

Calibrating Wireless Sensor Network Simulation Models with Real-World Experiments

Philipp Hurni and Torsten Braun

Institute of Computer Science and Applied Mathematics
University of Bern
{hurni, braun}@iam.unibe.ch

Abstract. This paper studies the energy-efficiency and service characteristics of a recently developed energy-efficient MAC protocol for wireless sensor networks in simulation and on a real sensor hardware testbed. This opportunity is seized to illustrate how simulation models can be verified by cross-comparing simulation results with real-world experiment results. The paper demonstrates that by careful calibration of simulation model parameters, the inevitable gap between simulation models and real-world conditions can be reduced. It concludes with guidelines for a methodology for model calibration and validation of sensor network simulation models.

Keywords: Wireless Sensor Networks, Energy Efficient Medium Access Control, Model Validation, Model Calibration, Networking 2009.

1 Introduction

In the past years, many energy efficient medium access and routing protocols for wireless sensor and actor networks have been proposed. Although few protocols have been implemented and evaluated on real sensor hardware testbeds, countless papers rely on network simulation results. Simulation tools are a valuable, manageable and yet cheap test environment for evaluating wireless sensor network mechanisms. Investigations on scalability issues, e.g studies on the behavior of network characteristics in large-scale ad hoc or wireless sensor networks, would simply be unfeasible without simulation tools. Simulations provide essential insights when developing and evaluating protocol mechanisms. Regrettably, evaluations on real-world systems are often neglected. Simulating performance improvements in protocols for wireless ad-hoc or sensor networks undoubtedly is easier and more convenient than realizing and proving them on real-world prototypes.

Simulation studies inevitably take assumptions and apply simplified models, e.g. 2-dimensional topologies and perfectly circular transmission ranges. The properties of the wireless channel, such as signal dispersion, environmental noise, multipath propagation, scattering and fading effects are often not incorporated at all. Model verification using real-world implementations is often omitted. The credibility of simulation studies has therefore frequently been questioned. Several

recent studies underlined the lack of rigor in the application of simulation tools. Inadequate simulation models and improper data analysis have been shown to produce inconsistent or misleading results. *Kurkowski et al.* [1] lately surveyed papers published in the MOBIHOC conference [2] between 2000 and 2005 and found severe flaws and inconsistencies in the simulation methodology. The survey concludes that the large majority of the research papers are not independently repeatable because of lack of documentation, omitted simulation input parameters, lacked statistical validity, inappropriate radio models and unrealistic traffic and/or mobility models. They emphasize that in any case of communication protocol study, researchers must validate the simulation model as a baseline to start any experimentation. *Andel et al.* similarly criticise the lack of realism of simulation studies in [3]. They emphasize that “properly validating simulation models against the intended or real-world implementation and environment can mitigate many of the problems of simulation”, such as incorrect and unrealistic parameter settings and improper level of detail.

In this paper we analyze the energy-efficiency and service characteristics of a recently developed energy-efficient MAC protocol. By conducting the same experiments in simulation and on real sensor hardware and reasoning over anomalies in the results, we outline a methodology to calibrate and validate wireless sensor network simulation models. We show that model validation and calibration is feasible with reasonable effort. By cross-comparing and simulation and real-world implementation of the energy-efficient MAC protocol, we show that careful investigations on parameter settings play a major role.

The paper first portrays the mechanism of the power saving protocol WiseMAC [4] in section 2.1. Section 2.2 describe the simulation model of WiseMAC, and 2.3 the WiseMAC prototype on real-world sensor hardware. Section 3 evaluates the protocol in a simple test scenario, and illustrates how we carried out our proposed model validation and parameter calibration process on the WiseMAC simulation and real-world implementation. Section 4 concludes the paper.

