Increasing ZigBee Network Lifetime with X-MAC

Pablo Suarez and Carl-Gustav Renmarker
Saab Communication
{suarez, renmarker}@saabgroup.com

Adam Dunkels and Thiemo Voigt
Swedish Institute of Computer Science
{adam,thiemo}@sics.se

Abstract

The ZigBee standard builds on the assumption that infrastructure nodes have a constant power supply. ZigBee therefore does not provide any power-saving mechanisms for routing nodes, which limits the lifetime of battery-powered ZigBee networks to a few days. This short lifetime drastically restricts the possible application scenarios for ZigBee. To expand the usefulness of ZigBee, we present a ZigBee implementation where we replace the default ZigBee MAC protocol with the power-saving MAC protocol X-MAC. Our results show that X-MAC reduces the power consumption for ZigBee routing nodes with up to 90%, leading to a ten-fold increase in network lifetime at the price of a slight increase in network latency. Furthermore, we are the first to experimentally quantify the energy-efficiency of X-MAC in a multi-hop scenario. Our results indicate that X-MAC reduces the power consumption of idle nodes that are within communication range of two communicating nodes, suggesting that X-MAC may be able to mitigate the hot-spot problem in sensor networks.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Network Operating Systems*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Wireless sensor networks, MAC protocols, ZigBee

1 Introduction

Communication is the largest power consumer in state-of-the-art sensor network hardware [9]. To achieve a long network lifetime, the sensor nodes must turn their communication device off as much as possible. To support multi-hop communication, receiving nodes must be awake when a sending node transmits packets. Several mechanisms for synchronizing senders and receivers have been developed [2, 8].

The ZigBee standard [15] is designed around the assumption that routing nodes have a constant power supply. Such nodes therefore never need to save energy by switching off their radio. This makes ZigBee unsuitable for applications where it is impossible to draw power cables to routing nodes.

To make ZigBee useful for a class of new application scenarios, we investigate the use of the X-MAC power-saving MAC protocol [2] in the ZigBee stack. We have implemented a ZigBee stack for the Contiki operating system [4], based on the Open-ZB stack [3], on top of the Contiki implementation of X-MAC. We quantify the power consumption with Contiki's built-in energy estimation feature [5] running on a set of Tmote Sky boards [9].

The contributions of this paper are threefold. First, we experimentally demonstrate that X-MAC reduces the power consumption of ZigBee routing nodes up to 90%, leading to a ten-fold increase in network lifetime. Second, we are the first to experimentally quantify the energy-efficiency of X-MAC in a multi-hop scenario and to experimentally confirm that the optimal listen-and-sleep cycle model by Buettner et al. [2] indeed provides power-efficient duty cycling configurations. Third, we experimentally quantify the trade-offs between energy consumption, X-MAC duty cycle, and network latency. Our results indicate that X-MAC reduces the power consumption of idle nodes that are within communication range of two communicating nodes. This suggests that X-MAC may be able to mitigate the hot-spot problem in sensor networks.

The rest of this paper is structured as follows. In Section 2 we present an overview of the ZigBee stack and the X-MAC protocol. After describing our implementation in Section 3, we present our experimental results in Section 4. Before concluding the paper in Section 6 we discuss related work in Section 5.

2 Background

ZigBee is an open standard for low-power, wireless networked monitoring and control [15]. The ZigBee stack consists of four layers, as shown in Figure 1: Application (APL), Network (NWK), Medium Access Control (MAC), and Physical (PHY).

Strictly speaking, only the NWK and APL layers belong

REALWSN'08, April 1, 2008, Glasgow, United Kingdom.

