Synchronization vs. Signaling: Energy-Efficient Coordination in WSN

Farhana Ashraf, Riccardo Crepaldi and Robin H. Kravets
University of Illinois at Urbana-Champaign
Email: {fashraf2, rcrepal2, rhk}@illinois.edu

***Abstract*— Synchronization and signaling are two common approaches used for coordination in duty-cycling wireless sensor networks. In this paper we analyze the trade off between these two approaches used for energy-efficient communication. Finally, we present Neighborhood-based Power Management (NPM), a hybrid MAC protocol that minimizes signaling overhead through opportunistically gained knowledge about neighbor wakeup schedules. Unlike the synchronization-based MAC protocols, NPM does not require a priori knowledge of the wakeup schedules. Using only a minimal exchange of schedule information, NPM reduces the signaling overhead by combining adaptive preambling with its neighborhood wakeup mechanism. Our simulations show that NPM outperforms popular B-MAC [1], X-MAC [2] and SCP [3] protocols under all network conditions.**

I. Introduction

Nodes in wireless sensor networks often adopt a periodic wakeup-sleep schedule to reduce the energy wasted during idle listening. Since both the sender and the receiver must be awake during communication, the challenge lies in coordinating their awake schedules. In an ideal scenario, a sender knows exactly when a receiver is awake either through a priori knowledge of or by synchronizing the wakeup schedules. Thus, senders and receivers wake up at the same time, transmit their data, and then go back to sleep. Such approaches require tight clock synchronization. But, real networks have clock drift which makes clock synchronization difficult and expensive to maintain in large sensor deployments [4]. Sensor nodes either have to use orthogonal control protocols to *synchronize* the wakeup schedules, or they have to *signal* their receivers to compensate for the unsynchronized wakeup schedules.

Synchronization-based coordination is achieved by periodically exchanging schedule information between the neighbors [5]–[8]. The cost incurred by a protocol that uses such coordination is determined by two factors: the protocol's ability to piggyback the schedule information onto data messages, and the frequency at which the protocol exchanges its synchronization messages. By periodically exchanging a dedicated beacon message every beacon interval, IEEE 802.11 PSM [5], [9] incurs unnecessarily high synchronization overhead even at small clock drifts and low traffic rates. Protocols such as S-MAC [6] and T-MAC [7] have lower synchronization overhead since they choose their synchronization period based on clock drift. Moreover, S-MAC and T-MAC piggyback their schedule information onto data messages whenever possible. This results in a low overhead at high traffic rates. But, nodes still must send dedicated synchronization messages when the traffic rate is low, causing the overhead to sometimes even dominate the energy saved by following a synchronized periodic wakeup schedule.

Signaling-based coordination eliminates the need for synchronization by allowing the nodes to have asynchronous wakeup schedules. Since nodes are completely unaware of the wakeup schedules of their neighbors, a sender transmits a signal (i.e. preamble) to let the receiver know that there is a transmission ready [1], [2], [10]. In this case, the average length together with the number of the preambles used, determine the overall signaling overhead. B-MAC [1] incurs high signaling overhead since it transmits a fixed length preamble (channel polling interval long) before each data message. Protocols such as X-MAC [2] and SpeckMAC [10] reduce the average signal length by sending short strobed signals directed towards the receiver. The receiver acknowledges the signal to stop the signal transmission and to initiate actual data transmission. Although such an approach reduces the signaling overhead for X-MAC and SpeckMAC, the shortened wakeup signal is achieved at the cost of higher idle listening at low traffic rates. X-MAC further reduces the signaling overhead by allowing any node to send all queued packets to the same receiver using a single signal. However, X-MAC can apply such an approach only when the traffic rate is very high. A sender's unawareness of its receiver's schedule information also limits these protocols' ability to reduce the signaling overhead by shortening their preambles.