2 WiseMAC

2.1 WiseMAC Protocol Operation

WiseMAC [4] is an unscheduled, contention-based sensor MAC protocol. It is very energy-efficient in scenarios with low or variable traffic. WiseMAC’s wake-up scheme consists of periodic duty cycles of only a few percents 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 interval T , but their wake-up patterns are independent and left unsynchronized. A preamble of variable length is prepended to each frame in order to alert the receiving node. When the receiver’s wake-up pattern is yet unknown, the duration of the preamble equals the full basic cycle interval 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. After successful frame reception, the receiver

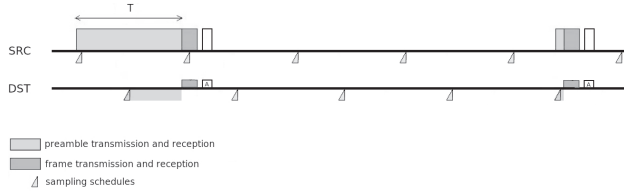


Fig. 1. WiseMAC

node piggybacks its own schedule to the respective frame acknowledgment. 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 receiver’s wake-up pattern, it only prepends a preamble that compensates for the maximum clock drift that the two involved node’s clocks may have developed during the time since the last schedule exchange, as illustrated in Figure 1 in the second transmission.

2.2 WiseMAC in the OMNeT++ Simulator

We implemented the WiseMAC protocol [4] in the OMNeT++ Network Simulator [5] using the Mobility Framework [6], which supports simulations of wireless ad hoc and mobile networks on top of OMNeT++. It calculates SNR (Signal-to-Noise) ratios according to a free space propagation model.

The energy consumption model calculates and sums up the amount of energy that is used by the transceiver unit. [7] models the energy consumption of a IEEE 802.11 wireless device with a transceiver state model with the three states sleep, idle, receive and transmit. Experimental results in [7] confirm the adequateness of the state model with three different energy consumption levels. As many low-power and low-bandwidth transceivers used in sensor networks are of very low complexity, the energy consumption in idle and receive mode is often almost equal and do not need to be treated differently. Pursuing the same methodology as [7], we modelled the energy consumption of the sensor nodes with a state transition model with respect to the time spent in three operation

Table 1. OMNeT++ parameters

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

Table 2. Transceiver parameters

supply voltage	3 V
transmit current	12 mA
recv current	4.5 mA
sleep current	5 μ A
recv to transmit	12 μ s
transmit to recv	12 μ s
recv to transmit	518 μ s
recv to sleep	10 μ s

Table 3. WiseMAC parameters

basic interval duration T	500 <i>ms</i>
duty cycle	1%
bit rate	19'200 <i>bps</i>
minimum preamble	1 <i>ms</i>
medium reservation preamble	u(0,2) <i>ms</i>
MAC header	104 bit
payload	96 bit

modes sleep, receive and transmit, weighted with the respective energetic costs. In a first step we applied the transceiver parameters of the TR1001 low-power radio transceiver module [8] (transmission rate, state transition delays, power consumption in sleep/recv/transmit states). The TR1001 is the radio chip of quite a few sensor nodes, including our own sensor hardware testbed. The parameters of the simulation environment and the energy consumption model are listed in Tables 1 and 2. WiseMAC-specific parameters are listed in Table 3.

2.3 WiseMAC on Embedded Sensor Boards

The simulation environment described in section 2.2 is an attempt to model a wireless sensor network, with respect to effects of signal dispersion, environmental noise, bandwidth limitation, energy constraints and clock drifts. Yet many other aspects that may play a role for wireless sensor networks are still left aside. Only measurements on real hardware make sure that all influences and side effects are taken into account. We therefore ported the original WiseMAC mechanism to the Embedded Sensor Boards (ESB) [9], a sensor node platform along with its own operating system, ScatterWeb OS [10]. WiseMAC requires a very accurate timing in order to keep track of and reach nodes in their particular wake-ups. In the presence of several sources of imprecisenesses (e.g. clocks drifts, software-based timers), this proved to be a challenging task. Yet the main features of WiseMAC outlined in [4] could be realized. We chose the same essential parameters of the WiseMAC simulation model as listed in Table 3.