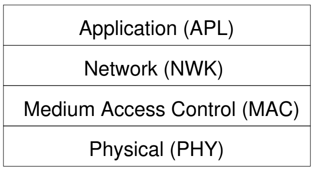

Figure 1. ZigBee protocol stack.

to the ZigBee stack. The PHY and MAC layers are specified by the IEEE 802.15.4 standard, but the ZigBee specification requires the NWK and APL layers to run over 802.15.4. ZigBee supports several network topologies and two types of devices: Full Function Devices (FFD), that can perform all available operations within the standard, including routing mechanism or coordination tasks; and Reduced Function Devices (RFD), that implement a limited version of the IEEE 802.15.4 protocol. The RFDs do not route packets and must be associated with an FFD. The current ZigBee standard requires FFDs to be always on, which in practice means that FFDs must be constantly powered. Battery-powered FFDs have a lifetime on the order of a few days.

X-MAC [2] is a power-saving MAC protocol that is designed to run on top of the 802.15.4 PHY. X-MAC reduces the power consumption by switching the radio on and off at regular intervals. To send a packet, a node broadcasts a train of short strobe packets. The strobe packet train is long enough to allow all nearby devices to be switched on at least once. After receiving a strobe packet, a node turns on its radio in preparation of receiving a full packet. As an optimization for unicast packets, the strobe packets include the address of the receiver of the full packet. When receiving a unicast strobe, a receiver immediately sends a short acknowledgment packet. The sender can then immediately send its full packet. Other nodes that happen to overhear the strobe packets can turn off their radios until the full packet has been transmitted.

3 Design and Implementation

We have implemented a ZigBee stack Contiki, based on the Open-ZB stack [3], but underlay the 802.15.4 MAC protocol with the power-saving X-MAC protocol. While our implementation is designed to run over any radio hardware, our target hardware is the Tmote Sky [9].

3.1 Upper Layers

The ZigBee specification separates the APL layer into three different sub-layers: the Application Support Sublayer, the ZigBee Device Objects, and Manufacturer Defined Application Objects. Since we are not opting for a specific application, we developed a simple application process to control and perform our experiments. These Upper Layer (UL) is located above the NWK layer. We have implemented the UL as a Contiki process that works independently from the rest of the system. The UL uses available stack functions that notify the UL of events such as incoming frames. The UL process is blocked until a packet arrives, or an incoming frame wakes up the UL. The UL then processes the received information.

The ZigBee NWK layer controls the IEEE 802.15.4 MAC layer and provides a service interface to the application layer. The network services include the generation of frames to transport the application layer protocol data units to an appropriate device, either the final destination or the next step in the communication chain. The network layer also controls a number of network management tasks, such as establishing new networks, joining and leaving networks, and neighbor and route discoveries.

We have implemented the NWK layer as a set of functions that manage the routing information and other network layer data. The network layer information is stored in three different tables: the neighbor table, the routing table, and the route discovery table. The three tables are implemented as arrays with a fixed number of elements.

3.2 Adding X-MAC to ZigBee

The IEEE 802.15.4 MAC layer implements a set of MAC-protocol functions and controls the physical layer and guaranteed time slot mechanisms. The IEEE standard includes two options for channel access: a beacon-based method, and a CSMA-CA channel access mechanism. We focus on the latter and therefore our IEEE 802.15.4 MAC layer handles frame generation, recovery, and acknowledgment control.

In state-of-the-art low-power sensor network hardware, the radio device has the highest power consumption [9]. Since the MAC layer controls the PHY, most power-saving mechanisms for sensor networks focus on the MAC layer. To increase ZigBee network lifetime, we add X-MAC into the ZigBee stack, placing it between the IEEE 802.15.4 MAC and the radio driver.

Our X-MAC implementation consists of three main functions that define the protocol operations: *send*, *read*, and *powercycle*. The *send* and *read* functions are called from the NWK layer. The *powercycle* function implements the duty cycling of the radio by periodically switching the radio on and off. The operation of the *powercycle* function is shown in Figure 2. The *powercycle* function does not control the transmission or reception of data, but only switches the radio on and off.

