

Research Article

Experimental Evaluation of Simulation Abstractions for Wireless Sensor Network MAC Protocols

G. P. Halkes and K. G. Langendoen

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2600 AA Delft, The Netherlands

Correspondence should be addressed to G. P. Halkes, g.p.halkes@tudelft.nl

Received 25 May 2009; Accepted 18 February 2010

Academic Editor: Faouzi Bader

Copyright © 2010 G. P. Halkes and K. G. Langendoen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The evaluation of MAC protocols for Wireless Sensor Networks (WSNs) is often performed through simulation. These simulations necessarily abstract away from reality in many ways. However, the impact of these abstractions on the results of the simulations has received only limited attention. Moreover, many studies on the accuracy of simulation have studied either the physical layer and per link effects or routing protocol effects. To the best of our knowledge, no other work has focused on the study of the simulation abstractions with respect to MAC protocol performance. In this paper, we present the results of an experimental study of two often used abstractions in the simulation of WSN MAC protocols. We show that a simple SNR-based reception model can provide quite accurate results for metrics commonly used to evaluate MAC protocols. Furthermore, we provide an analysis of what the main sources of deviation are and thereby how the simulations can be improved to provide even better results.

1. Introduction

Wireless Sensor Networks (WSNs) are networks of small cheap autonomous battery-powered sensor nodes. These nodes consist of a microcontroller and a radio for communication, as well as one or more sensors. When deployed in large numbers (hundreds or even thousands), sensor nodes provide a fine grained monitoring capability, which can be used for agriculture, intrusion detection, asset tracking, and many other fields. As the name implies, WSNs employ wireless communication to achieve their goal. Each sensor node contains a complete wireless networking stack, optimised for energy-efficient communication. This includes among others a MAC protocol, for coordinating the communication locally.

To evaluate a MAC protocol for a WSN it is required that one performs several experiments with different representative topologies. Furthermore, these experiments should ideally be repeated several times to obtain statistically relevant results. Performing these experiments in the real world is exceedingly time consuming and costly. Therefore, MAC protocol designers normally resort to using a simulator to evaluate their protocols. Using a simulator is a cheap

and quick way to perform many experiments with different topologies and parameter settings.

Simulators necessarily abstract away from reality in many ways. For example, radio propagation is not simulated by simulating the EM radiation through the air and obstacles from one antenna to the next, but by using a formula to calculate the received signal strength at the receiving radios. It is clear that these abstractions are required to make simulation feasible, and it is likely that many details can be ignored because of their limited impact on the simulation results. However, limited work has been done to validate the abstractions commonly used in simulators for WSN MAC protocols evaluation.

In this paper, we study the impact of two abstractions commonly used in simulations of WSNs. These abstractions are different ways to model the reception of signals at WSN nodes. First we evaluate the binary reception model that is used in the Unit Disk Graph (UDG) model. In this reception model, a signal is either received by a node at sufficient strength that perfect reception is guaranteed, or it is not received at all. Furthermore, all received signals have equal strength, so if two signals arrive at the same node at the same time the node will not be able to receive either signal. This

model is sometimes extended with an interference range. The signals arriving at nodes within the interference range are assumed not to be decodable, but strong enough to cause a collision with all other signals and therefore prevent reception.

The second reception model we evaluate is the SNR-based reception model. In this model, each signal is given a signal strength. If some signal arriving at a node is stronger than sum of all other signals at the node by at least the SNR ratio, the node can properly receive the signal. If the strength of the strongest signal versus the other signals is below the threshold, the node will only receive a garbled message. This model is commonly used in combination with Free space and Two Ray propagation models. However, for our evaluation of the SNR abstraction, we use measured signal strength from the testbed we use for validation.

We evaluate the accuracy of the physical layer abstractions within the context of MAC protocols for WSNs. Therefore, we focus on the performance metrics commonly used in evaluating MAC protocols. These are packet delivery ratio (a.k.a. packet reception rate or goodput), and energy consumption which is usually derived from the time spent in different radio states. Finally we investigate the average packet latency, which is also occasionally reported in the evaluation of MAC protocols.

The rest of this paper is organised as follows: in Section 2, we give an overview of relevant prior research in the area of simulation validation. In Section 3, we describe the setup of the experiments we performed, followed by the results in Section 4. In Section 5, we discuss differences between the simulated hardware and the real hardware, other than the reception models studied in this paper. Finally, in Section 6 we present our conclusions.

2. Related Work

Wireless simulation accuracy has been studied mostly in the context of Mobile Ad Hoc Networks (MANETs). Within this context, the work of Ivanov et al. [1] and Liu et al. [2] is most similar to our work. Ivanov compares the results of a real experiment with the results of the ns-2 wireless simulator for the same experiment, concluding that the simulated delivery ratio is quite accurate but latency results show much deviation from the real experiment. Furthermore, the study is limited to a single routing and MAC protocol. This limits the value as other studies have shown that different routing protocols are affected differently by different physical models [3].

The work of Liu et al. [2] provides some validation of the physical layer models used in the SWAN simulator, by using connectivity traces from a real experiment to drive the simulator. This study focuses solely on packet delivery ratio and parameter sensitivity, using different routing protocols.

Kotz et al. [4] provide a list of assumptions used by many MANET network simulators and provide a few small experiments to show that these assumptions can lead to erroneous results.

More specific studies into the accuracy of WSN simulators have been performed by Colesanti et al. [5], Lee et al. [6], Wittenburg and Schiller [7], and Pham et al. [8]. Colesanti studied the OMNeT++ MAC Simulator, but again only looked at packet delivery ratios and a single MAC protocol. The MAC Simulator uses the Unit Disk Graph (UDG) model. Colesanti et al. showed that by introducing probabilistic packet corruption derived from real-world experiments, the results of the UDG model could be made to approach the real-world experiments.

