

Energy Analysis of Four Wireless Sensor Network MAC Protocols

Brian Bates, Andrew Keating, Robert Kinicki
Worcester Polytechnic Institute

Abstract—Using the MAC Layer Architecture framework, four power-aware MAC protocols were implemented in TinyOS on TelosB motes. Indoor energy measurements over a single-hop network topology were conducted using three wireless sensor network traffic patterns (broadcast, convergecast and local gossip). AS-MAC was the most energy-efficient for convergecast and local gossip traffic, while SCP-MAC was best for broadcast traffic. Over the three traffic patterns, BAS-MAC used slightly less energy than Crankshaft and BAS-MAC significantly outperformed AS-MAC in the broadcast traffic scenario.

I. INTRODUCTION

Wireless sensor motes often run unattended on battery power for long periods. As radio usage consumes the majority of mote energy [1], developing medium access control (MAC) protocols which reduce radio energy consumption is important for wireless sensor networks (WSNs).

The most prevalent sources of energy waste in WSN radio communication are idle listening, overhearing and transmission collisions [2]. Since idle listening (i.e., listening to a wireless channel while no transmissions occur) occurs frequently in Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocols, carefully selecting when to receive can significantly improve network lifetime in a WSN. When a mote overhears a transmission intended for another node, it wastes receive-state energy. With the advent of packet-based radios where an entire packet must be received before its recipient header field is inspected, innovative overhearing avoidance strategies are needed. When simultaneous transmissions on the same channel collide, either extra mote energy is expended on MAC layer retries or without MAC layer ACKs, the transmissions will be lost unless higher layer ACKs are activated.

Early WSN MAC protocols used duty cycling [2], [3] to minimize energy waste by placing the radio in a low-power sleep state when not sending or receiving transmissions. The next generation of protocols introduced low-power listening (LPL) [4]–[6] with transmission preambles and channel polling to significantly reduce idle listening. Current state-of-the-art WSN MAC protocols consist of hybrid solutions which employ variations of these two techniques [7]–[9].

Comparing the energy-saving properties of published WSN MAC protocols is difficult due to varying network sizes, topologies and offered loads. Furthermore, these evaluations use a variety of simulators and few perform experiments on the same physical sensor hardware. Since these published experiments often involve intertwined factors such as multi-hop routing, sensitive parameter settings and hardware-specific

intricacies, extracting clear cause-and-effect performance relationships is nontrivial.

This paper provides an unbiased evaluation of energy consumption for four power-aware MAC protocols (AS-MAC, BAS-MAC, Crankshaft and SPC-MAC) using three standard WSN traffic patterns. While no one protocol is optimal under all circumstances, the measurements provide insight into general strategies for reducing energy consumption in future WSN MAC protocols.

II. RELATED WORK

Scheduled Channel Polling MAC (SCP-MAC) modifies LPL by having all neighboring wireless motes wake up to listen at the same time. This yields shorter message preambles and shorter duty cycles than LPL protocols such as the commonly-deployed BMAC [7]. However, since all neighboring SCP-MAC sensors share the same listening slot, overhearing occurs frequently.

Asynchronous Scheduled MAC (AS-MAC) [8] eliminates overhearing by assigning motes unique listening time slots. Synchronization information in each mote's Neighbor Table determines when a mote wakes up to either transmit or receive a packet. Waking periodically at its assigned time slot, each AS-MAC mote polls for packet receptions. Motes transmit in the transmission contention window that overlaps with the recipient's wakeup slot. If a mote loses contention, it retries during the recipient's next wakeup slot.

Crankshaft [9] also reduces overhearing in dense sensor networks by dividing time into frames which are sub-divided into receiver slots. Crankshaft frames include broadcast and unicast slots such that every neighboring mote wakes up for all broadcast slots and its own unicast slot, which is assigned based on the mote's physical address. The original Crankshaft protocol included MAC layer ACKs. While AS-MAC and Crankshaft both stagger receiver wakeup times, the two protocols are different. The Crankshaft ratio of broadcast slots to unicast slots per frame is configurable at compile time, and the number of unicast slots per frame is independent of the number of motes in the neighborhood. Thus in dense mote networks, multiple receivers will wake up and contend for receptions during the same slot. While Crankshaft relies on upper layers to manage clock synchronization, AS-MAC uses synchronization packets to align mote clocks and uses non-uniform offsets to provide unique receiver reception slots even in dense neighborhoods.