To be compatible with standard ZigBee implementations, the X-MAC strobe packets must conform to the 802.15.4 MAC format. X-MAC unicast strobe packets include the address of the receiver. We use the Frame type field of the frame_control field in the IEEE 802.15.4 MAC header to separate strobe packets from data packets. The frame field currently only uses four of the eight possible packet types: Data Frame, ACK Frame, Command Frame, and Beacon Frame. We define a new frame type, Preamble Frame, for the X-MAC strobe packets. While this solution is not compliant with the current 802.15.4 standard, it is easy to integrate into future versions of the standard.

The CC2420 radio, used on our Tmote Sky boards, automatically creates packets that conform to the 802.15.4 PHY, and adds the the checksum footer field, the frame length field and the a synchronization header field from the 802.15.4 MAC.

4 Evaluation

We use the energy estimation features in Contiki [5] to experimentally quantify the energy-efficiency of X-MAC for ZigBee on a set of Tmote Sky nodes. We compare the per-

```
while(1) {
   if(we_are_sending) {
      PT_WAIT_UNTIL(!we_are_sending);
   }
   radio->off();
   set_timer(xmac_off_timer);
   PT_WAIT_UNTIL(timer_expires(xmac_off_timer));
   if(we_are_sending) {
      PT_WAIT_UNTIL(!we_are_sending);
   }
   radio->on();
   set_timer(xmac_on_timer);
   PT_WAIT_UNTIL(timer_expires(xmac_on_timer));
}
```

Figure 2. X-MAC powercycle basic scheme.

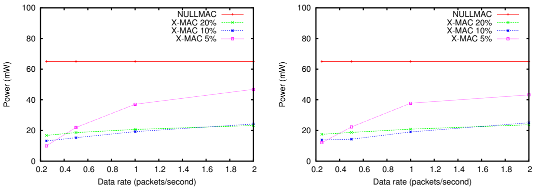

Figure 3. The radio listen power consumption for both the sender (left) and the receiver (right) increases with data rate.

formance of X-MAC with the always-on MAC protocol in Contiki, called NULLMAC. As expected, the power consumption for a node using NULLMAC is high, and battery lifetime is only around four days. NULLMAC is very similar to the CSMA-CA protocol in the ZigBee specification, which is the only available option for mesh networking in the current ZigBee standard. We performed our experiments indoors, inside the SICS office building.

We evaluate the energy-efficiency of X-MAC in both a single-hop scenario and a multi-hop scenario. In the single-hop scenario, we quantify the effect of contention on power consumption. Our experiments show that X-MAC decreases power consumption with up to 90%, that contention increases the power consumption slightly, and that, in a multi-hop scenario, power consumption is higher closer to the sink.

4.1 Single-hop Energy Consumption

Before quantifying the energy consumption of X-MAC in a multi-hop scenario, we first measure the energy consumption in a two-node scenario. This scenario gives us a baseline metric, that we can view as a lower bound on the energy consumption of X-MAC for ZigBee.

In this experiment, one sender sends packets to a sink. We vary the rate of the data generated by the sender. The data packets consist of 100 bytes ZigBee payload and a 802.15.4 header that includes 802.15.4 addresses in long format. The total packet length is 128 bytes.

To compare the energy consumption of X-MAC with the default energy consumption of ZigBee, we perform experiments with both the 100% duty cycle NULLMAC protocol and X-MAC. We vary the both the data rate and the X-MAC duty cycle, between 20%, 10% and 5%.

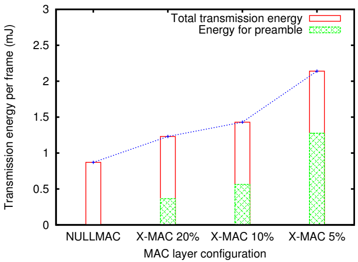

Figure 4. The energy consumption of the transmission of the preamble increases with decreasing duty cycle.

