Evaluation of WiseMAC on Sensor Nodes

Philipp Hurni and Torsten Braun

Universität Bern, Switzerland

Abstract. The WiseMAC protocol is one of the most energy-efficient medium access control protocols for wireless sensor networks. However, in many typical wireless sensor network scenarios, throughput is limited when high traffic occurs, e.g., if many sensors simultaneously detect and report an event to the base station. The paper proposes to improve the traffic-adaptivity of WiseMAC by extending the more bit mechanism supporting transmissions of frame bursts in WiseMAC. It allows bottleneck nodes to stay awake in situations of high traffic and temporarily abandon the periodic sleep-wake pattern. We evaluate WiseMAC's energy efficiency and compare the original and extended more bit by simulations as well as measurements in a real sensor experiment test-bed.

1. Introduction

The Wireless Sensor MAC (WiseMAC) [1] protocol is based on low duty cycles, periodic wake-ups and preamble sampling. Performance evaluations show that the required energy increases linearly with the traffic rate. However, the throughput is rather limited and packet loss already occurs with rather low traffic rate. The reason is that in tree-based wireless sensor network scenarios, nodes receiving traffic from several sources might become bottleneck nodes in case they wake up strictly in a periodic way. The more bit mechanism of WiseMAC allows exchanging additional traffic between a pair of nodes, but this only supports the exchange of large messages or frame bursts in point-to-point scenarios. This paper evaluates the more bit mechanism as well as the proposed extension to allow a bottleneck node to temporarily adapt its duty cycle in case of high traffic.

Section 2 describes the basic WiseMAC protocol and its more bit mechanism. Section 3 introduces the extended more bit mechanism. Section 4 presents simulation results, while section 5 discusses performance evaluation results from real-world experiments. Section 6 concludes the paper.

2. WiseMAC

WiseMAC is based on short, unsynchronized duty cycles and preambles exceeding the time of a node in sleep state (Figure 1). When transmitting a frame, a preamble of variable length is used to alert the receiving node in its wake-up interval not to return to sleep state, but to stay awake for the upcoming transmission. When the receiver's wake-up pattern is still unknown, the duration of the preamble equals the full basic cycle duration T, as illustrated in Figure 1 in the first transmission. The own schedule offset is then piggybacked to the frame and transmitted to the receiver.

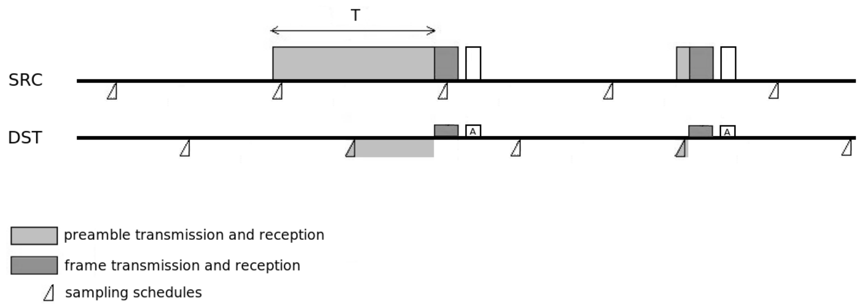

Figure 1. WiseMAC

After successful frame reception, the receiver node piggybacks its own schedule to the respective frame acknowledgement. Received schedule offsets of all neighbor nodes are subsequently kept in a table and are periodically updated. Based on this table, a node can determine the wake-up intervals of all its neighbors and minimize the preamble length for upcoming transmissions. It waits for its neighbor's wake-up and sends the frame just in the appropriate moment, only prepending a small preamble that compensates for the maximum clock drift that the two involved node's clocks may have developed since the last schedule exchange.

To increase the maximum achievable throughput in case of packet bursts and higher traffic load, WiseMAC suggests an optional fragmentation scheme called more bit mode. WiseMAC sets a flag (more bit) in a unicast MAC frame whenever a node has more packets to send. The more bit in the frame header signals to the receiving node that it shall not turn off the transceiver after receiving the frame, but switch to the receive mode again after frame acknowledgement in order to receive the next packet, cf. Figure 2. A sender does not need to wait for the next wake-up of the receiver to transmit the next frame. This increases the throughput. The scheme proved to be very effective in scenarios with varying traffic, especially with packet bursts generated by single nodes.

