# A Survey and Comparative Study for Performance Evaluation Technologies for Wireless Networks

**Mohammad M. M. Qabajeh† and Liana K. M. Qabajeh††**

†College of Engineering and Technology Palestine Technical University – Kadoorei (PTUK), Palestine
††College of Information Technology and Computer Engineering Palestine Polytechnic University – PPU, Palestine

**Summary**
Recently, the advances in portable computing and wireless technologies are opening up exciting possibilities for the future of wireless mobile networking. This rapid penetration has stimulated a change in the expectations of wireless users. Mobile Ad hoc Networks (MANETs) have evolved a great deal over the past two decades and considered as one of the most important and essential technologies to support future pervasive computing scenarios. MANETs have gained significant interest and popularity since they have enormous potential in several fields of applications. Absence of infrastructure, self-configuring and mobility are the main reasons behind this popularity. The contribution of this research is its brief description of the different evaluation schemes that are used for MANETs and other networks. Previous efforts have not surveyed these evaluation methods. Hence, this paper is very useful to research community to choose the suitable evaluation method for their research.

***Key words:***
*Network Simulator; Ad hoc networks; NS3; GloMoSim; QualNet*

## 1. Introduction

In the early 1972, DARPA pioneers its research in Ad hoc networks by deploying its Packet Radio Networks (PRnet) [1]. Since that time, the concept of Ad hoc wireless networks is introduced. Ad hoc networks are formed when a collection of mobile devices communicate with each other without pre-established infrastructure. Nodes in an Ad hoc network are often mobile, but it can also consist of stationary nodes [2][3].

Each of the nodes has a wireless interface and communicates with others over either radio or infrared channels.

Mobile Ad hoc NETwork (MANET) is a type of Ad hoc networks with rapidly changing topology. Formally, MANETs are collections of mobile nodes that communicate with each other over wireless links in the absence of any infrastructure or centralized administration [4][5]. This ensures that the network will not cease functioning just because one of the mobile nodes moves out of the range of the others. Each mobile node acts as a host generating flow, being the receiver of a flow from other mobile nodes, or as a router and responsible for forwarding flows to other mobile nodes [6]. Mobile nodes in Ad hoc networks have a limited transmission range, nodes that relies within the transmission range can communicate directly with each other, while intermediate nodes are needed to forward flow between nodes that are unable to communicate directly as shown in Figure 1. In MANETs, the mobile nodes may be laptops, palmtops, Personal Digital Assistants (PDAs), mobile phones, or pocket PC (Personal Computer) with wireless connectivity.

MANETs can be used in different applications that involve point-to-multipoint or multipoint-to-multipoint communication patterns. Disaster recovery, search and rescue efforts, military battlefields and temporary offices are common examples of such applications. One real life example is the attack on world trade center at New York in USA in 2001 [7].

The rest of the paper is organized as follows: In the consequent section, a detailed description of performance evaluation options is introduced. Section 3 provides a detailed description of the network simulators used in the research community. In section 4, a discussion of the simulators is provided. Finally, concluding remarks are summarized in section 5.

## 2. Performance Evaluation Options

In MANETs, evaluating and testing the routing protocols is a mandatory phase to its success in a real world application. To perform this evaluation, researchers have four options: using test-beds, emulators, analytical modeling or using simulation tools. The following subsections present an overview about each of these choices.

### A. Testbeds

Testbeds are often known as in-lab networks built and used by research community. In MANETs, the best way to predict the network behavior is to deploy it in a real environment which provides the best realism. Executing the real code on the real environment can detect more details that might be missed in the simulation [8].

However, testbeds have several drawbacks: First, the cost is very high, since building a testbed requires mobile terminals, wireless transmit equipments, analysis tools and high intensive labor with accumulated experience to monitor the

Manuscript received October 5, 2019
Manuscript revised October 20, 2019

testbeds, which is very expensive. Second, high complexity, since managing the deployment process, monitoring the testbeds and the wide range of mobility scenarios makes conducting testbeds a challenging task. Third, testbeds do not support scalability due to the challenges facing testbeds construction which makes it difficult to support scalable networks. Fourth, assurance of repeatability, since the nature of wireless environment makes it hard to ensure execution under similar conditions for each test run [9]. It is proven in the literature that no testbeds of more than 50 nodes were proposed [8][9][11].

