

Radiocommunication test suite for wireless sensor networks

Krisztian Veress
Department of Computer Science
University of Szeged
verkri@inf.u-szeged.hu

Miklos Maroti
Department of Algebra and Number Theory
University of Szeged
mmaroti@math.u-szeged.hu

Abstract

Medium Access Control (MAC) protocols play an undeniably important role in wireless sensor networks. If properly chosen for a given application, the underlying network's lifetime, robustness, scalability and bandwidth could be optimized. However selecting an appropriate protocol is hard, given the lack of deep and comprehensive research on MAC modelling and verification. It would be impressive to describe our needs the applied MAC protocol must conform to then automatically evolve and optimize a specific MAC protocol for our given application.

To facilitate this goal, we designed a test environment where multiple network topologies and communication protocols can be easily described, tested and evaluated under different circumstances in a controlled and reproducible way. As an *evaluate-function*, this will be used in future comparisons and automated optimization of MAC protocols.

Our proposed framework is designed to analyze MAC protocol behaviour by performing network tests focusing on the wireless communication. During our tests, simple messages with unique payloads are transmitted and statistics are collected about the communication in progress, allowing to characterize it with a large set of features rather than a pass/fail value. It supports all WSN platforms of TinyOS, dynamic network topologies and multiple communication modes.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communication; D.2.5 [Software Engineering]: Testing and Debugging—*Distributed debugging*; C.4 [Performance of Systems]: Measurement Techniques

Keywords

WSN, MAC, radiocommunication, measurement

1 Problem statement

There are various Medium Access Control (MAC) protocols specifically designed for use in Wireless Sensor Networks, sharing the common goals of reducing energy consumption [3], maintaining scalability and adaptability and providing good latencies and network bandwidth [15]. However the background of the great part of proposed protocols are generally based on experiments, observations or assumptions and are rather hand-written than an optimal result of a well-formed modelling process.

Several studies are available on comparing existing radiocommunication protocols [13], [7], however the methodology behind these comparisons and the extracted metrics are usually unascertained. In addition, it is also frequent that only prototype implementations are available for certain proposed MACs [6], [20], [2] which prevents their wide testing and usage on multiple platforms and in heterogenous networks, so real application measurements could be hardly made.

There are also – but few – papers on modelling existing low power MAC protocols in a whole [5] or certain parameters of them [17], [1] since creating universal models for medium access protocols is a real challenge. Then, our goal is to fill this gap by analyzing existing MAC protocols, extracting those features that best describe their behaviour and establish an easily configurable yet robust model for even complex MACs like full ZigBee with beacons and devices. Key research items to be explored and answered are

- *glue that holds together MAC protocols,*
- *MAC basic building blocks and their behaviour,*
- *fine-tune of existing MAC-protocols,*
- *application-based automated optimal MAC evolution.*

1.1 Our methodology

Our main objective is to develop an automated MAC protocol evolution framework being able to search for an optimal solution in the MAC-space for a given application or given constraints. Anticipated research challenges include the following entries that must be successfully rolled down to achieve this goal.

1. The first task is to provide a widely-usable test-suite capable of executing real wireless communication in a controlled manner. This framework should not intro-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSysDC'10, November 2, 2010, Zurich, Switzerland.
Copyright 2010 ACM 978-1-4503-0344-6/10/11 ...\$10.00

duce any constraints for wireless communication policies, be easily configurable, and should provide full-scale information about the performed communication. This framework will be the base of future comparisons, and stress-tests of existing and generated MAC protocols.

2. If we are able to evaluate MAC protocols, implementations of the latter must be ported to (if not available) for all widely used platforms in order to be evaluated with our previous framework. Deep comparisons should reveal the key features of MAC protocols in general that is essential for the next step.
3. Given the extracted features from the previous step, comprehensive and universal models are to be built up in an appropriate modelling language like TTCN or SDL. These models are expected to handle complex protocols, and constructed based on the requirements of medium access rather than properties or assumptions.
4. If models are available and application-defined needs are present, application- and/or network-specific MAC protocols could be programmatically generated using either genetic/evolutionary programming or other optimization methods.

In this paper we present our recent activity on achieving our first milestone, the development of a radiocommunication test suite.

2 Test suite concepts

The aim was to design and implement a test suite that can be run on real hardware, uses those drivers, layers and components that real applications do (similar to Twist [8]) and is built on top of the latest TinyOS operating system. When designing our test framework we focused mainly on the *simulation of an application's wireless communication*, not simulating the application itself. Application-logic unit testing and verification have been deeply studied by Beutel et al.[4] and Whitehouse et al.[18] and application simulators are also available like TOSSim, [11] or Avrora [16].

