Increasing Throughput for WiseMAC

Philipp Hurni, Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern

***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 detect and report an event to the base station. The paper proposes to extend and improve the optional more bit mechanism in WiseMAC. It allows bottleneck nodes to stay awake in case of high traffic instead of strictly sleeping periodically. Simulations show the effectiveness of the mechanism.**

***Index Terms*—Wireless sensor networks, Medium access control, Performance and throughput optimization**

I. Introduction

Energy efficiency is the main concern in wireless sensor networks. Many medium access control (MAC) protocols have been designed, implemented and evaluated for wireless sensor networks (WSNs). The Wireless Sensor MAC (WiseMAC) [1] [2] protocol has proven to be very energy-efficient. It 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 occurs with rather low traffic rates already. The reason is that in tree-based wireless sensor network scenarios, nodes receiving traffic from several sources might become a bottleneck node 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 in point-to-point scenarios. This paper proposes to extend this more bit mechanism by allowing a bottleneck to adapt its duty cycle in case of high traffic and to serve multiple neighbor nodes.

Section II describes the basic WiseMAC protocol and its more bit extension. Section III introduced the extended more bit mechanism, also called stay awake promise. Section IV discusses performance evaluation results and Section V concludes the paper.

II. WiseMAC

A. Basic Medium Access Mechanism

The WiseMAC protocol belongs to the unscheduled wireless sensor MAC protocols. WiseMACs wake-up scheme consists of periodic duty cycles of only a few percent in order to sense the carrier for a preamble signal, as depicted in Figure 1. All nodes in the network sample the medium with a common basic cycle duration T, but their wake-up patterns are independent and left unsynchronized.

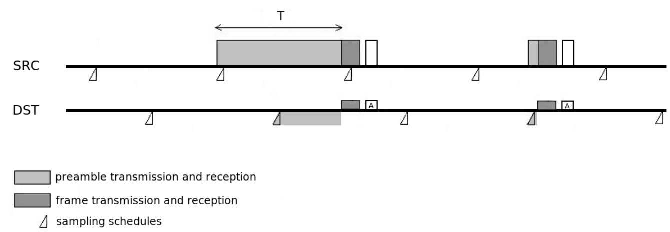

Fig. 1: WiseMAC

When transmitting a frame, a preamble of variable length is prepended for alerting the receiving node in its wake-up interval not to go to the sleep state. When the receivers 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 preamble is a simple bit sequence indicating an upcoming transmission to the nodes neighborhood. The own schedule offset is then piggybacked to the frame and transmitted to the receiver.

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 patterns of all its neighbors, which in turn allows minimizing the preamble length for the upcoming transmissions. As the sender node is aware of the receivers wake-up pattern, it only prepends a preamble that compensates for the maximum clock drift that the two involved nodes clocks may have developed during the time since the last schedule exchange. As illustrated in Figure 1 in the second transmission, WiseMAC minimizes the preamble and calculates its duration as follows:

$$T_{preamble} = min(4\theta L, T)$$

θ denotes the quartz oscillator clocks drift, L denotes the time since the last update of the neighbors wake pattern and T denotes the common basic cycle duration.

B. More Bit

Aiming to make the WiseMAC protocol more adaptive for varying or increasing traffic load, we first investigated the idea to let nodes react with increasing (e.g., by doubling) duty cycles when the traffic load exceeds a certain threshold. We soon detected that an increase of the duty cycles alone does not help at all, but only causes additional costs. If a node increases its duty cycle, it has to notify this increase to its neighbors. The approach did not lead to improved traffic adaptivity but to massive control message overhead. More energy was spent for control messages and overhearing. Delay and throughput could not be improved, but the mechanism performed even worse than the WiseMAC scheme. Therefore, we had to investigate other schemes.

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 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 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.