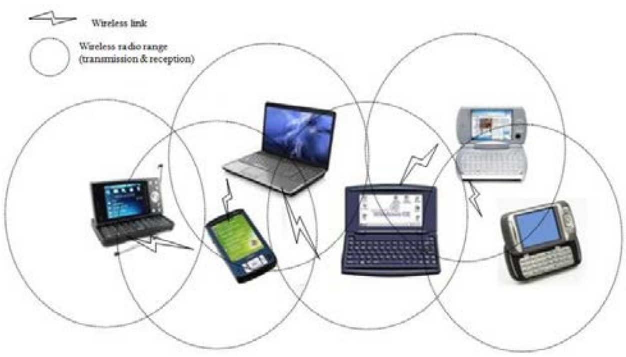


Fig. 1 Basic structure of an Ad hoc network

## B. Emulation

An emulator provides a combination of pure simulations and protocol implementation (testbeds). In particular, some of the network components are implemented in the real world and the others are simulated. The purposes of emulators are to allow testing the protocol in real hardware and to prepare for direct execution of the protocol in the real word. This allows setting and testing some underlying parameters or functionality of the proposed protocol in-lab without physically moving the nodes to the real environment [12]. For example, the properties of a physical layer can be emulated using Field Programmable Gate Arrays (FPGAs) [13].
Emulators have several advantages: First, using hybrid between network simulation and protocol implementation provides accurate evaluation of the MANET protocol in low cost and in large scale scenarios. Second, emulation is closer to realism compared with simulation, since mobile nodes and traffic pattern is real while only the link pattern is emulated [14]. Third, the cost of emulation is low, since it can be built in the lab environment and no additional equipment is needed. Forth, emulation provides seamless connection between protocol evaluation and implementation, since the software used in emulations can directly be used in the testbeds.

## C. Analytical Modeling

Analytical models use mathematical notions and models to describe certain performance aspects of a system under study.
They provide a best qualitative insight into the effects of various parameters and their interactions. Analytical modeling is a cost effective evaluation method because it gives a thorough understanding of the system. Also, it can often be quickly setup and evaluated, requires only paper, pencils and time to analyze the model. Therefore, analytical modeling is the cheapest and least time-consuming performance evaluation method compared to testbeds or simulation. Moreover, the results of analytical modeling can have better predictive values than testbed or simulation [15].
However, analytical models ignore network dynamic and cannot capture all the details that can be built into simulation models. Simulations can incorporate more details and require fewer assumptions than analytical modeling, and thus, more often are closer to reality. In addition, a solid mathematical background and probability theory are needed to build this kind of models. Many systems are too complex for analytical modeling, which requires too many simplifications, assumptions and approximations to turn out accepted results [15].

## D. Simulation

Simulation is defined as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for its control [16]. Simulation is the most widely used method in evaluating MANET routing protocols [17]. This is due to the following reasons: First, because it provides cheap, repeatable and controlled way with acceptable overhead required to carry out a simulation. Second, simulation allows evaluating scalable networks. Third, simulation enables experimentation with configurations that may not be possible with existing technology. Fourth, simulation allows continuous development of the models and can be considered as an early stage of the actual implementation which simplifies the real implementation of the models [18].
However, simulation results are not as accurate as real implementations because implementation can provide more reality than simulation tools. Also, simulation may be slow and needs hours of simulation time to examine simple effect.
Table 1. Summaries the comparison between the different evaluation approaches. The following section will discuss more details about the simulation tools.