The MAC Layer Architecture (MLA) [10] is a framework for optimized and reusable components to be leveraged across multiple MAC protocols. With hardware abstraction components, MLA provides hardware-dependent services such as alarming, local time and radio core access and hardware-independent services such as channel polling, preamble sending, time synchronization, slot handlers and low-level dispatchers. MLA-based MAC protocols have clean interfaces that are easily swapped in or out of a TinyOS configuration. Thus, MLA was used to compare the same application running on four MAC protocols.

III. METHODOLOGY

This research used MLA to implement four power-aware MAC protocols (SCP-MAC, AS-MAC, Crankshaft and Broadcastable AS-MAC (BAS-MAC)) on TelosB motes running TinyOS 2.1.0 and analyzed the energy consumption of each protocol under three distinct WSN traffic patterns: local gossip, convergecast and broadcast. By modifying the CC2420 radio drivers and using an oscilloscope, an energy profile of the TelosB radio states was developed. A TinyOS component that recorded the amount of time the radio spends in each state was created to measure energy consumption. This section describes our energy measurement techniques, the implementation details of the four protocols, and our experimental methodology.

A. Energy Measurement

WSN energy usage experiments require accurate measurement of energy consumed by a sensor over time. This energy expenditure depends on the radio's state distribution because individual radio states consume differing amounts of energy per unit time. Our approach to measuring energy is to record the cumulative time spent by a mote in each state and multiply this time by a fixed constant representing the power consumed in that state per unit of time. Gathering TelosB mote energy data required a new TinyOS component, the RadioRecorder, that hooks into the radio drivers provided by MLA and measures time spent in each radio state.

To determine the power consumed in each individual radio state, a mote running the AS-MAC protocol was connected to an oscilloscope. Since power consumption in a particular radio state is a function of the hardware and independent of the MAC protocol, it was sufficient to gather all the oscilloscope readings using just AS-MAC. Applying Ohm's law, the current flowing through the circuit was calculated. To measure power, RadioRecorder outputs the mote's radio state via three of the TelosB's GPIO pins. Since the oscilloscope displays digital and analog readings concurrently, the current drawn could be matched to distinct radio states. Both the current drawn by the microprocessor and the radio is used in forming the results because the processor is active during radio operations. The only scenario where the radio is on and the processor is idle occurs during long idle periods, but this situation does not occur in the protocols investigated.

Figure 1 shows a sample measurement from the oscilloscope, taken during the wakeup tone of AS-MAC. The three

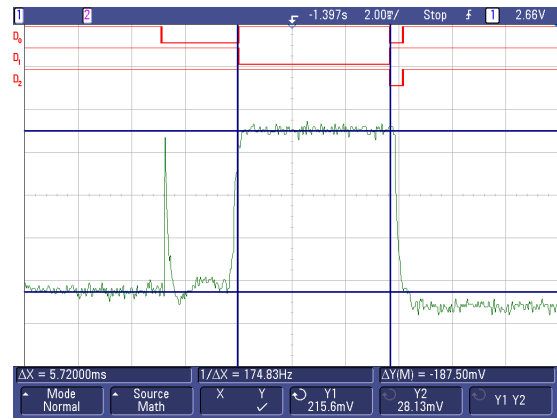


Fig. 1. Oscilloscope Snapshot During an AS-MAC Wakeup Tone.

lines across the top of the figure, labeled D0, D1 and D2, are digital output signals used to identify a particular radio state. This figure shows that the radio is in the idle state for the wakeup tone for approximately 5.7 milliseconds, with a voltage difference of 187.5 mV.

Table I summarizes the findings from the oscilloscope measurements. The stopped state refers to times when the radio is inactive. The starting and stopping states occur when the radio is warming up to begin listening or sending, and when the radio is powering down, respectively. In the idle state the radio is fully powered but not actively sending or receiving a message. In the receiving state the radio is actively receiving a message, and in the sending state the radio is actively sending a message. Table I also compares our findings with those in the CC2420 datasheet [11].

TABLE I
TELOS B RADIO STATE CURRENT DRAW

Radio State	Observed Value	CC2420 Datasheet Value
Stopped	0 mA	20 μ A
Starting	2.51 mA	N/A
Idle	20.07 mA	18.8 mA
Sending	17.01 mA	17.4 mA
Receiving	20.22 mA	18.8 mA
Stopping	10.04 mA	N/A

The idle and receiving state energy measurements are slightly higher than indicated in the datasheet column due to additional power consumed by the MSP430 microprocessor. These measurements determine the power spent while starting and stopping the radio and validate the CC2420 datasheet's accuracy.

