Reconsidering Power Management

Invited Paper

Cigdem Sengul, Albert F. Harris III, and Robin Kravets
Computer Science Department
University of Illinois at Urbana-Champaign
{sengul,aharris,rhk}@cs.uiuc.edu

***Abstract*— Power-management approaches have been widely studied in an attempt to conserve idling energy by allowing nodes to switch to a low-power sleep mode. However, due to the inherent inability of current approaches to match sleep schedules to different traffic patterns, energy is wasted switching needlessly from sleep to idle or large delays in traffic delivery are incurred due to being in the sleep state too long. In this paper, we explore such effects of various traffic patterns on current power management protocols. Our results show the importance of traffic information to obtain larger benefits from power management. While some proposals that exploit traffic information exist, they rely primarily on individual sender traffic patterns to develop sleep schedules, ignoring aggregate traffic observed by receivers. This deficiency motivates the design of a new power management protocol that use traffic information at the receivers to adapt sleep schedules.**

I. INTRODUCTION

Due to the energy-constrained nature of mobile devices, there has been extensive research on energy conservation throughout mobile systems. One specific target of energy conservation is network idle-time. While energy can be saved by keeping the wireless interface in a low-power sleep state in the presence of idling, the network performance might be negatively impacted since these cards have no communication ability when sleeping. Therefore, a wide range of power-management protocols have been proposed to efficiently wake up cards from the sleep state [1], [2], [3], [4], [5], [6], [7]. However, the majority of these protocols assume simplified energy models or traffic patterns. Changing these assumptions obviously affects the performance. The goal of our research is to evaluate the extent of such effects with the objective of designing better power-management protocols.

The main problem with current power management protocols is that each type of protocol makes assumptions about network traffic, which results in consuming energy either needlessly idling or switching between sleep and idle states. For instance, in synchronous protocols (*e.g.,* IEEE 802.11 PSM [1]), nodes wake-up periodically based on a schedule and stay awake for a fixed interval if they receive traffic announcements. Hence, long packet inter-arrival times incur high idling energy consumption. However, if the traffic is bursty, the awake intervals are utilized more efficiently. In asynchronous protocols, the receivers stay awake longer to overlap with senders [8], [9], [10], [11] or the senders stay awake longer and transmit a long preamble preceding each packet to wake up the receivers (*e.g.,* BMAC [2]). Therefore, receiver-based asynchronous scheduled protocols also have high idling overhead. For the sender-based protocols, when the packet inter-arrival times are short, the sender needlessly sends a long preamble even though its receivers are awake. Therefore, long inter-arrival times help amortize the preamble cost. Trigger-based protocols (*e.g.,* Wake-on-Wireless [3]) use a low-power radio to wake up the high-power data radio. Therefore, the trigger-based protocols incur high switching costs when the packet inter-arrival times are longer than the wake-up time of the high-power radio. Again, bursty traffic can take better advantage of the time the data radio is awake.

The main contribution of this paper is an extensive study of three representative power management protocols: IEEE 802.11 PSM [1], B-MAC [2], and Wake-on-Wireless [3]. We evaluate the effects of actual interface energy consumption with a specific focus on switching costs, as well as varying traffic patterns on energy-efficiency and delay of these protocols. To this end, we first present an energy model that accounts for the energy cost in each state of the wireless interface (*i.e.*, transmit, receive, idle and sleep). Furthermore, different than typical energy models, our model includes the energy consumption of switching into and out of sleep, idle and transmit. This energy model provides a comprehensive representation of the different components of the energy consumption of a wireless interface, which we also verify by energy measurements of IEEE 802.11 interfaces. Additionally, the time spent in each state also affects energy consumption and is determined by both the power-management protocol and the traffic patterns in the network. Therefore, we study a number of different traffic patterns.

Our study shows that violating the energy model and traffic pattern assumptions negatively impacts performance. While some research on using traffic shaping to augment power management exists [12], [13], [14], [15], [16], in these approaches, each sender independently affects sleep scheduling based on its traffic. However, the problems due to a mismatch of sleep-scheduling to current traffic would grow more severe when multiple senders, each with a unique traffic pattern to a single receiver, are considered. This observation suggests a new type of power management protocol, where the receiver determines sleep schedules and transmission times.

The rest of this paper is organized as follows. Section II presents our energy model, which is verified by experimental

This work is partly funded by Vodafone Fellowship and NSF CNS 0347468.

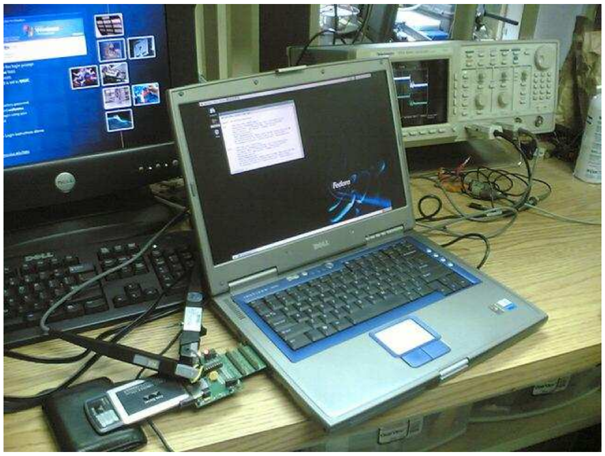

Fig. 1. Measurement setup used to determine energy characteristics of Aironet 350 wireless card in different modes.

TABLE I

TRANSMIT AND RECEIVE ENERGY COSTS (MW) FOR SELECTED CARDS.

Card	P_{tx}^{max}	P_{rx}
Cabletron [19], [20]	1400	1000
Lucent Wavelan [21], [22]		
2Mbps	1327.2	966.9
11Mbps	1346.1	900.6
Monolithics [4]	14.88	12.5
Mica Mote [23], [24]	81	30
Cisco Aironet 350 [18]	1850	1590

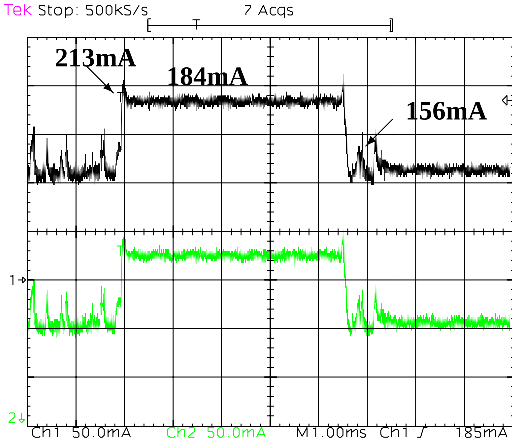

Fig. 2. Transmit energy: Transmit power level 100 mw, transmit rate 2 Mb/s.

measurements. Section III presents our categorization of power management protocols and the effects of violating traffic pattern assumptions on various power management schemes. Finally, Section V presents conclusions and future directions.

