Poster Abstract: Automatic Configuration of Controlled Interference Experiments in Sensornet Testbeds

Felix Jonathan Oppermann†, Carlo Alberto Boano†, Marco Zimmerling¶, and Kay Römer†

†Institute for Technical Informatics
Graz University of Technology
{oppermann, cboano, roemer}@tugraz.at

¶Computer Engineering and Networks Lab
ETH Zurich
zimmerling@tik.ee.ethz.ch

Abstract

Experiments under controlled radio interference are crucial to assess the robustness of low-power wireless protocols. While tools such as JamLab augment existing sensornet testbeds with realistic interference, it remains an error-prone and time-consuming task to manually select the set of nodes acting as jammers and their individual transmit powers. We present an automated configuration approach based on simulated annealing to overcome this problem. A preliminary evaluation based on two testbeds shows that our approach can find near-optimal solutions within at most a few hours. We believe our approach can facilitate the widespread adoption of controlled interference experiments by the sensornet community.

1 Introduction and Motivation

Radio interference negatively affects the performance of wireless sensor networks (WSNs). The presence of co-located wireless devices transmitting at higher power (e.g., Wi-Fi access points) or nearby appliances generating RF noise (e.g., microwave ovens) typically results in high packet loss rates, high latencies, and reduced energy efficiency. Consequently, there is a strong need to study the performance of a network in the presence of a congested channel, and to design protocols that can deliver high and stable performance despite interference. This requires testbed infrastructures in which realistic interference patterns can be created in a precise and repeatable way. Installing additional hardware for interference generation can be very costly (e.g., adding expensive VSG-based EMI generators [4]) and labor-intensive (e.g., adding Wi-Fi access points, microwave ovens [2]), and would not scale to large testbeds.

JamLab [1] is a tool to augment existing WSN testbeds with realistic interference generation without the need of extra hardware [1]. It employs a fraction of the existing nodes in a testbed to regenerate the interference patterns produced by common wireless devices, and essentially partitions the area of a testbed into cells, as shown in Figure 1. Each cell contains a jammer and a number of regular nodes that can be used for experimentation, such that each regular node can be interfered by the jammer despite its limited RF output power. The original JamLab work proposes a manual process to iteratively select a suitable set of nodes as jammers. However, when augmenting large WSN testbeds such as Indriya [5] and TWIST [3], this process is time consuming, as it would require several experiments, as well as knowledge about the position of the nodes.

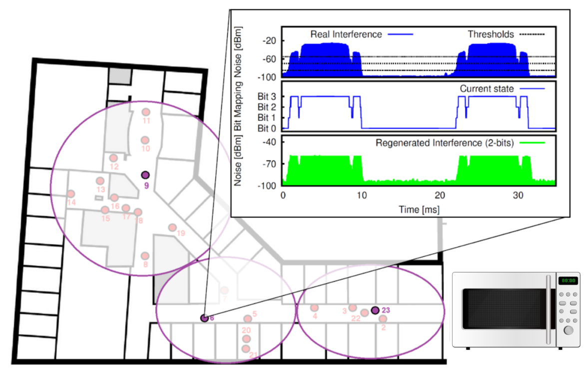

Figure 1. JamLab's architecture. A fraction of nodes acts as jammers, dividing the testbed area into different cells.

In this work, we propose an automatic jammer selection method for JamLab based on simulated annealing metaheuristic optimization, which can provide an optimal testbed configuration without the need for user interaction, and by limiting the effort to a one-time data collection. For the data collection we employ the same tool as used in the manual configuration process [1] to collect the received signal strength $rssi(y,x,p)$ when sending a message from node x with transmit power p to node y and to generate a signal strength matrix for the testbed.

2 Optimization Problem

When augmenting a WSN testbed using JamLab, the subset of jammers J selected from the set of all nodes N should be reduced as much as possible, as jammers are devoted to

SenSys'14, November 3–5, 2014, Memphis, TN, USA.
ACM 978-1-4503-3143-2/14/11.
http://dx.doi.org/10.1145/2668332.2668355

interference generation and cannot be used for experimentation. Furthermore, a number of constraints need to be fulfilled: (1) Each regular node in a cell needs to be within reach of a jammer; (2) To ensure all incoming messages are blocked, the jammer needs to produce an interfering signal with a strength that is at least $\tau = 3\,\text{dB}$ higher than the maximum strength of any other signal that a regular node may receive in order to block its reception[1]; (3) A regular node should be influenced by exactly one jammer in order to avoid cross-talk between different cells. To ensure this, the received signal strength of all other jammers needs to be at least 3 dB lower than that from any other node; (4) The network of remaining regular nodes should be connected.

