DARS: A Discrete Event Mobile Ad Hoc Routing Simulator

S. A. Kress, K. L. May, M. E. Moorman, G. Dimitoglou
Department of Computer Science
Hood College
Frederick, Maryland 21701, USA
{sak14, klm33, mem19, dimitoglou}@hood.edu

***Abstract*—The dynamic nature of Mobile Ad Hoc Networks (MANETs) makes them difficult to design, analyze and implement. Using simulations it becomes simpler, more cost-effective and easier to test, verify and reproduce results and develop better understanding of the performance, routing and particular characteristics of MANET environments. To address this need, the Dynamic Ad hoc Routing Simulator (DARS), an open source, platform independent discrete event simulation platform was developed. One goal for DARS was to balance the need for an intuitive, simple-to-deploy and easy to use versatile tool with a feature-rich graphical user interface. Another goal was to provide the necessary extensible infrastructure to perform reliable, scalable and realistic routing simulation experiments. After achieving both of the these goals, we present the potential of DARS as a useful tool for both research and educational purposes.**

Keywords-discrete event simulation; networking; wireless networks; routing protocols; mobile ad hoc networks

I. Introduction

Mobile ad hoc networks (MANETs) are wireless collections of network nodes. They are amorphous in structure since the nodes are mobile, therefore continuously moving and changing the network topology. The proliferation of highly capable mobile devices and the ubiquity of wireless network infrastructures has made MANETs an attractive solution with widespread applications and scope. Such networks can be deployed in a range of environments, ranging from large-scale, highly dynamic networks to small-scale, power-constrained networks. Examples range from uses in the military battlefield and emergency response to gaming [15], law enforcement and commercial applications for the exchange of information between taxicabs in a vehicle ad hoc network (VANET) [7, 12].

Designing applications for MANETs presents many different opportunities and challenges. Routing is a particularly interesting area since deploying MANETs requires the use of special routing protocols that enable participating nodes to route and exchange packets. Unlike other networks, ad hoc environments do not start with, or maintain, a known topology therefore the routing protocols have to discover the environment to operate. The possible large scale of such networks, with thousands or millions of nodes, makes it necessary to have the means to simulate an environment and study the performance and behavior of routing protocols.

This paper describes the Dynamic Ad Hoc Routing Simulator (DARS) which was developed to provide a platform-independent, easy to use, reliable and extensible, routing layer discrete event simulation platform. DARS has been implemented in Java, and offers a versatile front-end user interface (UI) that provides intuitive, drag-and-drop access to the back-end simulation engine. Besides extensive logging capabilities and interactive and batch mode scenario execution, some of the key features include simulation animation, pausing and replay.

The simulator was designed to serve both as a learning tool and an experimentation platform for students and researchers alike. It includes two routing protocols: Ad hoc On Demand Distance Vector (AODV) and Destination-Sequenced Distance-Vector Routing (DSDV) and comes with extensive code and user's guide documentation to support those interested in using it or extending its capabilities by developing new protocols. This paper is organized as follows: Section 2, provides some background of MANET protocols and simulators. Section 3 includes the description of the simulation environment including the software architecture and an overview of the main features. In Section 4 a sample simulation run is described, using AODV as the routing protocol. Finally, Section 5 concludes the paper with a summary and some ideas for possible improvements and future work.

II. Background

A. Mobile Ad Hoc Routing Protocols

Mobile ad-hoc networks are wireless networks with message exchanges based on multi-hop links. The lack of any other infrastructure (ex. router or base station) requires each node to act as a router by receiving and forwarding packets to and from other nodes. The implementation of routing is based on sets of rules (a protocol) that define the necessary node mechanisms and packet propagation methods. Numerous such protocols have been developed, each using a different strategy for determining the optimal path and moving packets between source and destination nodes. Unlike non-mobile networks that may use static routing protocols, in MANETs, dynamic routing protocols are used to match the dynamic nature of the environment which manifests with continuously changing routing tables. These protocols are typically classified under two categories, *Distance Vector* and *Link-State* protocols.

