is that for most of the rates and T_{ws} values tested, $p_w = 1$, so each wake-up results in exactly one packet being sent for CBR traffic. However, for Poisson traffic, occasionally, more than one packet may be sent per wake-up.

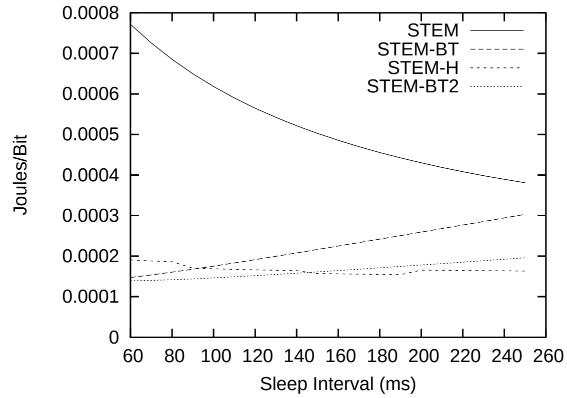
Unless otherwise noted, we used a per connection sending rate of one packet per second. Each data packet has a corresponding ACK packet that must be send back to the data sender. Thus, according to Table II, 128 total bytes are transferred per data packet. Thus, the channel utilization for one data packet per second is 5.33%.

A. Energy and Latency Comparison

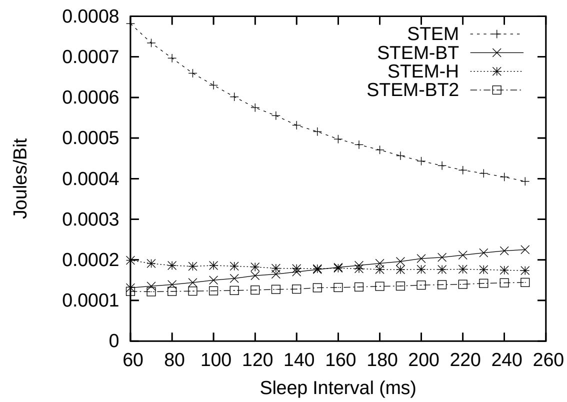
We adjusted T_{ws} from 60 ms to 250 ms to see the relative performance of the four protocols. Figure 6 shows that our analysis from Section III matches well with the simulation results.

We see that STEM uses the most energy of the protocols, but shows a large improvement as T_{ws} increases. Recall that STEM's major weakness is its large energy cost to monitor the wake-up channel in steady-state. As T_{ws} increases, the relative amount of sleep time for the monitoring process increases. Thus, the monitoring process uses less energy while the wake-up process uses only slightly more energy (due to the increase in T_{wt} for the sender). STEM-H consistently does much better than STEM. STEM-H's energy consumption also decreases as T_{ws} increases, though it is much less dramatic than STEM's decrease.

Figure 6 also shows that STEM-BT2 consistently outperforms STEM-BT. Both STEM-BT and STEM-BT2 show the same trend, a linear increase in energy as T_{ws} increases.



(a) Analytical results.



(b) Simulation results.

Fig. 6. Energy consumption of the protocols.

This is because the wake-up cost for these protocols increases while the monitoring cost only decreases slightly. In particular, the busy tone is transmitting for a longer time and, hence, neighbors have to keep their data radios idly listening (or probing) for a longer period of time on average.

In Figure 7, we see a linear increase in latency for all four protocols as T_{ws} increases. This is because the wake-up process takes longer as the sleep interval grows. STEM and STEM-H show a more gradual increase in latency because the time of their wake-up processes are proportional to roughly $\frac{1}{2}T_{ws}$, whereas STEM-BT and STEM-BT2's wake-up processes are proportional to T_{ws}. We also note that STEM-H and STEM-BT2 have a latency that is larger than that of STEM and STEM-BT, respectively, by a constant amount. This constant amount is approximately equal to T_F since STEM-H has to wait for an extra FILTER to be send on the wake-up channel (when compared to STEM) and STEM-BT2 has to wait for an extra FILTER to be sent on the data channel (when compared to STEM-BT).