Pham studied the channel model in the Castalia WSN simulator. The experiments used an unspecified tunable MAC protocol and were aimed at verifying the connectivity and fluctuations in connectivity by comparing simulations with the results of real experiments. The study found that even with the complex model used in the Castalia simulator, significant differences still occur.

The experiments done by Wittenburg and Schiller [7] focus on single link behaviour. The results show that given a reasonable propagation model, similar packet loss rates can be achieved as in real-world experiments. Lee et al. [6] provide a new trace-based noise model for wireless simulations. Through several experiments, Lee et al. show that their model can simulate single links more accurately with respect to packet delivery ratio than existing models. However, because both studies only consider a single link, effects such as collisions and the capture effect are unknown and no conclusions can be derived with respect to MAC protocol behaviour.

Because in MANETs energy efficiency is not an important metric, none of the MANET studies have considered energy consumption. The study by Colesanti, although focused on WSNs, also did not consider energy consumption. Heidemann et al. [9] have considered energy consumption but only to show that the energy consumed by nodes when waiting for packets to arrive is a significant factor and must be taken into account in simulations.

All MANET validation studies have used the 802.11 MAC protocol. As this is the de facto standard in MANETs, this is perfectly reasonable. However, as we show in this paper, and in a previously published condensed version of this study [10] with fewer experimental results, not all MAC protocols are affected equally by the choice of physical layer abstraction. Therefore, it is important to specifically study the impact of simulation abstractions on different MAC protocols.

3. Experiment Setup

To evaluate the simulation abstractions, we compare simulation results with results from our PowerBench testbed [11]. The testbed consists of 24 nodes installed in our offices. By configuring the send power to its lowest setting, we can create a multihop network. However, when using this setting we can only usefully employ 22 nodes. The nodes in our testbed are Tnodes, which use the same components as the mica2 nodes (Chipcon CC1000 radio, Atmel ATmega 128L processor).

In our testbed we use our TinyOS 2.x λ MAC framework, for which we have implementations of several MAC

protocols. The MAC protocols using the λ MAC framework represent different points in the MAC protocol design space. For our experiments, we use the B-MAC [12], T-MAC [13], Crankshaft [14], and LMAC [15] protocols. The B-MAC protocol is a representative of the Low-Power-Listening class of protocols. T-MAC is also a carrier-sense-based protocol, but instead uses frames with active and idle periods to reduce energy consumption. LMAC is an example of a TDMA protocol, and finally Crankshaft is a hybrid protocol using the slotted structure of TDMA protocols in combination with carrier sensing to achieve high energy efficiency. It should be noted that the LMAC implementation for the λ MAC framework uses a static slot assignment and uses a timer to detect the absence of packets in a slot rather than a carrier sense mechanism.

In order to limit as much as possible the influence of modelling differences between the software running on the real hardware and the simulation models, we have chosen to use the TinyOS 2.x simulator TOSSIM. TOSSIM uses the same code as is compiled for the hardware platform to compile the simulator, only replacing the hardware control modules with TOSSIM specific ones. The standard TOSSIM however does not provide a model for the CC1000 radio. Therefore, we used and modified the PowerTOSSIM extension for TinyOS 2.x as a basis to implement different reception models.

3.1. Traffic Pattern and Metrics. In our evaluation, we first consider the convergecast or to-sink traffic pattern. In this pattern, all nodes in the network send messages to a single sink node. This pattern is a representative of data collection in WSNs. Because we are considering a multihop network, a routing tree needs to be set up. To eliminate the influence of components other than the MAC protocol as much as possible, we use a fixed routing tree that we created off-line based on link quality measurements. Because different network setups have different characteristics, we use two different routing trees. The first routing tree has the sink in the centre of the network and has an average hopcount of approximately 1.86. The second routing tree has the sink at the edge of the network and has an average hopcount of approximately 2.48.

The second traffic pattern, we study is the broadcast flood pattern. This pattern is often used to disseminate data or commands in a WSN. We chose these two traffic patterns because they exercise different aspects of the MAC protocols.

As metrics we consider both delivery ratio and energy consumption. The delivery ratio for the convergecast pattern is simply defined as the fraction of messages sent by all nodes that arrive at the sink node. For the broadcast flood pattern the delivery ratio is calculated as the sum of all unique messages that arrived at the nodes divided by the number of messages injected in the network and the number of receiver nodes in the network:

$$d = \frac{\sum_{i=1}^N R_i}{(N - 1) * M}, \quad (1)$$

where d is the delivery ratio, N is the number of nodes in the network, R_i is the number of unique messages node i

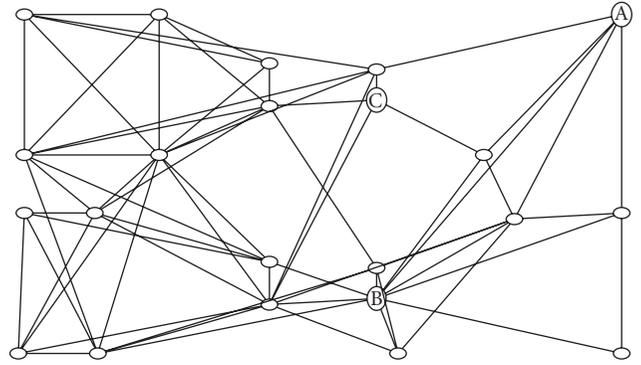


FIGURE 1: Connectivity in the PowerBench testbed. Links shown have at least 95% packet reception in both directions.

received, and M is the number of messages that the sink injected into the network. Note that the sink does not receive any of the messages it injects, so the maximum number of unique messages that can be received is $(N - 1) * M$. So if only half of the nodes receive a message injected, the delivery ratio would be 0.5.

The energy consumption in simulation is usually derived from the time the radio spends in transmit, receive, and idle states. In previous work [11] we have shown that using this simple three-state model yields accurate results for energy consumption. In this paper, we therefore use the time spent in the different states as our metric, rather than the combined energy consumption number. Using the separate states allows us to more precisely determine the causes of inaccuracy of the simulator.

