MobASim: a Software Platform for Mobile Ad Hoc Networks Modeling and Simulation

Andrzej Sikora and Ewa Niewiadomska-Szynkiewicz
Institute of Control and Computation Engineering
Warsaw University of Technology, 00-665 Warsaw, Poland
and
Research Academic Computer Network (NASK), 02-796 Warsaw, Poland
Email: A.Sikora@elka.pw.edu.pl, ens@ia.pw.edu.pl

Abstract—This paper addresses issues associated with modeling and simulation of wireless, mobile, and ad hoc networks. Particular attention is paid to an approach for federating parallel and distributed ad hoc networks simulators. We describe the design, functionality, implementation and performance of MobASim system. It is a Java-based software platform for MANET simulation performed on parallel computers or computer clusters. The practical application is provided to illustrate the operation and efficiency of the presented software tool.

I. INTRODUCTION

A mobile, wireless, and ad hoc network (MANET) consists of wireless mobile nodes that can dynamically and freely self-organize into temporary network topologies. The network's topology may change rapidly and unpredictably. There is no prearrangement assumption about specific role each node should perform. Each node makes its decision independently, based on the situation in the domain and its knowledge about the network.

Ad hoc architecture has many benefits, however its flexibility come at a price. Currently research effort is directed toward the specifics and constraints in mobile ad hoc networks, such as: limited transmission range, limited link bandwidth and quality of transmission, constrained resources, mobility and multihop nature of the network [1]–[3]. The complexity and scale of modern ad hoc networks limit the applicability of purely analytic approaches. Computer simulation is widely recognized as an important tool for researchers and engineers. It allows to design and analyze the behavior and performance of networks, and verify new ideas – new protocols, mechanisms, application scenarios, etc. The main difficulty in large scale ad hoc networks simulation is the enormous computation power, i.e., speed and memory requirements needed to execute all events involved by internodes communication, and time varying topology. Another problem is scalability, i.e. how a given simulator scales for large topologies and high speed channels. As a consequence, the developments of methods to speed up calculations has recently received a great deal of interest. Parallel and distributed discrete event simulation (PDES) has already proved to be very useful when performing the analysis of different network systems [2], [4], [5]. Parallel execution of simulation can improve the scalability of a network simulator both in term of network size and execution speed, enabling large scale networks and more network traffic to be simulated in real time.

Various simulators for wireless networks have been proposed [6]–[10]. These tools provide the facility to simulate the protocols in different layers (MAC protocols, routing protocols), and some of them simulate nodes’ mobility. There are a number of possible sets of criteria that could be used for network simulators comparisons, e.g. scalability, model size, execution time, memory requirements, programming interface, etc. Different tools are optimized for different purposes. However, most of available simulators require costly shared-memory supercomputers to run even medium size network simulation and do not support both the user interactions during the experiments and animation of network topology changes. Users set configuration parameters before starting the simulation, and they can see computation results after the experiment is terminated. Moreover, the source coding is usually specialized for a given simulator and it is not easy to learn how to use the tool. In addition, most existing ad hoc networks simulators focus on the wireless transmission modeling and MAC protocols implementation with the lack of the mobility modeling. Usually only simplified mobility models are provided.

This work deals with the description of an integrated software framework MobASim for parallel and distributed simulation of mobile ad hoc networks. We model the MANET simulator using discrete event systems methodology (DEVS) [11], and address the challenges to design high-performance simulation of MANET systems. Our goal was to develop simulator operating in real time, that can help testing of various protocols in different layers (MAC protocols, routing protocols), and some of them simulate nodes’ mobility. There are a number of possible sets of criteria that could be used for network simulators comparisons, e.g. scalability, model size, execution time, memory requirements, programming interface, etc. Different tools are optimized for different purposes. However, most of available simulators require costly shared-memory supercomputers to run even medium size network simulation and do not support both the user interactions during the experiments and animation of network topology changes. Users set configuration parameters before starting the simulation, and they can see computation results after the experiment is terminated. Moreover, the source coding is usually specialized for a given simulator and it is not easy to learn how to use the tool. In addition, most existing ad hoc networks simulators focus on the wireless transmission modeling and MAC protocols implementation with the lack of the mobility modeling. Usually only simplified mobility models are provided.