Given the dynamic load demand expected in wireless sensor networks, neither of the synchronization-based or the signaling-based coordination provides a complete and effective solution. Thus, the main challenge is to find a balance between signaling and synchronization. A hybrid protocol like SCP [3] assumes loose synchronization between the wakeup schedules of neighbors. The loose synchronization provides two benefits: senders can use shorter signals, and nodes can exchange the schedule information less frequently. However, due to since the periodic exchange of schedule information, SCP faces similar problems as the other synchronization-based coordination protocols like S-MAC and T-MAC.

Our research is based on two observations. First, high synchronization overhead stems from the periodic exchange of schedule information, even when there is no data to transmit. Schedule information can be obtained during active communication through piggy-backed timing information. Use of such *opportunistic scheduling* can achieve much of the benefits of full knowledge of schedules. Second, the main reason for high signaling costs is the inefficient use of the signal. Although, a signal wakes up all nodes that can hear it, only one sender-receiver node pair are allowed to exchange data. All other nodes go back to sleep, even if they have data. Our initial proposal, Neighborhood-based Power Management [11], allowed the signal to wake up and enable *opportunistic sending* for all nodes that hear the signal, amortizing the cost of signaling over multiple transmissions from multiple nodes. We extend our initial proposal and combine both opportunistic scheduling and opportunistic sending to effectively reduce the signaling overhead while incurring only minor synchronization overhead.

Our main contribution is the design of *Neighborhood-based Power Management (NPM)*, an energy-efficient hybrid MAC protocol that balances synchronization and signaling overhead. The novelty of NPM lies in its use of the wakeup signals. Unlike other signal-based approaches, NPM uses signals as neighborhood wakeup tones, enabling communication for all nodes in the neighborhood of the signaling node and amortizing the cost of signaling over multiple transmissions. Moreover, NPM uses opportunistically gained knowledge about the sleep schedules to adapt and ultimately reduce the signal length (i.e., preamble) based on one or a set of destinations. Although we show that the adaptive preambling provides the highest energy savings, the combination of the two mechanisms results in an energy-efficient protocol that supports efficient use of the communication channel with improved communication delay. Our evaluations show that by integrating neighborhood wakeup and adaptive preambling, NPM reduces the signal cost, resulting in higher energy-efficiency than signal-based protocols, B-MAC, X-MAC, and the hybrid protocol SCP. Our targeted approach to adaptive preambling, which only aims to wake up the destination and any other nodes that have similar wakeup times, outperforms SCP at high traffic loads, and B-MAC and X-MAC at all traffic loads, achieving the highest energy-efficiency.

The remainder of the paper is organized as follows. Section II describes the hybrid NPM and presents three policies to optimize signal length and reduce signaling overhead through *opportunistic synchronization*. In Section III, we evaluate NPM in comparison to other related approaches. Finally, Section IV concludes the paper and presents future research directions.

II. Neighborhood-based Power Management

Neighborhood-based Power Management (NPM) is a hybrid protocol that utilizes both signaling and synchronization mechanisms to coordinate nodes before a data exchange. Since nodes in NPM exchange schedule information only when there is data, synchronization cost is low. However, synchronization can drift when the network has low traffic or nodes have high clock drift. To account for such loose synchronization, nodes send wakeup signals (i.e., preambles) before sending their data messages to make sure that the receivers are awake.

