

A Simulation Template for Wireless Sensor Networks

Stefan Dulman, Paul Havinga
EEMCS Faculty, University of Twente, the Netherlands
Emails: {dulman, havinga}@cs.utwente.nl

Abstract—In this paper we introduce a new simulation template for wireless mobile sensor networks. It is built upon the OMNeT++ networks simulator and it contains the main features needed for simulating large ad-hoc networks of autonomous mobile nodes with sensing capabilities. Simulations can be performed both for the networking issues and for various distributed algorithms.

The template provides besides the easy integration of new code, a graphical interface useful for debugging or illustration purposes and a collection of tools that help modifying or implementing new features in an easy fashion.

The code is still under development. This paper gives an outline of what the template presently offers and on what features are planned to be added to it.

I. INTRODUCTION

In this paper we introduce a new simulation template for wireless mobile sensor networks [2]. It is built upon the OMNeT++ networks simulator [3] and it contains the main features needed for simulating large ad-hoc networks of autonomous mobile nodes with sensing capabilities. The wireless sensor networks are a particular example of autonomous decentralized systems. They are made up of hundreds of sensor nodes (having limited resources) that organize themselves into an ad-hoc network. There is no central server to control this network.

We have designed our template because the OMNeT++ simulator does not include support for mobile networks that communicate using radios. We chose to design this framework in such a way that allows easy modifications of the main parameters and, at the same time, the implementation details are transparent to the user.

The template provides besides the easy integration of new code a graphical interface useful for debugging or illustration purposes and a collection of tools that help modifying or implementing new features in an easy fashion. The code is still under development. This paper gives an outline of what the template presently offers and on what features are planned to be added to it.

This work is performed as a part of the European EYES project (IST-2001-34734) on self-organizing and collaborative energy-efficient sensor networks. It addresses the convergence of distributed information processing, wireless communication and mobile computing [1]. Part of the code was developed while visiting the Telecommunication Networks Group from the Technical University of Berlin.

II. TEMPLATE DESCRIPTION

In this section we will briefly introduce the OMNeT++ simulator and then the main characteristics of the template we are developing.

A. OMNeT++ simulator

The OMNeT++ simulator is a simulator originally designed for fixed, wired, distributed systems (such as: computer networks, multiprocessor systems, etc.). Its main features are:

- it is a discrete time simulator. The simulated objects communicate with each-other by exchanging messages at discrete moments of time;
- it is written in C++ and Tcl/Tk. One of the main advantages is the code portability: it runs on DOS, UNIX and WINDOWS without requesting the user to perform any changes;
- several graphical interfaces allows easy debugging and variables inspection. It also offers support for recording data vectors and scalars in output files;
- it offers support for parallel execution;
- simulated objects are represented by modules. The modules can be simple or composed (depth of module nesting is not limited). The modules communicate by messages (sent directly or via gates). Each module description consists of: an interface description and a behavior description;
- several random number generators (also from several distributions) with different starting seeds;
- simulations are easy to configure using .ini file. Batch execution of the same simulation for different parameters is also included.

All the simulated objects (modules, gates and links) can be created either statically (at the beginning of the simulation, from the configuration files), or dynamically (during the simulation).

B. Wireless Sensor Network Simulation Template

Our template is, in fact, a fully working project consisting of a network of mobile nodes that can communicate by wireless means.

The user has to write the code for his own protocols and to integrate it into the template (we provide tools that generate empty modules where this code should reside). Then the user has to specify the general network parameters and the simulations can start.