II. ENERGY CONSUMPTION FOR WIRELESS COMMUNICATION

To evaluate the energy-efficiency of different power management protocols [17], we summarize our energy model that captures the time a wireless interface spends in each state (*i.e.*, *transmit*, *receive*, *idle* and *sleep*) and switching costs between states. Our model is based on the energy characteristics a of typical wireless interface, which we validate through measurements. Based on the states of the wireless interface, we divide energy consumption into communication costs (*i.e.*, transmit and receive) and passive costs (*i.e.*, when it is not engaged in communication). The rest of this section presents these costs using the published information about various wireless cards, and our measurements of the Cisco Aironet 350 [18] card. In our measurement set-up (see Fig. 1), which is similar to [3], we used the PCCExtend extender to expose the connections of the wireless card, including the V_{cc}. The extender is inserted into the laptop and the card is inserted to the extender. The current is directly measured using a Tektronix TCP202 current probe and a Tektronix TDS654C oscilloscope.

A. Communication Costs

Communication costs of a node i, $E_{comm}(i)$, comprises of the energy consumed for transmit and receive.

$$E_{comm}(i) = t_{rx}(i) \cdot P_{rx} + \sum_{j \in NextHop} t_{tx}(i,j) \cdot P_{tx}(i,j), \qquad (1)$$

where node i's receive power is P_{rx}. $P_{tx}(i,j)$ is the total transmission power and is defined as $P_{tx}(i,j) = P_{base} + P_t(i,j)$, where P_{base} is the base transmitter cost and $P_t(i,j)$ is the transmit power level. Energy consumption is determined not only by the power level but also by how much time the device needs to remain in each state. Hence, in (1), $t_{rx}(i)$ is the total time spent in reception and $t_{tx}(i,j)$ is the time node i spends transmitting to node j. Since P_{rx} and P_{base} are fixed costs, $E_{comm}(i)$ is determined by how much data a node transmits and the transmission costs defined by the $P_{tx}(i,j)$'s. Essentially, the time spent in transmission and reception depends on the traffic pattern (i.e., packet size and burst length) and the channel quality, which determines how many (re)transmissions a node needs to handle.

Table I shows P_{tx}^{max} (energy at the maximum transmit power level) and P_{rx} for various cards. $P_t(i,j)$ attenuates with the n^{th} power of d, where d is the distance between nodes i and j, n is the path loss exponent and $2 \leq n \leq 4$ depending on the characteristics of the communication medium. Although $P_t(i,j)$ is a continuous function of distance, current cards support discrete transmit power level settings. For instance, Cisco Aironet 350 has five transmit power levels (mW): 100, 50, 30, 20 and 5. Fig. 2 shows $E_{comm}(i)$ for the Aironet 350. We measured the current that passed through the redundant lines on the extension board independently. Therefore, Fig. 2 depicts two identical curves and hence, the actual current consumption is twice what is shown in the labels. As in Table I, P_{tx} is higher than P_{rx} (2×184 mA and 2×156 mA respectively). Fig. 2 also depicts costs such as switching the transceiver on (2×213 mA). These costs are included in the transmission energy calculations for the Aironet 350 in Table I.

B. Passive Costs

$E_{passive}(i)$ represents the energy consumed when a node is not involved in reception or transmission. During this time, the wireless interface of a node can expend energy idling, sleeping

TABLE II

IDLE AND SLEEP MODE ENERGY COSTS (MW) AND TRANSITION ENERGY (J) FOR SELECTED WIRELESS CARDS.

Card	P_{idle}	P_{sleep}	$P_{sw} \cdot t_{sw}$
Cabletron [19], [20]	830	130	1.328
Lucent Wavelan [21], [22]			
2Mbps	843.7	66.3	0.6
11Mbps	739.4	47.4	0.6
Monolithics [4]	12.36	0.016	
Mica Mote [23], [24]	30	0.003	
Cisco Aironet 350 [18]	1150	140	0.6 (ad hoc) 0.19 (managed)

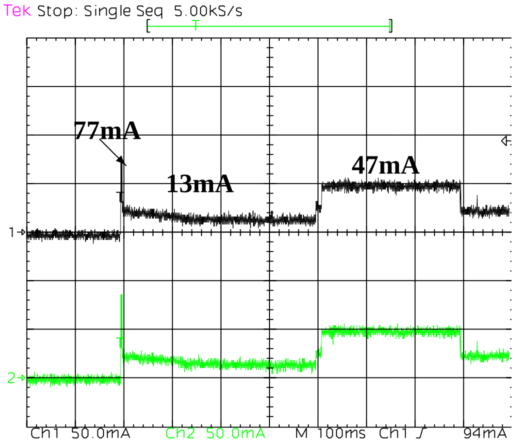

Fig. 3. Transition energy in managed mode.

or switching between these two states. Therefore,

$$E_{passive}(i) = t_{idle}(i) \cdot P_{idle} + t_{sleep}(i) \cdot P_{sleep} + t_{switch} \cdot P_{switch}, \tag{2}$$

where idling energy is determined by idle power, P_{idle}, and the duration of idling, $t_{idle}(i)$. Sleeping energy is a function of the sleep power, P_{sleep}, and the duration of sleep, $t_{sleep}(i)$. The cost of switching from idle to sleep is negligible. Hence, transition energy is the energy spent switching from sleep to idle, which is determined by switch power, P_{switch} and the duration of switching, t_{switch}. In our discussion of switching costs, we have only considered the actual device switching cost and not *the protocol cost* that includes the cost of finding out if the card can switch to a sleep state.

Table II lists P_{idle} and P_{sleep} for various cards. Except for a few cards, $E_{switch} = t_{switch} \cdot P_{switch}$ is typically not known. Therefore, we carefully measured P_{switch} for the Aironet 350. The measurements show that E_{switch} varies depending on whether the device is in managed or ad hoc mode since the Aironet 350 card goes through different states for different modes (see Figs. 3 and 4). The ad hoc mode cost is approximately the same as the Lucent Wavelan [21], [22], whereas the cost of transitions in managed mode are lower. Among the cards we considered, Cabletron [19], [20] is reported to have the highest transition energy costs.

