Poster Abstract: A Benchmark for Low-power Wireless Networking

Simon Duquennoy[1,2], Olaf Landsiedel[3], Carlo Alberto Boano[4], Marco Zimmerling[5],
Jan Beutel[6], Mun Choon Chan[7], Omprakash Gnawali[8], Mobashir Mohammad[7],
Luca Mottola[9,2], Lothar Thiele[6], Xavier Vilajosana[10], Thiemo Voigt[11,2], Thomas Watteyne[1]

[1]Inria, France [2]SICS Swedish ICT, Sweden [3]Chalmers University of Technology, Sweden
[4]Graz University of Technology, Austria [5]TU Dresden, Germany [6]ETH Zurich, Switzerland
[7]NUS, Singapore [8]University of Houston, USA [9]Politecnico di Milano, Italy
[10]Open University of Catalonia, Spain [11]Uppsala University, Sweden

ABSTRACT

Experimental research in low-power wireless networking lacks a reference benchmark. While other communities such as databases or machine learning have standardized benchmarks, our community still uses ad-hoc setups for its experiments and struggles to provide a fair comparison between communication protocols. Reasons for this include the diversity of network scenarios and the stochastic nature of wireless experiments. Leveraging on the excellent testbeds and tools that have been built to support experimental validation, we make the case for a reference benchmark to promote a fair comparison and reproducibility of results. This abstract describes early design elements and a benchmarking methodology with the goal to gather feedback from the community rather than propose a definite solution.

1. INTRODUCTION

The low-power wireless community has put a lot of effort in building testbeds and tools to enable realistic experimental validation [1, 3, 4]. While this was a necessary step to take, we argue the next step is long overdue. More than a decade after the first SenSys conference, new low-power wireless networking protocols continue to appear each year. Many of them are validated experimentally and at scale, but typically with different settings and testbeds than previously published protocols. This practice makes it difficult, if not impossible, to compare results and may even be unfair to some of the considered protocols.

To address these problems, we believe a standardized, yet constantly evolving benchmark for low-power wireless networking is needed. The ideal benchmark enables a fair comparison between new and existing approaches and provides tools to reproduce results across a variety of testbeds and application scenarios. This will raise the bar in the quality of experimental data, and provide researchers and engineers in both academia and industry with an objective view of the strengths and weaknesses among existing protocols.

2. GOALS

With the benchmark, we aim to provide researchers with a standardized way to evaluate their protocols and compare the results in a fair manner to the state of the art. To this end, the benchmarking suite should allow to run protocols on testbeds in a reproducible manner under different settings (*e.g.*, traffic load, traffic pattern, controlled interference) and measure key metrics including end-to-end packet reliability, latency, and radio duty cycle. The defined scenarios, settings, and testbeds should be maintained as a continuous community process, allowing the benchmark to evolve and adapt to technology and application trends.

The benchmarking suite should execute the selected protocols and scenarios automatically on behalf of the user. Ideally, users simply upload their firmware, select a benchmark scenario, and hit the *play* button. The testbed then takes care of running the corresponding experiment and collecting the results. The experimental setup and settings of each scenario in the benchmarking suite, such as traffic load, metrics of interest, and level of controlled interference, are defined using a scripting language (*e.g.*, Python).

The procedure is opposed to the classical approach where the experimental settings, such as traffic load and pattern, are encoded as "application logic" in the firmware. Instead, we move this logic out of the firmware and into the testbed infrastructure. The firmware is solely capable of handling requests received over serial or GPIO, such as *send to node #28*, whilst the testbed will collect and provide information as well as measure end-to-end packet reliability and latency.

Moving the definition and execution of experimental scenarios from the firmware to the testbed has several benefits:

- **Fairness:** The protocol developer is no longer responsible for embedding the experimental scenario into the firmware, a practice that may lead to ambiguous interpretations of experimental settings, such as "a data rate of 1 Hz with jitter." Instead, the testbed ensures that all protocols are transparently tested under the exact same settings.
- **Evolvability:** The benchmark can be changed simply by redefining or adding experimental scenarios specified in a scripting language. It thus becomes possible to evaluate existing protocols in a new scenario without having to access and modify the source code.

We discuss next a preliminary list of features that we consider essential for the benchmark. Note that the actual benchmark shall be evolutive and community-driven.