Finally, we also provide average packet latency as metric. Packet latency is usually traded for energy consumption in WSN MAC protocols, and therefore not considered a very important metric. However, it is interesting to see how much latency is incurred because of the trade off, and this does make latency an interesting metric.

3.2. Abstractions. In this paper, we study two reception models commonly used in WSN MAC protocol simulation. The first model is the binary reception model employed by the Unit Disk Graph (UDG) model. In this model nodes either receive a signal perfectly or not at all.

To arrive at a simulation of the binary reception model which can be compared with the results from our testbed, we cannot simply derive a connectivity graph from the node positions. As Figure 1 shows, the connectivity in our testbed network is very irregular. For example, there is a good connectivity between node A and node B, while the link between A and C, which is much shorter, allows virtually no messages to get through. Therefore, we first measured all link reception rates in our testbed. From this, we extracted the subset of links that show (near) perfect reception. These links are then taken to be usable for signal transmission, while all other links are discarded. In our experiments, we use the extended reception model that also implements an interference range. The links on which interference can

occur are the links for which the average signal strength is above a threshold. The threshold has been tuned to provide simulation results as close as possible to the real-world results. We have verified that the extended binary model yields slightly better results than the simple binary model, and therefore use the extended binary model in our simulations.

The second popular reception model is the Signal-to-Noise-Ratio- (SNR-) based model. In this model, all received signals have a different signal strength. To determine whether a signal can be received, it is compared to the sum of all other received signals and noise. If the signal is stronger than the combined signal by at least the SNR threshold, it is assumed to be decodable. Collisions therefore only occur when two signals arrive that are close in signal strength.

As with the binary reception model, we cannot simply use the positions of the nodes in our testbed to calculate received signal strengths as there is too little correlation between received signal strength and distance. This phenomenon has already been shown in [4]. We have therefore measured the (average) received signal strengths between all pairs of nodes in our testbed. We use this information as the received signal strengths in our simulations. Furthermore, we have experimentally determined the SNR threshold of the CC1000 radio on the Tnodes to be approximately 5 dB.

4. Results

We now present the results of our experiments. All experiments were repeated between 5 and 8 times, and the graphs show the mean and standard deviations.

4.1. Convergecast. For convergecast, we first show the experiment with the sink at the centre. This experiment shows the largest divergence between the simulations and the real-world experiments, and shows the most interesting effects. Then we will present the most interesting results from the convergecast experiment with the sink located at the edge of the network.

4.1.1. Delivery Ratio. Figure 2 shows the delivery ratio for the different protocols. From the graphs it is immediately clear that the binary reception model does not provide a good simulation abstraction. For all protocols except LMAC, the simulated delivery ratio is much worse than the measured delivery ratio. It is not surprising that the delivery ratio in LMAC is not affected by the binary reception model, as transmissions in LMAC are scheduled not to collide.

The large differences in delivery ratio for the B-MAC, T-MAC, and Crankshaft protocols are due to the all or nothing nature of the binary reception model. In real, life fewer collisions occur because weak signals do not interfere with strong signals. In the binary model, there is no distinction between weak and strong signals, which means all concurrent transmissions arriving at a single node will always cause a collision.

The delivery ratio for the SNR-based simulation for the most part approaches the measured results quite closely.

Notable exceptions are the B-MAC protocol at high message rates and the Crankshaft protocol at low message rates. The reason that B-MAC diverges at high message rates is because the real implementation detects more carriers than the simulator. Even though both use the same code for carrier detection, the fluctuations in (measured) signal strength that occur in real life are not simulated and therefore fewer carriers are detected when the signal strength is close to the detection threshold. We found that this abstraction is the cause of most differences between the measured results and the SNR-based simulation model.

The cause of the difference between the SNR-based simulations of the Crankshaft protocol and the measured results is very different. The sending of messages in the Crankshaft protocol is synchronised to a particular time in each slot. When equal length messages are sent as shown in Figure 3, both the actual messages and the acknowledgments are received error-free. However, when node A is slightly ahead of node D, the acknowledgement sent by node B in response to A's message will collide with the message from D to C. In simulation, the clocks of different nodes run exactly at the same rate. Therefore, once properly synchronised there is no chance of such a collision. However, in reality clocks on different nodes drift, which means that collisions of this type are likely to occur. At higher message rates, the extra carrier detections found in the real-life implementation prevent some hidden terminal problems, which compensates for the synchronisation problem.

4.1.2. Energy Consumption. Next we consider the time spent in the different radio states. Figure 4 shows the time spent in receive mode. For this metric, the difference between the binary reception model and the SNR-based model is very small. Although this may at first seem contradictory given the low delivery ratio for the binary model, one should take into account that collisions do cause the radio to remain in receive state.

Although the time spent in receive mode for the simulated experiment is very similar to the real-life situation, there are some differences that warrant explanation. The time the real B-MAC and T-MAC spend in receive mode is higher than in the respective simulations. This is again caused by the extra carrier detections. A similar explanation holds for Crankshaft. However, the difference here is that the offset between the real and simulated protocol is much more constant. There are two interrelated causes for the near constant offset. The first is that the Crankshaft protocol separates unicast and broadcast traffic and minimises overhearing for unicast traffic by alternating which nodes are awake to receive messages. This means that for unicast traffic, as is used in the convergecast pattern, there will be few extra carrier detections as most nodes will be asleep during a unicast transmission. The second reason is that the Crankshaft protocol uses broadcast packets for time synchronisation. The number of synchronisation messages will remain constant, regardless of the number of unicast messages being sent. As all nodes will be awake during times when broadcast messages may be sent, there will be a number