Table 1: Comparison of evaluation approaches

| *Approach* | *Advantages* | *Disadvantages* |
|---|---|---|
| ***Testbeds*** | • Allow a level of control over real world and use real devices „<br>• Can detect more details that might be missed in simulation.<br>• Ability to run real applications.<br>• More accurate than simulation | • The cost is very high<br>• high complexity which make it challenging task. „<br>• Difficult to support scalable networks „<br>• Mobility of nodes increase the complexity.<br>• Dependant on location. |
| ***Emulation*** | • Combine the repeatability, configurability, isolation and manageability of simulations and the realism of testbeds<br>• Provide a realistic physical layer<br>• Testing protocol parameters in-lab without physically moving the nodes<br>• Low cost and in large scale scenarios<br>• Test real systems effectively | • Emulators are really slow.<br>• Expensive<br>• Difficult in exploring a number of different dynamic |
| ***Analytical Modeling*** | • Predict system behavior<br>• Less expensive<br>• Cost effective<br>• Provide the best insight into the effects of various parameters and their interactions<br>• Quick evaluation | • Detailed system behavior cannot be captured<br>• Results obtained by it are not reliable<br>• Strong background in mathematics and probability theory needed |
| ***Simulation*** | • Configurability „<br>• Relatively slow and not accurate<br>• Relatively cheap, providing highly reproducible results,<br>• Scaling very well | • Do not operate in real-time<br>• A long simulation time is required<br>• Model setup may take a long time |

## 3. Overview of Simulation Approaches

Currently several simulation tools exist for Ad hoc networks, including DIANEmu [19], REAL [20], GloMoSim [21], GTNets [22], Jane [19], NS-2 [23], pdns [24], OPNET [25], OMNeT [26] and SWANS [27]. These tools differ in their simulation capabilities, features, environments and scalability. Some are dedicated to MANETs simulation such as Jane and SWANS. Other simulators result as extensions of wired network simulators such as NS-2 and general-purpose discrete-event simulation engines (such as PARSEC [28] and Maisie [29]).

In the following subsections, a review of the most popular simulation tools [8] is presented.

### 3.1 Network Simulator 2 (NS-2)

NS-2 [28] is considered as one of the most popular and well known simulation tool for implementing all types of network protocols. NS-2 is developed at Information Sciences Institute [30], and is supported by the Defense Advanced Research Projects Agency (DARPA) and National Science Foundation (NSF). NS-2 is a discrete-event network simulator organized according to the OSI model and initially intended to simulate wired networks [26]. After that 802.11 MAC Layer and important routing protocols needed in MANETs have been added to NS-2 [32]. The core of NS-2 is a huge piece of code written with C++ due to its quickness and Object-oriented possibilities. To ease the use of NS-2, it appears to the user as an Object Tool Command Language (OTCL) interpreter. It reads scenarios files written in OTCL and produces a trace file in its own format. This trace needs to be processed by user scripts or converted and rendered using the network animator, NAM [32], which permits to visualize the output, provide packet-level animation, and provide a Graphic User Interface (GUI) interface to design and debug network protocols. The combination of the two languages offers an interesting compromise between performance and ease of use; however this increases the complexity of the simulator and results in difficulties in learning and debugging NS-2 [33].
NS-2 is an open-source simulator, which makes it interesting on the one hand, but on the other hand there are some negative aspects that come along with it. Unfortunately NS-2 suffers from its lack of modularity and its inherent complexity. Indeed, adding components/protocols or modifying existing ones is not as straightforward as it should be.
Learning NS-2 needs a long period of time due to the lack of documentation in the source code and the usage of two programming languages. For a long time, NS-2 has been said to have few good documentation. Recently, the situation changed, as several users have put online their experience in the form of tutorials or example-driven documentations. Another well-known weakness of NS-2 is its high consumption of computational resources. A harmful consequence is that NS-2 lacks scalability, which makes it difficult to simulate large networks. NS-2 is typically used for simulations consisting of no more than a few hundreds of nodes [8][11].