Furthermore, simple unit tests provided by these frameworks are not enough for our needs, since we must characterize the whole network's communication in detail not just with a pass/fail value.

The proposed test suite has to be deployed on numerous (2+) motes that are going to build up the requested network and simulate the desired communication. Controlling these motes is done via one *BaseStation* mote connected to a laptop or PC and a PC-based Java application capable of re-setting and configuring the network and finally downloading the results collected during the testcases. Results are given either in raw (for quick tests) or XML + XSLT format (for benchmarks).

2.1 Modelling networks and communication

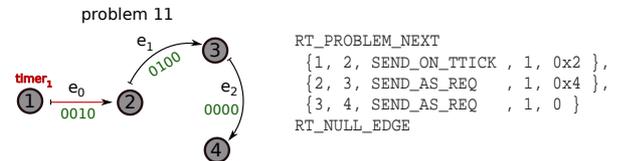
The model behind the scene is very simple, a $G = (V, E)$ directed graph having V vertices as motes, where an edge $(u, v) \in E$ is a communication line between mote u and v in such a way that messages transmitted by u are expected to be received by v . In this case u is the *sender* and v is the *receiver*

on the (u, v) edge. Thus, atomic communication (messages) are tied to the edges of the directed graph.

The main idea is to send unique sequential messages on these communication lines based on the applied communication policy. Then each and every node of the network track those messages that are tied to either an incoming or an outgoing edge of that node.

A communication policy has numerous global (network-wide), and local (per-edge) parameters. A policy describes *which mote* should send a message *to which mote(s)* on *what kind of an event firing*. Messages can be transmitted at the beginning of the test, based on timers or when messages are sent or received. Parameters include among others the communication mode (broadcasting, direct addressing, acknowledgements), Low-Power-Listening modes, timer frequencies, simulation time, etc.

A testcase in our test suite then consists of a network topology along with its communication policy. A sample testcase definition is given in the next example showing a multihop forwarding network using a timer in *mote 1* for sending periodic messages.



To avoid the need for reprogramming between tests, multiple testcases can be incorporated into the test suite. Then the PC application can select the desired testcase to be run.

2.2 Statistics collection

The core of the framework is our statistics collection algorithm. As we have already stated above, our framework's purpose is to collect communication statistics while the established network is online. The main goal was to compile statistical figures that allow us to deeply analyze the network's behaviour.

Since TinyOS allows us to track the state of any message sent or received by the radio chip, we count every occurrences of any distinct communication event and log every message sending attempt, success or failure.

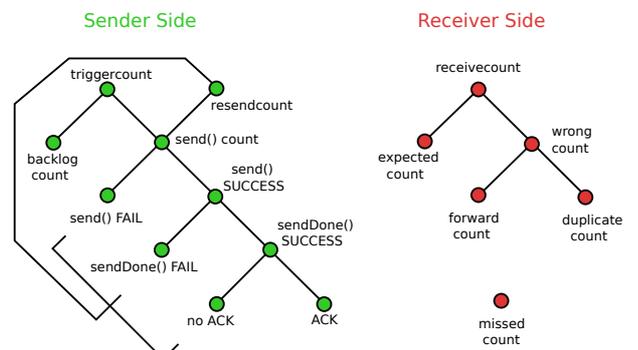


Figure 1. Relation between the statistics at both ends

We have created two classes of statistics, the sender's, and

those of the receiver. Actually, the statistics are collected per edge to let the implementation be as simple as it can, but eventually these numbers characterise the endpoint motes.

To ensure that our numbers are consistent, simple equations must hold between them. These are also shown in Fig. 1 by the hierarchical lines. Some examples:

$$\begin{aligned} \text{send} &= \text{sendFail} + \text{sendSuccess} \\ \text{trigger} + \text{resend} &= \text{backlog} + \text{send} \\ \text{resend} &= \text{sendFail} + \text{sendDoneFail} + \text{noAck} \end{aligned}$$

The final report of a test run contains these enlisted statistics (and some extra) for each edge separately. Furthermore, to understand the receiver side statistics, let us consider an artificial sniffing of a single edge communication the result of which is shown in figure 2.