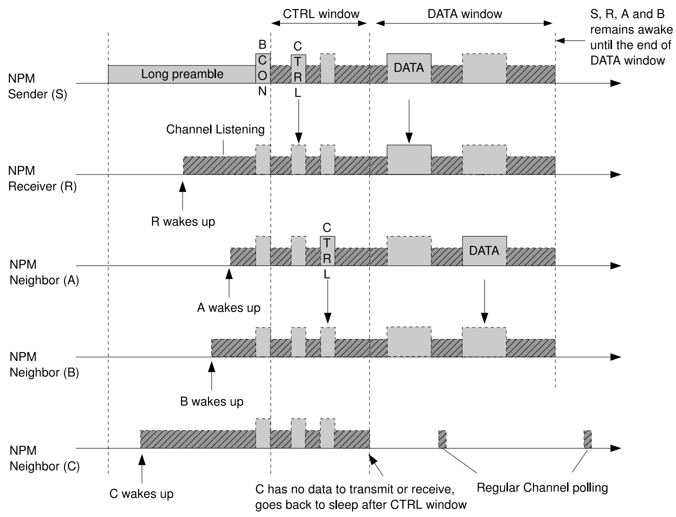

Fig. 1. Neighborhood wakeup with full preamble

However, unlike other signal-based protocols, NPM's preamble enables all nodes in the neighborhood of the signaling node to coordinate and exchange data, amortizing the signaling cost over multiple data transmissions. Nodes further reduce signaling costs by dynamically adapting preamble length using schedule information obtained from the low-overhead loose synchronization mechanism. In this section, we describe the two signaling mechanisms adopted by NPM: neighborhood wakeup and adaptive preambling. We also describe the details of NPM's beacon-based data-driven synchronization.

A. Signaling Mechanisms

All nodes in NPM wake up periodically and poll the channel for activity to receive incoming data messages. Due to the imperfect (out-of-date) synchronization information available to the nodes, NPM must use preambles before the actual data messages, to signal the receivers that they must stay awake until they receive the data messages. Since the control overhead due to signaling can be high in a network with unsynchronized wakeup-sleep schedules, NPM integrates two mechanisms to reduce the cost of signaling. While neighborhood wakeup reduces the total number of preambles required to send the data messages by allowing multiple data transmissions per preamble, adaptive preambling reduces the length of each preamble by utilizing the available synchronization information about the neighborhood.

1) Neighborhood Wakeup: When nodes in NPM detect a preamble during their periodic wakeup, they remain awake until the end of the preamble to receive and transmit data, if they have any. In this way, a single preamble could serve for multiple communications within the same neighborhood.

NPM can achieve lower signaling overhead since with one preamble it can send multiple packets to the same receiver or to different ones. It is also possible that the preamble wakes up both the sender and the receiver for a different flow, which can then initiate communication without the preambling phase.

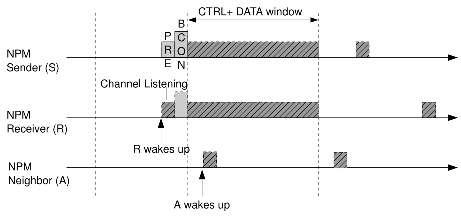

Fig. 2. NPM-Targeted Approach

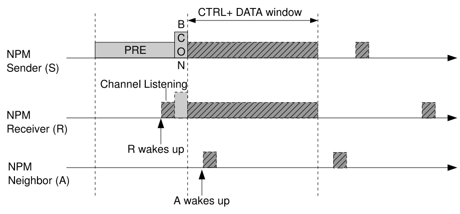

Fig. 3. NPM-Finish Early Approach

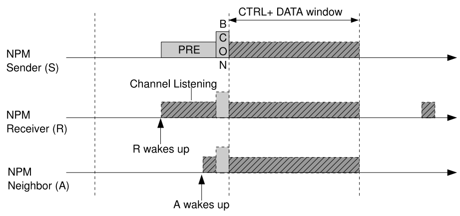

Fig. 4. NPM-Coverage Approach

NPM divides communication time into three components, the signal (i.e., preamble), the control window and the data window, which all together form a *transmission phase* (see Figure 1). After detecting a preamble, all nodes in the neighborhood of the preambling node coordinate to complete their data transmissions. Therefore, senders exchange *signal messages* with their intended receivers during the control window (similar to the ATIM window of IEEE 802.11 PSM). Nodes that have no data to send or receive, or cannot reach any nodes they have packets for, quickly go back to sleep after the control window.

