DOI: 10.15439/2015F52
ACSIS, Vol. 5

Simulation of Mobile Wireless Ad Hoc Networks for Emergency Situation Awareness

Andrzej Sikora[1] and Ewa Niewiadomska-Szynkiewicz[1,2] and Mateusz Krzysztoń[2]
Research and Academic Computer Network (NASK)[1]
Wawozowa 18, 02-796 Warsaw, Poland
Institute of Control and Computation Engineering[2]
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
Email: Andrzej.Sikora@nask.pl, ens@ia.pw.edu.pl

***Abstract*—Mobile self-organizing ad hoc networks (MANETs) can significantly enhance the capability to coordinate the emergency actions as well as monitor contaminated areas, explore unmanned space, inspect and control working environments. The management of networks that can dynamically and freely organize into temporary topologies raises interesting problems, that are particularly challenging for networks formed by mobile devices. Due to the inherent complexity of these systems, the development of applications relying on mobile network nodes and wireless communication protocols would be greatly simplified by the use of specific tools for supporting testing and performance evaluation. Modeling and simulation are widely used in the design and development of wireless ad hoc networks. In this paper we model mobile ad hoc network (MANET) using discrete event systems methodology (DEVS) and describe the functionality and performance of the Java-based simulation tool for the performance evaluation of self-organizing and cooperative networks for emergency situation awareness. The simulator can provide a useful support for the verification of the design of a network system, employed communication protocols, control and coordination algorithms, allowing the user to display a step-by-step evolution of the network in a suitable graphical interface. The practical case studies are provided to illustrate the operation and performance of the presented software.**

I. Introduction

THE AD hoc networking is a relatively new area of research that has become extremely popular over the last decade and is rapidly increasing its advance into different areas of technology. A mobile, wireless, and ad hoc network (MANET) is formed of wireless mobile devices (network nodes) that can dynamically and freely self-organize into temporary network topologies. The topology of MANET may change rapidly and unpredictably. Moreover, in many application scenarios there is no prearrangement assumption about specific role a given node should perform. Each device makes its decision independently, based on the situation in the domain and its knowledge about the network. Nodes communicate wirelessly and share the same radio channel. The devices located within their transmission range can communicate directly without the need for an established infrastructure and centralized administration. For communicating with devices located beyond the transmission range, the node needs to use intermediate nodes to relay messages hop by hop. Thus, in general, routes between mobile nodes may include multiple hops.

Currently research effort is directed toward the specifics and constraints in ad hoc networks, such as: limited transmission range, limited link bandwidth and quality of transmission, constrained resources, mobility nature of the network and transmission security [1], [2], [6].

To design a self-organizing MANET that can support the monitoring in emergency situations and/or support coordination of emergency actions the following problems have to be solved: (i) how to determine a minimum number of devices to monitor an area, explore an unmanned space or control working environment; (ii) how to determine the optimal positions for all devices and how to manage internode communication to imply connectivity among the working set of devices and a base station, and how to coordinate and control all devices; (iii) how to schedule devices to reach the destination positions, etc.

Design, development and evaluation of MANET is a nontrivial task, especially as it is envisioned to be deployed in a large scale. It is obvious that the complexity and scale of modern ad hoc networks limit the applicability of purely analytic analysis. Therefore, investigation of MANETs is achievable by resorting either to software simulators or to testbed networks. In most applications the lack of flexibility of testbeds and high costs of their development make simulation unavoidable.

In this paper, we model a MANET system using discrete event systems methodology. We describe a flexible interactive simulation environment for the development of self-organizing, cooperative wireless networks formed of static and mobile devices that can be used to monitor contaminated area, inspect a harsh environment, create a communication infrastructure to collect measurements and transmit them to the base station, and support coordination of emergency actions.

II. Wireless Networks Simulators

Due to the complex nature of MANET, its simulation is a challenging task. It needs models of hardware, wireless propagation, mobility, energy usage, decision making, etc. Hence, simulators are inherently complex and they generally require huge computational resources to execute. In general,

978-83-60810-66-8/$25.00©2015, IEEE

they rely on various techniques for improving their accuracy, usability, scalability, speed, etc.

A variety of software environments simulating wireless networks are available today. Simulators rely on various methods and technologies for improving their accuracy, usability, efficiency and scalability. An overview of state of the art ad hoc network modeling and simulation tools available commercially and from open source is presented in [16], [40] and [14]. Researchers and engineers can choose among publicly available products or alternatively, can develop their own simulator. The commonly used network simulators like Riverbed Modeler (OPNET) [36], ns2 [20], ns3 [21], OMNeT++ [30] or GloMoSim [11], and its commercial version QualNet [31] can simulate ad hoc networks. Moreover, several software tools for mobile robots simulation, like v-rep [39] can be used. These tools provide the facility to simulate the protocols in different layers (MAC protocols, routing protocols, etc.), and some of them simulate movement of nodes (wireless devices). There are a number of possible sets of criteria that could be used for network simulators comparisons, e.g. time of execution, memory requirements, scalability, available functionality, programming interface, etc. Different tools are optimized for different purposes.