Distance-vector routing requires from nodes to act as routers and periodically, or when network topology changes occur, inform their neighbors of the updated routing tables. Link-state routing protocols on the other hand require from a node acting as a router to inform all other nodes in a network of any topology changes. In MANETs, routing protocols can be further classified as on demand (reactive) and table-driven (proactive) routing protocols

Reactive protocols do not maintain routing information unless there is communication. When a node attempts to communicate with another node, then the protocol searches for the route and establishes the connection to facilitate the message exchange. This is usually achieved by flooding the network with routing requests [22]. The Ad hoc On-Demand Distance Vector (AODV) protocol is one of the most commonly studied and used distance vector, reactive protocols. Its implementation is simple, based on on-demand route discovery, route maintenance and hop-by-hop routing making it well-suited to dynamic network topologies. Some of the advantages of AODV are its simplicity and scalability to very large number of nodes due to its efficient (on-demand) bandwidth utilization and its robustness against nodes that disappear from the topology -- a common occurrence in MANETs. The limitations of the protocol are based on its inherent inability to reuse routing information, a relatively high route discovery latency due to its on-demand nature and its susceptibility to security attacks [13].

Proactive routing protocols maintain destinations and routes in routing tables before they are required and periodically distribute them to the entire network. This type of routing protocol often has the properties of link-state routing and only vary on the routing information that is being updated in each routing table. The Destination-Sequenced Distance Vector (DSDV) Routing protocol is such a protocol. Each node maintains in a routing table extensive routing and node information which is frequently advertised and either fully or incrementally transmitted to each of the node's neighbors. The design objective and advantage of DSDV was to guarantee loop-free paths [11]. Another advantage is maintaining only optimal paths instead of multiple paths to every destination. However, the continuous routing information advertising results in high bandwidth consumption, making DSDV inefficient, not scalable and not suitable for larger networks.

When comparing the two protocols it is obvious that DSDV works better in smaller networks while AODV seems more of a a general purpose protocol. Each one is a good representative of the two different classes, reactive and proactive protocols. As such, we decided to use them as the two example protocols that come with the first release of DARS.

B. Simulators

The examination of the operation and routing behavior of a real MANET can become very challenging. As topology sizes expand by including more nodes and the network structure changes as mobility increases, it becomes very difficult to obtain timely, meaningful and reproducible information from a real MANET environment. This means that the cost and difficulty of troubleshooting, verifying or improving the operation of the network also increase.

The use of simulation lends itself well when studying MANET routing protocols. This has led to the development of a number of simulation platforms varying in complexity, scope granularity and availability.

ns-2 is one of the most widely used open-source discrete event network simulators in the research community. It is based on the OSI model and was designed to simulate wired networks. Support for wireless networks and mobility models has been introduced either as extensions or by integration with other projects (i.e. CMU Monarch Project [20], Blueware [3], Graph Mobility project [21] and Obstacle Mobility [9]).

GloMoSim [23] is the second most popular open-source, discrete-event, OSI standard compliant, wireless network simulator. It is based on message-passing and was written in Parsec, a C-based simulation language [2]. While new protocols and modules have to be written in Parsec, the simulator has also been enhanced by the integration of external projects (i.e. Obstacle Mobility [9]) that have increased its usefulness. The lack of GloMoSim good documentation and a user interface has been addressed by QualNet [16], a commercial ad hoc network simulator that supports and extends protocols and models offered in GloMoSim.

Another commercial discrete-event network simulator is OPNet [6]. Written in C++ it offers a mature user interface, concurrent simulation scenarios execution and it supports an extensive list of protocols There are other simulators, such as J-Sim [10, 18], OMNet++ [8], GTNets [17], each offering a variety of models, options and features but they have received less attention by practitioners and researchers.

From the leading simulators, *ns-2* is powerful but complex, not very modular and computationally intensive. GloMoSim has poor documentation and was developed using Parsec [2], a non-mainstream programming language which makes its adaptability and use in the classroom less attractive. Overall, many of the community developed and open source simulators tend to lack good documentation and support. Commercial simulators mitigate these shortcomings but they are closed source and often less flexible in extending their capabilities.

Given the breadth and complexity of MANET simulation, we realized that it would be too ambitious to attempt to develop an all encompassing simulation platform that would mitigate all the issues with existing simulators. Inspired by the development of the AODV Simulator [19], we decided to focus on creating an extensible infrastructure that is focused on the simulation of MANET routing algorithms and protocols.

III. Simulation Environment

A. Software Architecture

DARS has been designed to enable users to interactively experiment with routing protocols, devise their own mobile topologies, develop communication scenarios and examine the generated network behavior.