All source-destination pairs complete their data transmissions using CSMA/CA in the data window. To support short term bursts of traffic, the nodes remain awake through the entire data window enabling NPM to opportunistically send newly generated data within the same data window without additional signaling. Transmissions that fail during opportunistic sending are rescheduled in the next transmission phase.

2) Adaptive Preambling: Nodes in NPM adopting the basic neighborhood wakeup mechanism always send full length preambles (see Figure 1). We refer to this protocol as NPM *full*. The *adaptive preambling* component of NPM utilizes the available loose synchronization information (described in section II-B) to dynamically adapt the length of the preambles.

To account for clock drift, nodes add a guard time to both ends of their preambles. Nodes only reduce the length of their preambles when they have recent synchronization information about their receivers. However, when the available synchronization information is out-of-date, nodes must transmit full length preambles. The adaptive preamble mechanism is also never applied to broadcast messages since there is no guarantee to wake up all nodes in the neighborhood if the preamble length is shorter than the nodes' sampling interval.

NPM supports three different approaches for adapting the length of the preambles:

- NPM *targeted* aims to wake up only one receiver. Since nodes transmit preambles only during the expected wakeup times of their receivers, NPM *targeted* achieves the shortest preambles (see Figure 2).
- NPM *finish early* aims to improve delay performance while utilizing the neighborhood wakeup mechanism. Thus, nodes stop transmitting preambles as soon as their receivers wake up, similar to X-MAC (see Figure 3).
- NPM *coverage* aims to fully utilize the neighborhood wakeup mechanism, while still reducing the length of the preambles. Preambles are long enough to wake up all known nodes in the neighborhood (see Figure 4).

By reducing the length of the preambles, adaptive preambling decreases the potential use of the neighborhood wakeup and opportunistic sending components of NPM. Essentially, the shortened preambles in NPM *targeted* wake up fewer nodes, increasing the total number of preambles required to complete all data transmissions. On the other hand, NPM *coverage*, which benefits the most from the neighborhood wakeup, has the longest preambles. The choice of the approach that most suits the user's needs is the results of a trade off between energy efficiency and latency, and will be thoroughly explored in Section III.

B. Synchronization Mechanisms

NPM uses beacon-based synchronization to obtain the wakeup-sleep schedule information. Instead of exchanging schedule information periodically (like S-MAC, SCP and IEEE 802.11 PSM) and spending a fixed overhead for synchronization, nodes in NPM exchange their schedule information only when there is data to send. This data-driven synchronization approach saves NPM from wasting energy on synchronization when there is no traffic in the network.

Schedule information (i.e., synchronization) is exchanged via two types of control messages: beacons and signal messages. Instead of sending a beacon at the beginning of each wakeup period and thus incurring a fixed synchronization overhead like IEEE 802.11 PSM, NPM nodes send one beacon message per transmission phase. The preambling node broadcasts the beacon message just after sending the preamble (see Figure 1) and thus provides its schedule information to all awakened neighbors. Thanks to neighborhood wakeup, the synchronization overhead due to beaconing is amortized over multiple data transmissions. Nodes in NPM also piggyback their schedule information when they reply to the signal messages during the control window. Although the synchronization overhead due to signal acks increases as more data messages are transmitted, the size of the overhead is very small (8 bytes for the schedule information).

Synchronization information is sent as the time difference between the local time of the sender and the time when the sender will wakeup for the next cycle according to its periodic wakeup schedule. Thus, the schedule information is not affected by the clock skews of the different nodes in the neighborhood. Each node keeps track of the synchronization information of each of its neighbors by maintaining a *schedule*

TABLE I

SIMULATION PARAMETERS

Duty cycle (awake/asleep):	
NPM, B-MAC, SCP	1ms/99ms
X-MAC	2ms/98ms
IEEE 802.11 PSM	100ms/600ms
NPM:	
(CTRL, DATA window)	(100ms, 600ms)

table. Each entry in the schedule table contains two pieces of information: the time difference between the neighbor's schedule and the local node's own wakeup schedule, and the age of the entry. Since synchronization information is not exchanged periodically, this information can become stale. The age information is used to purge stale data.