B. Protocol Implementation

For a consistent comparison of the four protocols, each protocol was implemented using TinyOS 2.1.0 and the MLA framework. To provide the fairest energy consumption comparison, minor modifications to the original protocol designs were deemed necessary.

1) *AS-MAC*: To keep the four protocols on a level playing field, the AS-MAC implementation does not include Hello messages. Instead, the motes only perform non-Hello wakeups. Moreover, due to AS-MAC’s inherently slow start-up times, initialization and synchronization were standardized across all protocols and experiments using a static initialization technique. Through static initialization, each AS-MAC mote’s Neighbor Table was pre-populated at compile time.

An additional layer was wired on top of the sending component to retro-fit broadcasting into AS-MAC. This layer recognized recipient broadcast addresses and sent individual unicast messages to each entry in the Neighbor Table.

2) *Crankshaft*: The original Crankshaft protocol uses the Sift distribution [12] for channel contention, weighting senders with an uneven probability to favor later contention slots. This reduces the probability of two senders selecting the same contention slot in a network with heavy traffic to a particular mote. To reduce complexity, our implementation uses a uniform distribution to determine contention slots. Since our experiments mitigate sender collisions by staggering transmission times, using the uniform distribution does not place Crankshaft at a disadvantage.

Since AS-MAC and SCP-MAC do not support MAC layer ACKs, Crankshaft was implemented without frame acknowledgments. Comparing energy usage for a protocol that uses ACKs to one that does not is unfair because the lack of an ACK signals a frame retransmission which trades increased energy consumption for increased reliability.

Lastly, the wakeup behavior of our base station (BS) node is different from the original Crankshaft implementation. Crankshaft was designed to have the BS listen during all unicast and broadcast slots. Its authors are assuming that, as in many WSN deployments, the base station is connected to a permanent power source. Noting that this is not always the case, we eliminate this expensive BS behavior in our Crankshaft implementation.

3) *SCP-MAC*: Our SCP-MAC implementation does not include adaptive channel polling, as this investigation does not consider multiple hop network topologies. Additionally, SCP-MAC’s two-phase contention resolution was not implemented because the staggering of mote transmissions avoids most of the transmission contention in our experiments. Finally, with the short duration of our experiments, SCP-MAC’s SYNC packets were not implemented because clock drift was not an issue.

C. BAS-MAC Design

While adding broadcast functionality to AS-MAC, it became clear that broadcasting would cause its energy usage, latency and throughput to spiral out of control in large networks. As no two AS-MAC receivers wake up in the same slot, broadcasting was emulated via multiple unicast transmissions. For a mote to broadcast in AS-MAC, it must stay awake for the individual wakeup time of every other mote in the network and possibly contend with other senders in the process. This

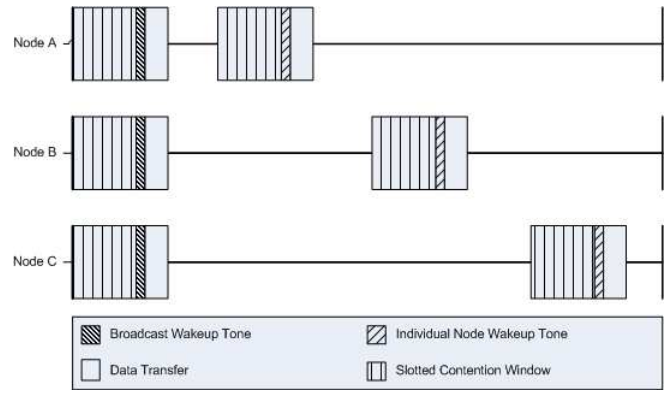


Fig. 2. BAS-MAC Scheduled Wakeups

observation led to the development of Broadcastable AS-MAC (BAS-MAC).

As in AS-MAC, each BAS-MAC mote has its unique scheduled wakeup interval and offset. The major innovation is the addition of a broadcast interval which defines a time slot when all neighboring motes will wake up simultaneously. Figure 2 depicts a wakeup cycle for three BAS-MAC motes in which all motes wake up for the synched broadcast slot and each mote also wakes up individually for its unicast slot.

