Accurate Timeliness Simulations for Real-Time Wireless Sensor Networks

Jérôme Rousselot
Jean-Dominique Decotignie
Real-Time Software & Networking
CSEM
Neuchâtel, Switzerland
jerome.rousselot@csem.ch

Marc Aoun
Peter van der Stok
Philips Research
Eindhoven, The Netherlands
marc.aoun@philips.com

Ramon Serna Oliver
Gerhard Fohler
Chair of Real-Time Systems
TU Kaiserslautern
Kaiserslautern, Germany
serna_oliver@eit.uni-kl.de

***Abstract*—The use of wireless sensor networks is rapidly growing in various types of applications that benefit from spatially distributed data collection. Some of these applications, such as industrial automation, fire detection or health monitoring, have strong timeliness constraints. Since field deployments are difficult to monitor and debug, the development of real-time communication protocols for wireless sensor networks necessitates accurate simulation models.**

This paper presents open source Omnet++ simulation models based on the IEEE 802.15.4 standard. Four evaluation scenarios are used to compare simulation timeliness and packet error rate results with experimental measurements. The small scale of the scenarios allows to isolate the effect of each system component. The comparison validates the models for timeliness estimates in sensor networks and pinpoints the variability of software implementations in embedded systems as a major cause of differences between simulated and measured results.

I. Introduction

Research in communication protocols makes extensive use of network simulators such as NS-2 [1] since they allow fast evaluation and comparison of novel algorithms. In the past few years, advances in miniaturization led to the emergence of wireless sensor networks (WSN) [2], cooperative embedded computer systems designed to monitor one or more parameters in their environment. Protocols research for WSN aimed at developing scalable solutions with ultra low power consumption in order to reach an operating lifetime of several years. These protocols have been evaluated using the same kind of simulators. However, the fact that these simulators were originally designed for the study of wired networks has cast some doubts on the results validity [3].

This paper describes simulation models[1] of wireless propagation, multiple access interference, radio state machine and the IEEE 802.15.4 non beacon-enabled MAC protocol [4] which were implemented in the Mobility Framework [5] for Omnet++ [6]. We follow the approach recommended in [7] and focus on simple setups to better identify any possible cause of deviation. We define four evaluation scenarios which are realized experimentally and reproduced in the simulator. They consist of two topologies of respectively two and three nodes. Timeliness metrics and packet error rates are compared when using broadcast and unicast traffic, and the impact of traffic load is considered.

This paper is structured as follows. Section II describes related work on the validation of wireless network simulation models. Section III presents the models used for the simulations. Section IV gives details on the measurement testbed, on the evaluation scenarios and on the metrics. Section V discusses the results and section VI concludes the paper.

[1]Available online at http://www.github.com/mobility-fw.

II. Related Work

Cavin et al. [8] attempted to simulate a common minimalistic scenario with three simulators (OPNET, NS-2, Glomosim). They obtained widely varying results. They attributed these variations to different levels of precision of the physical and medium access control (MAC) layer models, and concluded that these tools were not useful to wireless application developpers. Later, Kotz et al. [3] compared NS-2 simulations with experimental results and identified six common simplifying assumptions in the model as causes of errors. More recently, Ivanov et al. [9] have shown through simulation, emulation and experimentation that realistic results can be obtained with the NS-2 wireless model if the simulation parameters are properly adjusted. While they obtained good results for the packet delivery ratio and the connectivity graph, packet latencies were inaccurate.

Colesanti et al. [7] studied the validity of simulations using the OMNeT++ discrete event simulator and the MAC Simulator framework [10]. They observed that the simulation results tended to over-estimate the testbed results, and that the performance depends on a multitude of possibly interacting parameters. They recommend to focus on simple scenarios to reduce the scope of the problem. Halkes et al. [11] have studied the accuracy of the MAC protocol performance in the TinyOS 2.x simulator TOSSIM as a function of the receiver model. They found that for the Signal to Noise Ratio (SNR) based reception model, the accuracy was often below 5% for the delivery ratio and for energy consumption. Latency results were less accurate. They attributed all remaining deviations to unmodeled fluctuations in the received signal strength.

To the best of our knowledge, there is no other work which evaluates the accuracy of latency estimates in wireless

Parameter	Value
bit rate	250 kbps
delaySetupRx	1.792 ms
delaySetupTx	1.792 ms
delayRxTx / delayTxRx	192 µs

Table I: TI CC2420 radio model parameters.

network simulations.