The more bit scheme only serves to improve traffic adaptivity between one sender and one destination. In a wireless sensor network scenario, there are often nodes that have to forward data from large sub-trees. Such bottleneck nodes will have to forward messages generated by many other nodes. The more bit scheme does not help at all if several nodes aim to simultaneously transmit a packet to the same bottleneck node, as it can happen in a tree scenario as depicted in Figure 3. One node after the other will have to wait for a wake-up of the bottleneck node in order to forward a frame. The duty cycle of the bottleneck node, however, is not increased with the more bit scheme.

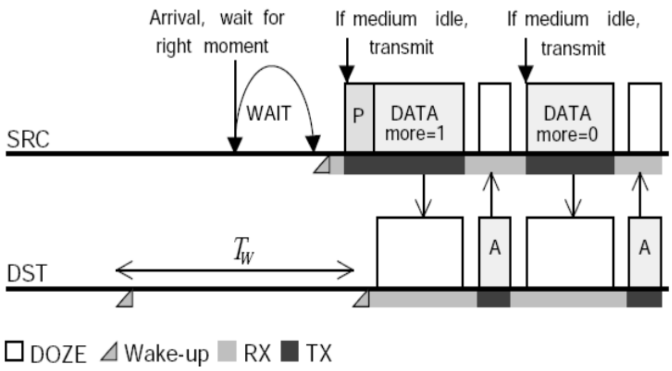

Figure 2. More bit in WiseMAC

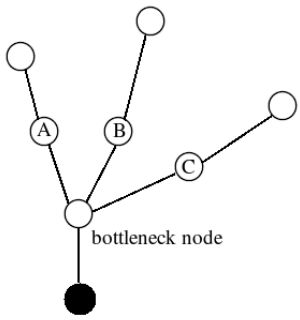

Figure 3. Tree structure in a wireless sensor network

3. Extended WiseMAC More Bit Scheme

We proposed a scheme, where nodes will automatically stay awake for a longer time than just the awake period when more traffic has to be handled and tell this to all nodes waiting to forward traffic to it [8]. Therefore, we extended the semantics of the more bit to a so-called stay awake promise bit. This is also called extended more bit hereafter. Figure 4 shows two sources SRC1 and SRC2 simultaneously aiming to transmit some packets to the same node DST, possibly because an event has occurred in their vicinity.

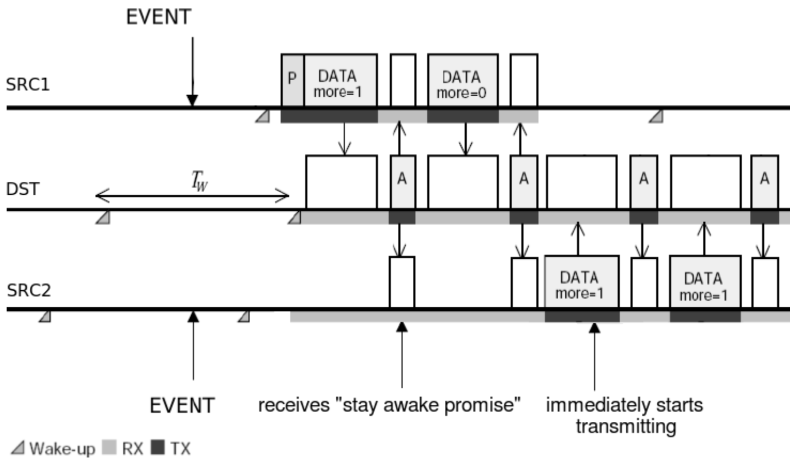

Figure 4. Extended more bit scheme based on stay awake promise

If SRC1 and SRC2 both aim to reach DST in the same wake interval, the medium reservation preamble will decide who is first. SRC1 wins the contention and sends its first two frames with the more bit set. The destination node acknowledges the more bit in the ACK packet and stays awake for at least a basic wake interval T. As SRC2 has lost the contention, it will wait and overhear the transmission from SRC1 to DST. By hearing the stay awake promise in the ACK, SRC2 knows that it can start sending its own data frames right after SRC1 has finished its transmissions. The advantage of this scheme is that no time is wasted for waiting, because the transmission of SRC2 can start immediately after the transmission of node SRC1. The mechanism is only activated when there is a node buffering more than one frame that requests its destination to stay awake for one next packet, which is an indication of increased load. The scheme is not applied after every unicast transmission, as this would lead to unnecessary energy consumption.