Frame Transmission. When a packet has to be sent, the network handler determines whether the frame receiver is already known and its schedule offset is already stored in the WiseMAC neighbor table. If the medium is free, the node calculates the necessary preamble duration, contends for medium access, switches to the transmit state and transmits preamble and frame subsequently. If the medium is not free, the node turns to the sleep state again and schedules the next transmission attempt for the next wake-up of the receiver. In case the receiver is unknown yet, the preamble duration is set to the basic interval duration T and the transmission attempt is initiated immediately.

Preamble Sampling and Frame Reception. As the transceiver switches need a certain turnaround time, and carrier detection is bound to the recognition

of a sequence of predefined start bytes, nodes need a certain minimum duty cycle to actually recognize whether a preamble is being sent. The duration of the wake duty cycle calculates as $\Delta t = T \cdot \text{dutycycle} = 5\text{ms}$. In fact, 5ms is only the time between the instants when the transceiver switches are initiated. The transition delay for changing from sleep state to the wake state has to be subtracted from the duty cycle. The net duty cycle therefore is in the range of only $3 - 4\text{ms}$ in each interval T . If a node does not recognize the start byte sequence within its duty cycle Δt , it immediately returns to the sleep state until the next wake interval. If it recognizes the start byte sequence, it stays in the receive state until preamble and frame are correctly received. After reception, the node checks the type of the frame. In case of a broadcast frame, the node immediately returns to the sleep state. In case of a unicast frame, it returns a 10-byte acknowledgement and goes back to the sleep state.

3 Comparing Simulation and Real-World Experiments

3.1 Measurement Methodology on the ESB

We investigated the energy consumption of ESB nodes via measurements on the node lifetime. This methodology is widely accepted and has been used in [11], [12]. It consists in charging so-called GoldCap capacitors and measuring the time a node can live on this given charge. When two nodes are charged with the same initial amount of energy, the node with a lower overall energy consumption can live longer. This allows evaluating the energy consumption of sensor MAC protocols in small-scale test scenarios.

Applying the GoldCap methodology, we obtained robust results with low variance. The following results form an entry point to the lifetime measurements on the ESB. In a first step, we compared the node's energy consumption in the three different states of the transceiver (sleep, receive, transmit). Figure 2 depicts the lifetimes of nodes in the particular states. The first bar illustrates the lifetime of an ESB node with a permanently turned-off transceiver. The second bar illustrates the lifetime of an ESB node running ScatterWeb CSMA, which keeps the transceiver permanently in the receive state. The third bar corresponds

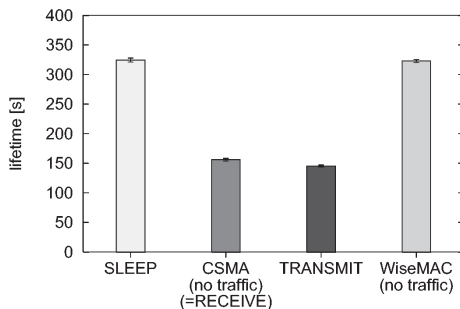


Fig. 2. Lifetimes of ESB nodes in different states

to a node in the most costly transmit state. As ESB nodes apply on-off keyed (OOK) modulation, the signal is simply turned on and off for bits '1' and '0', respectively. We therefore measured the transmit state (third bar) when sending a strictly alternating sequence of '1' and '0'. In regard of Figure 2, we can conclude that the energy consumption of the entire ESB node is highest in the transmit state. Receiving is almost equally expensive, and approximately twice as costly as the sleep state.