III. Models

This section presents the simulation models considered in this work. We used the discrete event simulator OMNeT++ and the Mobility Framework (a wireless network models library), to which we added our own models.

A. Radio

A TI CC2420 radio transceiver [12] was modeled using a detailed radio state model that includes transient states. While the default radio model of the Mobility Framework considers switching times between the steady states (Reception, Transmission, Sleep), these switching times depend only on the destination state. We introduced a radio state model (RadioAccNoise3) in which the switching time depend on both the current state and on the destination state. In particular, it takes less time to switch the radio to transmission mode when it is already in reception mode than when it is in sleep mode.

The timing and power consumption parameters used for the CC2420 are regrouped in table I. Timer values were taken directly from the datasheet. Power consumption values are obtained by multiplying the system voltage (3.3 V) by the current in steady state as defined by the datasheet.

When several transmitted signals arrive simultaneously at a receiver, it becomes more difficult to demodulate the information. Several types of models exist. We implemented a model that tracks the signal-to-interference-plus-noise ratio [13] during the whole frame reception (in SnrEvalRadioAccNoise3). The receiver model can then derive the bit error rate probability associated to each segment of the frame with a specific SNIR ratio (using an analytical model), and uses random numbers to decide wether or not each segment of the frame can be successfully demodulated (in DeciderRadioAccNoise3).

Concerning propagation, A free space pathloss model was used, with a path loss exponent of 2.5. While this is a very simple model that cannot account for multipath propagation and other complex propagation effects, it was considered enough to model our simple short distance and line of sight scenarios. More complex pathloss models are available in the simulator (log-normal, Rayleigh).

B. Medium Access Control

We implemented the IEEE 802.15.4 non beacon-enabled mode [4]. It is a Carrier Sense Multiple Access protocol with an exponential backoff window. Carrier sense protocols attempt to detect ongoing transmissions before emitting, in order to avoid collisions.

Parameter	Value
minBE	3
maxBE	5
maxCSMABackoffs	4
maxFrameRetries	3
AckWaitDuration	864 µs
SIFS	192 µs
aUnitBackoffPeriod	320 µs
CCADetectionTime	128 µs

Table II: IEEE 802.15.4 CSMA model parameters.

Each time that the protocol finds the channel busy, a backoff exponent counter BE is incremented, until it reaches a maximum value $maxBE$. This counter is used to determine when the next carrier sensing operation will be scheduled: an integer number is selected between 0 and $2^{BE}-1$, and this value multiplied by a backoff slot duration gives the backoff interval.

This operation is repeated a maximum of $macMaxCSMABackoffs$. If this threshold is reached, the transmission attempt is cancelled and a Channel Access Failure is reported to the upper layer.

If the protocol can successfully access the channel, the frame is transmitted. If the frame is a unicast transmission (addressed to a specific node), the protocol switches the radio into reception mode and waits for an acknowledgment for a maximum period of $macAckWaitDuration$ symbols. If no acknowledgment is received, a counter $FrameRetries$ is incremented and another transmission attempt is made, unless this counter reaches $MaxFrameRetries$.

If an acknowledgment is received or if the frame was broadcast, a transmission success is reported to the upper layer.

The values used for these parameters in our simulations are listed in table II, together with some key constants of the protocol.

C. Application

A simple application was considered. It takes a parameter T which is the time interval between two generated messages. First, a random time between 0 and T is selected, after which the first packet is sent. Then, after each time interval T, another packet is sent. The initial random time allows to avoid systematic collisions when more than one node are sending packets.

IV. Methodology

We defined four scenarios. The first two consider only one sender and one receiver, with a large time between two packets ($T=$100 ms). The other two scenarios consider two nodes sending packets to a common third node, and relatively high traffic rates (T =1, 5, 10, 25 ms). For each topology, we consider first broadcast traffic and then unicast acknowledged traffic. The packet size is always 60 bytes, and acknowledgments are 11 bytes long.

We set up these scenarios in a testbed using the Philips AG1 nodes (described below) and configured our simulator to reproduce these scenarios as closely as possible.

Traffic generating nodes recorded the time at which each packet was given to the MAC layer for transmission (this time is also recorded in the packet itself), and the time at which the MAC informed it of the transmission success or failure. We call the difference between these two times the *sender service time*. Similarly, we call *receiver service time* the difference between the moment the sender gives the packet to the MAC and the current reception time at the receiver. In addition, the number of channel access failures, received data frames, duplicates and acknowledgments are also recorded.