4. Simulations

4.1 Performance Evaluation Scenario

For performance evaluation by simulation, we used 90 nodes uniformly distributed in an area of 300 m x 300 m. Traffic using a Poisson model is generated for 1 hour at each node and sent towards a single sink. We use static shortest path routing. Each node uses a basic interval T = 250 between two wake-ups and a duty cycle of 5 %. We used the OMNeT++ network simulator [2] and the mobility framework [3], which supports simulations of wireless ad hoc and mobile networks on top of OMNeT++. The energy consumption model is based on the amount of energy that is used by the transceiver unit. CPU processing costs are considered as negligible. We used an energy consumption and state transition model with three operation modes sleep, receive and transmit, and applied the respective energy consumption values and state transition delays of the transceiver manufacturer [4]. Table 1 indicates the input current and the state transition delays of the simulations. The energy consumption during the state transition is assumed to be equal to the consumption of the respective higher state.

Table 1. Simulation parameters

carrier frequency	868 MHz
bit rate	19.2 kbps
packet size including header	160 bits
transmitter power	0.1 mW
SNR threshold	4 dB
sensitivity	-101.2 dBm
sensitivity carrier sensing	-112 dBm
communication range	50 m
packet loss coefficient α	3.5
carrier sensing range	100 m
supply voltage	3V
current	
transmit	12 mA
receive	4.5 mA
sleep	5 µA
state transition delays	
receive to transmit	12 µs
transmit to receive	12 µs
sleep to receive	518 µs
receive to sleep	10 µs
transmit to sleep	10 µs
packet queue length	15

4.2 Simulation Results

The good traffic adaptivity of WiseMAC is clearly visible in Figure 5, which depicts the overall energy consumption with the original WiseMAC approach dependent on the traffic rate. With no traffic, the energy consumption remains very low. With linear increase of traffic, WiseMAC is able to react with a more or less linear increase of the total energy consumption.

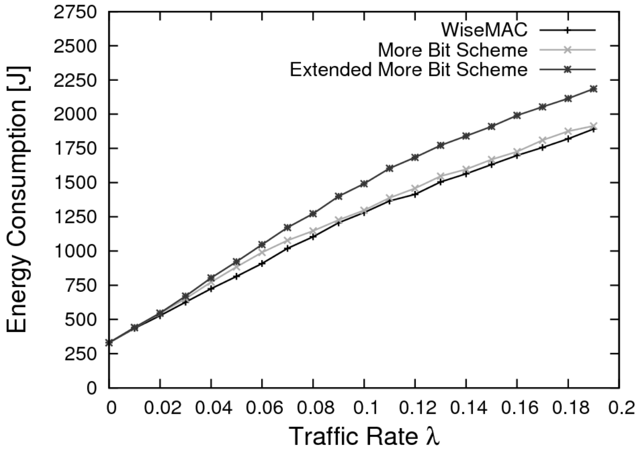

Figure 5. Energy consumption

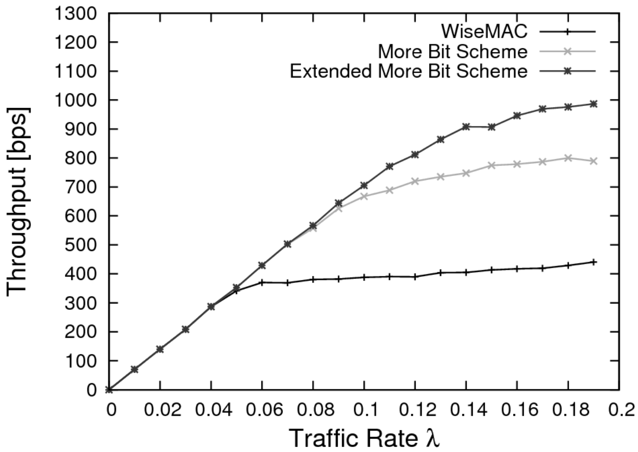

Figure 6. Throughput

Problems arise when dealing with packet bursts and when neighboring stations are also intending to send traffic. When a node wants to reach a station in its wake interval, but fails to access to the medium, it is quite likely that the preamble sampling period is missed and that the destination node goes back to sleep too early. The very short duration of the duty cycles to sense the carrier has an impact on the maximum traffic rate. The boundary values for the maximum traffic rate are limited by the duration of the sleep intervals to only a few percent of the cycle interval. Therefore, the bandwidth achievable with the basic WiseMAC scheme is limited and exceeding that limit results in higher packet loss. Figure 6 shows that an increase of maximum throughput is possible with the (extended) more bit, but the improved throughput comes with higher energy costs. But when we consider the ratio of throughput and energy, the extended scheme is even better than the more bit scheme for high traffic [8].