Obviously, $E_{passive}(i)$ is minimized if the network interface sleeps as soon as the node becomes idle and sleeps long enough to amortize the switching cost. However, the time in each state is defined by the interactions between network traffic and the power management protocol. Essentially, inter-arrival characteristics of the network traffic and how the power management protocol reacts to the current traffic determines the extent of idling and switching costs, and hence, $E_{passive}(i)$. Next, we discuss such protocol costs and compare different power management protocols under different traffic types.

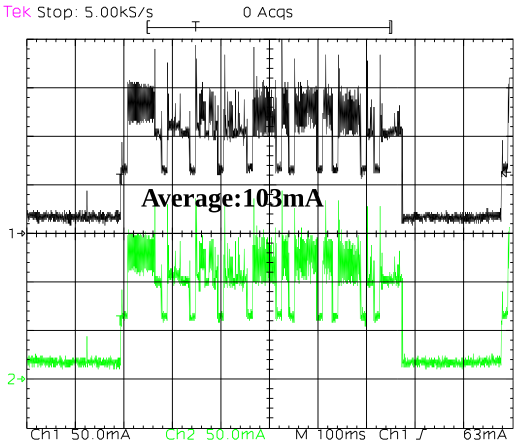

Fig. 4. Transition energy in ad hoc mode.

III. COST OF POWER-MANAGEMENT IN WIRELESS NETWORKS

Idle-time power management protocols can be divided into two categories: synchronous and asynchronous. Synchronous protocols use shared schedules, which can be either global (*i.e.*, defined across the network [1]) or local (*i.e.*, negotiated between neighbors [5], [6], [7]). Asynchronous protocols coordinate node wake up times without shared schedules [2], [3], [4], [7]. In this section, we discuss and evaluate the power-management protocols in each category in terms of their impact on idling energy consumption and network performance under different traffic patterns. Finally, we compare all protocols in terms of their switching costs. To this end, we next present the traffic patterns used in our evaluation.

A. Traffic Model

To evaluate the impact of various traffic patterns on different power management protocols, we use the following traffic types: CBR (constant bit rate), on-off and TCP. Using CBR, a node generates packets at every *interval*, which determines the rate of the flow. The parameters for on-off are *burst-time* and *idle-time*, which follow exponential distributions. Finally, a TCP flow from a client to a base station is based on measurement traces from an in-motion networking study [25]).

We used ns2 [29] to derive the send times for different traffic patterns: CBR with packet generation intervals of 0.5 s and 5 s, labeled as CBR-0.5 and CBR-5 respectively, on off traffic with 0.5 s and 5 s burst-time and 0.5 s of idle-time, labeled as OnOff-0.5 and OnOff-5 respectively, and finally, a TCP flow

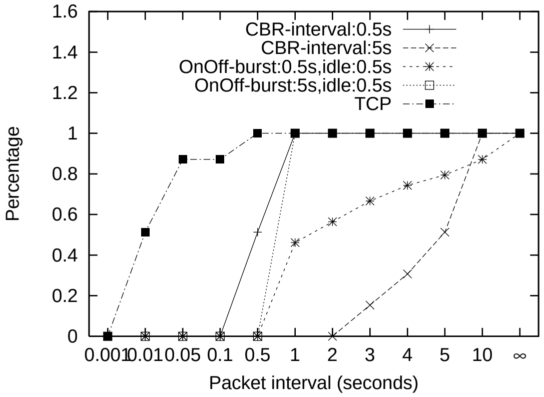

Fig. 5. Transmit patterns with different traffic types.

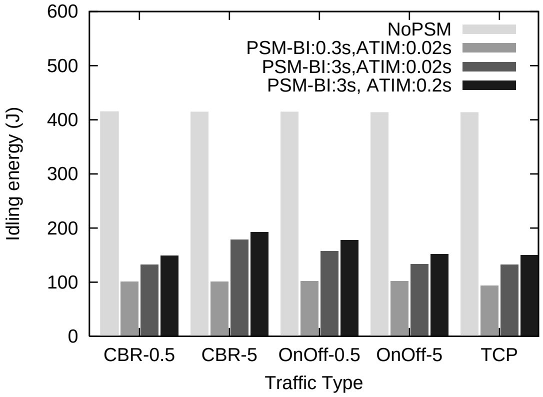

Fig. 7. IEEE 802.11 PSM: Idling energy consumption at the receiver.

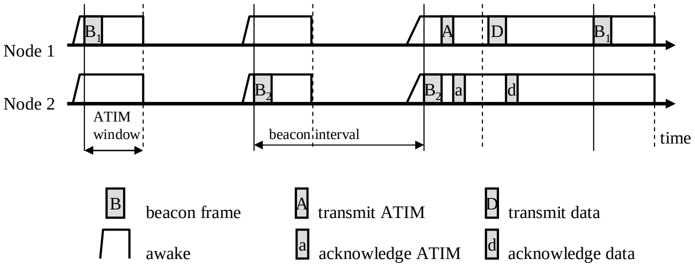

Fig. 6. IEEE 802.11 power-save mode

labeled as TCP. A sender sends 40 packets, each 128 bytes, are to a receiver. Simulation time is 500s.

Fig. 5 depicts the CDF (cumulative distribution function) of packet arrival times for these traffic patterns. Essentially, significantly different packet inter-arrival times are observed with each traffic pattern. For instance, while for the TCP trace 80% of the packets are sent within 0.05s, the packet inter-arrival time of the on-off traffic with burst-time of 0.5 s changes between 1 s and 5 s. Therefore, we next discuss the different classes of power-management protocols and how they are affected by different traffic patterns.

B. Synchronous power management

Synchronous power management schemes attempt to leverage shared schedules to reduce the amount of idling. To do this, a predefined schedule is distributed among nodes (either globally or locally). When a node has data to send, it waits for the next period that the intended receiver should be awake before transmitting. There are a number of trade-offs when designing synchronous power management schemes. First, synchronization is difficult to maintain in wireless networks in a distributed fashion. Second, matching the time between the awake periods of a node to its traffic pattern is difficult. If the nodes wake up far more frequently than data is sent, energy is wasted in needless switching. However, if nodes wake up too infrequently, large delays in delivery can occur.

IEEE 802.11 Power Save Mode (PSM) [1] is the standard synchronous power-management protocol, which also provides support for buffering packets for sleeping nodes. To maintain synchronization, beacon messages are sent at the beginning of every beacon interval. While the algorithm performs well in single-hop networks, it might not converge in dynamic environments. This problem motivated several asynchronous protocols, which are discussed in the next subsection.