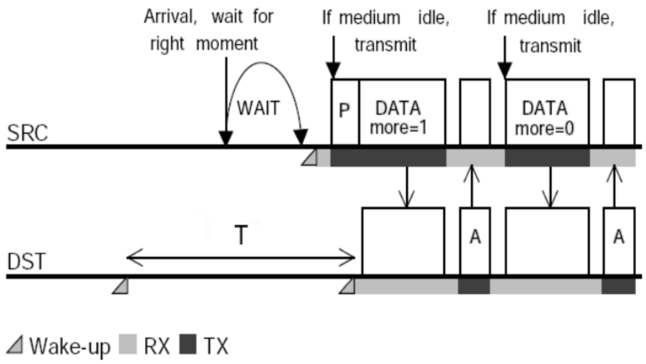

Fig. 2: More bit in WiseMAC

III. Improving WiseMAC by Stay Awake Promise

We tried to adapt the more bit idea one step further, because the scheme only serves to improve traffic adaptivity between one sender and one destination, and therefore has its limitations. In a wireless sensor network scenario, there are often nodes that have to forward data from large subtrees. 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 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. We aim to develop a strategy, where nodes will automatically stay awake for a longer time than just the sampling period when more traffic has to be handled and tell this to all nodes waiting to forward traffic to it. We therefore 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. 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 some a basic wake interval T. As SRC2 has lost the contention, it will wait and overhear the transmission 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, as it knows that DST will stay awake as it has given the promise with the acknowledged more bit. The advantage of this scheme is that no time is wasted for waiting, as the transmission of SRC1 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 a signal of increased load. The scheme is not applied after every unicast transmission, because it would lead to huge energy waste. The WiseMAC more bit mode and the extended more bit are evaluated in Section IV.

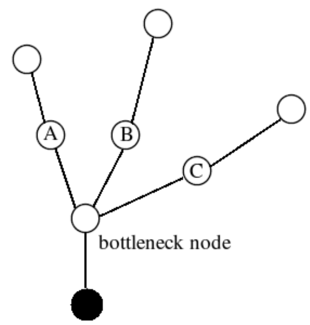

Fig. 3: Tree structure in a wireless sensor network

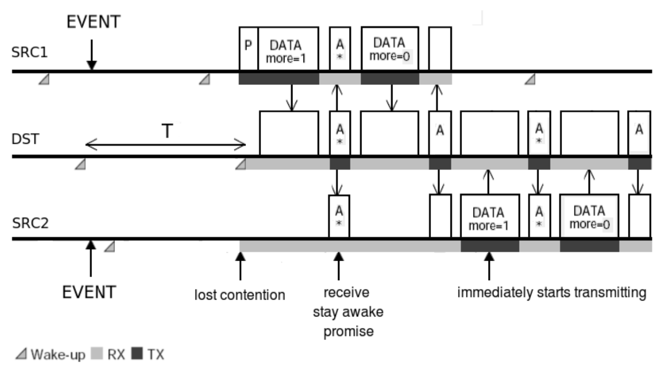

Fig. 4: Extended more bit based on stay awake promise

IV. Performance Evaluation

A. Performance Evaluation Scenario

For performance evaluation, we used 90 nodes uniformly distributed in an area of 300 m x 300 m. Traffic using a Poisson model is generated at each node and sent towards a single sink. We use static shortest path routing. Each node uses a basic interval T = 250 between 2 wake-ups and a duty cycle of 5%.

For the performance evaluation we used the OMNeT network simulator [6] and the mobility framework [7], which supports simulations of wireless ad hoc and mobile networks on top of OMNeT. This framework incorporates a sophisticated transmission model which is based on calculation of SNR (Signal-to-Noise Ratio) and SNIR (Signal-to-Noise-and-Interference Ratio) values according to a restricted free space propagation model. The Mobility Framework calculates the calculates the received power P_r on a node at distance d to $P_r(d) = P_t\lambda^2/((4\pi)^2 d^\alpha)$ where P_t is the transmitted signal power, λ the wavelength of the signal and α the path loss coefficient. The radio propagation model does not take multipath propagation or Doppler effects into account, but allows adjusting the path loss coefficient . Recent examinations of the signal attenuation in IEEE 802.11-based networks [9] conclude that a path loss coefficient between 3 and 4 is most suitable to model wireless propagation in office buildings and outdoor areas. Similar to other sensor network simulations (i.e. [10]), our simulations are based on a path loss coefficient of α = 3.5.