Most of available ad hoc networks simulators require costly shared-memory supercomputers to run even medium size network simulation. We are involved to large scale mobile network systems simulation and their practical applications, and our goal was to develop scalable simulator operating in real time. Hence, to provide high performance and scalability we utilized the paradigm of federating disparate simulators [7] and asynchronous distributed simulation technology [22], [40]. This is the main difference between our software and the other tools. Moreover, most existing ad hoc networks simulators focus on the MAC protocols implementation with the lack of the radio management and mobility modeling. Usually only simplified wireless transmission models and obstacles free simple mobility models are provided (ns-2, Castalia project in OMNeT++, v-rep). The other reason for developing a new simulator was the complicated architecture of available tools and limitations in results visualization and user-system interaction. In case of OPNET, OMNeT++ or ns-2 and ns-3 systems a user must read a large number of manuals to learn how to use the tool. The source coding is usually specialized for a given simulator and it is not easy to implement a given example and add modules developed by the user. Many network simulators do not support both the user interactions during the experiments and animation.

The current version of our simulator called MobASim provides implementations of radio propagation models, mobility models handling obstacles and tools for an environment (simulation scene) modeling. All these models are described in the next section. The open design of the architecture of MobASim, 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. The federated approach to simulation of networks and provided functionality make different our tool from mentioned popular software systems for simulation.

III. Ad Hoc Wireless Network Modeling

A. Network Modelling

The aim is to model an ad hoc network formed by N static and mobile, self-organizing and cooperating wireless devices D (network nodes) equipped with sensors. All nodes can move with the speed $v \in [v_{min}, v_{max}]$ in the workspace W avoiding the existing obstacles and communicate through radio.

In our research we use the discrete event systems methodology (DEVS) to model a mobile network, i.e., the network operation being modeled is understood to advance through events. The considered DEVS system is composed of several components responsible for different functionalities. We distinguish three types of such components: *node* – a mobile or static device that executes the assigned task, *mobility manager* that is responsible for tracking the nodes on the map and collision avoidance and *communication manager* that models the wireless communication between all nodes. Hence, each MANET simulator consists of three groups of logical processes LPs, responsibly Ns, MMs and CMs. The wireless communication and mobility models that are implemented in our simulation software are described below.

B. Mobile Network Node Modelling

A high quality mobility modeling is a critical aspect that has great influence on the performance characteristics of each network node and the whole ad hoc system simulation. In literature we can find several less and more realistic models, such as a random mobility model with Brownian-like motion, a random waypoint model (RWP) with randomly generated target point and velocity, a random direction model with randomly generated direction of movement. Map-based mobility models are used for applications in which nodes are constrained to move within defined paths. The surveys of mobility models and main directions to mobility modeling of moving wireless devices one can find in [2], [8], [19], [37].

Widely used approach to mobility modeling, is to apply a concept of an artificial potential field that can be viewed as a landscape where the mobile device moves from a high-value state to a low-value state, [5], [37]. The artificial potential function V is a differentiable real valued function, which value can be viewed as energy. The gradient of V results in a force F, which points in the direction that locally maximally increases V. The potential function can be constructed as a sum of repulsive V_- and attractive V_+ potentials. The obstacle repels the mobile device while the goal attracts it. The sum of V_- and V_+ draws a device to the target while deflecting it from obstacles. The concept of a potential function is used in particle-based mobility schemes [37], where network node considered as a self-driven moving particle is characterized by a sum of forces, describing its desire to move to the target and avoiding collisions with other nodes and obstacles.

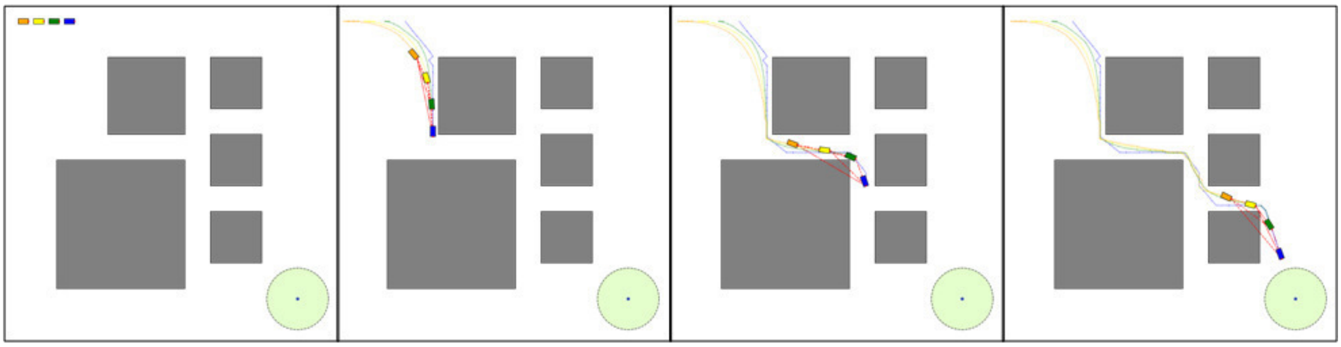


Fig. 1: Convoy formed by the team of mobile devices.

In our research we investigate a group mobility model for calculating of collision-free motion trajectories for N wireless devices that form a cooperative network. Our computing scheme adopts two techniques, the concept of an artificial potential field and the concept of a particle-based mobility. This model can be used to cooperative and fully connected networks design. The detailed description of our model is provided in [24], therefore in this paper we present the summary.

