MAP++: Support for Map-Based WSN Modeling and Design with OMNeT++

(Poster Abstract)

Piotr Szczytowski, Abdelmajid Khelil, Neeraj Suri
Technische Universität Darmstadt, DEEDS Group
Hochschulstr. 10
64289 Darmstadt, Germany
{szczytowski, khelil, suri}@informatik.tu-darmstadt.de

Categories and Subject Descriptors

C.2.1 [**Computer - Communication Networks**]: Network Architecture and Design - network communications, wireless communication.; I.6.5 [**Simulation and Modeling**]: Model Development - modeling methodologies.

General Terms

Wireless Sensor Networks Modeling and Simulation

Keywords

Wireless Sensor Networks, OMNeT++, Modeling and Visualization, Map-based Design, Performance Evaluation

1. INTRODUCTION

Wireless Sensor Networks (WSN) are receiving growing attention in the research community. As simulation is a frequently used to test and validate the different approaches, the flexibility of the simulation environments to support the varied WSN schema is desirable. The multitude works regarding WSN in general [7, 8], and OMNeT++ [1, 2, 5, 6] in particular, confirm the rising interest in providing the simulation environment for WSNs. As map based design is most appropriate for WSN[3], unfortunately, no project within the WSN community investigates map paradigms for simulations. The map paradigm builds on the region principle and therefore, provides excellent modeling primitives for WSNs. Global maps are created for the sake of network monitoring [3, 10] or of event detection [3, 9].

In this context, we propose MAP++ as a framework serving as an extension to the established network simulator OMNeT++. It provides following functionalities for support of WSN simulations:

- Extended topology generation, including setting of spatially correlated initial conditions and batch script generation for simplifying modeling and scenarios generation,
- Trace data visualization support that identifies spatial dependencies for design and debugging,
- Processing and analysis of trace data with support of relational databases for performance evaluation.

Research supported in part by DFG GRK 1362 (TUD GKmM), EC INSPIRE, EC CoMiFin.

OMNeT++ 2009, Rome, Italy.

2. MAP++ ARCHITECTURE

Figure 1 depicts the MAP++ framework architecture and its interactions with the OMNeT++ simulator. *Scenario Generator* is responsible for generation of the topologies designed by the user. *Batch Creator* creates simulation scripts for batch execution of defined scenarios. *Snapshots Configurator* configures and embeds the *Trace module* into the topology and provides means of configuring its behavior. The OMNeT++ simulator executes delivered scenarios, and embedded trace modules produce *Traces*. The *Visualization* and *Regioning* use the collected trace data for result representation and *SQL Converter & Database Interface* for performance analysis.

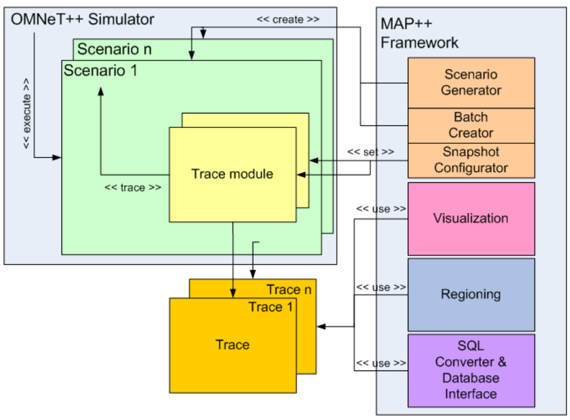

Figure 1: MAP++ Architecture

3. MAP++ BENEFITS

Usually, the simulation process is composed of three stages: modeling, execution and analysis. In this section we present the MAP++ benefits at each stage.

In the modeling stage MAP++ supports the definition of the modules, the generation of the topology and creation of scenario sets. Definition of modules consists of the identification of published attributes of certain module class. The topology generation results in automatized creation of the OMNeT++ NED files. Scenario sets correspond to producing the initialization file with sections describing varying parameters of execution. Also the visualization of scenarios is supported at this stage.

Regarding the execution stage of the simulation, our framework currently offers only assistance in the form of the trace module, which monitors execution of the simulation and produces trace data. However we envision extending the usability of the trace module in the future. Conceptually the trace module could evaluate the maps at the run-time. That would allow definition of conditions to which set of actions could be assigned. Other possibility of utilizing the framework is the run-time map visualization.