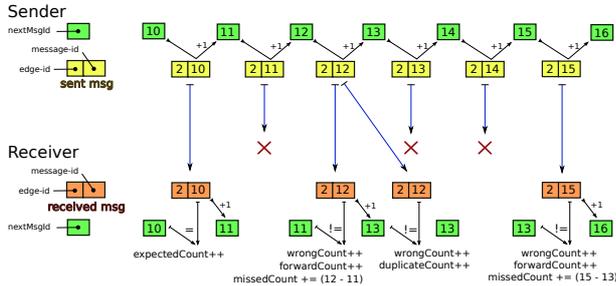


Figure 2. Receiver statistics collection demonstrated

It can be clearly seen that each message consist of two simple values: an *edgeid* and an auto-incremented *msgid*. On both ends of an edge, the carefully maintained *nextMsgId* values must match, otherwise different communication failures are detected based on the next received message.

3 Preliminary results

1. *Bandwidth measurements.* Defining testcases with policies containing 'continuously sending' edges let us measure the network's *maximal throughput*. Using problem 4 (see Fig. 3.), we could measure the effective speed of the Iris (18.17 kbps) and TelosA (14.33 kbps) platforms by evaluating the mean of results of multiple tests with packet size of 16 bytes (Fig.4).

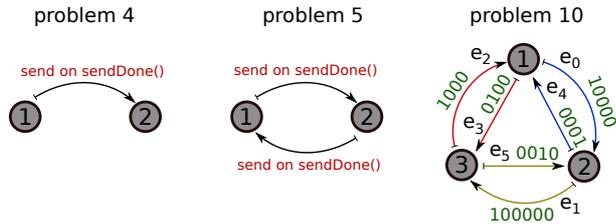


Figure 3. Heavy wireless traffic problems

2. *MAC parameter optimization.* Furthermore, after changing backoff parameters [19], [12] of the random CSMA/CA algorithm in the RF230 radio chip's driver, we measured 47.62 kbps on the Iris motes with the same configuration as before. This means a **162.1% performance increase** without introducing additional message loss and communication failures!

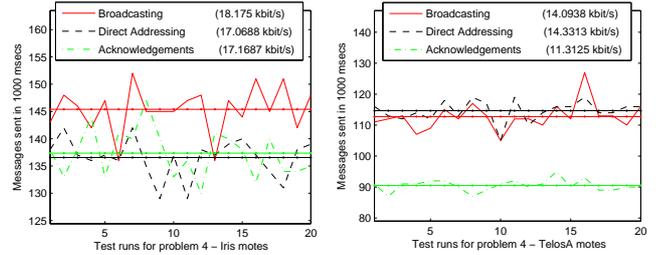
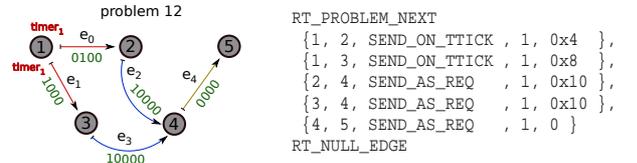


Figure 4. Effective speeds of Iris and TelosA

Packet loss information can also be derived from the missedCount and sendSuccessCount statistical values to characterise the wireless channel and to help choose the appropriate communication mode.

3. *Medium Access Control verification.* Furthermore we were able to use our tool with success for verifying the Low Power Listening [14], [9] components and radio stack layers. The LPL component must guarantee the proper message transmission while it duty cycles the radio chip to save power. If a message transmission request occurs in a particular mote, it wakes up and starts transmitting the message over and over again for at least *wakeup interval* time, which should guarantee in principle the reception of the message.

The best opportunity to analyze the behaviour of the LPL layer is to construct such networks where message transmission requests occur at known times and set the *wakeup interval* so that these two do not match [10].



As an example the results for the (*wakeup_interval* = 10msec, *timer_freq* = 50msec) pair are presented in table 1. It clearly shows that every message has been transmitted successfully either the LPL was enabled or disabled.

wi (msecs)	edge 0		edge 1		edge 2		edge 3		edge 4	
	0	10	0	10	0	10	0	10	0	10
trigger	19	19	19	19	19	19	19	19	38	38
backlog	0	0	0	0	0	0	0	0	10	6
sendDone	19	19	19	19	19	19	19	19	28	32
receive	19	19	19	19	19	19	19	19	28	31

Table 1. non-LPL and LPL values for $t_f = 50$ msecs

4. *Debugging* We have also been able to successfully debug acknowledgement related errors which occurred when Iris and TelosB motes were used in LPL mode in a multi-hop network. The problem was originated from the fact that Iris is faster than TelosB in responding acknowledgement packets, so it did not wait long enough for acknowledgements from Telos motes.