In IEEE 802.11 PSM, broadcast, multicast or unicast packets to a power-saving node are announced using ad hoc traffic indication messages (ATIMs) at the beginning of the beacon interval called the ATIM window. Only ATIM messages are allowed during this window. Fig. 6 shows the interactions between two nodes using IEEE 802.11 PSM in an ad hoc network. During the first two beacon intervals, no packets are pending for either node. The two nodes randomly send beacon messages to maintain synchronization. In the third interval, Node 1 has a packet to send to Node 2 and so sends a directed ATIM, which is acknowledged by Node 2. A node that receives a directed ATIM sends an acknowledgment and stays awake for the entire beacon interval waiting for a data packet to be transmitted. After the ATIM window, Node 1 sends the packet using normal channel access rules. Following the transmission of all announced packets, nodes can continue to transmit packets destined to nodes that are known to be awake for the current beacon interval.

In IEEE 802.11 PSM (and similar protocols that use an out-of-band channel to announce pending transmissions), the throughput of the network is limited to the amount of data that can be announced in the channel. If a node cannot send an indication message to wake up its destination, it must buffer its packets until the next beacon interval. If this continues to happen, the node's buffer eventually fills up and packets are dropped. Hence, packet inter-arrival times determine both communication performance and energy conservation (*e.g.,* depending on what extent idling energy consumption can be avoided at each beacon interval.)

The protocols parameters (i.e., the size of the beacon interval and atim window) affect idling and switching costs. For instance, regardless of the presence of traffic, each node stays awake the entire ATIM window in expectation of ATIMs. In IEEE 802.11 PSM, if there are no traffic announcements, the node stays idle at least the duration of the ATIM window. If the node stays awake to receive traffic at the end of the ATIM window, the amount of idling depends on how well the rest of

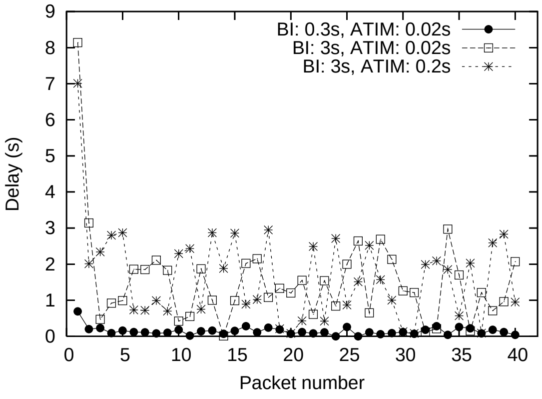

Fig. 8. IEEE 802.11 PSM: CBR traffic, packet interval 5 s.

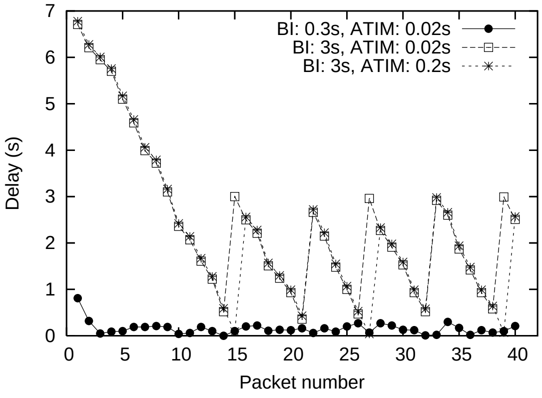

Fig. 9. IEEE 802.11 PSM: CBR traffic, packet interval 0.5 s.

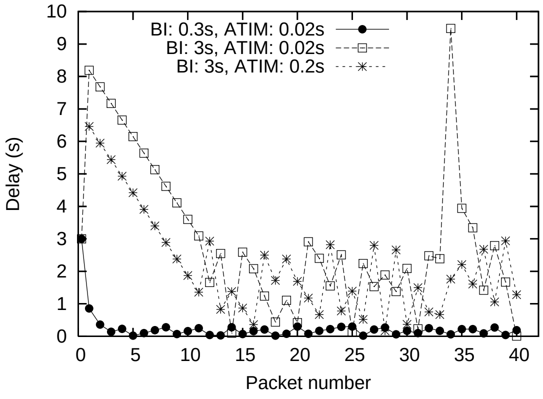

Fig. 10. IEEE 802.11 PSM: On Off traffic, burst interval 0.5 s.

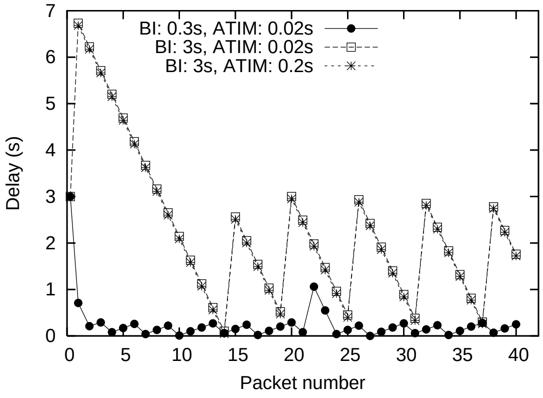

Fig. 11. IEEE 802.11 PSM: On Off traffic, burst interval 5 s.

the beacon interval is utilized. In our study, we experimented with 3 different beacon interval and ATIM window values (BI, ATIM): (0.3 s, 0.02 s), (3 s, 0.02 s) and (3 s, 0.2 s). Fig. 7 shows that while PSM effectively reduces idling energy, idling energy increases as BI increases, especially for traffic that has long inter-packet arrival times (*e.g.,* CBR-0.5). Idling energy is also expected to increase in the presence of broadcast/multicast packets, since the ATIMs for these messages cause all nodes (or just the nodes in the multicast group) to stay awake for the entire beacon interval. Since the neighbor sets of two nodes are not likely to be exactly the same, nodes rebroadcast in the next ATIM window even if all intended receivers have been covered in the current ATIM window, leading to unnecessary idling and reception costs.

Wake-up and idling patterns also affect the delay characteristics of the communication. Figs. 8, 9, 10, 11 and 12 depict IEEE 802.11 PSM performance for CBR-5, CBR-0.5, OnOff-0.5, OnOff-5 and TCP, respectively. We observe that with each traffic type, the first packet experiences a high delay since it takes longer to find a route when a power-management protocol is in place. Fig. 8 shows that when the interval between packets is long (*i.e.,* 5 s), each packet is delayed approximately the beacon interval duration. This also indicates that the receiver wakes up only for one packet and idles the majority of the beacon interval. On the other hand, in Fig. 8, the CBR traffic has a higher rate (*e.g.,* 0.5 s packet interval). Hence, we observe a triangle pattern, where the first packet observes a high delay, while the rest of the packets that can be sent in the same beacon interval have shorter delays. This also indicates a better utilization of the beacon interval. For on-off traffic, back-to-back packets result in a triangle pattern, and as expected this pattern becomes more observable as the burst-interval increases (see Figs. 10 and 11). Finally, TCP traffic exhibits the most interesting pattern. In this traffic model, 40 packets are sent with shorter intervals compared to the other traffic models. When the beacon interval is 3 s and the ATIM window is 0.02 s, all packets experience a delay of 8 s, which is the route discovery delay, and then the entire batch is sent in one beacon interval. However, when we increase the ATIM window size, there is less time to send data packets in a single beacon interval, and hence, we see a stair-step pattern.