5. Measurements in Real Test-Beds

5.1 Experimentation Environment

In order to examine the real-world behavior of the simulated wireless sensor network mechanisms, we implemented the original WiseMAC mechanism and the (extended) more bit on Embedded Sensor Boards (ESB). ESBs run the sensor node operating system ScatterWeb [5] and are equipped with a micro-controller MSP430, various sensors and communication interfaces such as an 868.35 MHz wireless transceiver. ESBs run with 3 AA batteries or external power source. The input voltage must be in the range of 3-5 V. The embedded voltage controller of the ESB then tailors the input voltage to 3V. The different sensors and the communication interfaces can be turned on and off. Depending on the operation mode of the sensors and the microcontroller, the ESB nodes have different energy consumption levels: Average power consumption for the ESB running with all communication interfaces is 45 mW. When all sensors are turned off and the TR1001 transceiver module is transmitting data, power consumption is 29 mW in average. With all sensors shut off and radio in sleep mode the ESB still consumes 14 mW.

5.2 Measurement Methodology

Measuring the current of a small device such as the ESB can be done with some inaccuracy using a cathode-ray oscilloscope. However, these devices are not intended to record and sum up the current and the energy consumption over a longer period of time. Equipping all nodes with replaceable or rechargeable AA batteries is not a suitable approach, because measurements of battery capacities have shown that the variance can be huge. The capacities of rechargeable batteries that have just been charged up also vary heavily, especially if some of them are new and some have already been used for many charging cycles. It is too impractical to use batteries or rechargeable batteries to make lifetime and energy-consumption measurements. With energy-saving sensor nodes, the respective lifetimes can last for days, weeks, or even months.

We therefore used another well-tested and established measurement methodology to investigate the energy consumption of the ESB nodes. The methodology was already applied by the developers of the ESB in [7] and likewise used in the investigation on different MAC protocols in [6]. The methodology uses so-called gold cap capacitors. These capacitors are a special kind of capacitors that come with high capacity of 1 Farad in our case. They can be charged quite quickly and power a sensor node for a reasonable amount of time. Such a capacitor stores up to 15 J for a charging voltage of 5.5 V. When charged with the same initial amount of energy, a node with a lower overall energy consumption can live longer. The methodology allows answering the question how much energy could actually be saved when applying energy-efficiency measures on the ESBs.

We charged the 1 Farad capacitors for a charging time of 120 seconds with a supply voltage of 5.5 V. Shutting down all sensors and unplugging the nodes from the RS232 Interface makes sure that only CPU and transceiver consume energy, besides some small amount of energy spent for the circuits on the board. We observed the supply voltage of the capacitor with a multi-meter. When unplugging the capacitor from the charging source, the voltage of the capacitor continuously keeps falling. We measured the time until the voltage drops below 3 V, which is the supply voltage the embedded voltage controller requires to power the node. Below this threshold, the node still runs for some small amount of time, but its behavior is unpredictable. By applying this methodology, we obtained robust and stable results with low variance, which allow comparing the ESB node's energy consumption in different operation modes. This allowed quantifying the energy efficiency gains for different traffic load levels.

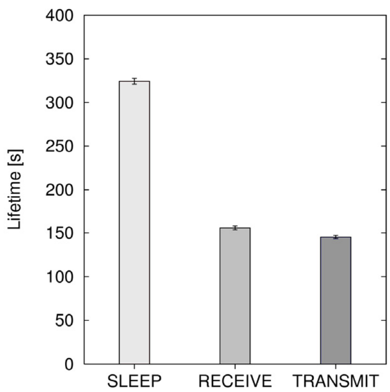

Figure 7. ESB operation modes

Figure 7 depicts the lifetime of an ESB node, when the transceiver is constantly in one of the three transceiver states sleep, receive and transmit. When comparing the sleep mode with the respective states receive and transmit, it is obvious that approximately half the energy of the capacitor is being used to power the ESB circuit, microcontroller unit, and memory. A node that is constantly in the sleep mode can live approximately twice as long as a node that is constantly in the receive state. The lifetime of nodes being constantly in the sleep state gives us upper boundary values for the energy measurements.