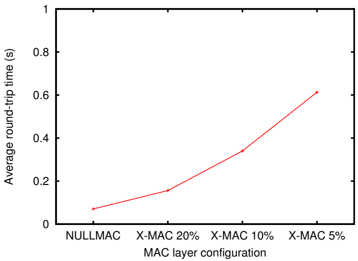

Figure 5. The average round-trip time increases with decreasing duty cycle.

Figure 3 shows the radio listen power consumption with varying data rate for both the sending and the receiving node, and Figure 4 shows the transmission energy per frame with increasing duty cycle. The radio listen power consumption increases with increasing traffic load. XMAC's transmission procedure requires that strobe packets are sent before the actual packet is transmitted. Hence, the sender must, after sending each of these preamble packets, turn the radio on in order to detect the strobe acknowledgment that signals that the receiver has detected the sender transmission and that it is ready to receive the packet. Therefore, both transmission and reception are sensitive to the traffic load of the network.

Figure 4 shows that the transmission energy per frame increases with decreasing duty cycle. The reason for the increase is that more strobe packets need to be transmitted when the receiving node has its radio switched off for a longer time.

Note also that a low duty cycle can decrease the performance because of the longer preamble sequence that is required as shown in Figure 4. If the network load is low, this is compensated by the energy savings achieved by turning off the radio for a long time. If the traffic load increases, the energy consumption is obviously affected.

As shown in Figure 5, the round-trip time is affected by the duty cycle scheme but not by the traffic load. Since there are no collision problems in this single-hop scenario, the transmission mechanism should take the same time on average with the same duty cycle.

4.2 The Effect of Contention

To study the effect of contention on the energy consumption of X-MAC, we use an experimental setup with two senders that transmit packets to a sink node. Our hypothesis is that collisions reduce the power-saving ability when several nodes try to access the channel.

According to the original X-MAC design by Buettner

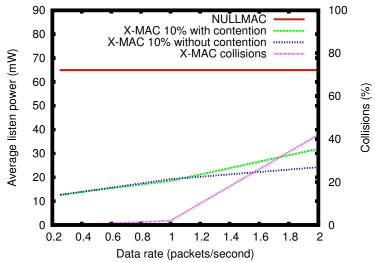

Figure 6. The listen power with two nodes contending for the medium. The number of collisions for X-MAC increases with the data rate.

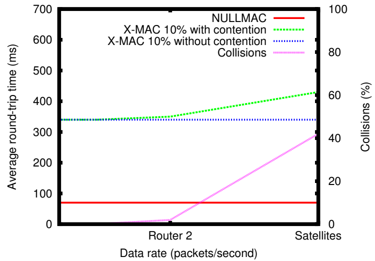

Figure 7. The average round-trip time and the amount of collisions increase with increasing data rate.

et al. [2], channel activity can be detected by listening for strobe packets. If a preamble has been detected by the node trying to send a packet, the packet should not be sent until a sleep period of the duty cycle expires. As a consequence, in high-traffic networks, we expect the latency to be higher and dependent on the traffic volume.

For the contention experiments, packets have the same format as in the previous experiments. Frames consist of a 100 bytes length payload and a 802.15.4 header with addresses in long format. The total length is 128 bytes.

Figure 6 shows that the energy spent for listening for traffic with contention is higher than without contention. The explanation is that, generally, when more traffic is present in the area, strobe sequences are longer since the destination node normally enters in a sleep period after having finished a previous transmission. Additionally, even during the node's own strobe period it is possible that it receives a preamble packet from another node that has not detected the sender's own preamble sequence. This results in the sender node having to restart its preamble sequence once the channel is free again.

We can also see that a high traffic load affects the transmit-mode energy spent per packet. Longer preamble periods logically mean that more strobe packets are transmitted per frame, so a higher amount of energy is consumed. This also means that a higher proportion of the transmit energy is spent in the preamble sequence instead of the data frame compared to the non-contention case as shown in Figure 8.

4.3 Multi-hop Energy Consumption

Figure 9 shows the experimental setup that we use to quantify the energy consumption of X-MAC in a multi-hop scenario. The setup consists of eight nodes, where three nodes form a routing path.