We model each mobile device D_i by a polygon in the workspace. We define its reference point $\mathbf{c}^i = [x^i, y^i]$, which is the location of its antena and a target location – the destination point $\mathbf{c}_g^i = [x_g^i, y_g^i]$. It is obvious that to maintain a constant communication with the base station at least one another device has to be within the transmission range of each D_i. Therefore, the motion trajectory of each D_i depends on the potential U_g^i between D_i and its target and the potentials U_k^i between D_i and other devices D_k, $k = 1, \ldots, N$, $k \neq i$. Hence, a simple artificial potential function U^i for ith device can been calculated

$$U^i(c^i) = U_g^i + \sum_{k=1, k\neq i}^{N} U_k^i(c^i), \qquad (1)$$

where

$$U_g^i(c^i) = \epsilon_g^i \Big(\frac{\overline{d}_g^i}{d_g^i} - 1\Big)^2, \qquad d_g^i = ||c^i - c_g^i||, \qquad (2)$$

$$U_k^i(c^i) = \epsilon_k^i \left(\frac{\overline{d}_k^i}{d_k^i} - 1\right)^2, \qquad d_k^i = ||c^i - c_k^i||. \qquad (3)$$

$\epsilon_g^i \geq 0$ and $\epsilon_k^i \geq 0$ in (2) and (3) denote weighting factors determining the importance of the target point g and the device D_k respectively. d_g^i and d_k^i are real Euclidean distances between $\mathbf{c}^i$ and $\mathbf{c}_g^i$ and $\mathbf{c}_k^i$ after a network transformation, $\overline{d}_g^i$ and $\overline{d}_k^i$ the reference distances calculated due to the current signal strength measurements.

Taking into account the potential function defined in (1) we can formulate the optimization problem (4) and calculate the expected position of the reference point $\mathbf{c}^i$ of D_i after moving the device:

$$\min_{\mathbf{c}^i} U^i(c^i). \qquad (4)$$

The Tangent Bug algorithm described in [5] is applied to avoid obstacles in the workspace while moving the devices.

In summary, the following algorithm is used to motion trace calculation of ith device at time instants $t_0, t_1, \ldots, t_i, \ldots$:

1) Calculate the reference distance $\overline{d}_g^i$ to the target point g and the reference distances $\overline{d}_k^i$ to all neighbour nodes due to the current maximal radio range and environment characteristics.
2) Calculate the displacement for the device D_i for results of step 1.
3) Move D_i to the new position in W.
4) Rotate D_i (if necessary).
5) Calculate and broadcast to the network the new position of the device D_i.
6) Return to step 1.

We assume that each network device can calculate its localization in the workspace based on the measurements from GPS or other localization systems ([15], [17], [29], [38]. The algorithm terminates when the minimal the potential function (1) reaches the minimum value, and all the devices don't change their positions. The example results of application of our model to calculate the motion patterns for a group of devices forming a convoy are presented in Fig. 1

C. Internode Communication and Reference Distance Calculation

A signal strength measurement using the RSSI (*Received Signal Strength Indicator*) are often used to estimate the reference distances d_g^i and d_k^i in real networks. For purposes of the simulation we applied the long-distance path loss model to radio transmission modelling, [35]. It indicates that received signal power decreases with a distance, both in outdoor and indoor environments. We assumed that the radio coverage region of the transceiver of D_ith device is a disc centered at $\mathbf{c}^i$. Hence, the signal degradation $PL(d)$ ("path loss") with distance d can be defined as follows $PL(d)[dB] = P_t(d)[dBm] - P_r[dBm]$, where P_t denotes power used by a sender to transmit the signal and P_r power of the signal received by a receiver. In the long-distance path loss model $PL(d)$ is modeled as a random variable with log-normal distribution ([2], [9], [32]).

$$PL(d)[dB] = PL(d_0)[dB] + 10nlog\left(\frac{d}{d_0}\right) + X_\sigma, \quad (5)$$

where d_0 denotes a reference distance (d_0=1 m for IEEE 802.15.4) and X_σ a zero-mean Gaussian distributed random variable with standard deviation σ (all in dB).

To enable the communication the signal strength received by neighboring nodes should exceed a receiver sensitivity P_s. The Q-function may be used to determine the probability that the received signal level will exceed P_s. It is defined as follows

$$Q(z) = \frac{1}{\sqrt{2\pi}}\int_z^x \exp\left(-\frac{x^2}{2}\right)dx, \quad Q(z) = 1 - Q(-z). \quad (6)$$

The probability that the received signal level P_r in distance d will exceed a value P_s can be calculated from the cumulative density function as

$$P[P_r(d) > P_s] = Q\left(\frac{P_s - \overline{P_r(d)}}{\sigma}\right). \quad (7)$$

A tabulation of the Q-function for various values of z is given by Rapaport in [35]. The received signal exceeds the receiver sensitivity with probability of 99% for

$$\frac{P_s - \overline{P_r(d)}}{\sigma} = -2.3, \quad \overline{P_r(d)} = P_t - PL(d). \quad (8)$$

From (8) we can calculate $PL(d)$

$$PL(d) = -2.3\sigma - P_s + P_t. \quad (9)$$

Using (9) and definition of PL, (5) we can estimate the reference distance $\hat{d_k^i}$ between i-th and k-th mobile nodes

$$\hat{d_k^i} = d_0 10^{\frac{-2.3\sigma - P_s^k + P_t^i - PL(d_0)}{10n}}, \quad (10)$$

where P_t^i denotes a transmission power of a node i and P_s^k the sensitivity of a node k.

The value of X_σ and n in (10) depend on the workspace conditions, and can be calculated using linear regression such that the difference between the measured and estimated path losses PL is minimized over a wide range of measurement locations. The values of n estimated for various environments are provided in [35].

D. Environment Modeling

To create a workspace of a network to be simulated a 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 [10] is the open source Java coded library containing standard methods for the manipulation of geospatial data, for example to implement GIS (Geographic Information System).