To gain a better understanding of the energy-latency trade-off for the protocols, we plot the average latency of Figure 7 versus the energy consumption of Figure 6(b) for each fixed T_{ws} value. The result is shown in Figure 8(a). We can see that STEM-BT2 outperforms all of the other protocols in this

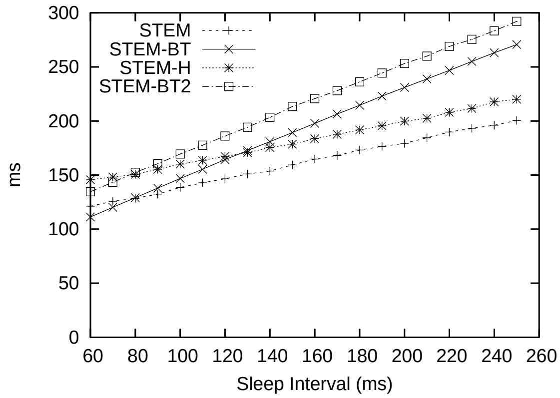
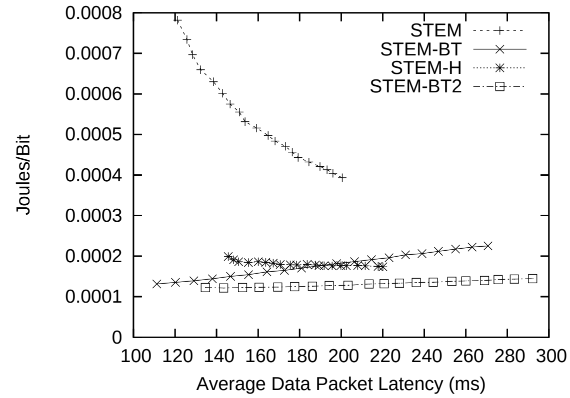
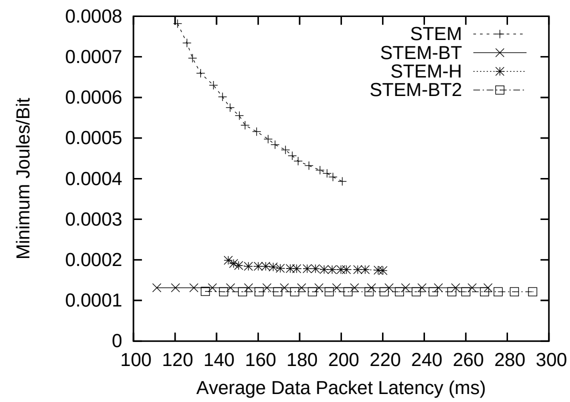


Fig. 7. Latency of the protocols.



(a) Exact energy for a given latency.



(b) Minimum energy for a given latency.

Fig. 8. Energy versus average latency of the protocols.

metric. However, we notice that STEM-BT and STEM-BT2 show an increase in energy as latency increases while STEM and STEM-H show a decrease in energy. Thus, comparing the energy of STEM-H to that of STEM-BT when the average latency is about 220 ms is misleading because STEM-H has lower energy consumption at that point, but STEM-BT has lower energy consumption when the average latency is smaller. For nearly all applications, desired latency is specified as less than or equal to rather than equal to. Thus, we present the results in a slightly different form in Figure 8(b). Here, we show the minimum energy consumption possible for each protocol to achieve a latency less than or equal to the x-axis value. From Figure 8(b), we see that STEM-BT and STEM-BT2 are nearly identical and consume less energy than STEM-H and STEM.

In Figure 9, we test the protocols at a higher sending rate. The rate is set to three packets per second (i.e., 16% channel utilization). From this graph, we see that the relative difference in energy consumption between STEM and STEM-H decreases. This is because a higher rate reduces the amount of monitoring time between wake-up procedures. Thus, STEM does much better because there is less monitoring time per packet arrival. STEM-H, however, shows less relative improvement since its monitoring cost is already low by design.

Also in Figure 9, we see that the relative difference between STEM-BT and STEM-BT2 increases at a higher rate. This is because, at a higher rate, the wake-up procedure becomes more frequent. Thus, STEM-BT2, which has a lower wake-up cost than STEM-BT, will further outperform STEM-BT.

B. The Effects of Spurious Wake-Ups

One of the disadvantages of doing the FILTER transmission detection on the wake-up channel in STEM-H is that interference in the frequency band may cause a node to detect the channel as busy when their is no FILTER being transmitted. For example, such interference may come from other sensor nodes that are not within communication range, but still transmit with enough power to interfere. Another example, is other electronic devices that may share the same unlicensed bandwidth as the sensor network may cause interference. Thus, it is instructive to study how the performance of STEM-H degrades in the face of such interference.