The architecture of DARS is based on a two-tier architecture. The first tier is the graphical user interface (GUI) which enables users to interact with the simulator while also providing the simulation animation rendering capabilities. The second tier is the simulation's main tier which includes an Event Handler module, the Simulation engine, a Node/Protocol

Interface Module, a Logging System and an an Error Handler module.

The Event Handler acts as a bridge between the tiers. It processes user events that occur on the GUI and is designed to interact with any GUI front-end that can generate and pass the necessary user events to the Simulation Engine. The Simulation Engine is the simulator's core component which performs the necessary discrete event computations and provides the necessary internal coordination, processing and replay capabilities The Node/Protocol Interface is designed to interact with the different routing definitions that describe how each protocol deals with node communication, path selection, and message exchanges. Finally, the Logging System creates a time-ordered record of all events and activities that take place during a simulation execution.

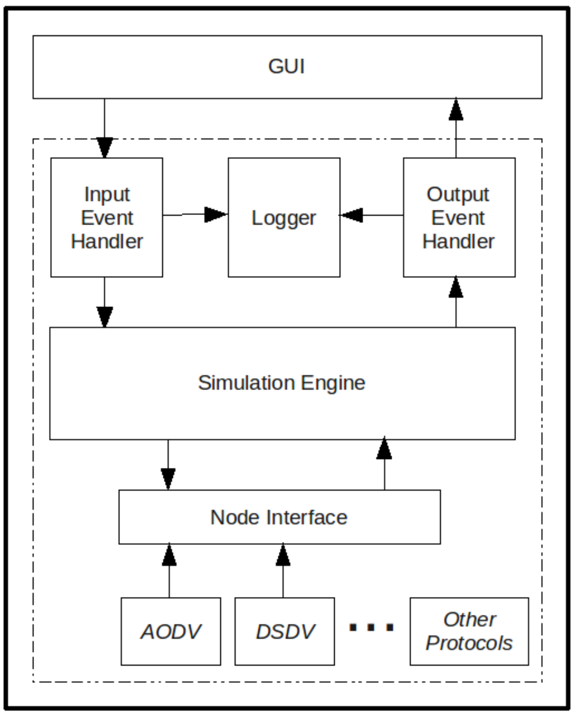

Figure 1. The DARS two-tier software architecture and its modules.

B. Features

DARS was developed as a platform independent application with three distinct overarching objectives.

First, to provide the necessary extensible infrastructure to perform reliable, scalable and realistic routing simulation experiments.

Second, to enable users via a GUI to setup, configure, and access both the simulation parameters and execution results.

Third, to provide a a user-friendly and intuitive GUI that could animate the routing process and provide to the user all the necessary information, including error handling, to assess the progress of the simulation. Following are some of the features and capabilities provided in DARS:

- **Intuitive Interface with Realistic Visualizations.** The DARS GUI provides a set of easy to use interface controls with drag-and-drop functionality, menu-driven navigation and keyboard shortcuts, giving easy access to topology configuration and node parameters. Distinct canvas areas enable users to focus on the GUI area of interest. The top area of the interface contains simulation control buttons and sliders. The middle (main) canvas area is where the network topology is configured and displayed During a simulation session, this graphic display provides the real-time animation of the MANET. The animation feature, while computationally intensive, provides a realistic view of the network. For larger networks, users have the option to disable the animation and conserve resources. The bottom canvas area is the DARS logger console, displaying detailed, timestamped, text-based information of the session in progress.

- **High Granularity of Node Control.** DARS allows the user to add, edit attributes and remove single or multiple nodes before and even during the simulation execution.

- **Flexible Topology Configuration.** Users, using drag-and-drop operations on the GUI, have complete freedom to setup, edit, save for later use and remove network topologies of any configuration. Node location and parameters are customizable and it is also possible to deploy random topologies with multiple nodes. Node parameters can be set manually or automatically for single or multiple nodes, or group of nodes using random but valid data within ranges.

- **Extensive Logging Capabilities.** DARS provides both real-time and off-line detailed logging of all events that take place at each increment of the simulation session including the state of each node and all the transmitted messages. The generated, verbose logs can be displayed on the GUI and filtered out according to the desired level of detail. These filtering options provide an invaluable tool for real-time monitoring and post-session analysis.

- **Simulation Replay.** The logs are constructed in such a way that can be later reloaded on DARS and have an exact replay of a previously executed simulation. During the replay, users are able to interact in real-time with the simulation, change and manipulate parameters to obtain different information.