This work was partially supported by the Warsaw University of Technology Research Program grant 2008.
distributed simulation technology. The current version of our simulator provides different models of radio management, mobility models handling obstacles, user-friendly interface and tools for dynamic visualization. The open design of its architecture, and its extensibility to include other open source modules or modules developed by the user, which are specific to a given application, was chosen in the hope that the system will be a useful platform for research and education in ad hoc networks modeling and applications. The federated approach to parallel and distributed simulation of networks and provided functionality make different our tool from the mentioned popular software systems for simulation.

II. MOBASIM SYSTEM DESIGN

A. Overview

The MobASim system provides a framework for mobile ad hoc networks simulation performed on parallel computers or computer clusters. It is a general purpose simulator of MANETs designed as federations of disparate simulators of subnetworks that compose the considered MANET or a federation of simulators of independent, geographically dispersed MANETs or WSNs (wireless sensor networks) that cooperate from time to time. The elements of such simulators can be easily reused in many computations. Each simulator can be executed on a separate processor or machine. The example of federation of three simulators – cooperating two MANETs and one WSN developed in MobASim is presented in Fig. 1.

![MobASim federated simulator](Image)

Fig. 1. MobASim simulator: federation of three wireless networks

B. Composition and Implementation

The MobASim simulator is composed of a runtime platform, a set of libraries for mobile applications (functions implementing nodes’ mobility and wireless communication), a set of tools mainly to support the interaction with a user (GUI) and runtime monitoring, and a database called MobASim_map. MobASim_map stores all geographical information – a map of a deployment area, and all network nodes’ positions. To create a map of the environment the MobASim user can define simple objects in the domain as polygons. For more detailed description of a terrain to be considered the MobASim simulator provides the interface to the GeoTools toolkit. The GeoTools [12] is the open source Java coded library containing standard methods for the manipulation of geospatial data, for example to implement GIS (Geographic Information System).

MobASim is completely based on Java. At the heart of the MobASim technology is the ASimJava library. Although ASimJava was described in [13], [14], we provide a brief summary of this technology.

1) ASimJava System Design: The ASimJava (Asynchronous Simulation Java) system can be used to develop general purpose discrete-event parallel and distributed simulators. It consists of two main components: ASimJava library – collection of Java-based procedures and functions that can be used to develop simulators of discrete event physical systems and ASimJava modeler – for simulated system configuration, simulation execution, results visualization and animation. The paradigm of federating disparate simulators described in [15] is utilized in ASimJava. It results in a set of simulators creating federation. It is assumed that its entire simulators are viewed as black boxes. The runtime infrastructures (RTI) is used for federates interconnecting. RTI implements the relevant services such as: calculation synchronization, efficient communication and scalable platform architecture.

![ASimJava federation](Image)

Fig. 2. A federation of simulators developed in ASimJava

The simulator that is built upon ASimJava classes has hierarchical structure. The simulated system is partitioned into several components (subsystems), with respect to their functionality and data requirements. Each component is implemented as a logical process (LP) that can be divided into smaller LPs. Hence, the logical processes are nested. LPs communicate with each other through message-passing. Each ASimJava application consists of elements corresponding to three level of hierarchy (see Fig. 2): logical process – the basic active element simulating a given physical process, domain (a collection of LPs and other domains), simulator (a collection of domains). Calculation processes that belong to the same level of hierarchy are synchronized. The synchronous and asynchronous variants of simulation are provided. In the synchronous variant the global clock is applied to processes synchronization, while in asynchronous realization – conservative, optimistic and hybrid protocols are utilized [4], [11].

The configuration of the system to be simulated can be loaded and saved into the disc file in the XML format. It may contain any number of user-defined parameters. ASimJava provides the bidirectional interface to XML file that uses
mobility models have been adopted in the simulation of MANETs. Experiments described in [5] demonstrate that federated, distributed simulation based on ASimJava can seriously speed up simulation, and allows to compute large scale models.

C. Composition of MobASim Application

The discrete event systems methodology (DEVS) is applied to model ad hoc network operation, i.e., the process being modeled is understood to advance through events. In our MobASim application the considered (DEVS) system is a wireless, mobile ad hoc network that is composed of several components responsible for different functionalities. We distinguish three types of such components:

- **Node**: a mobile device that performs the assigned task. It can change dynamically its position in the simulated area and can interact with other nodes in the system.
- **Communication Manager**: an object that models the wireless communication between all nodes.
- **Mobility Manager**: an object responsible for tracking the nodes on the map and collision avoidance.