III. EVALUATION

In our evaluation, we show that by integrating *opportunistic synchronization* to *signaling*, NPM can achieve energy-efficient communication. To characterize the protocol energy-efficiency, we analyzed the contribution of both signaling and synchronization to the total energy consumption.

As a baseline for our evaluation, we compare NPM to signaling-based protocols: B-MAC and X-MAC, a hybrid protocol: SCP , and a synchronization-based protocol: IEEE 802.11 PSM. We compare NPM to PSM to understand how NPM performs in a realistic unsynchronized network compared to PSM which is designed for a perfectly synchronized network and not implementable in a realistic networks with clock drift. To verify the effectiveness of the neighborhood wakeup, we compared NPM *full* to its dual B-MAC, and NPM *finish early* to its dual X-MAC. The detailed comparison between the NPM approaches, and the comparison between NPM *targeted* and the hybrid protocol SCP revealed the combined effect of neighborhood wakeup and adaptive preambling.

Network and Traffic Scenario

We evaluated the protocol behavior using *ns-2* in a square grid network with 100 nodes. To analyze the effect of different neighborhood size, we varied the spacing among the nodes to 200m, 140m and 75m. With a radio coverage range of 250m, these node distances created a neighborhood of size 4, 8 and 18 nodes respectively. Since we obtained similar trends for the different node densities, we present results for only 140m.

Nodes in the network follow random and unsynchronized wakeup times for all protocols except IEEE 802.11 PSM and SCP. PSM and SCP were always run with synchronized wakeup schedules. Table I summarizes the duty cycling parameters for each protocol. To make the simulations more realistic, all protocols (except IEEE 802.11 PSM) added random clock drifts (maximum drift 100 ppm = 100s drift per 1 million s) to each node. The two hybrid protocols, NPM and SCP added a 1ms guard time to its preambles to handle the clock drift. Moreover, NPM refreshed the schedule information inside the neighbor table every 60s.

We analyzed the protocols with CBR traffic at different traffic generation rates. We varied the CBR inter-arrival time

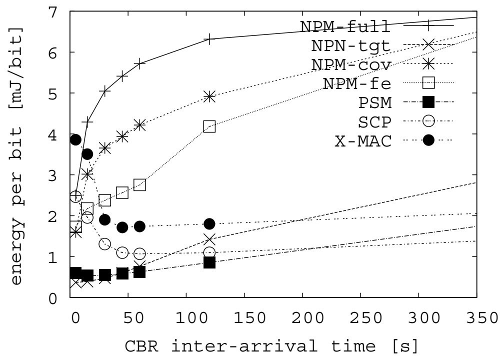

Fig. 5. Energy per bit

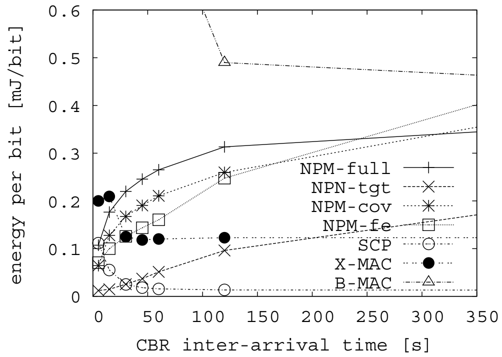

Fig. 6. Signaling Energy per bit

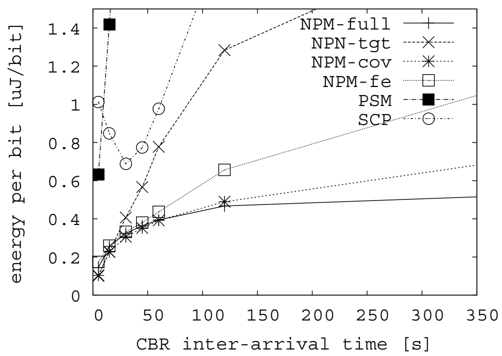

Fig. 7. Synchronization Energy per bit

from a 600sec for low load to a 5sec for high load. All data packets (size 80 bytes) were destined to a sink (transmit data rate of 250 Kbps [12]), which was placed in the distant corner of the grid. All protocols used shortest hop routing.

