Automated Wireless Sensor Network Testing

Jan Beutel, Matthias Dyer, Roman Lim, Christian Plessl, Matthias Wöhrle, Mustafa Yücel, Lothar Thiele
Computer Engineering and Networks Lab, ETH Zurich
8092 Zurich, Switzerland
beutel@tik.ee.ethz.ch

***Abstract*— The design of distributed, wireless, and embedded system is a tedious and error-prone process. Experiences from previous real-world Wireless Sensor Network (WSN) deployments strongly indicate that it is vital to follow a systematic design approach to satisfy all design requirements including robustness and reliability. Such a design methodology needs to include an end-to-end testing methodology. The proposed framework for WSN testing allows to apply distributed unit testing concepts in the development process. The toolflow decreases test time and allows for monitoring the correctness of the implementation throughout the development process.**

I. Introduction

Recent publications have reported, that several WSN deployments exhibit poor performance and unexpected failures [1]. In many cases, the root cause for the faulty operation remains largely unknown [2]. The characteristics of WSNs renders the design of applications inherently challenging. A WSN is a distributed system of tightly constrained nodes exhibiting a large degree of parallelism. The unreliable communication channel requires robust and resilient communication protocol design. Since many WSNs aim at continuous and autonomous operation for up to multiple years with only limited or no access and supervision, deploying a correct WSN application is of utmost importance. We argue that for implementing sustainable WSNs it is key to follow a systematic design approach accompanied by an end-to-end test methodology. We propose a novel test methodology and framework, bringing the central concepts of distributed unit testing to WSN software design.

II. Test methodology

In our approach, we apply distributed unit testing techniques to WSNs. Each test case applies well-defined stimuli to the application, monitors the results, and checks that the results conform to the rules defined by the requirements. The result of each test is a hard pass or fail condition. This methodology allows to use the same test cases for testing on different test platforms, such as simulation or execution on a testbed. The seamless transferability of test cases between test platforms allows for guiding the development process from specification to implementation while continuously ensuring the correctness of the implementation.

III. Framework for Automated WSN Testing

The methodology is implemented by integrating well-known WSN development tools, such as WSN simulators (TOSSIM [3]), instruction set simulators (AVRORA [4]) and testbeds (DSN [5]) into a comprehensive testing framework. In this demonstration, we present our current implementation of the test methodology for TinyOS 2 applications. As depicted in figure 1, the framework supports two test platforms: execution in the TOSSIM simulator and execution on Moteiv Tmote Sky targets connected to a wireless testbed (DSN).

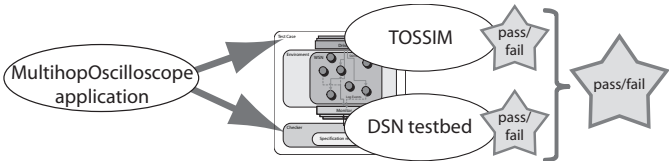

Fig. 1. A single application is checked on multiple test platforms using a single test specification. In this example a TinyOS 2 application is tested using a simulator (TOSSIM) and a testbed (DSN).

The testing framework presented is not limited to a single style of development, specific tools or languages (nesC) but allows to integrate a set of suitable and problem specific tools. The testing framework consists of (i) test drivers providing stimuli, (ii) test monitors collecting results, (iii) a testing environment, and (iv) test checkers based on the requirements. A detailed description including an implementation and case-study can be found in [6]. With the use of the demonstrated test methodology and underlying framework the tedious and error-prone process of developing WSN software is considerably enhanced, leading to a correct implementation based on reproducible testing techniques.

References

[1] K. Langendoen, A. Baggio, and O. Visser, "Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture," in *Proc. 20th Int'l Parallel and Distributed Processing Symposium (IPDPS 2006)*. IEEE, Piscataway, NJ, Apr. 2006, pp. 8–15, 14th Int'l Workshop Parallel and Distributed Real-Time Systems (WPDRTS 2006).

[2] J. Choi, J. Lee, M. Wachs, and P. Levis, "Opening the sensornet black box," Stanford Information Networks Group, Stanford University, CA, Tech. Rep. SING-06-03, 2006.

[3] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st ACM Conf. Embedded Networked Sensor Systems (SenSys 2003)*. ACM Press, New York, Nov. 2003, pp. 126–137.

[4] B. Titzer, D. Lee, and J. Palsberg, "Avrora: Scalable sensor network simulation with precise timing," in *Proc. 4th Int'l Conf. Information Processing Sensor Networks (IPSN '05)*. IEEE, Piscataway, NJ, Apr. 2005, pp. 477–482.

[5] M. Dyer, J. Beutel, L. Thiele, T. Kalt, P. Oehen, K. Martin, and P. Blum, "Deployment support network - a toolkit for the development of WSNs," in *Proc. 4th European Workshop on Sensor Networks (EWSN 2007)*, ser. Lecture Notes in Computer Science, vol. 4373. Springer, Berlin, Jan. 2007, pp. 195–211.

[6] J. Beutel, C. Plessl, and M. Wöhrle, "Increasing the reliability of wireless sensor networks with a unit testing framework," Computer Engineering and Networks Lab, ETH Zürich, Switzerland, Tech. Rep. 272, Mar. 2007.