5.3 Measurement Results

5.3.1 Power Consumption

Figure 8 depicts the measured lifetime of an ESB node when applying the methodology described in Section 5.2. The lifetime of a node applying the WiseMAC medium sampling technique with basic cycle duration T = 500 ms and 10% duty cycle is almost equal to the lifetime of a node with the permanently switched-off transceiver. Considering that the mechanism still allows reaching nodes within 500 ms, the cost for this connectivity is quite reasonable. When comparing the lifetime of the WiseMAC node to the lifetime of simple ScatterWeb CSMA, which keeps the transceiver permanently in the receive state, the lifetime could be increased by approximately 120 %.

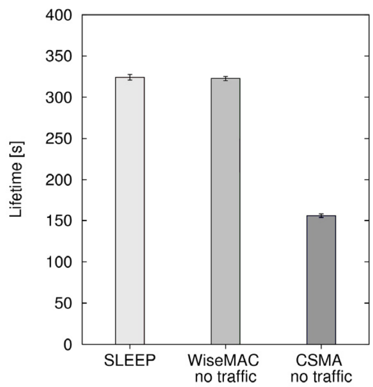

Figure 8: Power consumption for sleep mode, WiseMAC, ScatterWeb CSMA (receive mode)

The implementation parameters of the power saving WiseMAC protocol on ESB nodes listed in Table 2 led to stable and quite robust functioning of the prototype implementation on the ESB. We also implemented the more bit scheme and the extended more bit scheme using the stay awake promise on ESB nodes. Signaling to stay awake is achieved by altering a single bit in the MAC header. The buffer space for storing packets has been limited due to the small RAM in ESBs, and allows to store 20 frames. In case of buffer overflows, packets passed from the application layer are simply discarded.

Table 2. ESB prototype parameters

Basic interval duration T	500 ms
awake ratio	1%
retries	3
minimum preamble	5 ms
medium reservation preamble	uniform [0,6] ms
baud rate	19'200 bps
bit rate	9'600 bps
MAC header	104 bit
payload	96 bit
packet queue length	20

We measured the lifetime of WiseMAC in a chain scenario consisting of six nodes. Figure 9 depicts the lifetime of the selected ESB node 5 as a function of the traffic rate r when being charged with the initial amount of energy. As the node's energy consumption increases with increasing traffic along the chain, a more or less linear decrease of the node's lifetime can be observed. The lower curve in Figure 9 displays the lifetime of a node using ScatterWeb CSMA. ScatterWeb CSMA keeps the transceiver constantly in the receive state, applying no energy-saving scheme such as periodic switching between sleep and active states. As sending and receiving is more or less equally expensive, the traffic has no big impact on the lifetime of nodes.

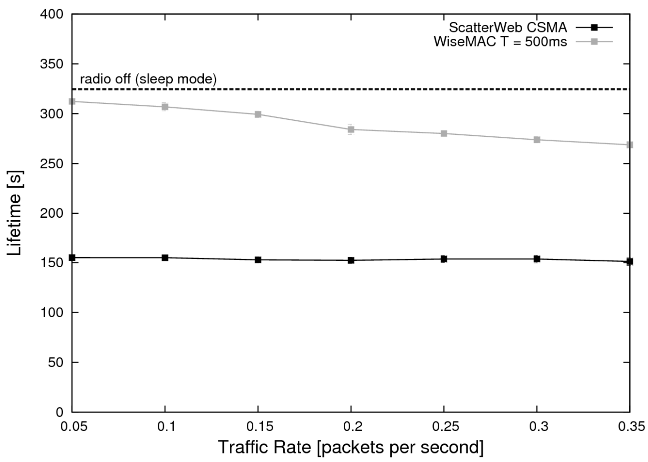

Figure 9. Lifetime of WiseMAC

5.3.2 Throughput

In another experiment, we measured the throughput of the two schemes more bit and extended more bit when generating traffic of equal rate from two senders to one receiver. When both senders aim to concurrently forward packets to the receiver, the receiver becomes a bottleneck, as both nodes aim to concurrently transmit their packets during the limited wake-up intervals. With the extended more bit scheme, the receiver node promises to stay awake for at least T = 500 ms by a single bit in the acknowledgement frame. Figure 10 shows the measured throughput in the given scenarios. The x-axis corresponds to the traffic generated by each of the two nodes.