The fourth bar in Figure 2 illustrates the lifetime of WiseMAC nodes in the absence of traffic. It becomes obvious that the WiseMAC implementation on the ESB with only 1% duty cycle leads to a very low idle energy consumption. Its lifetime is almost equal to the lifetime of a node with the permanently turned-off transceiver. We measured a mean lifetime reduction of 0.47% with a standard deviation of 1.19% in respect to the average lifetime in the sleep state.

Comparing the lifetime of the WiseMAC node to the lifetime of simple Scatter-Web CSMA, the lifetime could be increased by approximately 120%. To the best of our knowledge, this WiseMAC prototype is the most energy-efficient implementation of a sensor MAC protocol implemented on the ESB nodes research platform.

3.2 Evaluation Scenario

We evaluated the energy consumption and basic properties of the ESB WiseMAC prototype with increasing traffic load. We have chosen a linear chain topology with six nodes forwarding traffic from one source to one sink, as depicted in Figure 3. The nodes are all in the transmission range of each other. They build up a full mesh topology, but only the bold links are used. In the measurements on the simulator and the real-world implementation on the ESB nodes, we measured the one-way delay and the time until intermediate node 5 depleted.

An external node synchronizes the nodes by emitting a SYNC packet, upon reception all the nodes set back their internal clocks. Node 1 starts generating traffic and addresses all frames to node 2. The application layer in node 1 generates a packet and logs the exact time. It then passes the packet to the MAC layer, which buffers it and sends it at the next appropriate instant. Node 2 receives the frame and subsequently forwards all frames to node 3, until the packets reach node 6. When node 6 receives the frame, it passes it to the application layer, where it is decapsulated and the transmission time of node 1 is extracted. Like this, the one-way delay is calculated as the time between the instant when the application layer in node 1 passes the packet to the MAC and the instant when the application layer

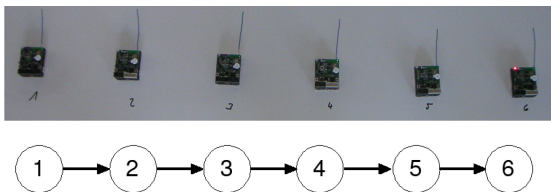


Fig. 3. 6 nodes chain topology

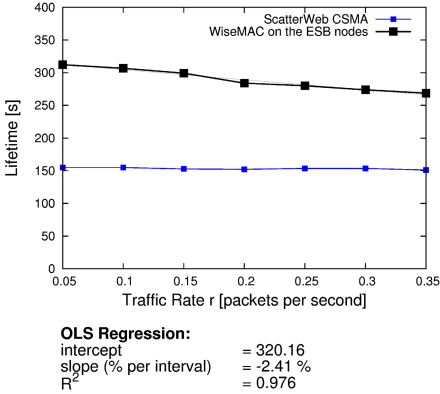


Fig. 4. WiseMAC & CSMA on ESB

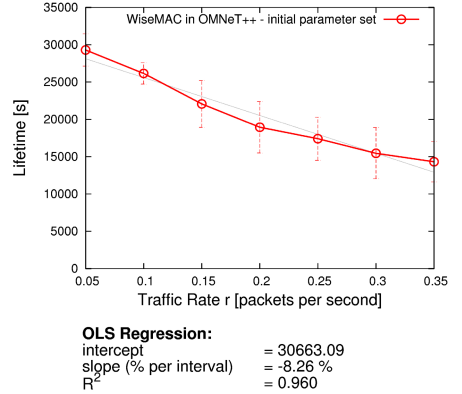


Fig. 5. OMNeT - initial parameters

in node 6 decapsulates the frame. The resulting inaccuracies of the one-way delay due to synchronization and packet processing are, given a 16 MHz processing unit on the ESB, in the range of some μs and can be considered negligible.

3.3 Node Lifetime in Simulation and Real-World Experiment

Many wireless sensor network simulation studies investigate on node lifetimes in the context of energy-balanced routing protocols, in most cases by assuming a linear battery model without self-discharge (e.g. [13]). We accordingly emulated the GoldCap lifetime methodology in the simulation model in OMNeT++, and measured the time until the intermediate node 5 depleted of an initial energy endowment of 20 Joules.