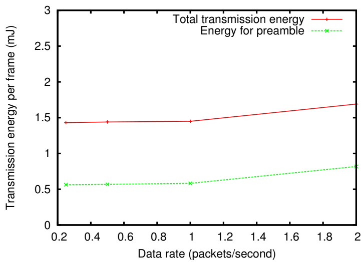

Figure 8. The transmission energy per packet increases due to the increase in collisions.

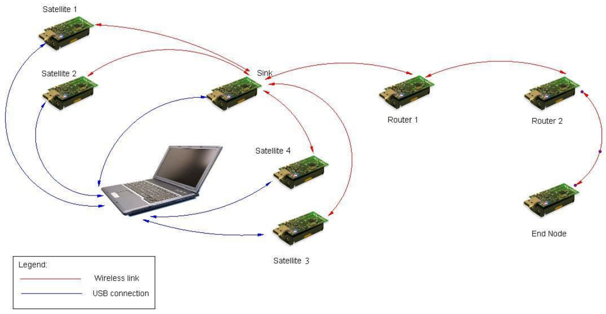

Figure 9. Setup for multi-hop experiment

There are four satellite nodes deployed close to the sink, within communication range. The satellite nodes are idle during the experiment. The purpose of the satellite nodes is to quantify the effect of nearby communication on idle nodes.

The Router 1 node forwards incoming messages from two nodes not in range of the sink: Router 2 and the End Node.

All nodes periodically transmit a data frame towards the sink node, and the sink replies with an ACK frame. Intermediate nodes route the packets so that they reach the sink.

Buettner et al. [2] propose an algorithm to determine the optimal values of the duty cycle scheme with varying traffic load. We have implemented this algorithm to obtain the optimal duty cycle for each traffic load. Given that each device receives different amounts of traffic, a weighted average was calculated to obtain a value valid for the whole network, resulting in a 4.3% duty cycle.

Figure 10 shows that in most cases the determined a duty cycle of 4.3% provides the best results. This demonstrates

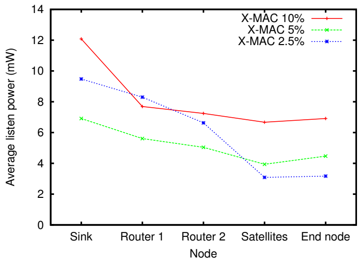

Figure 10. The radio listening power consumption decreases at nodes further away from the sink. An optimal duty cycle is somewhere between 2.5% and 5%.

the validity of the model proposed by the X-MAC designers [2]. Given that the mathematical expressions of the model are built using the traffic load as the main variable, for some of the nodes the averaged traffic load differs significantly from their actual situation and other configurations can reduce the energy consumption. For example, the satellite nodes and the End Node actually have a lower energy consumption with a 2.5% duty cycle. One practical conclusion of this is that having a homogeneous network topology makes the optimization estimations converge to the real solution. An alternative solution would consist in modifying the X-MAC protocol so that it can work with different duty cycles for different devices in the network

Figure 10 also shows that the End Node and the satellite nodes, that do not act as routers, consume less power than the router nodes. This is because the satellite close to the sink can turn off their radio when they detect a preamble sequence targeted to the sink, thereby saving energy. Given that all the traffic of the network has the sink node as the final destination, the satellite nodes detect a lot of traffic not destined for them so the described procedure occurs very often, which explains the lower consumption compared to the End Node. Note that this might imply that X-MAC is able to mitigate the hot spot problem [7] given a sufficient number of nodes are deployed close the sink. The hot spot problem occurs in in multi-hop sensor networks when the nodes close to the base station die early because they must relay more packets than nodes further away from the base station.