Figure 10 shows the energy consumption of STEM and STEM-H. For STEM-H, we vary the probability that when a monitoring node idly listens on the wake-up channel, it detects a FILTER transmission in error (i.e., a false positive). For STEM-H, the percentage values shown in the key of Figure 10 indicate the probability a false positive occurs each idle listening period. For example, “STEM-H, 5%” indicates that each idle listening period on the wake-up channel a monitoring node falsely detects and reacts to a FILTER transmission with probability 0.05. Thus, 0% false postive value indicates that *every* detection of the wake-up channel as busy is caused by a FILTER transmission. A 100% false positive value is the worst case where every idle listening period a monitoring node detects the wake-up channel as busy regardless of its actual state.

From Figure 10, we see that a low false positive percentage (e.g., less than 5%) does not affect the performance of STEM-H very much and it still significantly outperforms STEM. As the false positive percentage increase, the performance of STEM-H converges to that of STEM (the line for STEM and STEM-H with 100% false positives almost overlaps). This confirms to intuition discussed in Section III-C.4. Thus, with

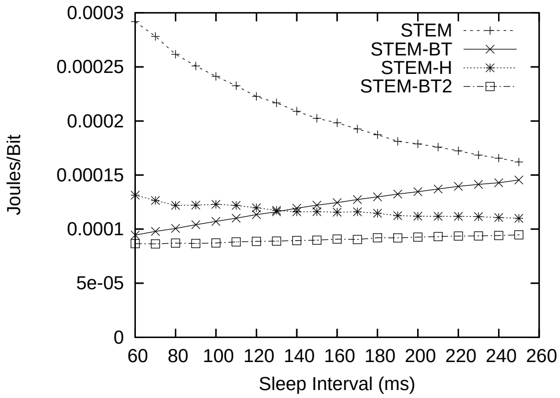


Fig. 9. Energy consumption of the protocols at a higher rate.

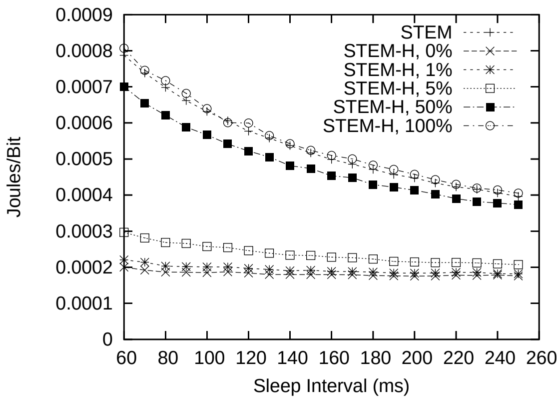


Fig. 10. Energy consumption of STEM and STEM-H with false positives.

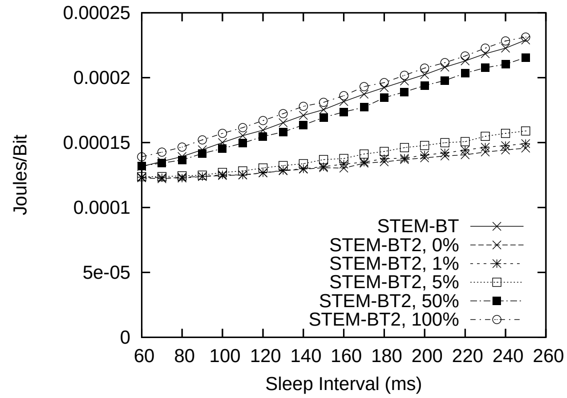


Fig. 11. Energy consumption of STEM-BT and STEM-BT2 with false positives.

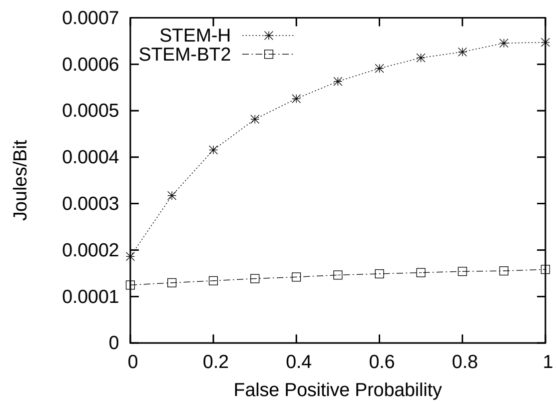


Fig. 12. Energy consumption of STEM-H and STEM-BT2 with false positives.

completely unreliable FILTER transmission detection, STEM-H performs no worse than STEM.