- **Multiple Simulation Execution Modes and Real-Time User Interaction.** In interactive mode, users can, during an active simulation session make any number of changes in the simulation parameters (e.g. simulation speed), the network topology, node properties or even add messages to be transmitted between nodes. It is also possible to pause an active session, make changes and then resume the simulation with the new changes going into effect. In batch mode, a user may reload an existing simulation session and just execute it from start to finish. In blended mode, the user may begin by loading a previously executed session but the user elaborates upon the loaded configuration by changing node or topology parameters.

- **Real Time Routing Tables.** DARS provides the ability to look at any nodes routing table at every given point during the simulation. The routing table will dynamically change based on the surrounding topology and the given protocol in use. Multiple routing tables can also be displayed simultaneously.

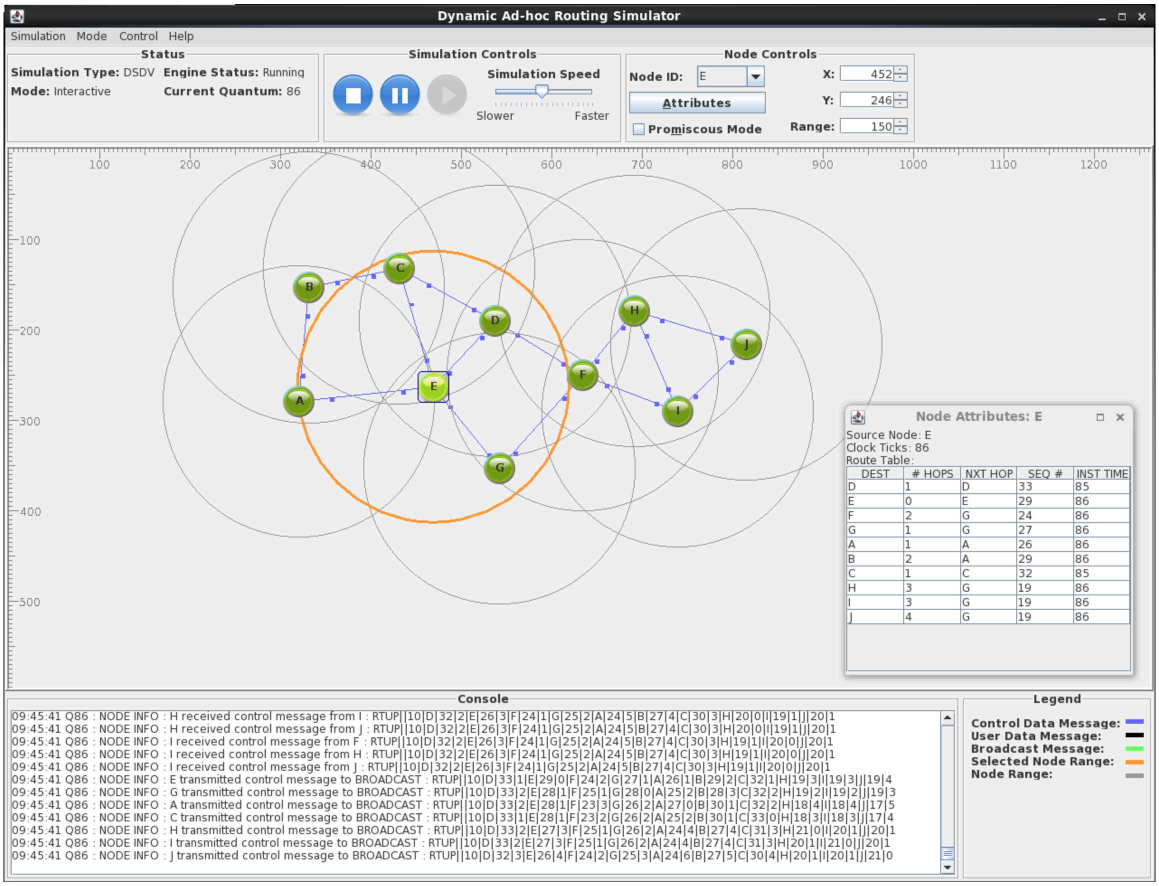

Figure 2. The DARS GUI consists of three panes: (a) the top pane which has simulation controls, (b) the middle pane which displays the network topology and the simulation animation during execution, and (c) the bottom pane which shows the real-time simulation log entries.