In the simulator, the data values were extracted by using the OMNeT++ API.

For the hardware platform, we used the Aquis Grain v1 (AG1) sensor nodes, developed at Philips Research. The AG1 nodes are equipped with an ATmega128L microcontroller and a TI/Chipcon CC2420 radio chip [12]. Time functionality in AG1 nodes is based on the oscillations of a crystal oscillator with a frequency f equal to 8Mhz. Local time is managed in software. A timer counter is incremented from 0, in steps of (1/f) µs, to generate an interrupt each 320 µs, corresponding to 1 backoff slot duration as defined in IEEE 802.15.4 [4]. Each time an interrupt occurs, a variable c is incremented by 1. To find the current local time of a node, the value of c is multiplied by 320 µs, and the current value d reached by the timer counter, where $0 \leq d < 320$ µs, is added to it. As the time measurements were made by the embedded software itself, they are dependent on the embedded system clock precision, on the processing speed, on the OS task scheduler and on interrupt routines.

V. Results

A. One source and one destination

Fig. 1 shows the empirical cumulative distribution functions for scenarios 1 (broadcast traffic) and 2 (unicast traffic).

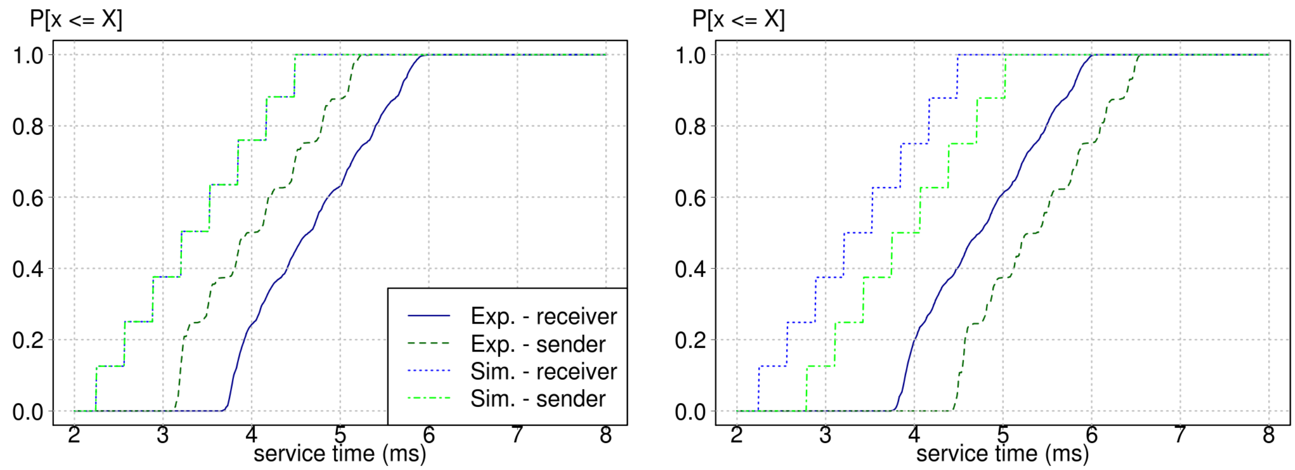
In the case of broadcast traffic (Fig. 1a), the simulated service time at the receiver and at the sender are more or less equal, to the point that the two lines completely overlap and are undistinguishable. This is due to the fact that the network simulator provides a single, network-wide time reference, and that processing times are not taken into account: while the source code of the simulation model is executed, the simulation time is stopped at a precise value, until all events scheduled at that simulation time have been processed. The simulation model does not attempt to capture processing times.

We also observe that eight values are possible for the simulated service time: approximately 2.25, 2.5, 2.9, 3.2, 3.5, 3.8, 4.2 and 4.5 ms. This is clearly due to the minBE parameter set to 3: the number of backoff slots is between 0 and $2^3 - 1 = 7$, thus 8 possible values with a slot size of 320 µs. The minimum service time of about 2.25 ms (when the backoff is equal to zero) can be explained as follows: an initial time to perform the clear channel assessment (128 µs), the time for the sender radio to switch from reception to transmission (192 µs) and the frame transmission at 250 kbps (1920 µs): 0.128+0.192+1.920=2.24 ms.