We perform similar tests with STEM-BT and STEM-BT2. Since both of these protocols do busy tone detection on the wake-up channel, we do not test this because it would affect both of the protocols similarly. In this case, STEM-BT2 would perform better relative to STEM-BT in this case since its spurious wake-ups are less expensive. Instead, we focus on false positives on the data channel since STEM-BT2 is periodically idly listening for FILTER transmission on the data channel. In contrast, monitoring nodes in STEM-BT constantly idly listens to the data channel once a wake-up occurs.

The results are shown in Figure 11. Again, we see that a small false positive percentage only has a small effect on STEM-BT2. As the false positive percentage increases toward the worst case scenario (i.e., 100%), STEM-BT2's performance converges to slightly worse than STEM-BT's performance. The reason for this is a 100% false probably percentage for STEM-BT2 exhibits nearly identical behavior to STEM-BT except that STEM-BT2 still has to send and receive two FILTER packets instead of one. Thus, a small amount of extra energy is used by STEM-BT2 for every wake-up to send and receive this second FILTER packet on the data channel.

Our final experiment to test spurious wake-ups is to determine the relation between the false positive percentage and energy consumption for STEM-H and STEM-BT2. In Figure 12, we fix $T_{ws} = 100$ ms and see that false positives have a more detrimental effect on STEM-H than STEM-BT. A major reason for this is that idle listening periods on the wake-up channel for STEM-H are much more frequent than idle listening periods on the data channel for STEM-BT2. Thus, the number of false positives occurring for STEM-H is significantly larger than for STEM-BT2.

C. Multi-Hop Performance

All of the results in previous sections have only considered single-hop networks with one flow. While this helps to gain a fundamental understanding of the protocols, it is unrealistic as sensor networks are designed to be multi-hop and operate with multiple, concurrent flows. Thus, we test multi-hop, multi-flow environments in this section. We note that this is the first time, of which we are aware, that such tests have been done on STEM or STEM-BT. In [1], [2], the simulation results are rather limited in that they only consider a single flow with an extremely low channel utilization.

In Figure 13, we set $T_{ws} = 100$ ms an incrementally

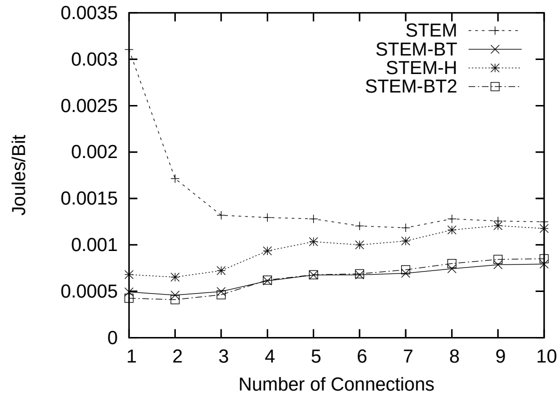


Fig. 13. Energy consumption of protocols in multi-hop scenarios ($T_{ws} = 100$ ms).

increase the number of concurrent flows in the network. We see that STEM-H's energy consumption converges to that of STEM with a larger number of flows. This is to be expected based on Figure 10 and Section III-C.4 since a larger number of flows implies that there will be more contending transmissions on the wake-up channel. Thus, monitoring nodes will wake up more frequently in response to FILTER packet transmissions for which it is not the intended receiver.

Similarly, Figure 13 also shows that STEM-BT2's performance converges to slightly worse than that of STEM-BT as the number of flows becomes large. Again, the increased flow contention means that busy tones are detected more frequently. When a node turns its data radio on in response, there is a greater probability that STEM-BT2 detects packet transmissions on the data channel which cause it to keep its data radio on. Thus, as explained in reference to Figure 11, STEM-BT2 ends up doing slightly worse than STEM-BT because it has to send two FILTER packets instead of one.