All results are obtained from an average of 10 simulation runs. For each run, all protocols were simulated for 1 hour.

A. *Simulation Results*

We evaluated the effect of *synchronization* and *signaling* in reducing the energy consumption during coordination. While synchronization overhead is determined by the frequency at which the synchronization information is exchanged, signaling overhead is determined by the length of the signals and the number of signals required to send all data messages.

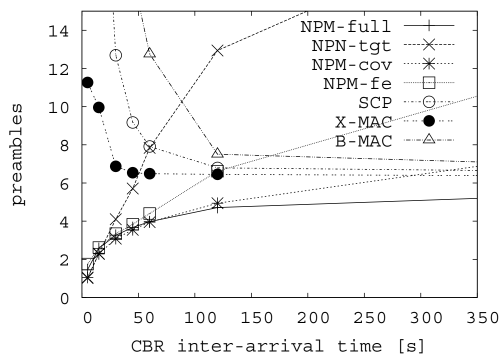

Fig. 8. Preamble count per data

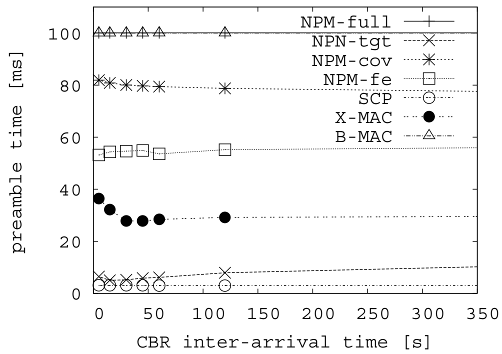

Fig. 9. Average Preamble Length

In order to capture the the contributions of both data transmission and coordination overhead to the total energy consumption, we used *energy per bit* as a metric. Any protocol that incurs a low control overhead, will have a low energy per bit value. For computing the total energy, we determined the time each protocol spends in the different radio states (transmit, receive, idle and sleep). For our evaluation, we used the power profile of MICAz radio [12] (current drawn in transmit mode = 17.4mA, receive mode = 18.8mA, idle mode = 20uA, and sleep mode = 0.1uA) with an external power source of 3.0V.

In this section, at first we compare the energy profiles of the different NPM approaches to evaluate the effectiveness of the two NPM signaling mechanisms: neighborhood wakeup and adaptive preambling. Next, we compare the NPM protocols with other base protocols to show the effectiveness of signaling and synchronization-based coordination mechanisms in reducing overall energy consumption.

1) Comparison between NPM approaches: We used *number of preambles per data message* and *average length of preambles* as metrics to quantify the effectiveness of neighborhood wakeup and adaptive preambling at varying load condition. An effective adaptive preambling approach will make the preamble shorter. On the other hand, an effective neighborhood wakeup will allow more transmissions using a single preamble, requiring fewer preambles to send all data messages.

The average length of preambles used by the NPM approaches solely depend on the distribution of the wakeup schedules of the neighbors. During our simulation, in a network with randomly unsynchronized wakeup schedules, we obtained an average preamble length of 100ms, 80ms, 50ms and 5ms for NPM *full*, *coverage*, *finish early* and *targeted* respectively (see figure 9). The effectiveness of adaptive preambling approach does not vary with traffic load, causing NPM *targeted* to be the most effective at all rates.