C. *Asynchronous power management*

Asynchronous power management protocols were designed to address two fundamental problems with synchronous protocols. First, synchronization is difficult to maintain. The loss of synchronization can cause message delay and energy overhead. Second, predefined wake-up schedules are difficult to match to future traffic patterns, causing further cost in terms of delay and energy. To answer these two problems, two

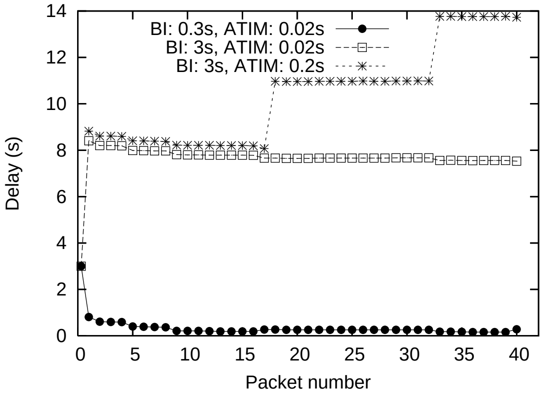

Fig. 12. IEEE 802.11 PSM: TCP traffic trace.

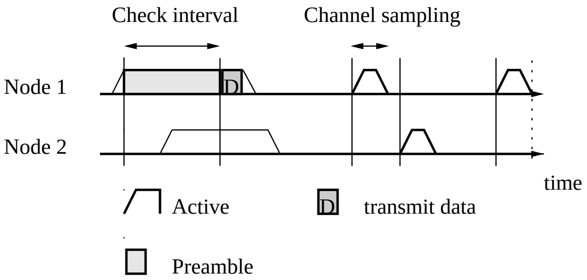

Fig. 13. B-MAC.

classes of asynchronous protocols were developed, which we call *asynchronous scheduled* and *trigger-based*. Asynchronous scheduled protocols solve the synchronization problem, but do not address the problems from potential mismatch between wake-up schedules and traffic patterns. On the other hand, the trigger-based schemes address both.

Asynchronous scheduled protocols [8], [9], [10], [11] operate on the idea that if nodes stay awake long enough, they are guaranteed to overlap with their neighbors. In the first proposed protocols, the receivers made sure that their awake periods cover all possible transmitters [8], [9], [10], [11]. However, later protocols [2], [7] moved the burden to the sender, requiring the transmission of a preamble that is long enough to ensure all possible receivers will hear it. Fig. 13 shows the operation of B-MAC [2], [7], where the receivers sample the channel periodically and stay awake if channel activity is detected. After reception, the node goes back to sleep. To provide reliability, the senders transmit a preamble that matches the length of the check interval of the receiver.

While asynchronous scheduled protocols remove any overhead from maintaining synchronization in the network, a node may spend significantly more time awake than in a synchronous approach. While in the receiver-based approach, nodes switch to idle when there is no traffic present, in the sender-based approach, nodes that are not the intended receivers also remain awake. This results in unnecessary switching and idling costs. Additionally, all current approaches incur more delay than a synchronous approach. Another drawback of these protocols is that broadcast support is only provided if the awake periods of all nodes within transmission range are identical. This is particularly a concern for routing protocols, since they typically discover and maintain routes by broadcasting requests through the network.

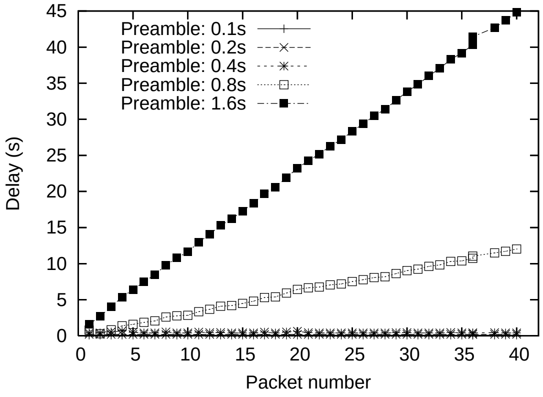

Fig. 14. B-MAC: CBR traffic, packet interval 5 s.

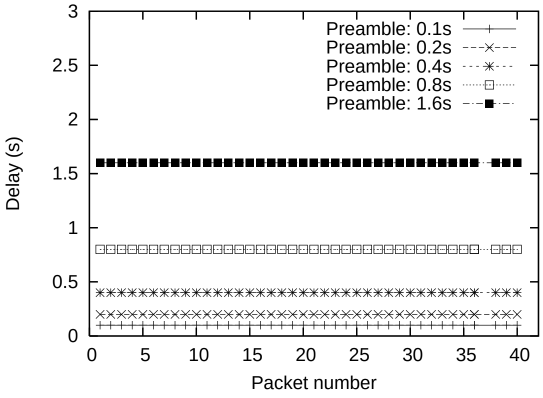

Fig. 15. B-MAC: CBR traffic, packet interval 0.5 s.

We show some of these effects by evaluating B-MAC under different traffic models. In our study, we used check intervals of 0.1 s, 0.2 s, 0.4 s, 0.8 s and 1.6 s, which are the suggested values in [2]. Since there is only one receiver, the preamble size and the check interval are the same. Figs. 14, 15, 16, 17 and 18 depict B-MAC performance for CBR-5, CBR-0.5, OnOff-0.5, OnOff-5 and TCP, respectively. In B-MAC, when the packet interval-arrival times are smaller than the preamble size, the delay increases linearly since each packet needs to wait for the previous packet to be sent (see Figs. 14, 17 and 18). On the other hand, when the interval between packets increases to 5 s for CBR traffic, each packet experiences a fixed delay that is equal to the preamble size. For on-off traffic with a burst-interval of 0.5 s, the linear increase in delay depends on the burst size at each burst interval (see Fig. 16).