Wireless devices are widely use to establish sensing systems for contaminated areas monitoring. To perform simulations with the MANET system for environmental monitoring it is necessary to model propagation of a pollution. Very dangerous for human beings are clouds created by heavy gases – gases with density greater than that of air. They can move close to the ground for significant time at high level of gas concentration [33].

The models of heavy gas dispersion are divided into categories based on different criteria. Three main groups: empirical, research and engineering models are distinguished [18]. Empirical models are developed based on environmental measurements and laboratory experiments. Research models – formulated by sets of partial differential equations dependent on time and three space coordinates – provide complete and detailed description of the physical process of the heavy gas dispersion. The trade off are engineering models that are widely used in practical applications.

The current version of the MobASim system provides the box model for instantaneous releases to simulate heavy gas dispersion. It belongs to the group of engineering models. It is a simple model that assumes that a gas cloud forms a uniform cylinder. The detailed description is in [23]. Hence, in this section we present the summary of the box model. In general, it is formed by a set of three linear ordinary differential equations:

$$\frac{dc_c}{dt} = v_c, \quad (11)$$

$$\frac{dr}{dt} = v_f, \quad (12)$$

$$\frac{dm_a}{dt} = \rho_{air}(\pi r^2)v_t + \rho_{air}(2\pi rh)v_e, \quad (13)$$

where $c_c = [x_c, y_c]$ denotes the position of the centre of a cloud, v_c, v_f, v_e, v_t denote following velocities: transport, gravitational and entrainment for edge and top of a cloud. r is the radius of a cloud, h its height and m_a is the entrained mass. ρ_{air} denotes the air density.

Equation (11) describes the spreading of the centre of a cloud, (12) the puff horizontal spreading influencing the cloud radius. Mass conservation is described by formula (13).

The gravitational velocity v_f can be calculated for a given standard gravity g, the height of a cloud h, a cloud ρ_c and an air ρ_{air} densities according to following formula:

$$v_f = C_F\sqrt{\frac{g(\rho_c - \rho_{air})h}{\rho_{air}}}, \quad (14)$$

where C_F denotes the Froude number of the front. In case of the dense gas models typically assumed $C_F = 1.1$ [23].

The relationship between the edge entrainment velocity v_e and the gravitational velocity is as follows:

$$v_e = \alpha v_f, \quad (15)$$

where $\alpha \in [0.6, 0.9]$ [18].

Finally, the entrainment velocity for a cloud can be calculated due to the following equation

$$v_t = u_* \left(\frac{\kappa}{1 + \beta \frac{g(\rho_c + \rho_{air})h}{\rho_{air} u_*^2}} \right), \tag{16}$$

where $\kappa = 0,4$ denotes the von Karman constant [18], u_* is the friction velocity [13], β denotes the parameter (suggested value $\beta = 0.125$ [3]).

To solve above equations the current cloud density ρ_c and its height h have to be determined. They depend on the concentration c of gas in a cloud. The value of c dynamically changes due to cloud mixing with an ambient air. It can be calculated due to the following formula

$$c = \frac{\frac{m_0}{M}}{\frac{m_0}{M} + \frac{M_a}{M_{air}}}, \tag{17}$$

where m_0 denotes a mass of contaminant gas, M and M_{air} the molar weights of the gas and air respectively.

The relation between the gas concentration and its density notable influences a cloud dynamics. This relation is affected by many factors. In our work we assume that mixing of gas with ambient air is the only source of density change. We neglect chemical reactions and occurrence of any aerosol formations in the cloud. Based on these assumptions we can calculate the density of gas

$$\rho_c = \rho_{air} \left(\frac{1 + c\frac{M - M_{air}}{M_{air}}}{1 + \frac{c\Delta H_0}{\left((1-c)M_{air}c_p^{air} + cMc_p\right)T_{air}}} \right), \tag{18}$$

where ΔH_0 denotes the enthalpy difference between the release material at the source and ambient conditions, c_p and c_p^{air} are specific heat capacities of gas and air respectively. T_{air} is the temperature of an ambient air.

Then, the height of a cloud h can be computed

$$h = \frac{V}{\pi r^2}, \qquad V = \frac{m}{\rho_c}, \tag{19}$$

where V is the volume of a cloud.

The Euler method is used to solve the model (11)–(13). It should be pointed that the presented model can be employed to simulate a gas dispersion in a flat area without obstacles. In general, the model is valid until the occurrence of one of two conditions [18]:

- the difference between the density of cloud and air is less than a small assumed value,
- the growth of a cloud radius in single step is small enough.

IV. Mobile Ad hoc Networks Simulation Platform

MobASim (*Mobile Ad hoc network Simulation*) [25] is a general purpose software framework for wireless, mobile ad hoc networks simulation and a library of Java classes that can be used to build a MANET simulator. Each simulator is designed as federation 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 components of such simulators can be easily reused in many computations. MobASim can run on a single or parallel machine or a computer cluster.