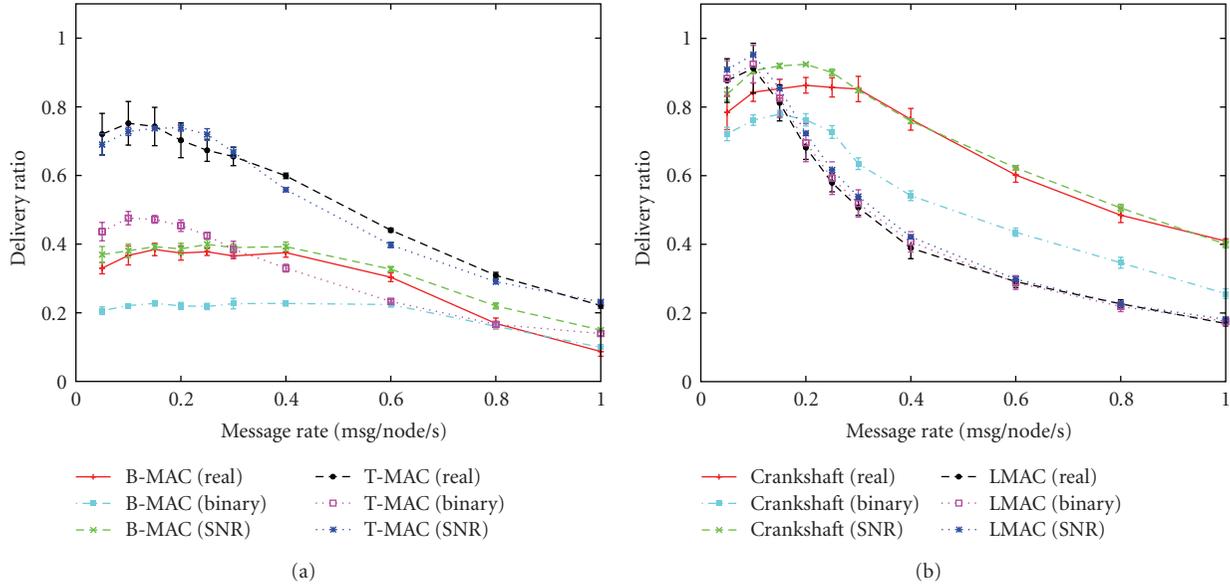


FIGURE 2: Delivery ratio for simulated and real convergecast experiments.



FIGURE 3: Potential collision situation for the Crankshaft protocol.

of extra carrier detections. The constant number of extra carrier detections caused by time-synchronisation messages and the relatively small number of extra carrier detections caused by unicast messages will make for a near constant offset between the real and simulated time spent in receive mode.

Figure 5 shows the time spent in transmit mode for the different protocols. Note that Figure 5 uses a different scale than the receive time graphs in Figure 4. Again the binary model shows the largest differences, with differences up to 20%. The largest difference occurs for the T-MAC protocol, which at all message rates spends less time in send state than the real-world implementation. Although the lower transmit time may seem logical given the lower delivery ratio achieved by T-MAC, this is only part of the story. The extra collisions in the binary model also cause more retries. These retries also cost extra send time. Based on this effect, one would expect more rather than less time spent in send mode. However, these retries are mostly RTS retries, which do not cost a lot of extra time. Furthermore, there are fewer messages being relayed. Relayed messages cost transmit time for every hop, while messages that get dropped after several retries only cost transmit at a single hop.

The transmit time results for the SNR-based model are almost all within 5% of the real-world results. Only the transmit time of the B-MAC protocol with the SNR-based model is 10% less at the highest rate. This is a result of the reduced bandwidth caused by the extra carrier detections.

4.1.3. *Latency.* Finally, Figure 6 shows the average packet latency. These graphs do not show the binary model anymore, because the divergence in packet delivery is very large. The large deviation in delivery ratio means that the average latency will be calculated over a very different set of messages, and therefore makes these results incomparable.

For T-MAC, the SNR-based latency results seem quite similar to the real-world results, except at low message rates. However, there are two effects here that cancel each other out at higher data rates. The already mentioned extra carrier detections on one hand cause extra latency in the real-world experiment (exposed terminal problem). On the other hand, they also prevent hidden terminal collisions around the sink, which effectively reduces latency. At lower data rates, hidden terminal collisions are less of a problem. Therefore, the latency in the real-world experiment is higher at low data rates.

B-MAC suffers from using long preambles. The real-world latency is much higher because each extra carrier that is detected defers the sending of a packet by a significant amount of time.

The difference between the real-world Crankshaft results and the SNR-based Crankshaft results is a direct consequence of the difference in the source of packet loss and retries. The packets lost in the real-world experiment are lost throughout the network. However, the SNR-based simulation experiences packet loss mostly around the sink. Because in Crankshaft the sink is assumed to be mains powered and therefore listens in every slot, a retry to the sink can be done in the next slot. However, a packet lost elsewhere in the network has to be retried in the next frame, and therefore has to be deferred for a lot longer.

4.1.4. *Topology Effects.* To determine how sensitive the results are to the specific topology, we also performed an experiment