SenSys '16 November 14-16, 2016, Stanford, CA, USA

ACM ISBN 978-1-4503-4263-6/16/11...$15.00

DOI: http://dx.doi.org/10.1145/2994551.2996692

Scenario Experimental scenarios are characterized by different traffic patterns and loads, which should be adaptable in an unconstrained way. Example traffic patterns that should be part of the benchmark are multipoint-to-point, point-to-multipoint, point-to-point, request-response, and multipoint-to-multipoint. Traffic can be periodic, sporadic (*i.e.*, with a known minimum inter-packet interval), or aperiodic. All traffic patterns are configurable with different packet arrival patterns across time and space, payload sizes, and on different testbeds. In addition to the traffic parametrization, scenarios might include events such as nodes joining and leaving the network.

Wireless Environment Experiments may be run in different wireless environments and levels of interference. For reproducibility of results, we build on JamLab [2] to inject controlled interference by replaying the RF noise of various sources such as IEEE 802.11 devices and microwave ovens. The testbed infrastructure will be in charge of generating the interference while a protocol is being benchmarked, and may also constrain the protocol into a given subset of channels. The test suite should also include runs on various topologies (*e.g.*, line, dense, sparse). As a complement to testbed experiments, we may also add simulations, making benchmark runs reproducible with a deterministic wireless environment.

Metrics Important metrics for the evaluation of low-power wireless communication protocols are reliability, latency, and radio duty cycle. Packet reliability and latency are obtained by looking at logs (transmission and reception events) and their associated timestamps. From the same information, other properties such as out-of-order delivery, duplicates, or the ability to meet given end-to-end deadlines can be inferred. Some testbeds allow to directly measure energy consumption. For other testbeds, we might define a common interface for all nodes to output their radio and CPU duty cycle, which can be used as a proxy for energy consumption and to measure channel utilization and other metrics.

3. METHODOLOGY

We outline next how we envision the use of the benchmark.

Scope The benchmark should complement – not replace – the evaluation of a protocol. Protocols often tackle specific challenges that may not be covered by the benchmark, requiring the use of specific setups, experiments, and metrics. The purpose of the benchmarking suite is to enable a comparison with other protocols across a set of basic scenarios and metrics. We nonetheless see evolvability as a major requirement to adapt the benchmark to future relevant technologies, application needs, and metrics.

Re-running vs. Comparing Results The benchmark provides a common framework to compare protocols. There are essentially two ways such comparison can be made: (1) by re-running other protocols or (2) by directly comparing against previously published benchmark results.

The former approach (re-running) is similar to the current practice in low-power wireless research. Its main advantage is that protocols are tested on the same physical topology and in the same time period. Re-running someone else's protocol is, however, a costly and error-prone procedure, as it requires the same source code, and as a slight misconfiguration may lead to a significant performance drop and hence to an unfair comparison and false conclusions.

The latter approach (comparing results) is closest to what other communities such as databases and machine learning do. By comparing against published benchmarking results obtained by the protocol's authors, this approach rules out problems related to the sub-optimal use of a protocol. It is, however, difficult to apply to wireless research, where experiments are run in an uncontrolled environment.

We propose a methodology that combines both approaches. Whenever there are results available for a protocol and the environment has not changed significantly, comparing numbers may be valid (unless the performance improvement is minimal, which would require additional experiments). If this is not the case, re-running is necessary, and the benchmarking suite will aid researchers by providing a collection of already configured state-of-the-art protocols, ideally including industry-relevant standards. Both approaches rely on an open platform for collecting benchmark scenarios, protocol source code, and experimental results.

Environment Dynamics Characterization To enable a fair comparison in the long run, we aim to formally describe the environment dynamics during an experiment and document it together with the benchmark results. To this end, we propose to devise a set of metrics that capture the overall state of the network, such as link qualities and their distribution in space in time, node density, and network diameter. We also plan to characterize environmental aspects such as temperature (and its gradient) or the level of RF noise in the surroundings using dedicated nodes.

With such meta-information at hand, one can decide whether to re-use published results or to re-run a protocol. We plan to keep track of testbeds evolution, both to detect permanent topology changes and to assess how the environment changes over the course of a day or a week. These data will enable us to better understand to what extent past results can be reused in present and future work.

4. NEXT ACTIONS

In this document, we have sketched a proposal for a low-power wireless networking benchmark. We invite the community to contribute and engage into a lively dialogue that helps to transform our initial ideas into a widely-accepted, standardized benchmarking suite that increases the rigor of experimental validation.

5. REFERENCES

[1] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne. FIT IoT-LAB: A large scale open experimental IoT testbed. In *IEEE WF-IoT*, 2015.

[2] C. A. Boano, T. Voigt, C. Noda, K. Roemer, and M. Zuniga. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *ACM/IEEE IPSN*, 2011.

[3] M. Doddavenkatappa, M. Chan, and A. Ananda. Indriya: A low-cost, 3D wireless sensor network testbed. In *ICST TridentCom*, 2011.

[4] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *ACM/IEEE IPSN*, 2013.