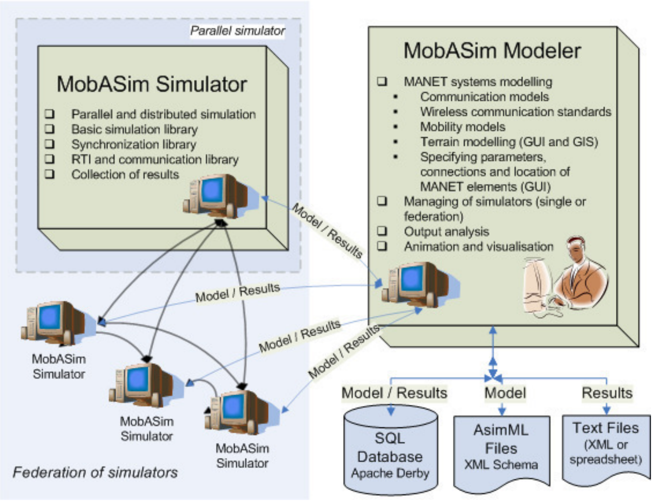

Parallel simulator
MobASim Simulator
❑ Parallel and distributed simulation
❑ Basic simulation library
❑ Synchronization library
❑ RTI and communication library
❑ Collection of results
MobASim Modeler
❑ MANET systems modelling
▪ Communication models
▪ Wireless communication standards
▪ Mobility models
▪ Terrain modelling (GUI and GIS)
▪ Specifying parameters, connections and location of MANET elements (GUI)
❑ Managing of simulators (single or federation)
❑ Output analysis
❑ Animation and visualisation
Model / Results
Model / Results
Model / Results
Model / Results
MobASim Simulator
MobASim Simulator
MobASim Simulator
Federation of simulators
Model / Results
Model
Results
SQL Database Apache Derby
AsimML Files XML Schema
Text Files (XML or spreadsheet)

Fig. 2: The MobASim application.

The MobASim system is completely based on Java. At the heart of the MobASim technology is the ASimJava library developed by the authors, and described in [26], [28]. MobASim is composed of four main components presented below.

A. MobASim Modeler

The model of the network considered can be generated using MobASim GUI or can be loaded from the disc file in the XML format. The configuration of the system is saved into the disc file for reuse in other experiments. The following attributes and parameters have to be provided by the user: number of wireless devices that compose a network, wireless transmission model, MAC protocol, radio communication range, type of the mobility model, minimal and maximal velocity, destination, simulation time horizon.

Three libraries are provided to implement a simulator: wireless transmission modeling library (WTML), Mobility Models Library (MML) and Environment Modeling Library (EML). WMTL contains a collection of classes implementing models of wireless transmission and wireless communication standards. The currently available version of MobASim provides the implementation of the long-distance path loss model described in section III-C and three classes of MAC protocols 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.

MML contains classes implementing three types of mobility models: random mobility model, waypoint model and group mobility model described in section III-B. All models utilize discrete event systems methodology. The state of each mobile node is described by three variables: location within the deployment region, orientation (an angle between X axis and the direction of a node movement) and velocity.

EML contains a collection of classes that allow to model a simulation scene. They provide interfaces to tools used to create a workspace. This library is under development. We plan to implement models for spread of contaminations and various types of gases dispersion. Nowadays, EML provides the heavy gas dispersion model described in section III-D.

a) MobASim Simulator: responsible for the simulator implementation and performance. It provides three libraries: basic library – a collection of classes implementing basic elements of a simulator (logical processes, events, etc.), synchronization protocols library – a collection of classes implementing synchronization algorithms and runtime infrastructure (RTI) for parallel and distributed simulation. The functions from these libraries are basic components of a given simulator.

b) MobASim Database: stores all geographical information – a map of a workspace, and all network nodes' positions.

c) MobASim User Interface: – an integrated GUI that can help a user to define a simulation scenario, create a network topology and provide all attributes of all devices that form a network to be considered. The GUI is organized in a set of nested setting and presentation windows. The setting windows are used to facilitate the configuration phase. The presentation windows are used to display the calculation results – network animation, statistics, etc.

V. Case Studies

In this paper we demonstrate some of the main capabilities of the MobASim platform by studying the performance of various ad hoc networks application scenarios concerned with an environmental monitoring and coordination of an emergency action. The results of experiments performed on single machine (AMD Sempron 1.67 GHz, 512 MB RAM) were compared with those obtained on parallel machine (Intel Core2 Duo 2.2 GHz, 2038 MB RAM), and computer cluster consisting of three machines (AMD Sempron, Intel Core2 Duo and AMD Athlon-M 1.2 GHz, 512 MB RAM). In parallel and distributed implementations the networks considered were divided into subnetworks simulated in parallel by three processors.

A. Rescue Action Coordination

The MobASim software can be used to support the design of mobile ad hoc network for the rescue actions – monitoring of the situation after various disasters: earthquake, fire, flood or explosion, etc. In this paper we present the results of hypothetical usage of MANET to support the rescue action in case of real life explosions, which occurred at Buncefield Oil Storage Depot, Hemel Hempstead, Hertfordshire in December 2005. The damages were significant, a large area around the site was evacuated, forty people were injured. The fire burned for several days, destroying most of the site and emitting large clouds of smoke into the atmosphere.

It is obvious that usually the most of the communication infrastructure, i.e., wired phone lines, base stations for cable networks, etc. are devastated after explosions. Hence, mobile, self-configuring and cooperative wireless network can be used to support the rescue action. Let us consider that we plan to send several rescue teams to work on the disaster scene. A coordination action can be achieved only if rescuers are able to communicate, both within their team, the members of the other teams and an emergency action coordination center. Thus, one of the priorities in the disaster management is to reinstall the communication infrastructure as quickly as possible and monitor the situation. It can be done by deploying temporary communication equipment, e.q. vehicles or robots equipped with radio transceivers and sensors, and form an ad hoc network. Computer simulation can be used to support the decision about the number of devices used to create the given network, their destination positions, and finally number of rescue teams [27].

A series of experiments for various network topologies used for reestablishing the communication infrastructure and monitoring were performed. Each network system was built by three types of nodes:

A – mobile devices equipped with transceivers and a set of sensors,
B – static devices equipped with transceivers and a set of sensors; a few of them were mobile (backup devices),
C – base station (emergency action coordination center).