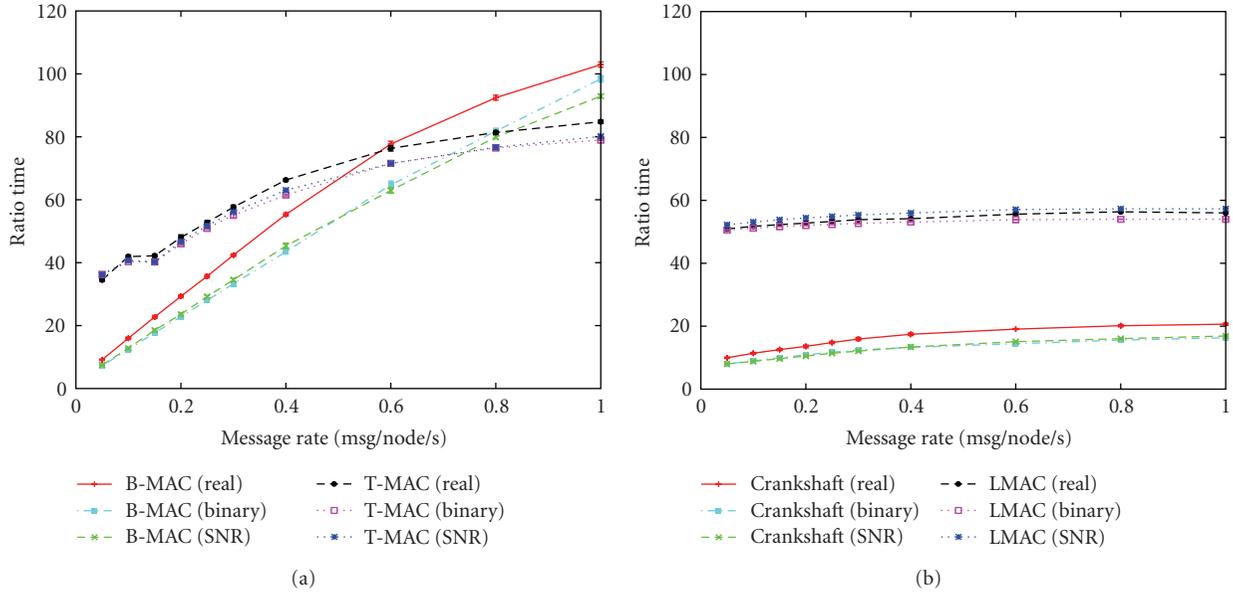


FIGURE 4: Radio receive times for simulated and real convergecast experiments.

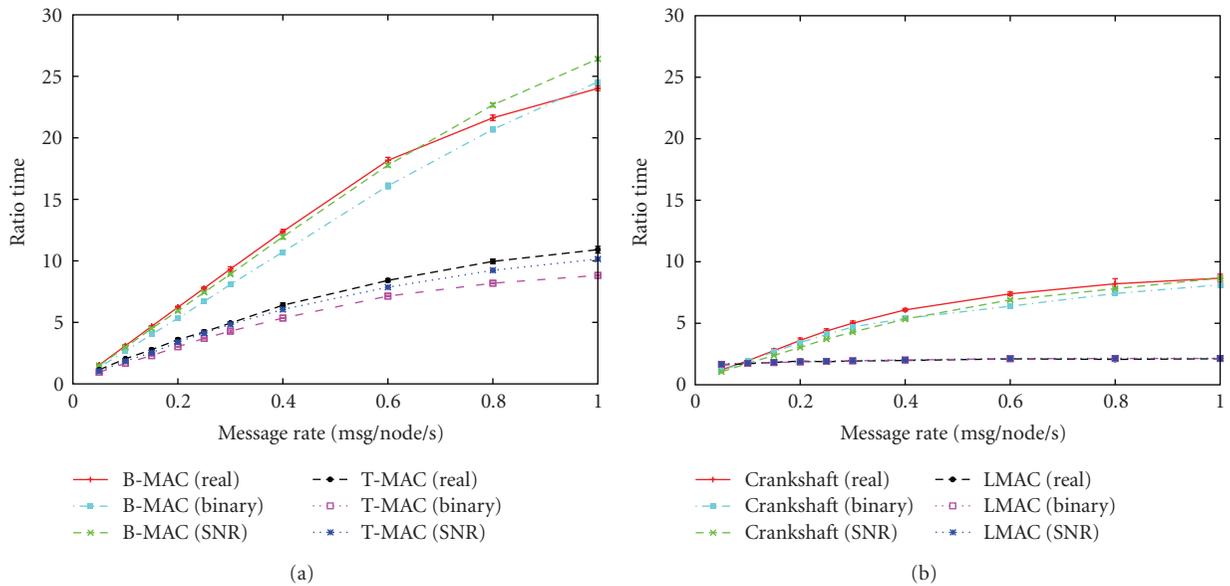


FIGURE 5: Radio transmit times for simulated and real convergecast experiments.

where we placed the sink at the edge of the network. In this section, we only present the interesting differences with respect to the previously shown experiment.

Figure 7 shows the delivery ratio for the different protocols. Compared to the experiment with the sink in the centre of the network (as shown in Figure 2), the binary reception model results are much closer to the real-world results for all protocols except LMAC. The other notable point in this graph is that the SNR-based results for the Crankshaft protocol do not match the real-world results as closely as in the previous experiment. As noted in the analysis in Section 4.1.1, there were two opposite effects that played

a part in the good result for the SNR-based model in the previous experiment: fewer collisions caused by small desynchronization and more collisions due to hidden terminal problems. As the sink is located at the edge of the network in this experiment, there are fewer hidden terminal problems. This results in fewer collisions in the network for the SNR-based simulations, which means a higher delivery ratio. However, the results are still within 10%.

For the latency results in Figure 8, we again see that the reduced hidden terminal problem changes the relative performance of the SNR-based model. In this case, the T-MAC protocol is affected most clearly. Where in the