Figure 4 depicts the lifetime of node 5 applying WiseMAC on the ESB measured with the GoldCap capacitors. Figure 5 depicts the lifetime of the same node in the same simulation experiment in OMNeT++. The y-axis corresponds to the traffic rate r being generated by the source node 1. In Figure 4 the lifetime is measured as the time the intermediate node 5 can live of its initial GoldCap charge ($t_{charge} = 120s$). We focus on the slope of the lifetime curves in Figures 4 and 5. With traffic increasing linearly, a linear decrease of the lifetime could be observed in both the real-world and the simulation experiment. We applied OLS regression analysis to measure intercept, slope (measured as the relative change per measurement interval in respect to the intercept) and the correlation coefficient R^2 to assess the *goodness of fit* of the linear model. Notice that the absolute values of the lifetimes are of no particular importance. As we could not assess the absolute value of the charge of the GoldCap capacitor, the absolute values of the lifetime curves can not be put into relation.

3.4 Impact of Inappropriate Simulation Parameters

Figure 5 displays the lifetime curve when applying the initially chosen parameters (c.f. Table 2) to the WiseMAC simulation in OMNeT++. As one can clearly see

by comparing Figure 5 to Figure 4, the impact of the traffic applied to the chain is much stronger in the simulation model than in the real-world implementation of WiseMAC. The WiseMAC ESB lifetime curve decreases with -2.41% of the intercept per measurement interval ($\Delta r = 0.05$). The high correlation coefficient R^2 approves the appropriateness of the linear model. Applying the same linear OLS regression model, we measured a slope of -8.26% with the curve of the simulation model and a similarly high correlation of the linear model.

We investigated on the reason for the much higher negative slope in the simulation experiment and found that some of the initial assumptions of the simulation model had been too simplistic: as we had relied the choice of the parameters only on the technical specification of the transceiver module (see Table 2), as done in many other simulation studies of wireless sensor MAC protocols, we had omitted the energy consumption of the CPU and board circuitry. With the initial parameter set listed on Table 2 in Section 2.2, receive and transmit states are approximately 1000 times more costly than the sleep state. These parameters only account for the energy spent by the transceiver unit, and neglect the energy consumed by CPU and board circuitry.

The sad truth is that the major portion of nowadays wireless simulation studies base on such inappropriate simulation parameters and assumptions. Relying on the scarce information of the wireless transceiver manufacturer’s datasheets and feeding those parameters into simulators has become the de-facto standard procedure of most wireless sensor MAC protocol studies. When comparing the slopes of lifetime curves of the real world experiment in Figure 4 with Figure 5, one must admit that the simulation model basing solely on the datasheet parameters of the transceiver chip does not yet deliver a reasonable energy model for sensor network simulations.

3.5 Simulation Model Calibration

As we measured the ratio between sleep and receive and transmit on the ESB nodes with the GoldCap capacitor methodology in Section 2 to be merely in the range of 1:2.25:2.5, we went on to calibrate the energy parameters of the simulation model accordingly. We find that the cross-comparisons between real-world

Table 4. Calibrated OMNeT++ simulation parameter set

supply voltage	3 V	recv to transmit	4 ms
transmit current	5.0 mA	transmit to recv	2 ms
recv current	4.5 mA	sleep to recv	1 ms
sleep current	2.0 mA	recv to sleep	1 ms
WiseMAC parameters:			
basic interval duration T	500 ms		
duty cycle	1%		
minimum preamble	5 ms		
medium reservation preamble	u(0,6) ms		