### 3.2 Network Simulator 3 (Ns-3)

NS3 is a discrete-event network simulator. It is open source and licensed under the GNU GPLv2 license. It was developed in 2008 to replace its predecessor NS2, to improve the scalability and performance and to reduce the compilation time by using C++ in combination with the scripting language OTCL. NS3 is intended to be used with Linux, although it is possible to run it on Windows by using cygwin or MiniGW. Even though it does not offer any graphical user interface it has proven to be comprehensible and easy to handle. The entire code is documented and can be accessed via Doxygen, which make it easy to read and understand the code of an NS3 simulation [34].

As stated before, NS3 is based on the concept of discrete simulation. This means that a point in simulation time is assigned to every event, events are initiated and triggered consecutively and “simulation time moves in discrete jumps from event to event.

NS3 simulation stands for a communication point, such as an end system or a router. NS3 is event-base simulator having base for any event and interaction. Functionality and properties are added to these nodes. The nodes in NS3 are interconnected by channels, which represent the different forms and media of data transmission. Two of the C++ classes in NS3 that describe channels are the point to point channel for wired communcation and the Wifi channel for wireless connection [35].

NS3 offers many different applications for all kinds of network functionality, which can be configured and adapted to the intended network behavior. In NS3, the user can create and configure nodes, channels, net devices and applications separately or this can be done by using the extensive and powerful Helper-API of NS3 with relatively low effort. Also, Helper-API can add a protocol stack and address to a set of nodes.

### 3.3 Global Mobile Information System Simulator (GloMoSim)

GloMoSim is a scalable simulation environment for wireless and wired network systems that was developed at University of California, Los Angeles. GloMoSim aimed at stimulating models that may contain as many as 100,000 mobile nodes with a reasonable execution time [36]. It is the second most popular wireless network simulator [8]. GloMoSim is written in the parallel discrete-event simulation capability provided by a C-based parallel simulation language; Parallel Simulation Environment for Complex systems (PARSEC) [36] and hence benefits from the latter’s ability to run on shared-memory symmetric processor computers.

GloMoSim respects the Open Systems Interconnection model (OSI) standard [11] and has been developed using languages, libraries and frameworks dedicated to discrete-event simulation. These middleware technologies typically focus on performance, concurrency and distribution [8]. Standard Application Programming Interfaces (APIs) are used between the different layers. This allows the rapid integration of models developed at different layers by different users [33]. Two versions of the simulation tool exist: the academic research version, which is for academic use only, and a commercial version, which is distributed as the QualNet software package.

GloMoSim uses parallelism which refers to the simultaneous execution of different instructions of the same program. Parallelism is used to quicken simulations and allow GloMoSim to model networks made of tens of thousands stations [8]. The parallelization technique used by GloMoSim is to split the network into different sub-networks, each of them being simulated by distinct processors. The network is partitioned in such a way that the number of nodes simulated by each partition is homogeneous.

Source Code is written primarily in C and the PARSEC compiler is used to create executable files. For the development of custom protocols in GloMoSim, some familiarity with PARSEC is required. Most protocol developers will write purely C code with some PARSEC functions for time management. PARSEC code is used extensively in the GloMoSim kernel, but it is not required to know and understand how the kernel works.

### 3.4 Netsim Simulator

Network Based Environment for Modeling and Simulation (NetSim) is a discrete event simulator developed by Indian Institute of Science in 1997. NetSim simulates Cisco system networking hardware and software and is used to analyze computer networks with supreme depth and flexibility.

NetSim has a built-in development environment with excellent GUI support. Users can benefit from a simple drag-and-drop pattern of network construction process. It comes with abundant protocol libraries and models including many wireless supports such as WLAN, IEEE 802.11 a/b/g/n, GSM, CDMA, Wi-Max, MANET, Wireless Sensor Network, and Zigbee [37].

NetSim is more compatible with Windows system than Linux system, and it is more friendly to Visual Studio IDE. Thus, for those who do not use Windows system, they have to run a Windows virtual machine or Windows enabled platform on Linux system to enable NetSim for simulation, which slightly limits its use in academic studies. On the other hand, with support of Visual Studio, it is very user friendly for debugging and testing. NetSim has gained popularity on simulating cognitive radio as either teaching instrument or research tool.