In this paper the results of tests performed for two network configurations are presented:

- Net 1: 23 nodes A, 12 nodes B, 1 node C (36 LPs).
- Net 2: 81 nodes A, 46 nodes B, 1 node C (81 LPs).

In both cases the maximal velocity of all mobile nodes was equal to 10 m/s. Three variants of implementation – sequential, parallel and distributed on three hardware platforms depicted in Table I were compared. The goal of all experiments was to simulate 900 seconds of physical ad hoc network operation. The execution times of each experiment are given in Table II.

TABLE I: Computer systems: sequential, parallel and distributed implementations.

Variant	AMD Athlon-M 1,2 GHz 512 MB RAM	AMD Sempron 1,67 GHz 512 MB RAM	Intel Core2 Duo 2,2 GHz 2038 MB RAM
Sequential		X	
Parallel			X
Distributed	X	X	X

The last two columns collect the speedup for parallel and distributed calculations, defined as $s(p) = T(1)/T(p)$, where

TABLE II: Performance evaluation – comparison of execution times.

Ex.	Simulation time [s]			s(2)	s(4)
	S	P	D	P	D
Net 1	64.2	49.2	49.9	1.31	1.29
Net 2	182.2	152.8	118.9	1.19	1.53

$T(1)$ denotes execution time in case when all calculations were performed on a single processor and $T(p)$ execution time in case of p processors or machines.

During the experiments the animation of time varying network topologies – 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 are depicted in Figures 3 and 4.

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. We can observe that federated, parallel and distributed simulation developed based upon MobASim software can speedup simulation of MANETs operation w.r.t. sequential implementation. As expected, the calculation speedup depends on the size of network model and assumed degree of parallelism – the speedup factor increases with the problem dimension and complexity. It can be observed that the sequential part of execution concerned with MobASim internal servers and database initializations depends on the size of problem to be considered, and seriously influences the whole computation time. As a final observation, we can point that we obtained the speedup of calculations even in case of small dimension and strongly interconnected network models. We expect much better results for higher dimension networks operating in inherently parallel environments, e.g. cooperating MANETs or sensor networks.

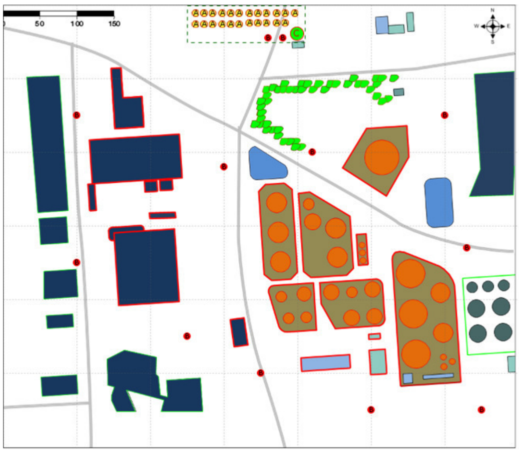


Fig. 3: The snapshot of initial network topology.

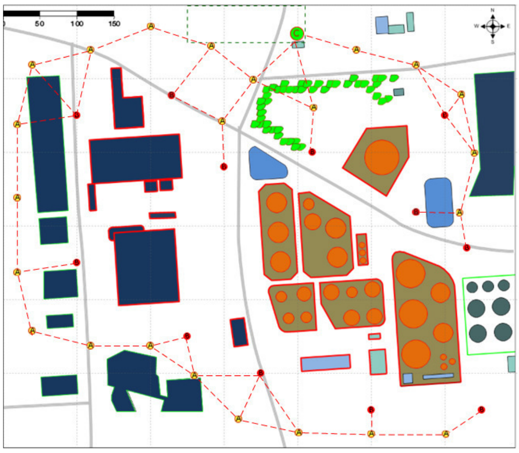


Fig. 4: The snapshot of final network topology.

B. Gas Cloud Detection and Exploring

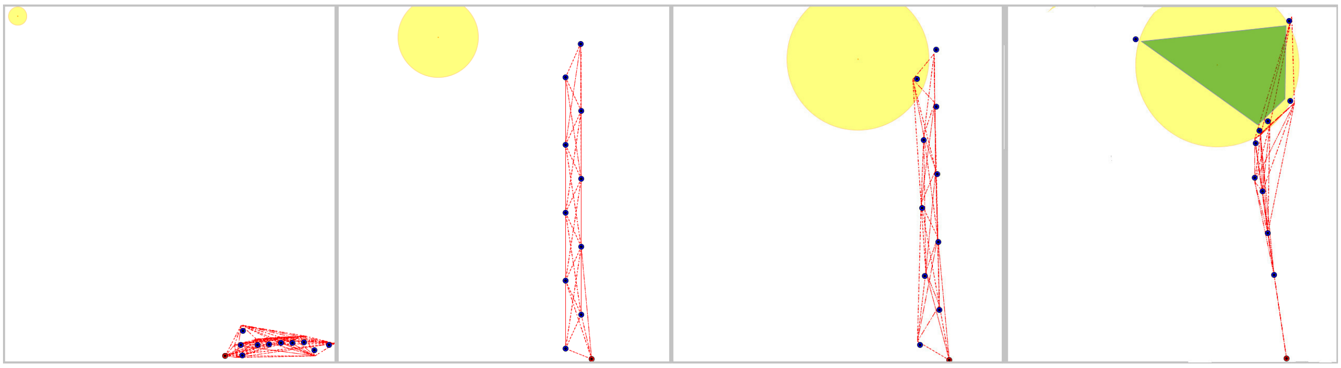
A wireless network formed by mobile devices can reduce lack of situation awareness in area prone to emergencies such as environmental pollution. The MobASim platform was used to simulate various scenarios of application of MANETs for environmental monitoring. The goal of the first series of experiments was to design and develop robot-assisted wireless sensor networks for outdoor and indoor monitoring. The simulation was used to determine the optimal number of sensing devices, calculate collision and obstacle-free motion trajectories for mobile robots carrying the sensor devices and calculate the optimal positions for all sensors (network nodes). Various deployment strategies were considered, i.e., regular and pre-defined grid, self-configuring and hybrid deployments. The results are described and discussed in [27]. The objective of the second series of experiments was to design and develop a MANET for detection, exploration and tracking of a heavy gas cloud. Such cloud may be created as an effect of an instantaneous release of gas heavier than air from a tank [4].