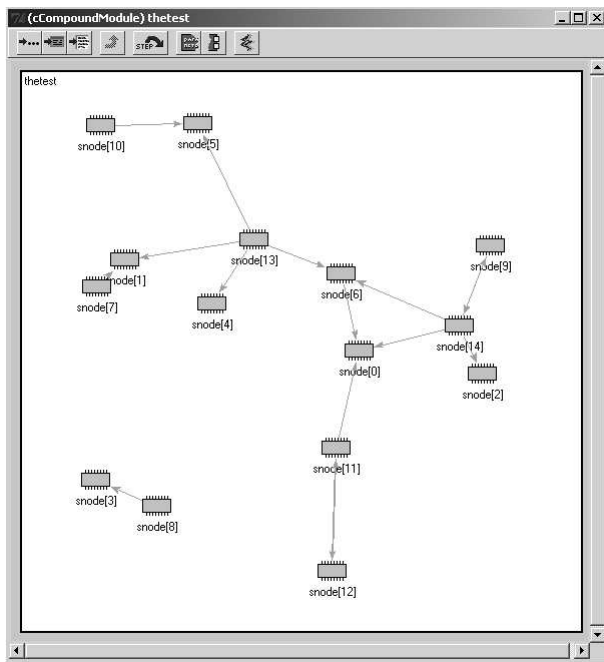


Fig. 1. Simulation Example

All the parameters of the network can be changed without any need of rewriting any code. We have defined (a lot of) easy to use macros for accessing the main functionalities of the code. This way, even if we are making changes to the template and improving the functionality, the already existing code created by users still runs on a newer version.

The already implemented features are:

- mobility is implemented (Random Way Point algorithm by default). Each node is responsible for defining its own trajectory and announcing it to the simulator;
- nodes exchange messages using wireless communication. A message will be heard by all the neighbors situated within the transmission range (the modules within transmission range are connected automatically to each-other);
- the user can specify if unidirectional or bidirectional links have to be used. Each node can specify and update its transmission range independently;
- some basic energy management is also included;
- the nodes have different kinds of failing probabilities (individual failures, failures that affect regions of the map, etc.) Maps for area failures can be specified and used. Other maps can easily be used for obstacles, fading, etc.

In order to perform all of these features we have chosen to use a central manager that takes care of storing the connectivity map and updating it upon request. It also takes care of reading additional configuration files and sending the information to each sensor node. The manager is hidden from the display and all the communication between the nodes and the manager is hidden from the user. This way, the graphical display represents only the network the user is interested in (see Figure 1).

Each sensor node has an architecture similar to the one

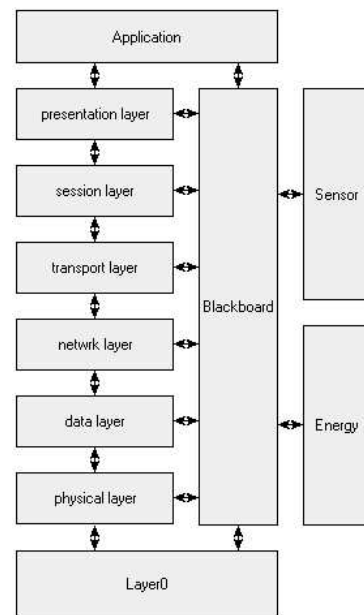


Fig. 2. Sensor Node Description

presented in Figure 2. The OSI network layers are displayed only as an example. The user can modify them by adding, removing or replacing any of them. The internal modules inside each sensor node can exchange information using the module called *blackboard*.

There is another module in there called *layer 0*. It is the interface between two sensor nodes and also between each sensor node and the central manager. It provides all the functionality described above to the custom network layers. When the lower (in the protocol stack) custom layer wants to send a message to the neighbors within transmission range, it just forwards this message to the *layer 0* module.

III. CONCLUSIONS AND FUTURE WORK

The template that we have just briefly described is still under development. In our research group, we have been implemented and tested already several algorithms using this software (MAC protocols, routing algorithms, clustering algorithms, localization algorithms, etc.). It proves to be quite fast and reliable. New features are added upon request.

As future work we plan to add some more features. The most important one is related to making a library with the algorithms we have already implemented. Other algorithms designed for the general OMNeT++ simulator will be adapted to work with the new template. Other additional features include a new traffic generator, an automatic statistical data collection and representation, a central database for data collection, etc.

REFERENCES

- [1] EYES project.
<http://eyes.eu.org>.
- [2] Simulation template website.
<http://wwwhome.cs.utwente.nl/~dulman/tools.htm>.
- [3] OMNeT++ website.
<http://whale.hit.bme.hu/omnetpp/>.