The effectiveness of neighborhood wakeup however depends on the network traffic load. An effective neighborhood wakeup will ensure better utilization of the NPM DATA window. At high traffic rates (i.e. when the CBR inter-arrival is low), all NPM approaches can effectively utilize their entire DATA window. Thus, all NPM approaches send the same number of data messages per preamble (see figure 8). However, as the rate decreases (i.e. CBR inter-arrival time increases), the opportunity for effectively utilizing the DATA window decreases. The effectiveness of neighborhood wakeup now depends on the protocol's ability to wakeup more nodes. Since a longer preamble has a higher probability to wake up more nodes, the effectiveness of neighborhood wakeup at low rates follows a reverse trend to the effectiveness of the adaptive preambling (see figure 8, 9). Thus, NPM *targeted*, having extremely short preambles, wakes up very few nodes and has the least effective neighborhood wakeup at low rates.

We have analyzed the contributions of neighborhood wakeup and adaptive preambling on the overall signaling overhead at varying traffic load condition. Since the effectiveness of adaptive preambling does not depend on traffic load, the overall signaling overhead is determined by the effectiveness of the neighborhood wakeup mechanism. Thus, at high rates, NPM *full* has the highest signaling overhead over all NPM approaches, and NPM *targeted* has the lowest (see figure 6). At low rates, NPM *coverage* and NPM *finish early* have increasingly higher signaling overhead than NPM *full*. NPM *targeted*, although it has the most inefficient neighborhood wakeup at low rates, incurs the lowest signaling overhead of all NPM approaches due to its extremely short preambles.

The low signaling overhead of NPM is achieved at the cost of synchronization overhead. The schedule information piggybacked onto data messages results in a fixed synchronization cost. However, the overhead due to beacon message exchange is incurred after each preamble. Thus, an NPM approach incurs more synchronization overhead when the nodes send more preambles. We can verify this by observing the similarity between the synchronization overhead and the number of preambles per data messages trends. NPM *targeted* incurs the most overhead, whereas NPM *full* pays the least synchronization overhead.

The signaling overhead and synchronization overhead of NPM follows completely inverse trend. However, due to the data-driven synchronization in NPM, the synchronization overhead is very low (in the range of uJ/bit, see figure 7) compared to signaling overhead. Thus, the energy performance of NPM (see figure 5) is determined by the signaling overhead, resulting in NPM *targeted* to be the most energy-efficient at all traffic rates.

2) Comparison with base protocols: We compare the energy profiles of different protocols to show that by effectively combining opportunistic synchronization with signaling, NPM can achieve low overhead coordination at varying traffic loads.

As the traffic rate increases, protocols follow two interesting trends with respect to the total energy per bit (see figures 5). Energy for the protocols that allow neighborhood wakeup, such as PSM and NPM, either remain stable or have a decreasing trend with higher rates. For the rest of the protocols, B-MAC, X-MAC and SCP, energy consumption increases as the rate increases. To understand these trends, we analyzed the signaling and the synchronization overhead incurred by each of these protocols (see figures 6 and 7).

Signaling overhead is the only component of B-MAC and X-MAC's control overhead. At the low and moderate traffic rates, due to the one preamble per packet policy, both of these protocols consume fixed signaling overhead per bit at low rates. However, as the network gets overloaded, more packets get dropped due to collision and contention, increasing the energy consumed per received data bit (we omit B-MAC from the total energy per bit graphs due to its very high and skewed values relative to the other MAC protocols). The strobed preambles of X-MAC allow it to avoid overhearing and to have shorter preambles, resulting in much lower signaling overhead than B-MAC. For the same reason, the increasing trend of X-MAC starts at a much higher traffic rate than it does for B-MAC.