path loss coefficient α	3.5
carrier frequency	868 MHz
transmitter power	0.1 mW
SNR threshold	4 dB
sensitivity	-101.2 dBm
sensitivity carrier sensing	-112 dBm
communication range	50 m
carrier sensing range	100 m

TABLE I: Simulation parameters

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 and not taken into account. For performance evaluation of power-saving MAC and protocols one has to carefully model the transceivers energy consumption in its respective operation modes and state transition phases, as well as the transition delays and their respective costs. 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 [11]. Table 2 indicates the input current (supply voltage: 3V) 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, i.e. when switching from sleep to receive, the cost during the 518 μs is equal to the same amount of time spent in the receive state. Then, the sum of the energy consumption simply equals the sum of the energy spent in the respective states.

current:	
send	12 mA
recv	4.5 mA
sleep	5 μA
state transition delays:	
recv to send	12 μs
send to recv	12 μs
sleep to recv	518 μs
recv to sleep	10 μs
send to sleep	10 μs

TABLE II: Energy and switching simulation parameters

B. Performance Results

WiseMAC proved to be suitable and more energy-efficient than existing MAC protocols under variable traffic conditions, such as S-MAC [3] or T-MAC [4]. 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, which is a desirable property.

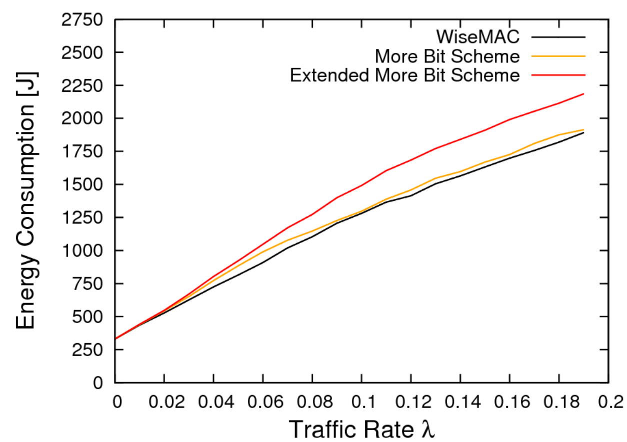

Fig. 5: Energy consumption

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 gain access to the medium, because there is another station sending to the node, 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 a vast impact on the maximum traffic rate. By limiting the duration of the sleep intervals to only a few percent of the cycle interval, the boundary values for the maximum traffic rate are determined. Therefore, the bandwidth achievable with the basic WiseMAC scheme is limited and exceeding that limit results in higher packet loss as shown by Figure 6 and Figure 7. Figures 5-8 show that an increase of maximum throughput is possible with both the more bit and the stay awake promise bit. Compared to the more bit scheme, the stay awake promise approach is superior in terms of all considered performance metrics such as throughput, packet loss, and one-way delay. However, the improved performance comes with higher energy costs. But when we consider the ratio of throughput and energy, the stay awake promise scheme is even better than the more bit scheme for high traffic.