All described components are implemented as logical processes LPs. They are divided into three groups of computing processes, adequately responsible for:

- **N** – simulation of tasks to be performed by mobile nodes,
- **CM** – simulation of internode wireless communication and updating of the network communication topology,
- **MM** – simulation of mobile nodes movement and providing the access to the information about environment and other nodes’ positions in the network.

The processes from the group CM implement the mobility models provided in MobASim. In current version of the software two models are implemented. It is possible to combine various mobility models in one simulator, i.e., the model of mobility can switch w.r.t. the current state of the node. The processes from the group CM implement two wireless communication models. The user can select one of three types of MAC protocols.

The structure of a given application (MANET simulator) implemented under MobASim is presented in Fig. 3. We can see that every simulator of MANET is composed of one logical process from the group CM, one process from the group MM, and several processes from the group N.

In summary, MobASim logical processes utilize the classes from ASimJava library for simulation of discrete events execution in parallel environment, and classes provided in MobASim libraries designed for wireless, mobile applications.

1) **Mobility Models Library**: In general, two types of mobility models have been adopted in the simulation of MANETs [2], [3]. Motion traces that provide accurate information about mobility patterns (positions of nodes in time) and behavior of the nodes in the considered environment, and synthetic models – analytical random-motion models that describe mobility without using real traces. In literature we can find several less and more realistic synthetic models, such as: a random mobility model – a discrete implementation of Brownian-like motion, a random waypoint model with randomly generated destination point and velocity, a random direction model with randomly generated direction of movement. The special group is formed by the map-based mobility models that are used for applications in which nodes are constrained to move within defined paths. Most of presented models describe an obstacle-free movement. The popular commercial and publicly released software tools for networks simulation support mobility models, see OPNET [10], ns-2 [9], GloMoSim [8]. Most of them are limited to motion traces and simple obstacle-free synthetic models.

MobASim simulator provides three types of mobility models. In all cases the obstacles are included. The obstacles are generated by the user or are loaded from a real map. It is assumed that wireless signal is obstructed by the obstacles, too. All models utilize discrete event systems methodology (DEVS). The state of each mobile node is described by four variables:

- location within the deployment region,
- orientation (an angle between X axis and the direction of a node movement),
- velocity,
- energy stored in the node.

It is assumed that generated paths are dynamically changed taking into account the state of the nodes and surroundings (obstacles and other nodes in the network). All data concerned with the environment and all nodes in the network are stored in the data base (MobASim_map) served by the logical process MM (mobility manager). The following types of events are defined:

- MStart – start the movement.
- MC – continue the movement.
- MStop – stop the movement.
where $d$ denotes the distance between two considered nodes, $P_t$ power used by a sender to transmit the signal and $P_r$ power of the signal received by a receiver. $PL(d)$ is called "path loss" with distance $d$.

Modeling path loss is difficult but very important task. If we know the model of $PL(d)$ we can predict the occurrence of a radio channel between any two nodes in the network. Over time, many less and more detailed propagation models have been introduced [2], [3]. In practice, three techniques for path loss estimation are extensively used: long-distance path loss model, log-normal shadowing, and fading model. The long-distance models predict variations of the signal intensity over large distances. They have been developed as a combination of analytical and empirical methods. The log-normal shadowing model considers the fact that the transmission area of a transmitter may be different at two different locations, which leads to measure signals that are different than the average value calculated by the long-distance path loss model. In this model path loss at distance $d$ is modeled as a random variable with log-normal distribution. The fading models predict variations of the signal intensity over very short distance.

Most of the available software platforms for mobile ad hoc network simulation implements only long-distance model in its simplest version. Our simulator implements two models: long-distance and log-normal shadowing. The currently available version of MobASim implements the simplified models of the physical layer and the interference management. It provides implementation of MAC protocols from three categories based on the method that they handle the hidden and the exposed terminal problems: class 1 – protocol assuming random access to the wireless channel (the hidden and exposed node problem is unsolved), class 2 – the protocol solves the hidden node problem but leaves the exposed node problem unsolved, class 3 – the protocol solves both the hidden node and the exposed node problems, but requires the deployment of an additional signaling channel. The MobASim user can choose the protocol suitable to designed application. We assume that the accurate model of MAC layer can be adopted from the other open source simulators, if necessary.