V. Conclusions and Future Work

In this paper, have categorized wake-up protocols for sensor and ad-hoc networks into three basic types: synchronous, asynchronous, and out-of-band. After presenting many examples of each type, we focus on improving the energy consumption out-of-band wake-up protocols. The trade-offs of using an out-of-band protocol, as opposed to synchronous or asynchronous, are discussed in Section I.

We presented a description of two previously proposed out-of-band protocols, STEM and STEM-BT [2]. In addition, unlike previous work, we did extensive analysis on the expected energy consumption per data bit of these protocols. Based on this analysis, we observe a way that each protocol could be more energy efficient. In terms of energy consumption, STEM has a high steady-state wake-up channel monitoring cost and STEM-BT has a high cost each time a wake-up is performed. Thus, we propose two new protocols, STEM-H and STEM-BT2, to lower the energy consumption of STEM and STEM-BT, respectively. These new protocols are also analyzed for comparison with STEM and STEM-BT.

With a theoretical understanding of all four protocols, we implement each of them in the *ns-2* simulator and perform extensive tests for comparison. Our results show that, in terms of energy, STEM-H and STEM-BT2 virtually always outperform STEM and STEM-BT, respectively. This is true even when STEM-H and STEM-BT2 detect a large percentage of false positives during their channel probing periods. In addition, we also test the protocols in a more realistic multi-hop, multi-flow sensor network environment, unlike previous work.

For future work, we would like to see how the new protocols work in combination with other synchronous and asynchronous wake-up techniques. In particular, we plan to adapt the techniques from [5], [6] to STEM-H and STEM-BT2. Also, we would like to investigate the performance of the protocol in the context of more realistic sensor network applications, instead of only using random Poisson and CBR connections.

References

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

[2] ——, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, January-March 2002.

[3] MICA2 Mote Datasheet, `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-05_A_MICA2.pdf`.

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

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

[6] ——, "A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio," *IEEE Transactions on Mobile Computing*, 2004, accepted for publication.

[7] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous Wakeup for Ad Hoc Networks," in *ACM MobiHoc 2003*, June 2003.

[8] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," in *IEEE Infocom 2002*, June 2002.

[9] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms," in *ACM MobiHoc 2004*, May 2004.

[10] M. J. McGlynn and S. A. Borbash, "Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks," in *ACM MobiHoc 2001*, October 2001.

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

[13] 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.

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

[15] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," in *ACM SenSys 2003*, November 2003.

[16] M. L. Sichitiu, "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks," in *IEEE Infocom 2004*, March 2004.

[17] M. J. Miller and N. H. Vaidya, "On-Demand TDMA Scheduling for Energy Conservation in Sensor Networks," University of Illinois at Urbana-Champaign, Tech. Rep., 2004.

[18] C. R. Lin and M. Gerla, "Real-time Support in Multihop Wireless Networks," *ACM Wireless Networks*, vol. 5, no. 2, pp. 125–135, March 1999.
[19] L. M. Feeney, "A QoS Aware Power Save Protocol for Wireless Ad Hoc Networks," in *IFIP Med-Hoc-Net 2002*, September 2002.
[20] L. Gu and J. A. Stankovic, "Radio-Triggered Wake-Up Capability for Sensor Networks," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2004*, May 2004.
[21] C. F. Chiasserini and R. R. Rao, "Combining Paging with Dynamic Power Management," in *IEEE Infocom 2001*, April 2001.
[22] C. S. Raghavendra and S. Singh, "PAMAS – Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks," *ACM Computer Communications Review*, July 1998.
[23] R. Zheng and R. Kravets, "On-demand Power Management for Ad Hoc Networks," in *IEEE Infocom 2003*, April 2003.
[24] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks," USC/Information Sciences Institute, Tech. Rep. 527, 2000.
[25] ——, "Geography-informed Energy Conservation for Ad Hoc Routing," in *ACM MobiCom 2001*, July 2001.
[26] 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.
[27] X. Yang and N. H. Vaidya, "A Wakeup Scheme for Sensor Networks: Achieving balance between energy saving and end-to-end delay," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) 2004*, May 2004.
[28] `ns-2` – The Network Simulator, `http://www.isi.edu/nsnam/ns`.
[29] TinyOS Community Forum, `http://webs.cs.berkeley.edu/tos`.