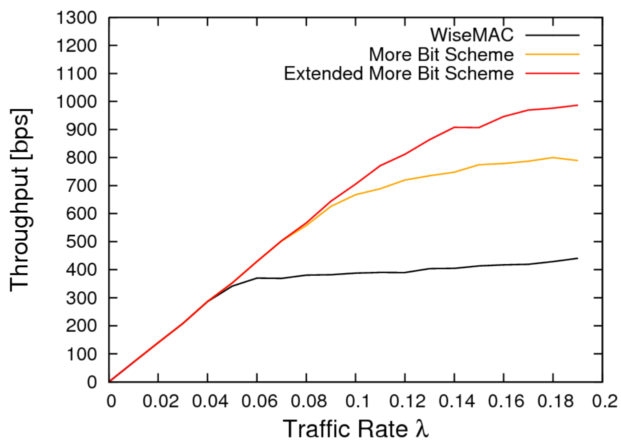

Fig. 6: Throughput

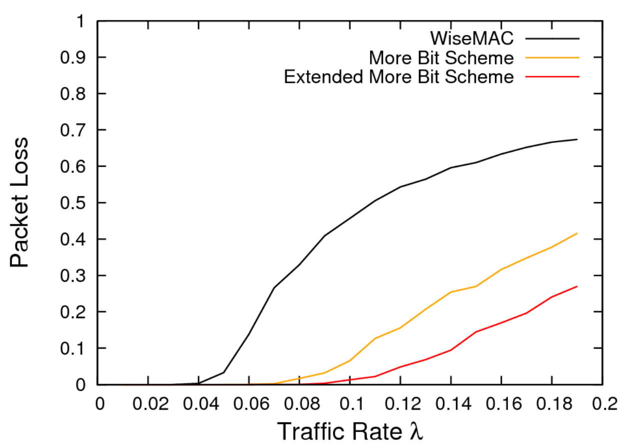

Fig. 7: Packet loss

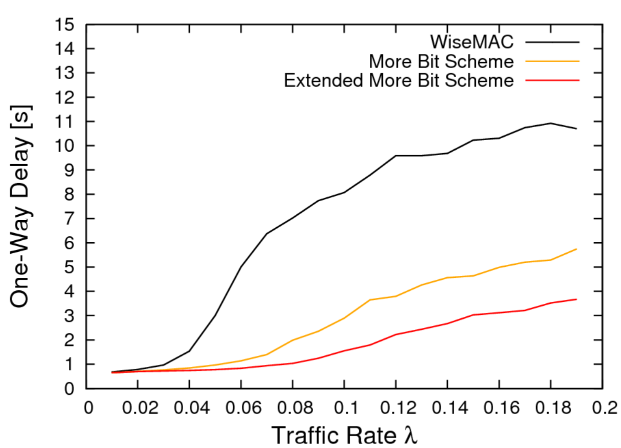

Fig. 8: One-way delay

V. Conclusion

The paper proposed a scheme to improve the possible throughput of the WiseMAC protocol in wireless sensor networks. The scheme is based on allowing bottleneck nodes in a wireless sensor networks to stay awake for a longer time and to forward traffic for several neighbor nodes by dynamically adjusting the duty cycle when needed. The scheme shows clearly benefits in terms of throughput, packet loss and delay. This comes at the cost of somewhat increased energy usage. We intend to implement the scheme on real sensor nodes [8] and confirm the simulation results by real measurements. Another issue is to investigate other algorithmic variants of the stay awake scheme, e.g., to trigger the scheme not only in case of fragmented data, but in other detected overload situations.

References

[1] El-Hoiydi, A.; Decotignie, J.-D.: *WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks*, ALGOSENSORS, 2004

[2] El-Hoiydi, A.: *Energy Efficient Medium Access Control for Wireless Sensor Networks*, PhD Thesis, École Polytechnique Fédérale de Lausanne, 2005

[3] Ye, W., Heidemann, J., Estrin, D.: *An Energy Efficient MAC protocol for Wireless Sensor Networks*, IEEE Infocom, 2002

[4] Dam, T.V., Langendoen, K.: *An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks*, ACM Conference on Embedded Network Sensor Systems (SenSys), 2003

[5] Demirkol, I., Ersoy, C., Alagoz, F.: *MAC Protocols for Wireless Sensor Networks: A Survey*, IEEE Communications Magazine, 2006

[6] Varga, A: *The OMNeT++ Discrete Event Simulation System (http://www.omnetpp.org)*, European Simulation Multiconference (ESM2001), Prague, June 2001

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

[8] 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

[9] Faria, D.B.: *Modeling Signal Attenuation in IEEE 802.11 Wireless LANs*, Technical Report TR-KP06-0118, Stanford University, 2005

[10] Ganeriwal, S., Kumar, R., Srivastava, M. B. *Timing-sync protocol for sensor networks* ACM Conference on Embedded Networked Sensor Systems (SenSys), 2003

[11] RF Monolithics: *Datasheet for TR1001 868.35 MHz hybrid transceiver* (http://www.rfm.com/products/data/TR1001.pdf)