While BAS-MAC ‘borrows’ the broadcast slot concept from Crankshaft, by inheriting AS-MAC’s offset-based implementation of receiver slots, the intent is for BAS-MAC to be more flexible than Crankshaft’s concept of fixed broadcast slot ratios within frames. The BAS-MAC broadcast wakeup frequency can be set higher, equal to, or lower than the individual mote wakeup frequency depending on the importance of broadcasting in WSN deployment.

D. Parameter Selection

Each of the four protocols have unique parameters which affect their behavior, and altering parameter settings can yield significant differences in energy consumption. Our configured parameters aim to minimize any biases they might introduce.

The four protocols share several parameters, but configuring these parameters was difficult because each protocol was designed with differing network topologies and tasks in mind. Our four implementations and parameter settings focused on standardizing wakeup intervals, tone and contention check lengths, contention window sizes and behaviors, and initialization mechanisms.

The concept of a wakeup interval, the amount of time to complete one repeating cycle of the protocol, is found in all four protocols. In AS-MAC, this is the amount of time for all individual motes to wake up once. In SCP-MAC, it is the time between two scheduled wakeup tones. For Crankshaft, this is a single frame. In BAS-MAC, we define the wakeup interval as the length of time for all individual motes to wake up plus the time required for a broadcast wakeup. To ensure standardization, the wakeup interval is fixed across all protocols. This means that in SCP-MAC, there is one scheduled wakeup per interval. For AS-MAC and BAS-MAC,

the number of unicast slots varies with the size of the mote neighborhood. This relationship between the number of slots and motes holds for Crankshaft when the mote neighborhood is sparse. Thus for our ten-mote experiments, these three protocols all contain ten unique unicast slots.

We minimized the impact of distinct synchronization techniques in the original versions of each protocol. By implementing SCP-MAC as a variant of Crankshaft, synchronization packets and the normal SCP-MAC bootstrapping phase were avoided. Hello wakeup times were removed from AS-MAC and BAS-MAC and these protocols used a predetermined, static Neighbor Table to control mote wakeup times. Since Crankshaft is the only one of the four protocols to include MAC layer ACK packets, we disabled these for the experiments.

As the primary focus of this research was to compare energy usage across protocols, it was important to standardize wakeup tone length and contention window slot size. For all protocols, we defined a minimum amount of time needed to sense the channel for activity. Considering our oscilloscope measurements, lack of clock drift and our synchronization techniques, the wakeup tone length was set at 5ms. This 5ms time interval was also used to define the slot time in the sender contention window. Moreover, with our experiments utilizing 10 motes, all four protocols were implemented with 16 5ms contention slots using a uniform distribution to pick a slot and a total contention window size of 80ms.

However, the packet-based CC2420 radios in TelosB motes put SCP-MAC at a disadvantage. This radio cannot take advantage of SCP-MAC's overhearing avoidance optimization, which inspects packet headers on the fly and avoids additional overhearing before the entire packet has been received. We justify this decision based on the relative popularity of packet-based radios over byte-based radios.

E. Experiments

A thorough set of indoor experiments was run to measure the energy consumption of AS-MAC, BAS-MAC, Crankshaft and SCP-MAC. Each experiment ran for three minutes and was repeated five times. The reported results are averaged over the five runs. Fixed parameters across all experiments included: 50-byte packets, 16 5ms contention window slots, and a 5ms wakeup tone. We found that 50-byte packets produced minimal transmission errors.

Ten-mote experiments were executed using local gossip, broadcast and convergecast traffic patterns. In the local gossip experiments, five motes were designated as senders with corresponding receivers spaced two meters apart. Senders transmitted once every 10000ms with a 2000ms wakeup interval. In the broadcast experiments, a BS mote broadcasted periodically to nine leaf motes, each two meters from the BS, once every 10000ms with a 1000ms wakeup interval. In the convergecast experiments, nine motes periodically transmitted to a single BS mote, located two meters away, once every 10000ms with a 1000ms wakeup interval. Note, these choices were intended to avoid heavy transmission contention that

might induce backlogs and the added complexity of sending queues at the motes.

IV. RESULTS

This section analyzes energy usage results for the local gossip, convergecast, and broadcast experiments. Sender and receiver measurements are separated in local gossip experiments. In the convergecast and broadcast experiments, base station results are separated from those of the leaf nodes. Total energy consumption in mJ, shown as a stacked bar graph of energy consumed in each radio state, is the average of all nodes of the represented type.