Let us assume that the emergency team has to find, surround and suppress a chlorine gas cloud. The network formed by mobile devices equipped with punctual gas sensors can successfully support the team in detecting and surrounding the cloud. Hence, the set of self-organized mobile devices were deployed in a suspicious area, and formed a coherent network to discover the cloud. After detecting the cloud the devices were divided into two groups. The devices from the first group were responsible for exploring and discovering the shape of the cloud. The task of the remaining devices was to maintain communication between the first group of sensors and the base station.

The results of simulations performed with the example ad hoc network are shown in Fig. 5. The values of all parameters occurring in the equations (18)-(13) are given in Table III [12], [34], [23].

VI. Summary and Conclusions

The evolution of wireless, mobile ad hoc networks and improved designs will strongly depend on the ability to predict their performance using simulation methods. In this paper we described the application of our software platform MobASim to design mobile ad hoc networks used to detect threats and support decision makers in emergency situations. The presented case studies confirm that the MobASim system can support researches and engineers during the design and implementation of MANETs applications and verification of new MANET's technologies. The tool is especially useful in



(a) (b) (c) (d)

Fig. 5: MANET for gas cloud detection and tracking: (a) initial network topology, (b) cloud detection, (c) cloud was detected, (d) exploring the cloud and discovering its shape (in green the detected cloud shape).

TABLE III: Values of parameters of the chlorine gas cloud simulation model.

Symbol	Value	Units
c_c	[200, 200]	$[m,m]$
v_c	[3, 1]	$[\frac{m}{s}, \frac{m}{s}]$
r	10	m
m_0	2000	kg
m_a	0	kg
g	9.81	$\frac{m}{s^2}$
ρ_{air}	1.20	$\frac{kg}{m^3}$
M	71	$\frac{g}{mol}$
M_{air}	29	$\frac{g}{mol}$
c_p	0.48	$\frac{kJ}{kg*K}$
c_p^{air}	1.01	$\frac{kJ}{kg*K}$
u_*	0.15	$\frac{m}{s}$
ΔH_0	661	$\frac{kJ}{kg}$
T_{air}	293.15	K
α	0.9	-

large scale applications in which the speed of simulation is of essence, such as real time ad hoc networks simulation.

References

[1] Aggelou G., *Mobile Ad Hoc Networks. From Wireless LANs to 4G Networks*, McGraw-Hill, USA 2005.

[2] Basagni S., Conti M., Giordano S. and Stojmenovic, I., *Mobile Ad Hoc Networking*, Wiley-Interscience, IEEE Press, 2004.

[3] Britter, R.E. and Simpson, J.E., *Experiments on the dynamics of a gravity current head*, Journal of Fluid Mechanics, Vol. 88, pp. 223-240, 1978. doi:10.1017/S0022112078002074

[4] Cameron, I. and Raman, R., *Process Systems Risk Management*, Elsevier Academic Press, pp. 246–247, 2005.

[5] Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., and Thrun, S., *Principles of Robot Motion*, The MIT Press, Cambridge, 2005.

[6] Daniluk, K., and Niewiadomska-Szynkiewicz, E., *Energy-efficient security in Implantable Medical Devices*, in Proc. FedCSIS, pp.773-778, 2012.

[7] Ferenci, S.L., Perumalla, K.S. and Fujimoto, *An Approach for Federating Parallel Simulators*, Proc. of PADS 2000, Bologna, pp. 63–70. doi:10.1109/PADS.2000.847145

[8] Fongen, A., Gjellerud, M. and Winjum, E., *A Military Mobility Model for MANET Research*, Proc. of IASTED International Conference on Parallel and Distributed Computing and Networks, 2009.

[9] Forouzan, B.A., *Data Communications and Networking*, McGraw-Hill, 2004.

[10] GeoTools The Open Source Java GIS Toolkit, http://geotools.codehaus.org/ .

[11] GloMoSim homepage, http://pcl.cs.ucla.edu/projects/glomosim/ .

[12] Hanna, S.R. and Drivas P.J., *Guidelines for use of vapour cloud dispersion models*, Center for Chemical Process Safety, Institute of Chemical Engineers, New York, 1989.

[13] HGSYSTEM, *The Heavy Gas Dispersion Model HEGADAS*, HGSYSTEM Technical Reference Manual, http://www.hgsystem.com/tech_ref/Chap07.pdf .