- **Extensibility via an exposed Application Programming Interface (API).** The DARS architecture is modular and has been designed to allow users to extend the system both in the front and the back-end. The GUI module can be easily changed or even replaced with a completely different interface. Similarly, users are able to extend the two current available classes of protocol types (one proactive and one reactive) by adding more protocols within these two classes (e.g. Clusterhead Gateway Switch Routing [5], Distributed Bellman-Ford Routing Protocol [4], Hierarchical State Routing [14], Robust Secure Routing Protocol [1] etc) or by adding a new protocol class type (e.g. adaptive, flow-oriented, power aware, etc.). Adding a protocol is simple, well-documented and requires just three steps: (a) using a "Node Factory" class to create a new node, (b) extending a "Node Class" to actually implement a protocol and (c) creating a dialog box that describes the node attributes.

IV. Experimentation with DARS: AODV

To exercise the features of DARS, two protocols have been included with the initial release: AODV and DSDV.

Fig. 3 shows an animation snapshot of the main pane from the execution of the AODV routing protocol.

In this instance, ten mobile nodes were deployed. The circles around the nodes represent the communication signal strength and the lines connecting the nodes indicate that route connectivity has been established between nodes and messages can be exchanged. Users have the ability to display in real-time the routing table information of any node on the network. In AODV, routes are discovered on an "as-needed" basis and only a limited horizon of one hop is maintained instead of the entire route.

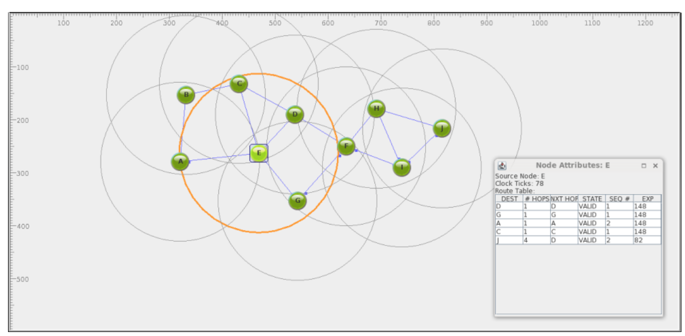

Figure 3. A snapshot of an AODV-based MANET simulation execution with ten nodes.

The numbered tick marks on the left and top side of the pane are used to provide coordinate information on the location of a node. While in this example the particular network is very small in size and the location of the nodes is obvious, DARS can accommodate network topologies with thousands of nodes, exchanging large numbers of messages.

V. Conclusion and Future Work

Simulations are very useful to understand and examine issues in mobile ad hoc networks. A variety of simulation platforms exist but the majority were developed for wired network topologies. The proliferation of interest in wireless networks has generated the need to enhance these platforms and extend them in order to accommodate network node mobility. Some of these platforms have been successful in attracting MANET researchers but each platform seems to have its own shortcomings. The most common limitation is the increased complexity of many platforms or the difficulty in extending features and capabilities.

We developed DARS, a discrete event simulator, with the goal of creating a versatile, extensible and scalable but simple to use simulation platform. For the first release we included two routing protocols to ensure the feasibility of running MANET topologies on the simulator. Preliminary results are very encouraging and DARS has met the original software requirements and will hopefully also meet the requirements of a wider user base.

For the immediate future, there are numerous potentially interesting areas of work. First, it would be useful to have a verification study that would evaluate and benchmark DARS against certain criteria and against other simulation platforms that can handle the mobility and dynamism of ad hoc networks. Second, the usefulness of DARS will increase if more routing protocols are added to the existing suite. A geographically aware and a secure routing protocol would be two desirable top candidates that could enrich the current version of the simulator. Third, it would be very beneficial to conduct a performance study with extensive testing in order to establish the extent of the capabilities of the software and the physical requirements and boundaries (e.g. RAM, CPU) imposed by hardware. Currently, only anecdotal evidence of preliminary load and stress testing results have indicated that DARS scales well (thousands of nodes) but a more systematic and comprehensive analysis should be undertaken.

Overall, DARS seems to have fulfilled the motivating objective of creating an intuitive, versatile and extensible simulation tool for mobile ad hoc networks. The software source, documentation and executables for various platforms can be found on the project page at: http://dars.sourceforge.net/

References

[1] Afzal, S.R., Biswas, S., Koh, J., Raza, T., Lee, G. and Dong-kyoo Kim, RSRP: A Robust Secure Routing Protocol for Mobile Ad hoc Networks, Proceedings of Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE, pp. 2313-2318, March 31 2008-April 3 2008.