The bar graphs are broken purely across radio states and not radio functions. For example, a transmitting mote spends some initial time in the idle state and oscillates between idle and sending states while transmitting preambles. Thus, when a mote transmits more frequently, both sending and idle state times increase. Similarly, between reception of preambles a receiving mote spends a portion of time in the idle state.

One general observation from the graphs presented is that the four protocols spend differing amounts of time in the starting and stopping states. These variations, due to the asynchronous nature of TinyOS and differences in code structure between the four protocols, yield small discrepancies that do not significantly affect the results.

In all the experiment sets, the variance between the five runs was very low with most of the variation across trials attributable to the propagation of wireless signals through the air. This high consistency within the measurement instances is due to the small distance between motes.

A. Local Gossip

Figures 3 and 4 depict the energy usage for the ten-mote local gossip experiment involving five motes each sending staggered messages once every 10 seconds to their corresponding receiver. Under this traffic pattern, AS-MAC uses the least energy, with senders consuming approximately 110 mJ of energy and receivers consuming about 65 mJ. SCP-MAC consumes roughly 2.5 times as much energy for receivers and 1.75 as much energy for senders. SCP-MAC's overhearing issue yields more energy consumption in the receiving state for both senders and receivers.

The energy consumption of Crankshaft and BAS-MAC is very similar for both senders and receivers. This is expected since both protocols contain a single broadcast slot and a unicast slot for each receiver. Note, the extra energy consumed by Crankshaft and BAS-MAC compared to AS-MAC is due to time spent in the idle state. Local gossip traffic has no broadcasts. Thus, without an explicit broadcast slot, AS-MAC does not spend idle time associated with the extra wakeup tone in the broadcast slots of Crankshaft and BAS-MAC.

Across the four protocols, sending local gossip messages consumes more energy than receiving these messages. Although TelosB sending is less costly per unit time, a sender is in the sending state longer than a receiver is in the receiving state due to time spent sending preambles over the contention window.