Although NPM uses a combination of both signaling and synchronization mechanisms, it is the signaling overhead that shapes the total energy per bit of NPM (explained in section III-A.1). The high energy efficiency of NPM *full* compared to its dual B-MAC for all traffic conditions, shows the effectiveness of neighborhood wakeup in reducing total energy consumption. However, the efficiency of NPM *finish early* with respect to its dual X-MAC is dependent on both the type and the rate of traffic. NPM *finish early* consumes less energy than X-MAC only when the effectiveness of NPM's neighborhood wakeup outperforms the benefit of X-MAC's very low signaling overhead. Achieving this goal becomes even more difficult at extremely low rates. Thus, despite the neighborhood wakeup, NPM *finish early* consumes higher energy per bit for CBR traffic until the rate becomes sufficiently high. NPM *targeted*, however, by using very low overhead for signaling, consumes much less energy than X-MAC for almost all traffic rates except for extremely low rates. At extremely low rates, NPM *targeted* has very few opportunities to use its neighborhood wakeup, causing *targeted* to consume higher energy than X-MAC.

SCP's low total energy per bit stems from its very low signaling overhead. SCP achieves this by spending more energy in synchronization. At low rates, SCP does better than the NPM *targeted*. But, as the rate increases, NPM's neighborhood wakeup amortizes the cost of both of its signaling and synchronization over multiple transmissions, causing NPM *targeted* to consume less energy than SCP.

NPM *targeted*, which is the best in terms of energy among all NPM approaches, reaches very close to PSM in a perfectly synchronized network (we consider no control overhead for PSM due to explicit synchronization). For higher rates, NPM *targeted* performs even better than PSM. This proves the effectiveness of opportunistic synchronization and opportunistic sending in achieving high energy efficiency.

IV. Conclusion

This paper presents a hybrid protocol for energy-efficient communication in wireless sensor networks, called Neighborhood-based Power Management (NPM). By combining *synchronization* and *signaling* effectively, NPM reduces energy consumption across a large variety of network conditions. The *data-driven synchronization* mechanism in NPM allows nodes to opportunistically gain schedule information about their neighbors. Using this information, NPM reduces the signaling cost by applying adaptive preambling. Moreover, by strategically using the wakeup signal as a neighborhood wakeup tone, and thus allowing all awake neighbors to *opportunistically send* using that signal, NPM further reduces signaling costs by amortizing it over multiple nodes and multiple transmissions.

Our current prototype chooses the *preambling node* randomly from the multiple nodes in the neighborhood that have data to send. However, a more informed selection of the preambling node may improve the system throughput and fairness. We will extend our NPM protocol to include a distributed local algorithm for choosing preambling nodes. Our future works also include developing an analytical model for NPM and investigating the protocol behavior more thoroughly.

References

[1] J. Polastre, J. L. Hill, and D. E. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys 2004,*. ACM, 2004.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *SenSys*, 2006.

[3] W. Ye, F. Silva, and J. S. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *SenSys*, 2006.

[4] K. Romer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," *Handbook of Sensor Networks: Algorithms and Architectures*, 2005.

[5] B. O'Hara and A. Petrick, *The IEEE 802.11 handbook: A designer's companion*. IEEE Computer Society Press, 2004.

[6] Y. Wei, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks." IEEE, 2002.

[7] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *SenSys03*, 2003.

[8] Y. Sun, S. Du, O. Gurewitz, and D. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, 2008.

[9] C. Hu, R. Zheng, J. Hou, and L. Sha, "A microscopic study of power management in IEEE 802.11 wireless networks," *International Journal of Wireless and Mobile Computing*, 2006.

[10] K.-J. Wong and D. K. Arvind, "SpeckMAC: Low-power decentralised MAC protocols for low data rate transmissions in specknets," in *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, New York, NY, USA, 2006.

[11] F. Ashraf and R. Kravets, "Poster abstract: neighborhood-based power management," *Mobile Computing and Communications Review*, 2009.

[12] M. Datasheet, "Crossbow Technology Inc," *San Jose, California*, 2006.