[2] Bagrodia, R., R. Meyer, M. Takai, Y., Chen, X. Zeng, J. Martin and H. Y. Song, Parsec: A parallel simulation environment for complex systems, Computer 31 (1998), pp. 77–85.

[3] Barr, R., (2002) Blueware: Bluetooth simulator for ns, Technical report, MIT, Cambridge (2002).

[4] Bartsekas, D. P., Gallagher, R.G. (1987) Distributed Asynchronous Bellman-Ford Algorithm, Data Networks, pp. 325-333, Prentice Hall, Enlgewood Cliffs, 1987, ISBN 0-13-196825-4.

[5] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, Mario Gerla (1997) Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel, IEEE Singapore International Conference on Networks, SICON'97, pp. 197-211, Singapore, 16.-17. April 1997, IEEE.

[6] Desbrandes, F., S. Bertolotti and L. Dunand, Opnet 2.4: An environment for communication network modeling and simulation, in: Proceedings of European Simulation Symposium. Society for Computer Simulation, 1993, pp. 64–74.

[7] Giordano, S. (2002) Mobile Ad Hoc Networks, in Handbook of Wireless Networks and Mobile Computing (ed I. Stojmenović), John Wiley & Sons, Inc., New York, USA. doi: 10.1002/0471224561.ch15

[8] Imre, S., Keszei, Horv´ath, Holl´os, Barta and Kujbus, Simulation environment for ad-hoc networks in OMnet++, in: IST Mobile Summit 2001, 2001, pp. 135–140.

[9] Jardosh, A., E. M. Belding-Royer, K. C. Almeroth and S. Suri, Towards realistic mobility models for mobile ad hoc networks, in: MobiCom '03: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (2003), pp. 217–229.

[10] J-Sim, http://sites.google.com/site/jsimofficial/, (visited 20-March-2011).

[11] Perkins, C. E. and Bhagwat. P. (1994). Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In Proceedings of the conference on Communications architectures, protocols and applications (SIGCOMM '94). ACM, New York, NY, USA, 234-244.

[12] Perkins, C.E. (2001). Ad hoc networking: an introduction. In Ad hoc networking. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA 1-28.

[13] Ning.P. and Sun.K. (2003) How to misuse AODV: A case study of insider attacks against mobile ad-hoc routing protocols. Technical report, Comput. Sci. Dept., North Carolina State Univ., Raleigh, NC, USA, 2003.

[14] Guangyu Pei and Mario Gerla and Xiaoyan Hong AND Ching-Chuan Chiang, A Wireless Hierarchical Routing Protocol with Group Mobility, IEEE WCNC'99, New Orleans, USA, September 1999.

[15] Riera, S. M., O. Wellnitz and L. Wolf, (2003) A zone-based gaming architecture for ad hoc networks, in: NETGAMES '03: Proceedings of the 2nd workshop on Network and system support for games (2003), pp. 72–76.

[16] QualNet, http://www.scalable-networks.com/, (visited 25-March-2011).

[17] Riley, G. F., The Georgia Tech network simulator, in: MoMeTools '03: Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research (2003), pp. 5–12.

[18] Sobeih, A., Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan and Honghai Zhang, J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks, in the IEEEWireless Communications Magazine, August 2006.

[19] The AODV Simulator, http://aodvsimulator.sourceforge.net/, (visited 25-March-2011).

[20] The CMU Monarch Project's Wireless and Mobility Extensions to NS. (1998), http://www.monarch.cs.rice.edu/, (visited 25-March-2011) .

[21] Tian, J., J. Hahner, C. Becker, I. Stepanov and K. Rothermel, (2002) Graph-based mobility model for mobile ad hoc network simulation, in: Annual Simulation Symposium, 2002, pp. 337–344.

[22] Tseng Y.C., Shen C.C, and Chen W.T. (2003) Mobile IP and ad hoc networks: An integration and implementation experience. Technical report, Dept. of Comput. Sci. and Inf. Eng., Nat. Chiao Tung Univ., Hsinchu,, Taiwan, 2003.

[23] Zeng, X., R. Bagrodia and M. Gerla, Glomosim: A library for parallel simulation of large-scale wireless networks, in: Workshop on Parallel and Distributed Simulation, 1998, pp. 154–161.