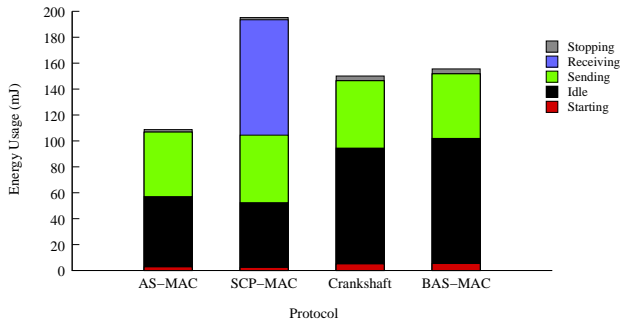


Fig. 3. Local Gossip Senders

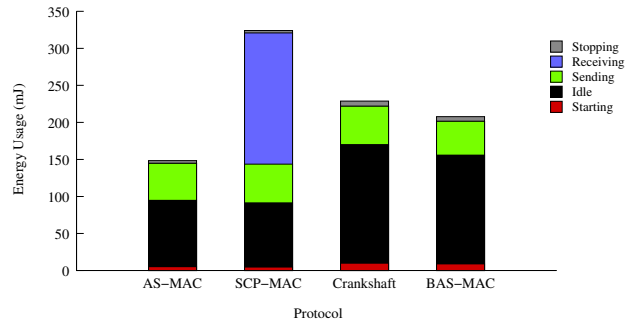


Fig. 5. Convergecast Senders

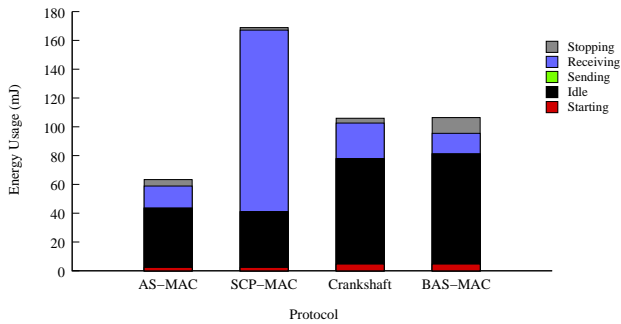


Fig. 4. Local Gossip Receivers

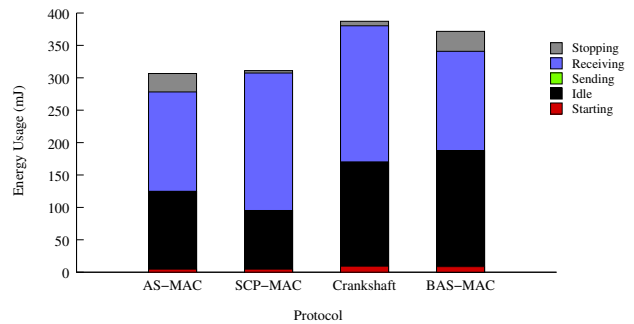


Fig. 6. Convergecast Receiver

B. Convergecast

Figures 5 and 6 graph energy usage for the convergecast experiments involving nine nodes each sending staggered messages to a BS node every 10 seconds. In this scenario AS-MAC again exhibits the best overall performance. Similar to the local gossip results, most of the difference in energy usage between AS-MAC and SCP-MAC is due to overhearing, with roughly as much SCP-MAC sender energy consumed in overhearing as through all other radio activity. Crankshaft and BAS-MAC senders both consume more energy than AS-MAC due to having two wakeup times per wakeup interval. On the receiver side, the AS-MAC advantage displayed under local gossip traffic has been reduced and SCP-MAC's energy consumption is close to AS-MAC. Note, the average receiver energy consumption in a convergecast scenario is more than double the receiver consumption under paired local gossip regardless of the MAC protocol used.

C. Broadcast

Energy consumption for broadcast traffic is displayed in Figures 7 and 8 respectively. These experiments utilize a star topology in which a single base station broadcasts to nine leaf nodes at a rate of one message every 10 seconds. Figure 7 highlights AS-MAC's inefficient broadcasting mechanism and demonstrates the value of having BAS-MAC provide a single broadcast slot into each AS-MAC wakeup interval.

Since all nodes wake up simultaneously in SCP-MAC, it is natural that SCP-MAC senders use the least energy when only BS broadcast traffic is sent. Receivers of broadcast traffic

perform similarly in AS-MAC and SCP-MAC. This is because each of these protocols have a single wakeup slot per interval. However, this slot is a broadcast slot in SCP-MAC but a unicast slot in AS-MAC. Because AS-MAC only has unicast slots, this puts a greater burden on the broadcasting node, causing the greatly increased energy consumption seen in Figure 7. Furthermore, there is only one wakeup slot per wakeup interval in SCP-MAC, as opposed to Crankshaft and BAS-MAC, which require each node to wakeup for both a broadcast and a unicast slot per interval. This causes SCP-MAC to outperform these two protocols in both sending and receiving.

The broadcast traffic energy usage results indicate that the best power-aware protocol to use depends whether the BS is connected to a centralized power source. If energy usage

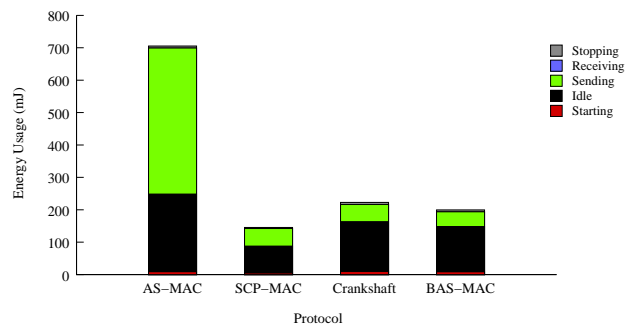


Fig. 7. Broadcast Sender

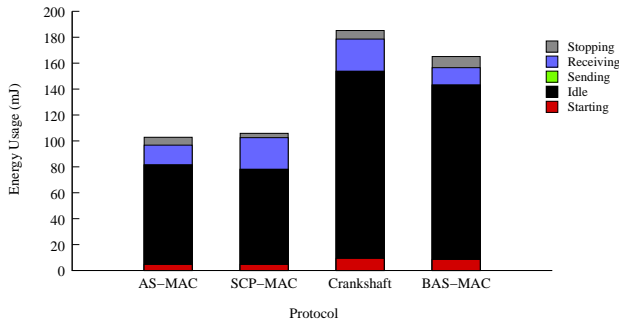


Fig. 8. Broadcast Receivers

of the broadcasting BS is not a concern, then AS-MAC is a good choice. However, in multi-tiered WSN topologies where battery-powered mid-tier nodes forward broadcasts to neighborhoods of leaf nodes, AS-MAC would be a bad choice. For example, SCP-MAC would be a better protocol choice when Internet node queries were regularly sent to motes in a multi-tiered WSN via a broadcast flood.