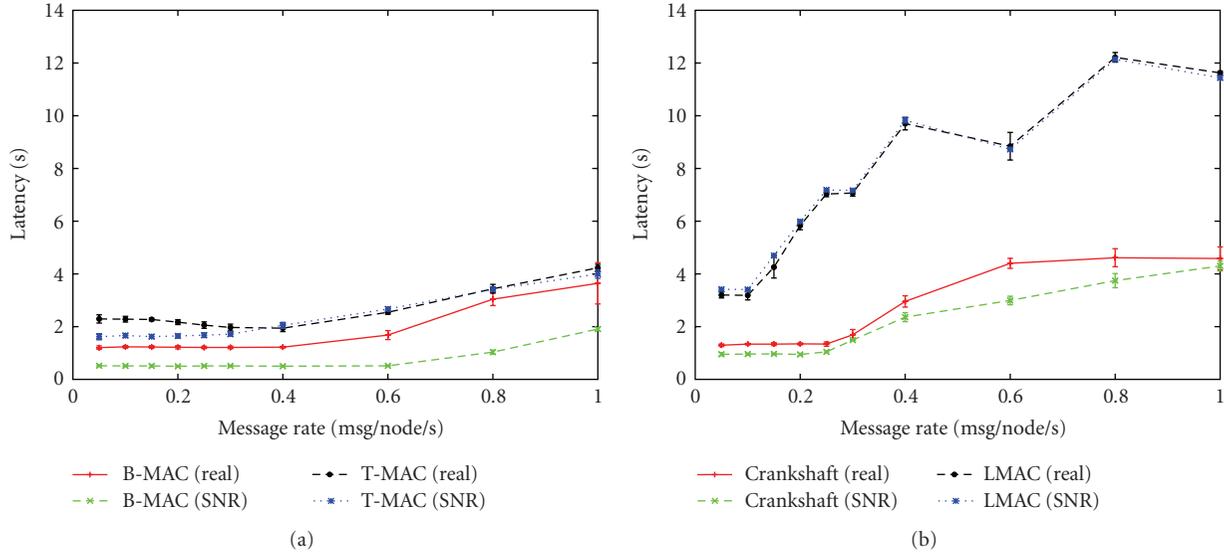


FIGURE 6: Average packet latency for simulated and real convergecast experiments.

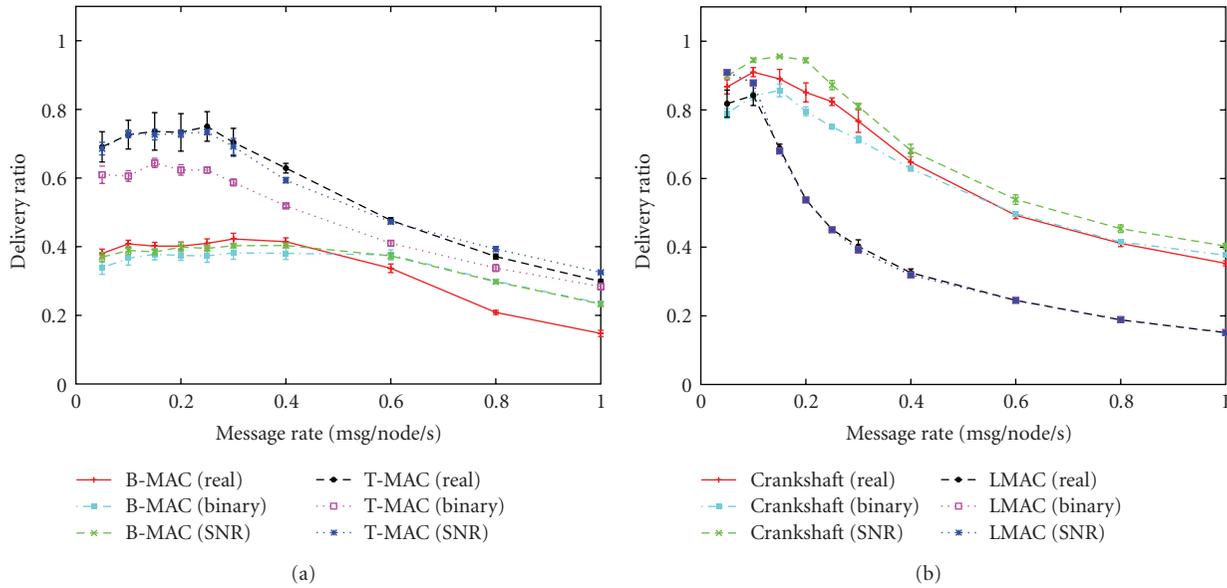


FIGURE 7: Delivery ratio for simulated and real convergecast experiments. with the sink at the edge of the network.

experiment with the sink at the centre of the network, the extra carrier detections reduced hidden terminal collisions in the real-world experiment and thereby lowered the latency at higher message rates, this is no longer the case with the sink at the edge of the network. This results in a uniformly higher latency for the T-MAC protocol in this experiment.

The final interesting difference we see in this experiment is that the latency for the SNR-based simulation of the Crankshaft protocol is very close to the measured latency. Although encouraging, it should be noted that the average hopcount for messages arriving at the sink for the real-world experiment is lower than for the simulation. As the latency difference is higher further away from the sink, the different distribution hides the real latency differences. Although most

differences are at most 10%, for the first hop and the nodes at 4 hops from the sink, the difference can be up to 50%.

4.2. Broadcast Flood. As a second test, we used the broadcast flood traffic pattern. The delivery ratios for all protocols except B-MAC are within a few percent of the real-world experiment for the SNR-based reception model (Figure 9). Extra carrier detections are again the cause of the difference in performance. Although Crankshaft also suffers from this problem, the problem is most pronounced in B-MAC because B-MAC uses long preambles which exacerbate the problem. When a node running B-MAC has to defer a packet because it detects the channel is busy, it has to wait for a significant amount of time. If instead it detects the channel as