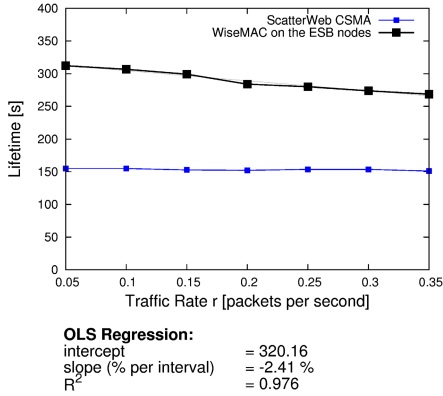


Fig. 6. WiseMAC & CSMA on ESB

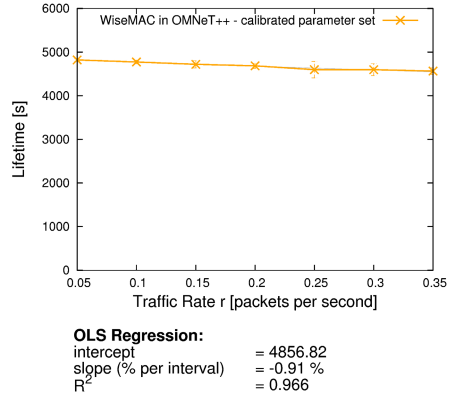


Fig. 7. OMNeT - calibrated parameters

implementation results of the WiseMAC ESB prototype and the OMNeT++ simulation results of Section 3.3 were of enormous value and helped to find suitable and realistic simulation parameters. We adjusted the energy consumption parameters of Table 2 to reflect the measurement results of the experimental evaluation of the nodes lifetimes in the different operation modes with the Gold-Cap methodology. Measuring the currents in receive and transmit modes with a customary multimeter is almost impossible because of high variation during signal (de)modulation. With the GoldCap methodology, we obtained stable average values for the energy consumption of the entire node in each state of the transceiver. We measured the node’s energy consumption in sleep state to roughly 2 mA, and accordingly estimated transmit and receive currents with 4.5 mA and 5 mA.

In a next step, we calibrated the state transition delays to realistically reflect the properties of the medium access mechanism of the ESB nodes. Switches generally require more time than indicated by the technical datasheet of the transceiver manufacturer [8]. To switch from receive to transmit, the network interface driver of the ESB nodes needs to go through different implementation-specific steps (e.g. disable certain interrupts, initialize and tune the radio interface). This procedure requires roughly 4 ms, whereas the datasheet of the transceiver only accounts for 12 μs. Similarly, switches from transmit to receive and from sleep to receive need more time. The calibrated parameter set containing all adjusted values is listed in Table 4.

3.6 Simulation Model Validation

For the ease of illustration, Figure 6 again depicts the measurements on the ESB nodes. One can clearly see the astonishing impact of the adaptation of the simulation parameters in Figure 7. The figure depicts the simulation results when applying the *calibrated* parameter set. The lifetime decreases only slowly with a slope of -0.91% per measurement interval, and reflects the real-world

measurements of Figure 6 far better than the curve obtained with the initial datasheet-based parameter set in Figure 5. Both Figures 6 and 7 illustrate a similarly low lifetime decrease with increasing traffic along the chain, as well as a similarly high correlation coefficient R^2 . As transmitting and receiving is only twice as expensive as the sleep state, the increasing traffic has a lower impact on the curve, unlike with the initial datasheet parameters of Table 2. The difference between the slopes of the real-world curve of -0.91% and the calibrated parameter set in Figure 7 of -2.41% can be explained by the retransmissions which occur in the real-world experiment due to the inherently unreliable channel, and the absence of retransmissions in the small-scale simulation, as well as the slight self-discharge of the GoldCap capacitors in the real-world experiment.

We underline that the investigations on the energy-consumption and transition delays of the real-world testbed noticeably paid off, as calibration and validation of the simulation model led to simulation results which come very close to the real-world experiment results. The ratio between the energy consumption of the transceiver states proved to be the decisive parameters for the slope of the lifetime curve. With the calibrated parameter set, we definitely obtained a more realistic energy consumption model for WiseMAC on the ESB nodes than with the initial datasheet-based parameter set.