Note, this investigation has focused on simple segregated traffic patterns. If the WSN had consisted of a cluster-based tiered topology with battery-powered cluster heads, the energy analysis would need to account for both convergecast traffic going from leaf nodes through cluster-heads to a master BS and broadcast traffic going in the opposite direction down through the cluster-heads to the leaf nodes.

It is also interesting to note that there exist slight differences between Crankshaft and BAS-MAC receiver results, despite the similarity of the protocols. Neither our analysis nor intuition explains the slight advantage of BAS-MAC in this investigation. These results may motivate future research comparisons of synchronous frame-based and asynchronous offset-based WSN MAC protocols.

V. CONCLUSION

This paper compares energy usage of four state-of-the-art, power-aware MAC layer WSN protocols running TinyOS on TelosB motes. Using the MLA framework, AS-MAC, BAS-MAC, Crankshaft and SCP-MAC were implemented on the same ten TelosB motes. Developing the four protocols within a common framework and standardizing parameter settings enabled extraction of cause-and-effect protocol behavior from energy usage experiments. By minimizing the possibility of transmission collisions, this investigation clarifies the primary sources of energy waste when each of the four MAC protocols handles three distinct sensor network traffic patterns.

Due to its inherent overhearing avoidance, AS-MAC performs best under local gossip and convergecast traffic. However, AS-MAC consumes the most energy during a broadcast traffic scenario. Conversely, SCP-MAC performs best under broadcast traffic, where its built-in overhearing is beneficial. Crankshaft and BAS-MAC behaved similarly across all the experiments with BAS-MAC consuming slightly less energy in most of the experiments.

This fair comparison of four power-aware MAC protocols yields no single WSN MAC protocol which performs best under all traffic situations. Moreover, given the large variety of applications, topologies and traffic patterns that modern WSNs face, it is unlikely that one MAC protocol will be the best in all circumstances. This re-enforces utilizing a framework such as the MLA to provide a suite of power-aware MAC protocols that could eventually be dynamically swapped into motes as the WSN circumstances and objectives change.

Future work directions include: analysis of MAC protocol energy consumption over multi-hop networks, studying the tradeoffs associated with including or omitting ACKs in the MAC layer, and understanding the impact of contention window size on energy usage and packet delivery ratios amongst single-hop neighbors.

REFERENCES

- [1] J. Hill and D. Culler. Mica: a wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, November 2002.
- [2] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, USA, June 2002. USC/Information Sciences Institute, IEEE.
- [3] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Sensys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, New York, NY, USA, 2003. ACM.
- [4] A. El-Hoiydi, J. Decotignie, C. Enz, and E. Le Roux. Poster abstract: wisemac, an ultra low power mac protocol for the wisenet wireless sensor network. In *Sensys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 302–303, New York, NY, USA, 2003. ACM.
- [5] M. Buettner, G. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Sensys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320, New York, NY, USA, 2006. ACM.
- [6] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Sensys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM.
- [7] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 321–334, Boulder, Colorado, USA, 2006. ACM.
- [8] B. Jang, J. Lim, and M. Sichitiu. As-mac: An asynchronous scheduled mac protocol for wireless sensor networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pages 434–441, Atlanta, Georgia, USA, 2008. IEEE.
- [9] G. Halkes and K. Langendoen. Crankshaft: An energy-efficient mac-protocol for dense wireless sensor networks. In *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN '07)*, Delft, The Netherlands, 2007.
- [10] K. Klues, G. Hackmann, O. Chipara, and C. Lu. A component-based architecture for power-efficient media access control in wireless sensor networks. In *Sensys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 59–72, New York, NY, USA, 2007. ACM.
- [11] Texas Instruments Inc. Chipcon cc2420 datasheet. <http://focus.ti.com/lit/ds/symlink/cc2420.pdf>.
- [12] K. Jamieson, H. Balakrishnan, and Y. Tay. Sift: A mac protocol for event-driven wireless sensor networks. In *Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN '06)*, pages 260–275, Zurich, Switzerland, 2006.