5 Related Work

Cunha et al. have implemented an open-source implementation of IEEE 802.15.4/ZigBee for TinyOS [3]. They evaluate algorithms that enable the usage of beacon-mode in cluster-tree topologies [6]. Others have analyzed the capacity audio streaming capabilities of ZigBee networks [1]. In contrast to these efforts, we concentrate on adding low power routing to ZigBee by integrating a power-efficient MAC layer into the stack. We experimentally evaluate the power-efficiency of the resulting stack using Contiki's built-in energy estimation feature [5]. Staub et al. [11] have previously experimentally compared the lifetime of two MAC protocols: LMAC [13] and TEEM [12], an enhancement of S-MAC [14]. Their comparison was performed by powering the sensor nodes with GoldCap capacitors which enables lifetime comparisons within a reasonable amount of time [10].

6 Conclusions

We show that by integrating the popular X-MAC power-saving MAC protocol into the ZigBee stack, we can significantly prolong the lifetime of battery-powered ZigBee networks. Our experiments show the reduction in energy consumption can be up to 90%.

We are the first to experimentally quantify the energy-efficiency of X-MAC in a multi-hop scenario. Our results show that the energy consumption is slightly higher for nodes closer to the sink and that the algorithm by Buettner et al. [2] for finding optimal duty cycles is sound.

Finally, our results indicate that X-MAC reduces the power consumption of idle nodes that are within communication range of two communicating nodes. This suggests that X-MAC may be able to mitigate the hot-spot problem in sensor networks. We see this as an interesting area for future work.

7 Acknowledgments*

This work has been performed within the SICS Center for Networked Systems, partly funded by VINNOVA, SSF and KKS.

8 References

[1] D. Brunelli, M. Maggiorotti, L. Benini, and F. L. Bellifemine. Analysis of audio streaming capability of zigbee networks. In *Proceedings of the European Conference on Wireless Sensor Networks, EWSN 2008*, 2008.

[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 *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320, Boulder, Colorado, USA, 2006.

[3] A. Cunha, A. Koubaa, R. Severino, and M. Alves. Open-ZB: an open-source implementation of the IEEE 802.15. 4/ZigBee protocol stack on TinyOS. *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1–12, 2007.

[4] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors*, Tampa, Florida, USA, Nov. 2004.

[5] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. Software-based online energy estimation for sensor nodes. In *Proceedings of the Fourth IEEE Workshop on Embedded Networked Sensors (Emnets IV)*, Cork, Ireland, June 2007.

[6] A. Koubâa, A. Cunha, and M. Alves. A Time Division Beacon Scheduling Mechanism for IEEE 802.15. 4/Zigbee Cluster-Tree Wireless Sensor Networks. *in Euromicro Conference on Real-Time Systems (ECRTS 2007), Pisa (Italy), July*, 2007.

[7] M. A. Perillo, Z. Cheng, and W. B. Heinzelman. An analysis of strategies for mitigating the sensor network hot spot problem. In *MobiQuitous*, pages 474–478, 2005.

[8] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, 2004.

[9] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proc. IPSN/SPOTS'05*, Los Angeles, CA, USA, Apr. 2005.

[10] H. Ritter, J. Schiller, T. Voigt, A. Dunkels, and J. Alonso. Experimental Evaluation of Lifetime Bounds for Wireless Sensor Networks. In *Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN2005)*, Istanbul, Turkey, Jan. 2005.

[11] T. Staub, T. Bernoulli, M. Anwander, M. Waelchli, and T. Braun. Experimental lifetime evaluation for MAC protocols on real sensor hardware. *Proceedings of ACM Workshop on Real-World Wireless Sensor Networks (REALWSN'06), Uppsala, Sweden, June*, 2006.

[12] C. Suh and Y. Ko. A traffic aware, energy efficient MAC protocol for wireless sensor networks. *IEEE International Symposium on Circuits and Systems*, pages 2975–2978, 2005.

[13] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks. *1st Int. Workshop on Networked Sensing Systems (INSS 2004)*, 2004.

[14] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 12(3):493–506, 2004.

[15] Zigbee. Web page. 2007-11-21. http://www.zigbee.org.