Trigger-based protocols notify nodes when they should wake up through the use of a second control channel [3], [4], [26], [27], [28]. To be effective, the control channel must consume less energy than the main channel. Furthermore, the two channels should be orthogonal (*e.g.*, transmitting in the 915MHz [3], [27] or using RFID technology [26] does not interfere with IEEE 802.11). RTS [28] or beacon messages [3],

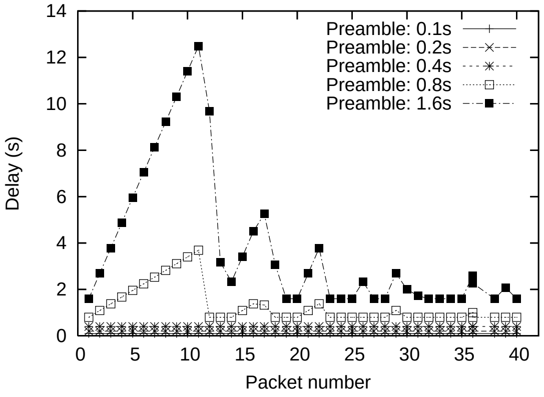

Fig. 16. B-MAC: On Off traffic, burst interval 0.5 s.

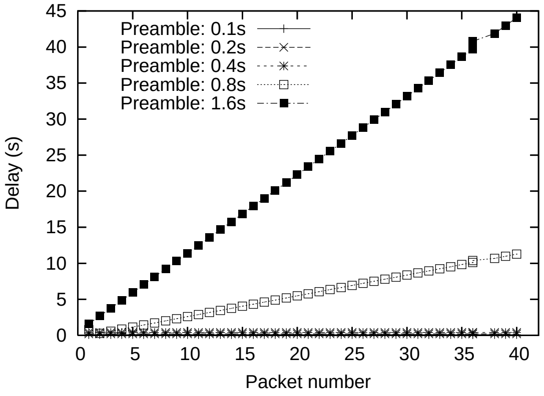

Fig. 17. B-MAC: On Off traffic, burst interval 5 s.

[4] are sent using the control channel to wake up intended receivers, which first respond in the control channel and then turn on their main channel to receive the packet. After the packet transmission has ended, the node turns its radio off in the main channel. The out-of-band signaling used by trigger-based protocols avoids the extra awake time needed by asynchronous scheduled protocols. Additional savings can be achieved on the control channel using any of the synchronous approaches (*e.g.*, STEM [4]). Fig. 19 shows the state transitions of data and wake-up radios in the Wake-on-Wireless protocol [3]. The wake-up radio stays in receive mode to listen to traffic announcements for a fixed duration. If no announcements are heard, it stays asleep until a sleep timeout occurs. If an announcement is heard, the data radio is woken up. The data radio is turned off if the idle timer expires.

The limitations of trigger-based protocols come from the complexity of requiring two radios on one node. Additionally, to save energy, the data radio should not be woken up too frequently since switching from off to idle costs more time and energy than switching from sleep to idle. Finally, there is no guarantee that the data channel is usable even if the wake-up radio can successfully transmit to the receiver, causing the receiving node to wake up and the sending node to try to transmit uselessly. Similarly, a usable data channel is not accessible if the wake-up channel is not usable.

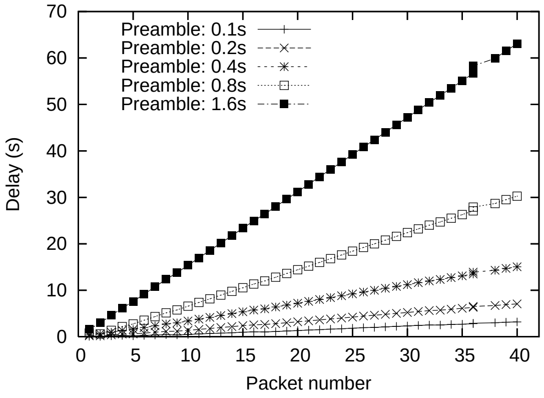

Fig. 18. B-MAC: TCP traffic trace.

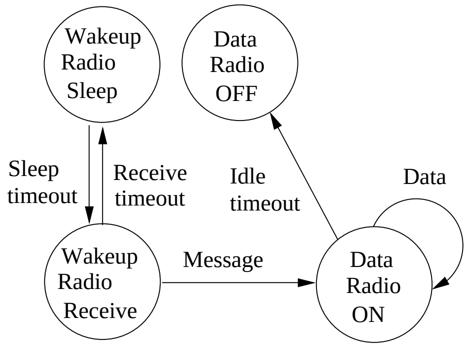

Fig. 19. State transitions in Wake on Wireless.

In our evaluations, we experimented with different data radio wake-up times: 0.5 ms, 5 ms, 50 ms, 500 ms and 5 s. Figs. 20, 21, 22, 23 and 24 depict Wake-on-Wireless performance for CBR-5, CBR-0.5, OnOff-0.5, OnOff-5 and TCP, respectively. While high data radio wake-up time incurs fewer transitions, it incurs high delay (5 s in comparison to 0.5 s). Essentially, when the data radio wake-up time is on the order of the ATIM window of the wake-up radio, we observe less interesting delay patterns (e.g., only wake-up times higher than 500 ms allows reducing the number of transitions). However, when the wake-up time increases to 5 s, we observe similar triangle delay patterns depending on how closely the packets are generated at the sender.

D. The cost of transitions

While power-management protocols save energy from putting nodes to a low-power sleep state, energy is expended for switching the node from sleep to idle. Fig. 25 shows the measurement results for Aironet 350 when the card operates in power-save mode with a beacon interval of 2 s. Although, the wireless interface has nothing to send or receive, it incurs 100 mA for switching from sleep to idle. In Fig. 25, Aironet 350 draws 66 mA on average during the ATIM window for around 13 ms. After receiving a beacon, the interface switches to sleep state and draws 14 mA. Therefore, switching, in addition to device switching costs, affects idling costs depending on the type of power-management protocol. Hence, in this

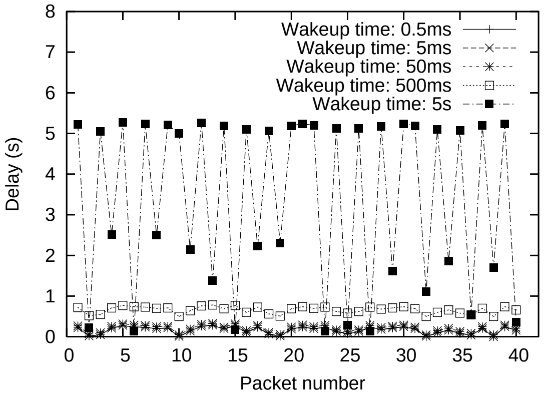

Fig. 20. Wake on Wireless: CBR traffic, packet interval 5 s.