The testbed measurements lead to larger service times, around 1.5 ms more for the receiver and 0.9 ms for the sender. This can be attributed to a number of factors: software interrupts handling, MAC processing time, data exchanges between the radio chip and the microcontroller (through the SPI bus), system task scheduler, and software processing time of the measure itself. After the radio finishes receiving a frame, it informs the microcontroller through an interrupt request. This triggers the execution of an interrupt handling routine, which reads some control data from the radio concerning the frame such as its size and configures the SPI bus to send the frame to the microcontroller. Transferring a 60 bytes packet on a 2 MHz SPI bus alone takes more than 200 µs. After the MAC processing time, the timestamping of the packet at the application takes around 50 µs. While we cannot estimate the costs of each software routine, we attribute the remaining deviations to software issues. For instance, when the MAC sends the data frame to the application, it does so by creating a task in the system. The task handling code is executed every 320 µs and will cause here on average an additional delay of 160 µs. The variability of these causes also explains why the curve is smooth and why the discrete backoff values are barely noticeable. This delay can be reduced by using a system on chip platform, in which the radio directly writes the received bytes in central memory [14], and by optimising the inter process communications (such as between the MAC and the application).

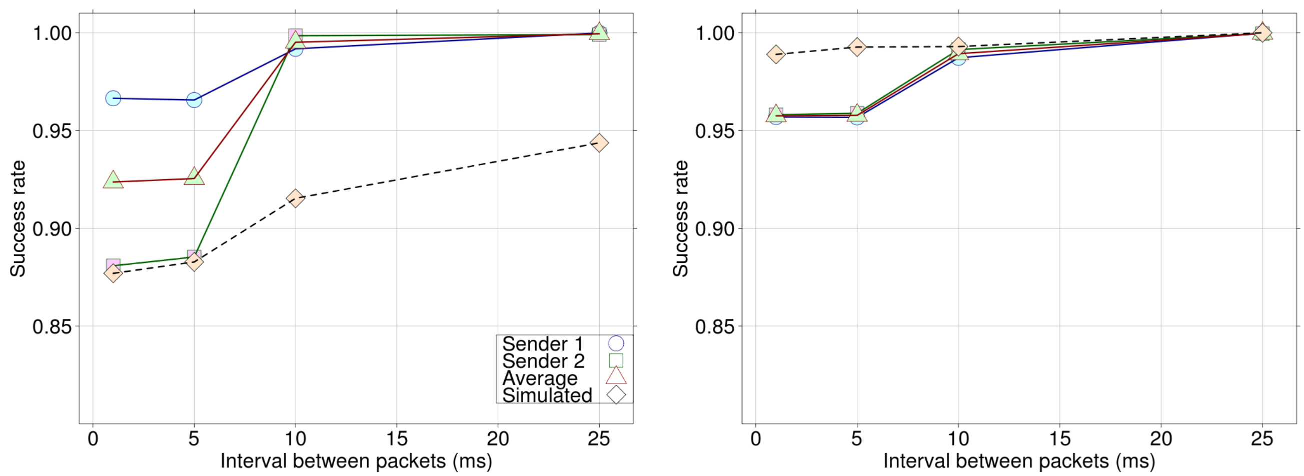
The smaller service time measured at the sender is due to the fact that the sender does not have to transmit the whole data frame back from its radio to the microcontroller. The discrete values of the backoff windows can also be seen on the measured sender service time.

On Fig. 1b, the simulation results are now clearly different between the source and the destination. This is due to the acknowledgment transmission time which is included in the service time experienced by the sender, while the receiver computes its service time immediately upon reception of the data frame at the microcontroller. The radio model takes 192 µs to switch from transmission to reception mode, and the transmission of an 11 bytes acknowledgment takes 352 µs, leading thus to a difference of 544 µs, in line with what we observe on the figure.



(a) Broadcast traffic. (b) Unicast traffic.

Figure 1: Cumulative distribution function of the service time at the sender and at the receiver. The impact of the slotted backoff algorithm can clearly be seen at the transmitter side and in simulation.



(a) Broadcast traffic. (b) Unicast traffic.

Figure 2: Success rates with two sources and one destination. The use of acknowledgments and retransmissions in the case of unicast traffic clearly improves the success rate and the protocol fairness.

Both the measured and simulated receiver service time do not differ from what we obtained in the first scenario. The measured service time of the sender increased slightly, but not as much as in the simulator. We explain this by a faster radio switching time.