3.7 One-Way Delay

Figure 8 depicts the one-way delay of the packets sent along the six nodes chain, measured both on the ESB prototype and in the OMNeT++ simulation. As expected, the delay proved to be independent from the examined traffic rates, as no congestion effects yet occur. Obviously, the results of simulation and the ESB implementation fit quite well. The per-hop delay of roughly $300ms$ could be obtained both in simulation and on the ESB nodes, and can be explained as follows: If a packet has to be sent, the sender node first waits for the next wake-up of the receiver. The expected time to wait for the next instant of the next

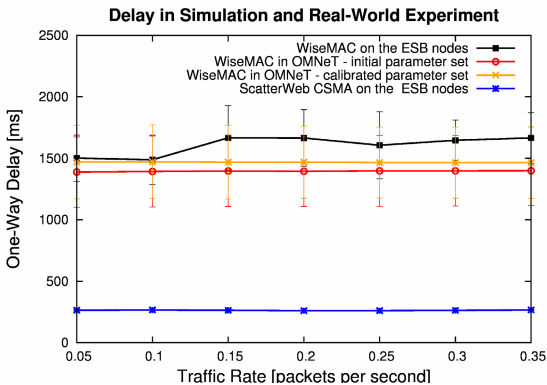


Fig. 8. One-way delays of WiseMAC in simulation and the ESB prototype

node is $E(t_{wait}) = \frac{T}{2} = 250ms$. The expected delay per hop $E(d_{hop})$ therefore consists of t_{wait} and the delays caused by the actual frame transmission and acknowledgement, i.e. the time for the medium reservation preamble t_{MRP} , the minimum preamble t_{MP} , the transmission delay of the frame t_f , the transceiver switches t_{rxtx} and t_{txrx} and transmission delay of the acknowledgement t_{ack} . With the examined traffic rates and a clock drift of $\theta = 30ppm$, the preamble does not yet exceed the minimal preamble t_{MP} of $5ms$.

We analytically obtain an expected per-hop delay $E(d_{hop})$ of

$$\begin{aligned} E(d_{hop}) &\cong E(t_{wait}) + E(t_{MRP}) + t_{MP} + t_f + t_{rxtx} + t_{txrx} + t_{ack} \\ &\cong 250ms + 3ms + 5ms + 20ms + 4ms + 2ms + 10ms \cong 294ms \end{aligned}$$

This results in a 5-hop delay of $1470ms$, which is the latency that was actually measured in simulation using the calibrated parameter set. The gap between the delays of simulation and real-world experiment has been slightly reduced by applying the calibrated parameter set in Figure 8.

The measurements of the one-way delays on the ESB are between $1501ms$ and $1665ms$, and differ from the simulation results by 2-13%. The reasons for the slightly higher values in the one-way delay of the ESB WiseMAC prototype are manifold. First, transmissions on the ESB still take longer as the calibrated parameters model it. There is an additional delay between frame transmission and acknowledgement reception, as ESB nodes first prepare and buffer the acknowledgement frame and then pass it to the network interface driver. Other implementation-specific issues may also play a role, i.e. the packet scheduling was implemented to include a safety margin of some milliseconds, such that the sender can carefully check the carrier before accessing it. In addition, retransmissions in the real-world implementation increase the average one-way delay. As nodes have to wait for $T = 500ms$ for the next duty cycle after each transmission attempt, retransmissions inevitably increase the one-way delay. We intend to integrate an error and packet loss model in future investigations, which however will require additional investigations on channel behavior and parameters (e.g. error model, packet error rate).