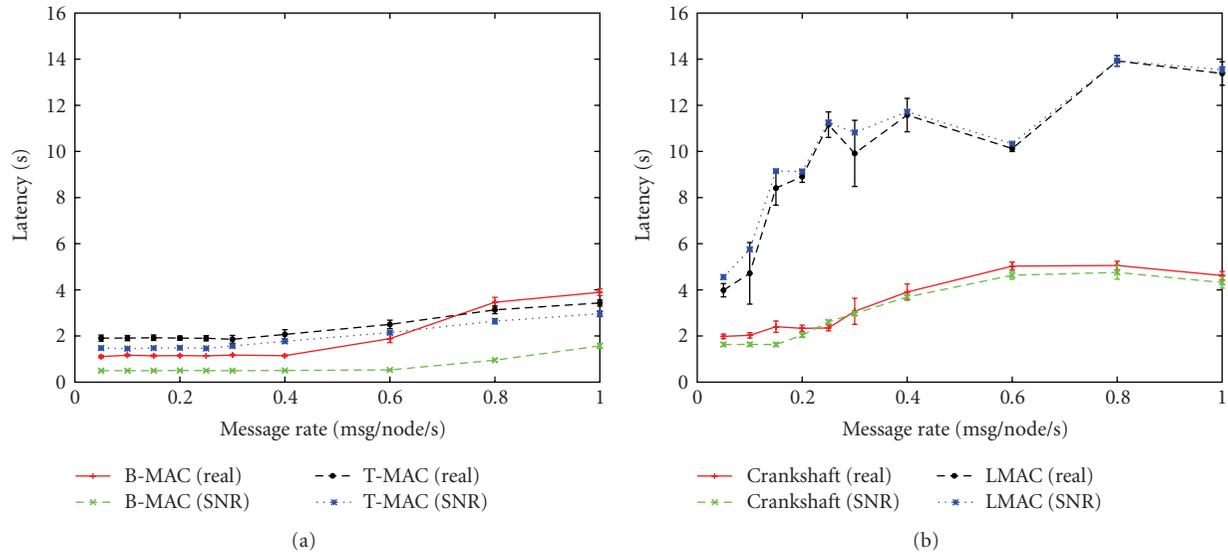


FIGURE 8: Average packet latency for simulated and real convergecast experiments. with the sink at the edge of the network.

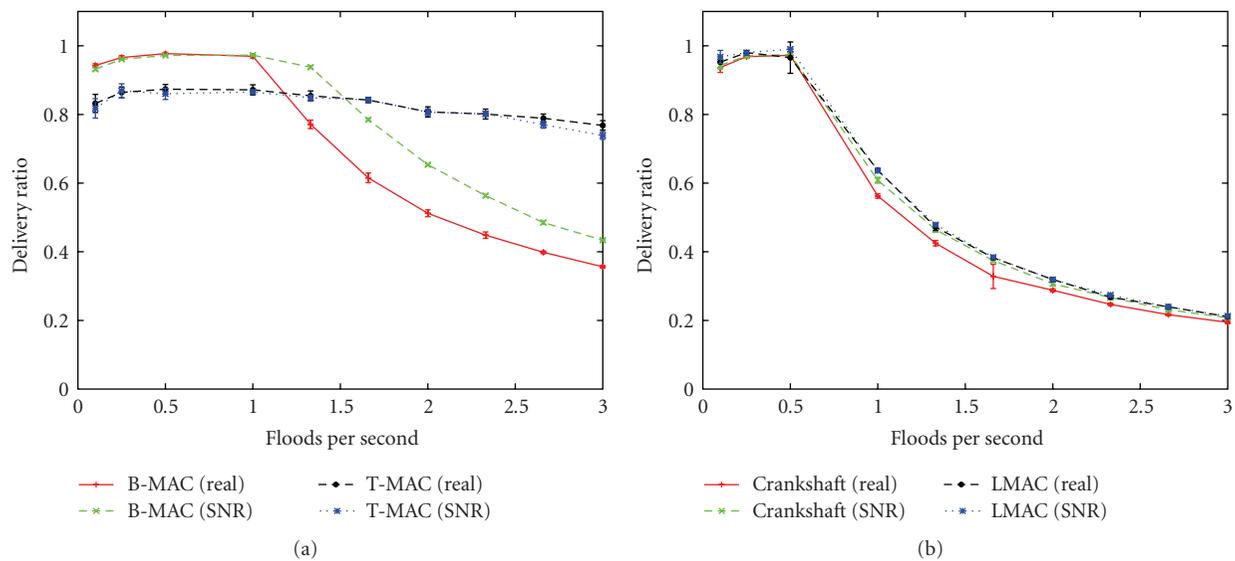


FIGURE 9: Delivery ratio for the broadcast experiment.

idle, as it does in the simulated case, the available bandwidth is increased.

Note that for T-MAC the extra carrier detections can actually increase the delivery ratio. For T-MAC, the available bandwidth is not a problem. What is a problem though for T-MAC is nodes switching off the radio when there is no traffic detected. When a carrier is detected, a node will reset its timeout and will only start the timeout again when the carrier disappears. Then, when the message is resent by another node, the node with the reset timeout may be able to receive the resent message resulting in an increased delivery ratio. The effect however is small.

We have also studied the time the radio spends in different states. As the results provide no extra insights, we do not show them here.

5. Modelling Differences

As we mentioned in Section 3, we used the TOSSIM simulator which is compiled from the same code as is used to compile the hardware code. This limits the differences between the code executing in the simulator and the code running on the hardware. However, this does not mean there are no differences. The modules that normally interact with the hardware have to be replaced by modules that change the simulator state to reflect the requested changes. For example, the radio driver module has to simulate the transmission of signals to the other radio driver module instances when requested to transmit bytes.

For the time spent in the different radio states, there is another factor that plays an important role: the simulation

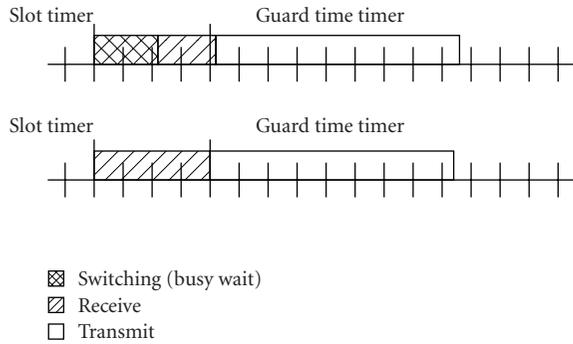


FIGURE 10: Radio states at the start of an LMAC slot for hardware implementation (top) and simulation (bottom).

of busy waiting. When switching the radio between radio states, the processor has to wait for short amounts of time at several points. This is implemented by busy waiting in the driving code. For the simulator, busy waiting is transformed to a no-op, because busy waiting cannot be directly simulated in a discrete event simulator such as TOSSIM. This modelling difference can result in significant differences in the simulation results.