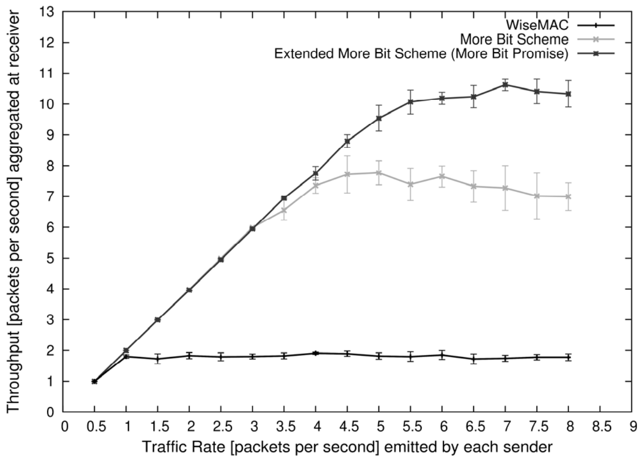

Figure 10. Throughput for WiseMAC, More Bit and Extended More Bit on ESBs

The WiseMAC protocol without the more bit scheme can only deliver one packet per wake-up, and therefore, throughput is limited to two packets per second with a basic cycle duration T = 500 ms. When increasing the rate, packets are subsequently queued in the buffer. When the buffer is full, packets are simply dropped.

When two stations apply the (extended) more bit scheme, they can alternately empty their transmit buffers by packet bursts for increasing traffic. More than one packet can be sent during each wakeup interval of the receiver. The sending station receives packets from its application layer and buffers them until the receiver node's next wake-up. The sender then transmits frames with the more bit set, listens for the acknowledgement and continues sending the next packet in line, until its buffer is empty.

By applying the (extended) more bit scheme, we could increase the throughput to much higher values. The throughput reaches nearly 8 packets per second for the more bit scheme and exceeds 10 packets for the extended more bit scheme using the stay-awake promise bit. The throughput increase for the extended more bit scheme compared to the original more bit scheme exceeds 20 %. The throughput improvement of 20 % is similar to what has been achieved in the simulation scenario as shown in Figure 6, although the measurement scenario is much simpler.

6. Conclusions

The paper evaluated WiseMAC and extensions on ESB sensor nodes. The measurement results underline and usefulness of the energy-conserving WiseMAC compared to a MAC scheme without integrated power saving mechanism. The paper evaluated two schemes to improve throughput in scenarios with multiple senders and bottleneck destination nodes. The results obtained in simulation and sensor testbed confirm that the extended more bit basing on the so-called stay-awake-promise performs better than the original WiseMAC more bit scheme. The superior performance of 20% has been found similar in both simulation and real-world experiments.

7. References

[1] El-Hoiydi, A.; Decotignie, J.-D.: WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks, ALGOSENSORS 2004, First International Workshop on Algorithmic Aspects of Wireless Sensor Networks

[2] András Varga: The OMNeT++ Discrete Event Simulation System (http://www.omnetpp.org), European Simulation Multiconference (ESM'2001), Prague, Czech Republic, June 2001

[3] Mobility Framework for OMNeT++ (http://mobility-fw.sourceforge.net)

[4] RF Monolithics: Datasheet for TR1001 868.35 MHz hybrid transceiver, http://www.rfm.com/products/data/TR1001.pdf

[5] Schiller, J. Liers, A. Ritter, H. Winter, R. Voigt, T.: ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing, 38th Annual Hawaii International Conference on System Sciences, 2005

[6] Staub T., Bernoulli T., Anwander M., Waelchli M., Braun T.: Experimental Lifetime Evaluation for MAC Protocols on Real Sensor Hardware, ACM Workshop on Real-World Wireless Sensor Networks REALWSN, 2006

[7] Ritter H., Schiller J., Voigt T., Dunkels A., Alonso J.: Experimental evaluation of lifetime bounds for wireless sensor networks, European Workshop on Wireless Sensor Networks EWSN, 2005

[8] Philipp Hurni, Torsten Braun: Increasing Throughput for WiseMAC, IEEE/IFIP WONS 2008 , Garmisch-Partenkirchen, Germany, January 23 - 25, 2008