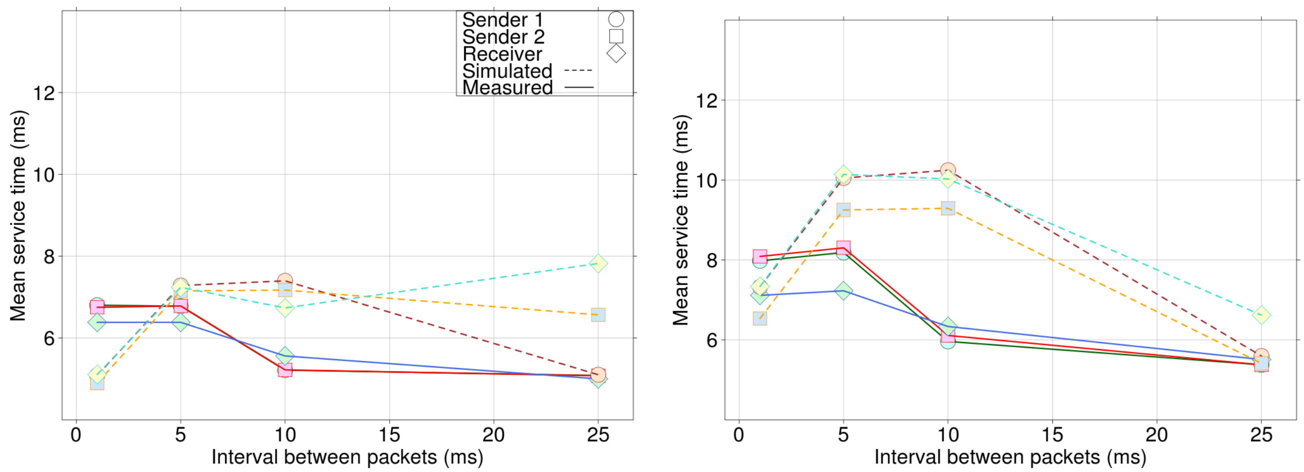
B. Two sources and one destination

We now consider the effect of sharing the medium between two source nodes trying to access a common destination node. First with broadcast transmissions, and then with acknowledged unicast transmissions.

Fig. 2a shows the transmission success rates measured at the receiver (without duplicate frames) from both nodes as well as the average value, as a function of the time between two packets at a sender (the total traffic rate is thus twice higher). The average success rate value is given for the simulation results.

The performance begins to degrade when considering intervals equal to or less than 10 ms. Even at 1 ms, the success rate remains good. Simulation results are slightly degraded but remain at about 5% of the measured success rate. We attribute the difference to conservative assumptions in the PHY layer model.

The service times are shown on Fig. 3a with the measured



(a) Broadcast traffic.

(b) Unicast traffic.

Figure 3: Mean service times with two sources and one destination. The increased reliability of unicast transmissions comes with a degradation of the mean service time.

data plotted using solid lines and the simulated data using a dashed line.

The measured service times increase with a higher traffic because the channel is found busy more often, leading to increased backoff durations. The simulated sender service times initially increases with the traffic load but decreases with the maximum traffic rate. We attribute this effect to contention access failures: the channel is often found busy, and a frame is dropped rather than transmitted. The time lost while trying to send such a packet is not included in the computation of the mean service time. The rate of contention access failures for each node can be seen on Fig. 4. We observe that this rate is higher in simulation than with the testbed, confirming this hypothesis. A higher traffic in the experiment would presumably lead to the same counter-intuitive result of decreasing service time.

Fig. 2b shows the transmission success rate with unicast transmissions. Acknowledged traffic leads to improved success rates, and this time the simulation results are even closer to measurements. With high data rates, we observe a receiver success rate higher than that of the senders. We attribute the difference to transmissions during which the data frame was correctly received but for which the acknowledgment was not received by the sender. It seems that this case did no occur as often, or at all, during the experiment. Longer measurements and simulation repetitions with different random seeds would probably lead to closer results.