The optimal value of the receive check interval of LPL was also measured with this framework.

4 Reproducibility

While we designed the proposed framework to be run on real hardware to acquire real measurements rather than laboratory-sealed ones, there was also a requirement that performing the same test multiple times in the same environment should give us the same results. Note that differences may occur in the statistical numbers however we would better measure real than conceptual values.

Without this important fact, we could not rely on the values in comparing Medium Access Control protocols. For example, bypassing the wireless channel's unreliability – ex. using a cable – was a no option because real RSSI, LQI values are anticipated.

5 Ongoing and future work

There are some known limitations of the current implementation. In the near future

- multi-edges (for broadcast communication links),
- independent communication modes and trigger periods
- random timer seeds for simulation clock skews and random timing values

will be supported as well. Besides, the framework does not scale well during the configuration and downloading phases – since nodes must be in the reception range of the *Base-Station*, however the simulation phase is ready for multi-hop operation.

A separate library component – being able to measure power consumption, task latency, atomic section lengths and interrupt-context lengths – will also be incorporated.

6 References

- [1] R. Ahmad, F. Zheng, M. Driberg, S. Olafsson, and M. Fitch. Modelling energy consumption of relay-enabled MAC protocols in ad hoc networks. In *ISWPC'09: Proceedings of the 4th international conference on Wireless pervasive computing*, pages 236–241, Piscataway, NJ, USA, 2009. IEEE Press.
- [2] G.-S. Ahn and et. al. Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 293–306, New York, USA, 2006. ACM.
- [3] S. Aslam, F. Farooq, and S. Sarwar. Power consumption in wireless sensor networks. In *FIT '09: Proceedings of the 6th International Conference on Frontiers of Information Technology*, pages 1–9, New York, NY, USA, 2009. ACM.
- [4] J. Beutel and et al. Automated wireless sensor network testing. In *INSS'07: Networked Sensing Systems*, pages 303–303, 2007.
- [5] R. Chau. Modelling of medium access control (MAC) protocols for mobile ad-hoc networks. Technical Report DSTO-TN-0637, 06 2005.
- [6] S. Ergen and P. Varaiya. PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 5(7):920–930, 2006.
- [7] G. Halkes and et al. Comparing energy-saving MAC protocols for wireless sensor networks. *Mobile Networks and Applications*, 10:783–791, 2005.
- [8] V. Handziski and et al. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 63–70, New York, NY, USA, 2006. ACM.
- [9] R. Jurdak, P. Baldi, and L. C. Videira. Adaptive low power listening for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 6(8):988–1004, 2007.
- [10] S. Lee, J. Choi, J. Na, and C.-k. Kim. Analysis of dynamic low power listening schemes in wireless sensor networks. *Comm. Letters.*, 13(1):43–45, 2009.
- [11] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137, New York, NY, USA, 2003. ACM.
- [12] Y. Li, K.-P. Long, W.-L. Zhao, and F.-R. Yang. RWBO(pd,w): a novel backoff algorithm for IEEE 802.11 DCF. *J. Comput. Sci. Technol.*, 20(2):276–281, 2005.
- [13] J. Mo, H.-S. W. So, and J. Walrand. Comparison of multichannel MAC protocols. In *IEEE Transactions on Mobile Computing*, volume 7, pages 50–65, 2008.
- [14] D. Moss, J. Hui, and K. Klues. *Low Power Listening*. Core Working Group, tep 105 documentary for tinyos 2.x edition.
- [15] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.
- [16] B. Titzer, D. Lee, and J. Palsberg. Avrora: scalable sensor network simulation with precise timing. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 67, Piscataway, NJ, USA, 2005. IEEE Press.
- [17] Q. Wang. A realistic power consumption model for wireless sensor network devices.
- [18] K. Whitehouse and et al. Marionette: using rpc for interactive development and debugging of wireless embedded networks. In *ISPN'06: Information Processing in Sensor Networks*, pages 416–423, 2006.
- [19] L. Yun, L. Ke-Ping, Z. Wei-Liang, and C. Qian-Bin. A novel random backoff algorithm to enhance the performance of IEEE 802.11 DCF. *Wirel. Pers. Commun.*, 36(1):29–44, 2006.
- [20] M. Zhao, Z. Chen, and Z. Ge. QS-Sift: QoS and spatial correlation-based medium access control in wireless sensor networks. *Int. J. Sen. Netw.*, 2(3/4):228–234, 2007.