For results analysis the visualization is provided. It is rendered by loading the topology definition and creating its two dimensional representation. The area occupied by nodes is fragmented into Voronoi [4] polygons, bringing the additional benefit of reflecting the node density. The individual polygons are filled with the shade of grey color corresponding to the value of selected property at given time instant. Basic operations like creation of differential maps are supported. Scrolling along time axis displays development of phenomenons taking place in simulation. The multitude of the data collected during the simulations is not easy to process. The researchers are mostly interested only in certain information. Our framework overcomes this problem by exporting trace data to SQL database which simplifies querying this data.

4. MAP++ USAGE SCENARIOS

In this section, we present the main usage scenarios of our framework, which include modeling, design and debugging and performance evaluation.

4.1 Modeling

Modeling begins by generating the topology corresponding to the assumed system model. Our framework allows the specification of the communication range, size of the sensor field, number of nodes and type of topology, e.g., grid or random. MAP++ also supports the creation of the heterogenous scenarios by specifying and adding supplemental types of nodes to already created topologies.

After topology generation, accurate modeling of the factors influencing the behavior of WSN is needed. Because of their spatial nature, the natural approach is an iterative generation of the maps for different topologies and simulation parameter sets followed by their comparison. Our framework partially automatizes this process. Varying the values of the simulation properties provides batch generation of scenarios, which allow investigation of their impact.

4.2 Design

The core contribution of our framework in the design phase is the visualization of the maps. WSN designers can use our visualization tools to test tentative solutions and their preliminary performance.

The execution of the batch of scenarios creates trace data that can be loaded into the visualization tool. For example displaying the residual energy map along with topology information may clarify, whether the reason of the observed energy hotspots is the routing algorithm or sensing activity.

Static snapshot does not provide enough insight into the dynamics of the network therefore our visualization supports smooth scrolling through the trace data over time. This procedure unveils the causality of the events during the lifetime of the network.

We also allow for map-based tracing. Such a trace gives the set of regions constituting the map and list of region splitting and merging operations for a certain time interval. In order to set region information we implemented a regioning technique. Regioning algorithm logically groups neighboring nodes belonging to the same value class.

4.3 Debugging and Performance Evaluation

The map abstraction localizes the spatially correlated errors. The accuracy of in-network sensor data aggregation can be evaluated by comparing the ideal map with the one created by the aggregation algorithm. Comparison is the differential map of both views.

Besides visualization, the trace data can be evaluated using the MAP++ database interface. After selecting the active database, the query tool automatically searches for a set of changing parameters and creates a list of their distinct values. A user may query only particular nodes belonging to some region from certain snapshot, to which the user navigated using the visualization. The query tool automatically generates query that uses current time index and addresses only the selected nodes.

5. CASE STUDY

As a case study we took the visualization of *eScan* [10] algorithm accuracy. eScan is an energy map construction algorithm based on polygon aggregation. We realized that relative differential map, represents a powerful tool for illustrating the accuracy of the eScan algorithm. Analyzing the created maps we could determine the optimal parameters for the protocol in the considered setting and observe spatial concentration of the measurement errors. Because of lack of map-based simulation support the designers of eScan were not able to do such investigations.

6. REFERENCES

[1] Mobility Framework for OMNeT++. `http://mobility-fw.sourceforge.net/`.

[2] OMNeT++ Community Site. `http://www.omnetpp.org/`.

[3] A. Khelil et al. MWM: A Map-based World Model for Wireless Sensor Networks. In *Autonomics*, 2008.

[4] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[5] Köpke et al. Simulating Wireless and Mobile Networks in OMNeT++ The MiXiM Vision. In *OMNeT++ Workshop*, March 2008.

[6] J. Lessmann and T. Heimfarth. Flexible Offline-Visualization for Mobile Wireless Networks. In *UKSIM*, 2008.

[7] S. McCanne and S. Floyd. NS Network Simulator. `http://www.isi.edu/nsnam/ns/`.

[8] P. Levis et al. Tossim: accurate and scalable simulation of entire tinyos applications. In *SenSys*, 2003.

[9] W. Xue et al. Contour Map Matching For Event Detection in Sensor Networks. In *ACM SIGMOD*, 2006.

[10] Y. Zhao et al. Residual energy scan for monitoring sensor networks. In *IEEE WCNC*, 2002.