

Shawn: The fast, highly customizable sensor network simulator

Sándor P. Fekete and Alexander Kröller
Institute for Mathematical Optimization
Braunschweig University of Technology
Braunschweig, Germany
Email: {s.fekete,a.kroeller}@tu-bs.de

Stefan Fischer and Dennis Pfisterer
Institute for Telematics
University of Lübeck
Lübeck, Germany
Email: {fischer,pfisterer}@itm.uni-luebeck.de

Abstract—Shawn is a discrete event simulator for sensor networks. Due to its high customizability, it is extremely fast but can be tuned to any accuracy that is required by the simulation or application.

Keywords—Simulation; Discrete Events; Rapid Algorithm Engineering, Wireless Sensor Networks

I. INTRODUCTION

Nowadays, a plethora of tools to simulate wireless sensor networks exists. The most prominent one is the Network simulator-2 (Ns-2)[1]. The utmost goal of these well-known simulators is to be as “realistic” as possible by simulating physical effects, data and message encoding, wireless interference effects, processor limitations, etc. However, high accuracy comes at the price of long simulation times. When developing sensor network algorithms on the application layer, most often this accuracy is not required. E.g., during testing and improving an algorithm, the developer is usually only interested in its results, quality and whether it terminates correctly. Furthermore, the large number of factors that influence the behavior of the whole network renders it nearly impossible to isolate a specific parameter of interest.

II. SHAWN

Against this background, we have developed our proposed simulation framework called Shawn [2]. Its central idea is to replace low-level effects with abstract and exchangeable models, such that the simulation can be used for huge networks in reasonable time whilst keeping the focus on the actual research problem.

For maximum performance, Shawn is written in C++ and is known to run under Windows and many flavors of Unix/Linux. Released under the BSD license, it is currently in active development and use by different universities and companies to simulate wireless (sensor) networks. It was and continues to be an invaluable tool for many research publications and it has been successfully applied in numerous bachelor and master theses.

Shawn’s internal architecture differs substantially from those of other simulators and performance and flexibility have been our primary design guideline. It finishes simulations

in less than a minute where Ns-2’s running time already exceeds 25 hours [3]. Shawn is tailored for simulations of huge networks with a few 100,000 nodes and more, where physically accurate simulators fail completely.

The design rationale behind Shawn is to *simulate the effects of a phenomenon and not the phenomenon itself* by using abstract models and implementations of these models. A user can select the actual granularity and behavior and is hence able to adapt the simulation to his specific needs. E.g., to let message transmissions always succeed within constant time, with probabilistic message loss or, if physical accuracy is really necessary, by simulating the effects of a complete OSI stack. For almost every aspect in Shawn, one can choose between an abstract, simplified and a realistic model implementation. The former is used if the particular aspect is not important in the current development phase, and generally extremely fast. The latter is preferable when this certain simulation aspect is the current focus of investigations.

Another major benefit of Shawn is that users are not forced to implement every detail of the simulation in a distributed manner. The developer has full access to the communication graph, all nodes and their data. So, he can simply implement some details as centralized graph algorithms and defer the transformation into a distributed variant until this is actually required. This makes it possible to begin with a simple algorithmic sketch and to turn it step-by-step into a full-blown distributed protocol. This leads to a rapid and less error-prone development cycle.

Other simulators often trigger their simulated nodes periodically (e.g. every few ms). So, even inactive nodes slow down the simulation, and effects happening in less than this fixed interval are hard to grasp. Shawn, in contrast, uses the concept of discrete time to speed up simulations where node polling happens at much larger intervals, but events can be scheduled to happen in between at any time with any precision.

REFERENCES

- [1] Ns-2: Network simulator-2, <http://www.isi.edu/nsnam/ns/>.
- [2] Shawn, <http://shawn.sf.net>.
- [3] A. Kröller, D. Pfisterer, C. Buschmann, S.P. Fekete, and S. Fischer, *Shawn: A new approach to simulating wireless sensor networks*, Proceedings of the Design, Analysis, and Simulation of Distributed Systems (DASD '05), San Diego, 2005.