**D. MobASim GUI**

The MobASim user interface is graphical and written in Java language. It is organized in a set of nested windows. We can distinguish setting and presentation windows (see Fig. 4). **Setting windows** are used to introduce the configuration parameters about an experimental environment, simulated ad hoc network, individual nodes and experiment setups. We can define:

- Simulated area: the flatland in which ad hoc network is deployed, the user can generate a simple map or the geographical data are loaded from GeoTools data base;
- Ad hoc network: initial number of nodes (wireless hosts), type of wireless transmission model, MAC protocol;
- Nodes: radio communication range, energy reserve, mobility model, minimal and maximal velocity, destination node;
- Experiment setup: simulation time horizon, number of processes, number of machines, IP of the machines.
After completing the initial settings, MobASim starts the simulation experiment. The results of simulation – time varying topology and adequate statistics are presented in the display windows. Display window is used to present the animation of network topology, current traffic, critical paths, collision alert, and information about network nodes status.

![MobASim graphical user interface](image)

It is possible to change some configuration parameters during the experiment. A user can change the mobility model, node’s velocity, radio communication range and energy resources.

### III. Simulation under MobASim

MobASim software was used to perform simulation of several ad hoc network topologies. In this paper we present the application of our tool to support the design of mobile ad hoc network for the rescue actions – monitoring of the situation after disasters like: earthquake, fire, flood or explosion.

Let us consider the following situation. The explosion in the chemical factory devastated the factory surrounding, among other things, most of the communication infrastructure, i.e., wired phone lines, base stations for cable networks, etc. We plan to send several rescue teams to work on the disaster scene. First, it is necessary to recognize the hazards for people in the contaminated area. Secondly, the efforts of the rescue teams should be coordinated, and a coordination action can be achieved only if rescuers are able to communicate, both within their team, the members of the other teams and the central station. Thus, one of priorities in the disaster management is to reinstall the communication infrastructure as quickly as possible, which can be done by deploying temporary communication equipment, e.g., vehicles or robots equipped with radio transceivers. So the mobile ad hoc network can be built. Computer simulation can be used to support the decision about the number of nodes necessary to create the wireless network, number and destination points of nodes equipped with sensors for monitoring the environment, and finally number of rescue teams. The simulation experiment was performed under MobASim. The network of 10 mobile nodes equipped with transceivers and 4 mobile nodes equipped with transceivers and a set of sensors was used for reestablishing the communication infrastructure and monitoring. All configuration parameters were introduced via setting windows (see Fig. 4) and saved to the database. The simulation experiment was executed. The animation of time varying network topology – all nodes moving from the source to their destination, avoiding the obstacles – was presented in the MobASim display window. The snapshots of initial, temporary and final network topologies calculated for 0, 10, 15, 20, 40 and 50 steps of simulated time are depicted in Fig. 5. From the simulation results we see that by using multihop wireless communication and mobile nodes, the communication between the base station and rescuers will be possible without the need for reestablishing the fixed communication infrastructure.

Table 1 collects the number of active (moving) nodes, number of wireless connections and number of nodes in interfering range in case of different MAC protocol categories. The results are presented for several time steps (every 5 units of simulated time). It can be seen that, as expected, the number of interfering nodes is highest in MAC class 1 and lowest in MAC class 2.

<table>
<thead>
<tr>
<th>Time step</th>
<th>Number of nodes</th>
<th>Wireless connections</th>
<th>Interferencing nodes (MAC classes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>class 1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>13</td>
<td>23</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>15</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>27</td>
<td>7</td>
</tr>
<tr>
<td>35</td>
<td>15</td>
<td>26</td>
<td>6</td>
</tr>
<tr>
<td>40</td>
<td>15</td>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>45</td>
<td>15</td>
<td>19</td>
<td>5</td>
</tr>
<tr>
<td>50</td>
<td>15</td>
<td>19</td>
<td>6</td>
</tr>
</tbody>
</table>

### IV. Summary and Conclusions

In this paper, we described the MobASim software platform for mobile ad hoc networks modeling and simulation. MobASim can support researches and engineers during the design and implementation of MANETs applications and verification of new MANET’s technologies. MobASim user interface provides tools for the tested ad hoc network configuration, and presents graphs and statistics of simulation results. MobASim is especially useful in large scale applications with a lot of hosts in which the speed of simulation is of essence, such as real time mobility ad hoc network simulation. The software will be free available for researchers and students.

### References

Fig. 5. Example: application of ad hoc network. Initial, temporary and final network topologies calculated by MobASim simulator.