As an example, we show the start of an LMAC slot (see Figure 10). The radio states shown are for a node which owns the slot. In the real-world implementation, the switch from off state to receive state takes just over 2 msec (cf. Figure 10 top). This switch is started when the slot timer fires. Then, when the guard time timer fires, the radio is switched to transmit state. The time the radio requires to switch is timed through a busy-waiting loop in the code. As already mentioned, this is transformed into a no-op in the simulator. The result is that the switch is effectively performed immediately in the simulator (cf. Figure 10 bottom). The total time spent in receive mode would therefore be longer in the simulation than in the real-world, even if the simulation is otherwise perfect.

In the specific example, we could easily resolve the problem because the guard time is calculated taking into account the switch time. By overriding the switch time constant with a value of zero for the simulation, this specific instance of the busy-waiting problem was resolved. There are also several other points in the radio switching code where busy-waiting loops are used, but the impact of those loops is much smaller as their waiting time is in the order of 0.1 msec. Furthermore, not all protocols are impacted by the demonstrated off-to-receive switch problem. If the timing for events after the switch is done relative to the completion of the switch, for example, by waiting until a certain number of bytes have been read from the radio, the time difference is effectively zero as well.

6. Conclusions

In this paper, we have presented the results of an experimental study of two often used reception models in the simulation of WSN MAC protocols, namely, the binary

reception model and the Signal-to-Noise Ratio (SNR) model. We have compared simulations that use these models with data from a testbed. The comparison focused on the metrics most commonly used in evaluating WSN MAC protocol performance: delivery ratio and energy consumption. We have also studied the average packet latency as that is also sometimes used as a metric in evaluating WSN MAC performance.

The results show that the binary reception model used in, for example, the Unit Disk Graph simulation model results in significant deviations from the real-world results. The delivery ratio in particular showed differences of up to 50% for B-MAC and T-MAC. Furthermore, different protocols were impacted differently, which means that the binary model is also unsuitable for relative comparisons.

The SNR-based model, however, can provide quite accurate results for metrics commonly used to evaluate MAC protocols. Most results are within 5%, while virtually all are within 15%. For latency, the results are less accurate. The main cause of remaining deviation between real-world results and the SNR-based model is the unmodelled fluctuations in (measured) signal strength.

It should be noted that a study such as ours is necessarily limited in scope. For example, we have not studied the effects of mobility. As mobility also impacts signal propagation, simulations which include mobility may also require more extensive reception models.

We conclude that simulation should not be based on the binary reception model as the results obtained with such simulations deviate too far from reality. The SNR-based reception model provides results that are reasonably close to reality.

Acknowledgment

This research has been partially supported by the 7th Framework Programme of the European Union under Grant ICT-2008-224460 (WISEBED).

References

- [1] S. Ivanov, A. Herms, and G. Lukas, "Experimental validation of the ns-2 wireless model using simulation, emulation, and real network," in *Proceedings of the 4th Workshop on Mobile Ad-Hoc Networks (WMAN '07)*, Berne, Switzerland, March 2007.
- [2] J. Liu, Y. Yuan, D. M. Nicol, et al., "Empirical validation of wireless models in simulations of ad hoc routing protocols," *Simulation*, vol. 81, no. 4, pp. 307–323, 2005.
- [3] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 87–94, ACM, October 2001.
- [4] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proceedings of the 7th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04)*, pp. 78–82, October 2004.

- [5] U. M. Colesanti, C. Crociani, and A. Vitaletti, "On the accuracy of OMNeT++ in the wireless sensor networks domain: simulation vs. testbed," in *Proceedings of the 4th ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '07)*, pp. 25–31, Crete Island, Greece, October 2007.
- [6] H. Lee, A. Cerpa, and P. Levis, "Improving wireless simulation through noise modeling," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, pp. 21–30, Cambridge, Mass, USA, April 2007.
- [7] G. Wittenburg and J. Schiller, "A quantitative evaluation of the simulation accuracy of wireless sensor networks," in *Proceedings of the 6. Fachgespräch "Drahtlose Sensornetze" der GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme"*, pp. 23–26, Aachen, Germany, July 2007.
- [8] H. N. Pham, D. Peditakis, and A. Boulis, "From simulation to real deployments in WSN and back," in *Proceedings of the 8th ACM International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, pp. 1–6, Helsinki, Finland, June 2007.
- [9] J. Heidemann, N. Bulusu, J. Elson, et al., "Effects of detail in wireless network simulation," in *Proceedings of the SCS Multiconference on Distributed Simulation*, pp. 3–11, Phoenix, Ariz, USA, January 2001.
- [10] 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 '09)*, Pisa, Italy, June 2009.
- [11] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, "PowerBench: a scalable testbed infrastructure for benchmarking power consumption," in *Proceedings of the International Workshop on Sensor Network Engineering (IWSNE '08)*, pp. 37–44, Santorini Island, Greece, June 2008.
- [12] J. Polastre and D. Culler, "B-MAC: an adaptive CSMA layer for low-power operation," Tech. Rep. cs294-f03/bmac, University of California, Berkeley, Calif, USA, December 2003.
- [13] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 171–180, Los Angeles, Calif, USA, November 2003.
- [14] G. Halkes and K. Langendoen, "Crankshaft: an energy-efficient MAC-protocol for dense wireless sensor networks," in *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN '07)*, pp. 228–244, Delft, The Netherlands, January 2007.
- [15] L. van Hoesel and P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks," in *Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS '04)*, Tokyo, Japan, June 2004.