The system selects from the set of all nodes N a subset J as jammers. In addition, the sending power level $power(j)$ of each jammer $j \in J$ can be adjusted in order to control the cell size. Nodes not selected as jammer belong to set R. The power level setting of these nodes is fixed and $power(r)$ returns a constant value for all regular nodes $r \in R$. The function $rssi(x, y, p)$ returns the previously collected received signal strength of the link between nodes x and y with a sending power setting of p at node x measured using the tool from [1] as described above. Nodes x, y with a link that has a received signal strength above the receiver's sensitivity threshold σ are considered to be neighbors, hence $neighbor(x, y) = true$. Based on these definitions, we can formulate the following constrained optimization problem. Each constraint maps directly to the respective textual description introduced above.

$$
\begin{aligned}
\text{minimize} \quad & \|J\| \\
\text{subject to} \quad & \forall n \in R\ \exists j \in J: && (1)\\
& \quad \text{neighbor}(j,n), \\
& \forall n_1, n_2 \in R\ \exists! j \in J: && (2)\\
& \quad \text{neighbor}(j,n_1) \wedge \text{neighbor}(n_2,n_1) \\
& \quad \rightarrow \text{rssi}(j,n_1,\text{power}(j)) > \text{rssi}(n_2,n_1,\text{power}(n_2)) + \tau, \\
& \forall n_1, n_2 \in R\ \exists! j_1 \in J\ \forall j_2 \in J: && (3)\\
& \quad \text{neighbor}(j_2,n_1) \wedge \text{neighbor}(j_1,n_1) \wedge \\
& \quad \text{neighbor}(n_2,n_1) \wedge j \neq k \\
& \quad \rightarrow \text{rssi}(n_2,n,\text{power}(n_2)) > \text{rssi}(j_2,n_1,\text{power}(j_2)) + \tau, \\
& \forall n_1, n_2 \in R\ \exists (k_1,k_2,\ldots,k_p) \forall 0 < i < p: && (4)\\
& \quad k_1 = n_1 \wedge k_p = n_2 \wedge \text{neighbor}(k_i,k_{i+1}) \\
\text{with} \quad & \text{neighbor}: N \times N \rightarrow \mathbb{B} \\
& (x,y) \mapsto \begin{cases} \text{true} & \text{rssi}(x,y,\text{power}(x)) > \sigma \\ \text{false} & \text{otherwise} \end{cases}
\end{aligned}
$$

3 Prototype Implementation

The optimization process is implemented as a Python program that employs the simulated annealing meta heuristic to find a near-optimal solution for the above optimization problem. As the default algorithm of simulated annealing does not support constraints, we transformed all constraints into additional objectives. To still be able to use a single-objective algorithm, these are combined in a single utility function, by forming a weighted sum of the normalized metrics. Constraints are given more weight than the original objective of minimizing $||J||$, to ensure that infeasible solutions are very unlikely to be accepted. As a complete elimination of cross-talk turned out to be impossible for larger testbeds and the synchronous version JamLab can handle a limited amount of cross-talk, it was only given a very low weight in order to find solutions with few jammers.

[1]Typical WSN radios are able to receive the stronger out of two signals if the second signal is at least 3 dB weaker (co-channel rejection threshold [1]).

Table 1. Properties of exemplary configurations for two testbeds (mean values, min. and max. in parentheses).

Testbed	Number of jammers	Available nodes	Nodes affected by cross-talk
TU Graz Testbed	1 (1, 1)	16 (16, 16)	0 (0, 0)
TWIST Testbed	33 (31, 34)	44 (43, 46)	44 (43, 46)

4 Preliminary Evaluation

For a preliminary evaluation of the performance of the automatic configuration process, we executed it with data from a smaller testbed at TU Graz consisting of 17 nodes in a single room and the TWIST testbed. The latter consists of 102 nodes, out of which 77 were operational, set up in a grid-like network spread over several rooms on three floors. The optimization process was executed eight times for each testbed. Within a maximum of two hours, the current algorithm is able to find suitable configurations as can be seen in Table 1. The number of jammers is low enough to leave a sufficient number of nodes for the actual experimentation. While the simple solutions for the local testbed with just a single jammer do not suffer from cross-talk, this is an issue of the proposed configurations for the TWIST testbed. In the TWIST testbed, most nodes are affected by cross-talk, but typically only by one or two additional jammers with a weak signal. The synchronous variant of JamLab is able to handle cross-talk, so these configurations are still useful.

5 Next Steps

We plan to optimize the execution time and stability of the configuration process by systematically adjusting the parameters of the optimization process and by conducting a more thorough evaluation on a larger set of WSN testbeds.

Acknowledgements

The research leading to these results has received funding from the European Union 7^{th} Framework Programme (FP7-ICT-2011-8) under grant agreement n° 317826 (RELYonIT) and from Nano-Tera, projects X-Sense and OpenSense.

6 References

[1] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Proc. of the* 10^{th} *IPSN Conference*, Apr. 2011.

[2] F. Lemić et al. Demo: Testbed infrastructure for benchmarking rf-based indoor localization solutions under controlled interference. In *Proc. of the* 11^{th} *EWSN Conference*, Feb. 2014.

[3] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proc. of the* 2^{nd} *REALMAN Workshop*, May 2006.

[4] J. Slipp et al. WINTeR: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments. In *Proc. of the* 6^{th} *CNSR Conference*, May 2008.

[5] M. Doddavenkatappa et al. Indriya: A low-cost, 3d wireless sensor network testbed. In *Proc. of the* 7^{th} *TridentCom Conference*, Apr. 2011.