[14] Kasch, W.T., Ward, J.R. and Andrusenko, J., *Wireless Network Modeling and Simulation Tools for Designers and Developers*, Comm. Mag., Vol. 47, Issue 3, pp. 120–127, IEEE Press, March 2009. doi:10.1109/MCOM.2009.4804397

[15] Kasprzak, W., Szynkiewicz, W., Karolczak, M., *Global colour image features for discrete self-localization of an indoor vehicle*, Lecture Notes in Computer Science 3691, pp. 620627, Springer Verlag, Berlin Heidelberg, 2005.

[16] Kurkowski, S., Camp, T. and Colagrosso, M., *MANET Simulation Studies: The Incredibles*, SIGMOBILE Mob. Comput. Commun. Rev., Vol. 9, Issue 4, pp. 50–61, ACM, 2005. doi:10.1145/1096166.1096174

[17] Mao, G., Fidan, B. *Localization algorithms and strategies for wireless sensor networks*, Inforamtion Science Reference, USA, 2009. doi:10.1007/BF02136831

[18] Markiewicz, M.T.,*Mathematical modeling of heavy gas atmospheric dispersion over complex and obstructed terrain*, Archives of Environmental Protection, Vol. 36, no. 1, pp. 81–94, 2010.

[19] Musolesi, M. and Mascolo, C., *Mobility Models for Systems Evaluation. A Survey*, State of the Art on Middleware for Network Eccentric and Mobile Applications (MINEMA), Springer, 2009.

[20] Network Simulator ns-2 homepage, http://www.isi.edu/nsnam/ns/.

[21] Network Simulator ns-3 homepage, http://www.nsnam.org/.

[22] Nicol, D.M. and Fujimoto, R. (1994) *Parallel Simulation Today*, Annals of Operations Research, Vol. 53, pp. 249–285. doi:10.1007/BF02136831

[23] Nielsen, M., *Dense Gas Dispersion in the Atmosphere*, Riso National Laboratory, Roskilde, Denmark, 1998.

[24] Niewiadomska-Szynkiewicz, E. and Sikora A., and Kołodziej, J., *Modeling Mobility in Cooperative Ad Hoc Networks*, Mobile Networks and Applications, Vol. 18, No. 5, pp. 610-621, 2013. doi:10.1007/s11036-013-0450-2

[25] Niewiadomska-Szynkiewicz, E., Sikora, A., *A Software Tool for Federated Simulation of Wireless Sensor Networks and Mobile Ad Hoc Networks*, Applied Parallel Scientific Computing K. Jonasson (ed.), LNCS7133, part I, pp. 303-313, Springer-Verlag, 2012.

[26] Niewiadomska-Szynkiewicz, E. and Sikora A., *ASim/Java: A Java-based Library for Distributed Simulation*, Journal of Telecommunications and Information Technology, No. 3, pp. 12-17, 2004.

[27] Niewiadomska-Szynkiewicz, E. and Sikora, A., *Simulation-Based Evaluation of Robot-Assisted Wireless Sensors Positioning*, Progress in Automation, Robotics and Measuring Techniques, Vol. 351, pp. 181-190, Springer International Publishing, 2015. doi:10.1007/978-3-319-15847-1_18

[28] Niewiadomska-Szynkiewicz, E. and Sikora, A., *A Federated Approach to Parallel and Distributed Simulation of Complex Systems*, International Journal of Applied Mathematics and Computer Science, Vol.17, Issue 1, pp. 99106, 2007. doi:10.2478/v10006-007-0009-0

[29] Niewiadomska-Szynkiewicz, E. *Localization in Wireless Sensor Networks: Classification and Evaluation of Techniques*, International Journal of Applied Mathematics & Computer Science, University of Zielona Gora Press, vol. 22, No 2. pp. 281-297, 2012.

[30] OMNeT++ homepage http://www.omnetpp.org/.

[31] QualNet homepage http://web.scalable-networks.com/content/qualnet.

[32] Santi, P., *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, Ltd, 2006.

[33] Scargiali, F., Di Rienzo, E., Ciofalo, M., Grisafi, F., and Brucato, A., *Heavy Gas Dispersion Modelling Over a Topographically Complex Mesoscale: A CFD Based Approach*, Process Safety and Environmental Protection, Volume 83, Issue 3, Pages 242-256, ISSN 0957-5820, May 2005. doi:10.1205/psep.04073

[34] Schmittinger, P., Florkiewicz, T., Curlin, L.C., Lke, B., Scannell, R., Navin, T., Zelfel, E. and Bartsch, R., *Chlorine*, Ullmann's Encyclopedia of Industrial Chemistry, Wiley-VCH Verlag, 2006.

[35] Rappaport, T.S., *Wireless Communications. Principles and Practice*, Prentice Hall, USA, 2009.

[36] Riverbed Modeler (OPNET) homepage http://www.riverbed.com/products/performance-management-control/network-performance-management/network-simulation.html.

[37] Roy, R.R., *Handbook of Mobile Ad Hoc Networks for Mobility Models*, Springer, USA, 2010.

[38] Van Haute, T., Rossey, J., Becue, P., De Poorter, E., Moerman, I., and Demeester, P., *A hybrid indoor localization solution using a generic architectural framework for sparse distributed wireless sensor networks*, Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Vol. 2, pp. 1009–1015, 2014. doi:10.15439/2014F20

[39] V-rep (Virtual Robot Experimental Platform) homepage, http://www.coppeliarobotics.com/.

[40] Zeigler, B.P., Praehofer, H. and Kim, T.G., *An Overview of MANETs Simulation*, Jour. of Electronic Notes in Theoretical Computer Science (ENTCS), Vol. 150, Issue 1, pp. 81-101, 2006. doi:10.1016/j.entcs.2005.12.025