The service times (shown on Fig. 3b) follow the behaviour previously observed with broadcast traffic. This time, the measured results actually begin to decrease, confirming our hypothesis in the previous section on the experimental results

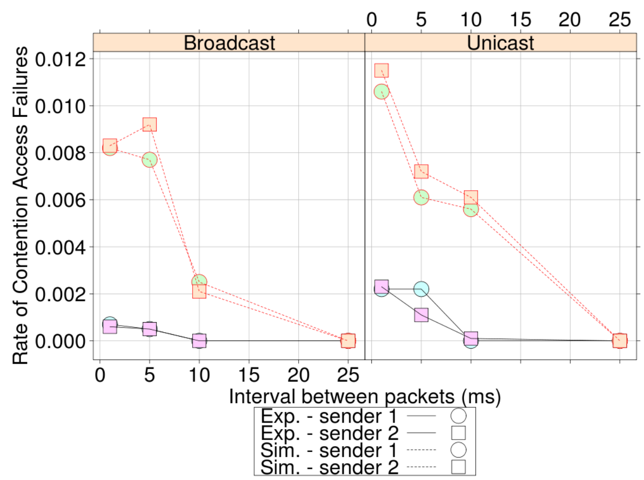


Figure 4: Rate of contention access failures for both senders (two sources and one destination). Experimental measurements shown with solid lines and simulation results in dashed lines.

behaviour with higher traffic: unicast traffic leads to higher channel usage. The higher rate of channel access failures (Fig. 4) confirms the increased channel usage compared to broadcast traffic. The mean service times are a few hundreds of microseconds higher than in the broadcast case.

VI. Conclusion

This paper evaluated simulation models of wireless sensor networks by considering small networks of 2 and 3 nodes with broadcast and unicast traffic. The evaluation scenarios were implemented in a small testbed network. The metrics focused on timeliness and success rate. We found that the simulation results matched experimentation closely. When deviations were found, such as for the increased service time in the experimental setup, they were attributed to in-system effects that are difficult to estimate and thus to model, and often dependent on the software implementation.

This work showed that accurate timeliness simulation results can be obtained for wireless sensor networks, and that discrepancies between the model and the experiment can even lead to the identification of small embedded software defects rather than modeling errors. Further work will consider more complex scenarios and power consumption estimates.

VII. Acknowledgment

The work presented in this paper was supported (in part) by the European Commission under the Framework 6 IST Project *Wirelessly Accessible Sensor Populations* (WASP, www.wasp-project.org) and by the National Competence Center in Research on Mobile Information and Communication Systems NCCR-MICS, a center supported by the Swiss National Science Foundation under grant number 5005-67322.

References

[1] K. Fall and K. Varadhan, *The NS Manual*, 2002.

[2] Akyildiz, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, p. 393, 2002.

[3] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, (New York, NY, USA), pp. 78–82, ACM, 2004.

[4] *IEEE Std 802.15.4-2006, IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).*

[5] W. Drytkiewicz, S. Sroka, V. Handziski, A. Koepke, and H. Karl, "Mobility framework (mf) for simulating wireless and mobile networks using omnet++," in *Proceedings of the 3rd International OMNeT++ Workshop*, 2003.

[6] A. Varga, "The omnet++ discrete event simulation system," in *Proceedings of the European Simulation Multiconference (ESM'2001)*, pp. 319–324, 2001.

[7] U. M. Colesanti, C. Crociani, and A. Vitaletti, "On the accuracy of omnet++ in the wireless sensor networks domain: simulation vs. testbed," in *PE-WASUN '07: Proceedings of the 4th ACM workshop on Performance evaluation of wireless ad hoc, sensor,and ubiquitous networks*, (New York, NY, USA), pp. 25–31, ACM, 2007.

[8] D. Cavin, Y. Sasson, and A. Schiper, "On the accuracy of manet simulators," in *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, (New York, NY, USA), pp. 38–43, ACM, 2002.

[9] S. Ivanov, A. Herms, and G. Lukas, "Experimental validation of the ns-2 wireless model using simulation, emulation, and real network," in *In 4th Workshop on Mobile Ad-Hoc Networks (WMAN'07*, pp. 433–444, 2007.

[10] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Network*, vol. 43, no. 4, pp. 499–518, 2003.

[11] G. P. Halkes and K. G. Langendoen, "Experimental evaluation of simulation abstractions for wireless sensor network mac protocols," in *Proceedings of the 14th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD 2009), Pisa, Italy*, 2009.

[12] Texas Instruments, *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, 2007. http://www.ti.com.

[13] E. Ben Hamida, G. Chelius, and J. M. Gorce, "Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation," *SIMULATION*, 2009.

[14] V. Peiris, C. Amre, and S. Bories, "A 1v 433/868mhz 25kb/s-fsk 2kb/s-ook rf transceiver soc in standard digital 0.18μm cmos," *Proceedings of the International Solid State Circuits Conference (ISSCC 2005)*, pp. 258–259, 2005.