4 Conclusions

This paper illustrates by case study how cross-comparisons of real-world experimental results gained in small-scale experiments can to help to calibrate and validate wireless sensor network simulation models. We propose the following five steps as a baseline for simulation experiments with sensor network protocols:

1. Development and implementation of a simulation model with respect to the most important physical real-world and hardware constraints (e.g. signal dispersion, transceiver transition delays, energy consumption, etc).
2. Implementation of a prototype on real sensor hardware with the basic functionality of the proposed protocol mechanism.

3. Definition of small-scale experiment scenarios. Measurement and estimation of the prototype parameters (e.g. energy consumption, transition delays, bandwidth, packet error rate, etc) by analyzing the protocol behavior in small scale.
4. Calibration of the simulation model by reintegration of the measured or estimated real-world parameters.
5. Validation of the calibrated simulation model by cross-comparison with the real-world results of the small-scale experiments.

Careful investigations on realistic models for wireless ad-hoc and sensor network simulations are valuable and inevitable when confronting the legitimate scepticism against simulation studies. In order to achieve and provide confidence in own simulation studies' results, researchers should exert themselves to cross-compare their proposed ideas and mechanisms with at least small-scale real-world experiments. Although this might be more costly, time-consuming and exhausting than relying on pure simulation results, it leads to more valuable and more solid research results.

Mechanisms that only pay off and that can only be reproduced in simulation are of no particular value. Without any validation, simulations of wireless sensor networks mechanisms only produce unverifiable and possibly even misleading results. Anchoring the simulation model to real-world experiments undoubtedly increases trust and confidence into simulation results, although scalability effects might still be unaccounted for. The methodology applied in this case study shall therefore be a guideline for model calibration and validation of sensor network simulation models.

References

- [1] Kurkowski, S., Camp, T., Colagrosso, M.: MANET simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.* 9(4), 50–61 (2005)
- [2] MobiHoc: The ACM International Symposium on Mobile Ad Hoc Networking and Computing, <http://www.sigmobile.org/mobihoc>
- [3] Andel, T.R., Yasinsac, A.: On the Credibility of Manet Simulations. *IEEE Computer Magazine* (2006)
- [4] El-Hoiydi, A., Decotignie, J.D.: WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks. In: Nikolettseas, S.E., Rolim, J.D.P. (eds.) *ALGOSENSORS 2004*. LNCS, vol. 3121, pp. 18–31. Springer, Heidelberg (2004)
- [5] Varga, A.: The OMNeT++ Discrete Event Simulation System. In: *European Simulation Multiconference* (2001), <http://www.omnetpp.org>
- [6] Drytkiewicz, W., Sroka, S., Handziski, V., Koepke, A., Karl, H.: A Mobility Framework for OMNeT++. In: *3rd Intl. OMNeT++ Workshop* (2003)
- [7] Feeney, L.M., Nilsson, M.: Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. *IEEE INFOCOM* (2001)
- [8] RF Monolithics: Datasheet for the TR1001 868.35 MHz hybrid transceiver, <http://www.rfm.com/products/data/TR1001.pdf>

- [9] Schiller, J., Liers, A., Ritter, H., Winter, R., Voigt, T.: ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing. In: 38th Annual Hawaii International Conference on System Sciences (2005)
- [10] Schiller, J.H., Liers, A., Ritter, H.: ScatterWeb: A wireless sensor network platform for Research and Teaching. Elsevier Computer Communications (2005)
- [11] Ritter, H., Schiller, J., Voigt, T., Dunkels, A., Alonso, J.: Experimental evaluation of lifetime bounds for wireless sensor networks. In: European Workshop on Wireless Sensor Networks (EWSN) (2005)
- [12] Staub, T., Bernoulli, T., Anwänder, M., Waelchli, M., Braun, T.: Experimental Lifetime Evaluation for MAC Protocols on Real Sensor Hardware. In: ACM REALWSN (2006)
- [13] Chang, J.H., Tassiulas, L.: Maximum lifetime routing in wireless sensor networks. IEEE/ACM Transactions on Networking (2004)