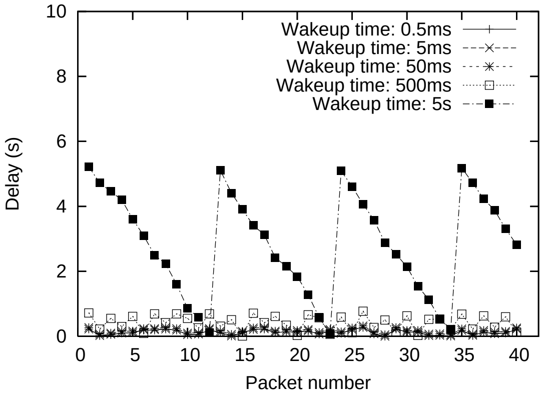

Fig. 21. Wake on Wireless: CBR traffic, packet interval 0.5 s.

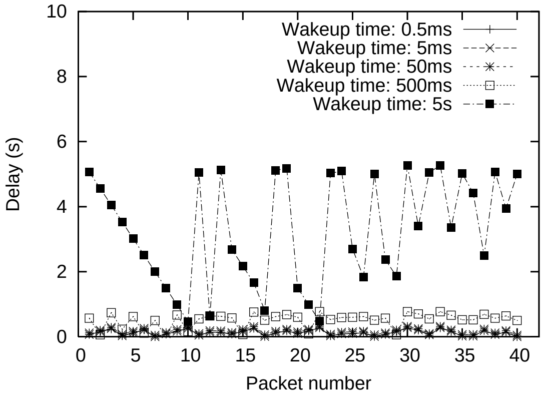

Fig. 22. Wake On Wireless: On Off traffic, burst interval 0.5 s.

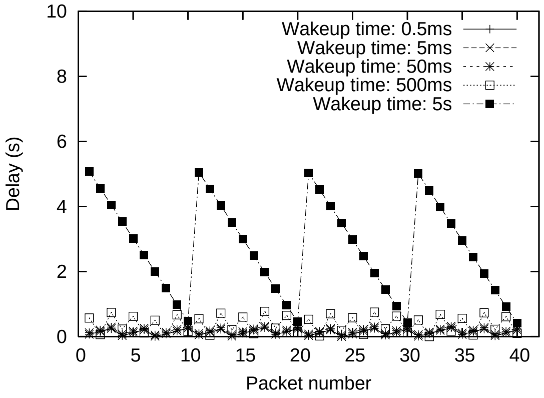

Fig. 23. Wake on Wireless: On Off traffic, burst interval 5 s.

section, we compare IEEE 802.11 PSM, B-MAC and Wake-On-Wireless, in terms of their switching energy consumption based on different protocol parameters.

In IEEE 802.11 PSM, as the beacon interval decreases, the node becomes more responsive to current traffic but incurs high switching costs. For instance, when the beacon interval (BI) is 0.3 s, the average number of transitions is 1597, and when BI is 3 s, the node switches to idle 135 times for all traffic types (see Fig. 26). On the other hand, the Wake-On-Wireless protocol requires fewer transitions for the data radio since the wake-up radio signals the exact time the data radio is needed. However, depending on the power management protocol the wake-up radio uses, it also incurs transition costs. In our model, the wake-up radio followed IEEE 802.11 PSM with a beacon interval of 0.3 s. Furthermore, as the wake-up time of the data radio increases, the number of transitions decreases. For instance, if the data radio can be switched on in 0.5 ms, then the radio makes exactly 40 transitions (*i.e.,* the number of packets). However, if the data radio takes 5 s to switch on (the expected wake-up latency in [3]), then for all traffic types, enough data accumulates to keep the radio awake for an extended time and the number of transitions reduces. For instance, only 4 transitions are necessary for CBR-0.5 and 1 transition for TCP. Finally, once the data radio is on, it idles for a fixed time before switching to sleep (0.2 s in our experiments). Hence, the idling energy cost is strictly tied to the number of transitions. Similarly, in B-MAC, the number of transitions decreases with the check interval size. When the check interval is 0.1 s, the nodes wake up 5000 times, and for a check interval of 1.6 s, the number of transitions is 313. In B-MAC, a node immediately switches to sleep if it senses the channel idle, hence B-MAC has an idling cost on the order of milliseconds. However, if the node senses the channel busy, then it needs to stay awake until it receives the packet. Hence, the idling time can be as long as the preamble size.

Our results show that while different types of power-management protocols address separate issues about reducing idle-time energy consumption, each of these protocols have their own shortcomings in the presence of different traffic types. Therefore, we need to reconsider the design of power-management protocols, which incur less switching and idling costs to accommodate different types of traffic.

IV. TRAFFIC SHAPING TO IMPROVE POWER MANAGEMENT

Our performance study reveals that traffic information has a critical impact on the energy-savings gained through each power-management protocol. Previous research has also looked into using traffic shaping to augment power management. In [12], application-specific information is used to balance energy savings and delay in a Wireless LAN. To avoid

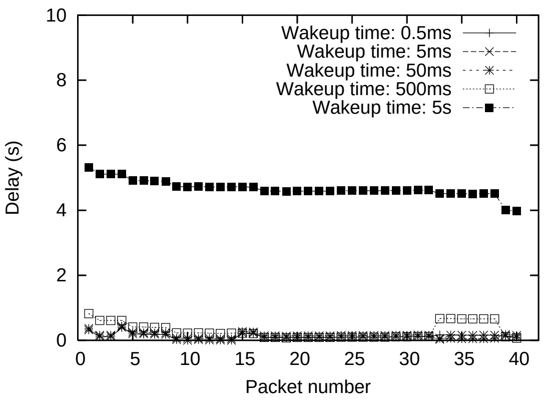

Fig. 24. Wake on Wireless: TCP traffic trace.

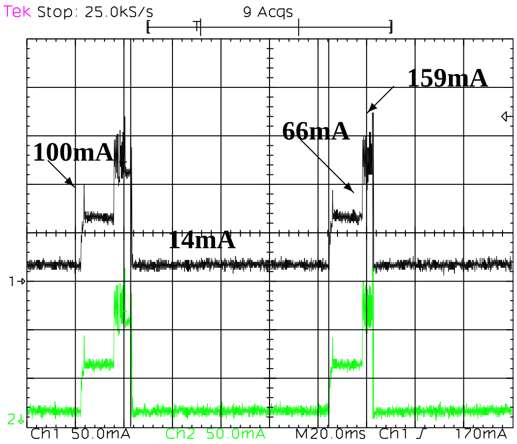

Fig. 25. Measurement of Aironet 350 in PSM: Beacon interval is 2 s.

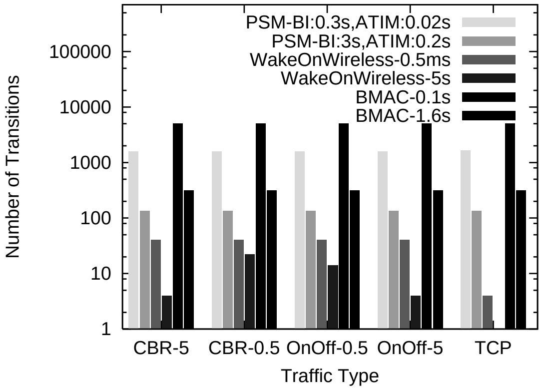

Fig. 26. The number of transitions with different traffic types and power management protocols.

delays from power-management in an ad hoc network, On-Demand Power Management (ODPM) [13] proposes to switch a node to active mode whenever it receives a packet. The node goes back to power-save mode only when it remains idle for a fixed period of time. This approach also reduces the number of sleep-to-idle transitions. However, since the time to switch back to power management mode is fixed, this protocol may incur high idling costs depending on the traffic pattern (*i.e.*, packet inter-arrival time). Similarly, the Bounded-Slowdown (BSD) protocol [14] keeps nodes awake for a protocol-dependent time when they are involved in communication. When there is no network activity, nodes back off and wake up less frequently. However, BSD is best suited for Web-traffic, where small round trip times and short connections are expected. Finally, to adapt power management to observed traffic patterns, [15], [16] propose more explicit traffic shaping, where the network is accessed in bursts to reduce the number of transitions from and to the sleep state. The goal is to buffer packets as long as possible before their deadlines, so that a burst of packets can be sent reducing energy consumption.

However, these approaches are all sender-driven. Although this seems like an obvious choice, since the sender is aware of its own traffic pattern, this model overlooks the fact that often in multi-hop wireless networks, receivers take part in multiple flows from many different senders. The aggregate traffic pattern observed by the receiver is not known by the individual senders and has the highest potential to improve sleep schedules. Therefore, a new receiver-based power-management protocol should be designed to achieve higher energy efficiency.

V. CONCLUSION

Power management protocols aim to save idling energy. However, one challenge to creating efficient power management protocols is matching the sleep schedules to traffic patterns, since poor matches create increased switching costs, needless idling, and long delays. In this paper, we analyze the effects of different traffic generation patterns on the efficiency of various power management schemes. To this end, we present an energy consumption model derived from experimental measurements. We show that each type of power-management protocol, while performing well under certain traffic patterns, performs poorly under others. This leads us to consider protocols that take traffic pattern information into account. However, current approaches to using traffic-shaping to augment power management are sender-driven. We believe a receiver-driven power management protocol has more potential to conserve energy, due to the fact that the aggregate traffic pattern observed by the receiver is not known by any of the individual senders. Such receiver-driven schemes would also avoid the problems associated with synchronized power management schemes. Our future work in power management protocol design includes development of such protocols and their evaluation under many different network scenarios.

VI. ACKNOWLEDGMENTS

The authors would like to thank Jonathan Kimball of the Grainger Center for Electric Machinery and Electromechanics for his help with the Aironet 350 measurements.

REFERENCES

[1] IEEE 802 LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard 802.11, 1999.

[2] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2004.

[3] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: An event driven energy saving strategy for battery operated devices," in *8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, September 2002, pp. 160–171.

[4] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Topology management for sensor networks: Exploiting latency and density," in *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002.

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

[6] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *1st international conference on embedded networked sensor systems (SenSys)*, November 2003.

[7] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2006.

[8] L. M. Feeney, "A QoS aware power save protocol for wireless ad hoc networks," in *Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2002.

[9] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," in *International Conference on Parallel Processing (ICPP)*, 2003.

[10] 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.

[11] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2003.

[12] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, July 2000.

[13] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *IEEE INFOCOM*, March 2003.

[14] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," in *8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, September 2002.

[15] C. Poellabauer and K. Schwan, "Energy-aware traffic shaping for wireless real-time applications," in *10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*, May 2004.

[16] P. J. M. Havinga and G. J. M. Smit, "Energy-efficient wireless networking for multimedia applications," *Wireless Communications and Mobile Computing*, pp. 165–184, Apr-Jun 2001.

[17] C. Sengul and R. Kravets, "Heuristic approaches to energy-efficient network design," in *International Conference on Distributed Computing Systems (ICDCS)*, June 2007.

[18] "Cisco aironet 350 series client adapters data sheet," http://www.cisco.com/en/US/products/hw/wireless/ps4555.

[19] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, July 2001, pp. 85–96.

[20] R. Braynard and C. Ellis, "Asynchronous and asymmetric communication for balancing energy consumption in sensor networks," Duke University, Tech. Rep. CS-2004-11, 2004.

[21] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE INFOCOM*, Apri; 2001, pp. 1548–1557.

[22] K. Jamieson, "Implementation of a power-saving protocol for wireless ad hoc networks," Master's thesis, MIT, 2002.

[23] "Mica2 mote datasheet," http://www.xbow.com.

[24] "Mote hardware," http://research.cs.ncl.ac.uk/astra/motes.htm, 2004, based on an extract from a dissertation written by Geoff Martin.

[25] R. Gass, J. Scott, and C. Diot, "Measurements of in-motion 802.1 networking," Intel Research Technical Report, Tech. Rep. IRC-TR-05-050, 2005. [Online]. Available: http://www.cambridge.intel-research.net/~rgass/papers/tr/irc-tr05050-in%motion.pdf

[26] S. Agarwal, S. V. Krishnamurthy, R. H. Katz, and S. K. Dao, "Distributed power control in ad-hoc wireless networks," in *IEEE Symposium on Personal Indoor and Mobile Radio Communication (PIMRC)*, 2001.

[27] J. Rabaey, J. Ammer, J. L. da Silva Jr., and D. Patel, "PicoRadio: ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes," in *IEEE Computer Society Annual Workshop on VLSI (WVLSI'00)*, 2000.

[28] S. Singh and C. S. Raghavendra, "Pamas: Power aware multi-access protocol with signalling for ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 28, pp. 5–26, July 1998.

[29] "The network simulator-ns2 notes and documentation," http://www.isi.edu/nsnam/ns.