NetSim comes with an in-built development environment, which serves as the interface between User’s code and NetSim's protocol libraries and simulation kernel. De-bugging custom code during simulation is an advanced feature. This can be carried out at various levels including at a per-packet interval [37].

### 3.5 QualNet

Scalable Network Technologies developed QualNet [38] as a commercial version of GloMoSim network simulator which is mainly used for wireless networks. QualNet offers more features than GloMoSim. These features include extensive documentation and technical support, user-friendly tools as well as tools for building scenarios and analyzing simulation output. QualNet also offers a large set of modules and protocols for both wired and wireless networks (local, Ad hoc, satellite and cellular). QualNet runs on all common platforms (Linux, Windows, Solaris). Since QualNet is built on top of GloMoSim, it is written in PARSEC [36]. PARSEC is used to provide event scheduling and parallel simulation services.

There are three available libraries of QualNet. A standard library which offers most of the protocols and models required for research and business-oriented activities in both wired and wireless networks. A MANET library which provides very specific additional components for Ad hoc networks other than those already present in the standard library. A QoS library which includes quality-of-service specialized protocols. Also, QualNet includes a Digital Elevation Model [32] to make nodes and radio waves moving in non-flat terrains with specified radio absorption characteristics. QualNet seems to be the most complete network simulator, in terms of available protocols, models and tools for what concerns mobile Ad hoc networks [39].

### 3.6 Optimized Network Engineering Tools (OPNET)

OPNET [25] is a discrete-event network simulator first proposed by Massachusetts Institute of Technology (MIT) in 1986. OPNET is written in C++ and is a well-established and professional commercial suite for network simulation. It is actually one of the most widely used commercial simulation environments [8]. One of the most interesting features of OPNET is its ability to execute and monitor several scenarios in a concurrent manner.

OPNET comes along with a large number of predefined functions, protocols, devices and behaviors, which make it a powerful program just from the start up and without big effort. Additionally, the opportunity to implement new algorithms is given. Also, several tools and editors are provided. The aim is to make use of the numerous existing components that are part of OPNET in order to decrease the developers’ effort, shrink implementation time, and reduce the number of errors. OPNET provides a hierarchal GUI feature and a lot of documentation comes along with it.

Nevertheless, OPNET is not an open-source software and therefore users and companies need to purchase licenses. Hence, the cost of the software could discourage many developers, since open-source solutions are available. Additionally, the main disadvantage is its relative complexity to model a particular system. The time required to learn it and achieve the modeling of a system can be very long, especially for new developments [40]. Furthermore, it is reported that the OPNET simulator is pretty memory consuming and that it is difficult to modify the library models [8].

### 3.7 OMNeT

OMNeT is a discrete event simulator that has been publicly available since 1997. In particular, it is a general-purpose simulator capable of simulating any system composed of devices interacting with each other's. OMNeT uses C++ programming language and object-oriented design.

OMNeT has been used in several research areas including wireless and ad-hoc networks, sensor networks, IP and IPv6 networks, multiprocessors and other distributed hardware systems, wireless channels, peer-to-peer networks, storage area networks (SANs), optical networks, queuing networks, file systems, validating hardware architectures and high-speed interconnections (InfiniBand) [26]. In general, OMNeT is not designed specially for telecommunication networks.

OMNeT is a component-based simulator and the basic entity in OMNeT is a module. Modules are composed of submodules or they can be atomic. The atomic modules capture the actual behavior. Modules communicate with each other via messages through gates. Gates are linked to each other using connections. For example, the protocol models can be combined into a compound module representing a host node [41].

## 4. Discussion and Simulator Selection

MANETs simulators have different features and models, so selecting the proper simulation tool depends on several factors. First, choosing a simulation package depends on the research requirement. The availability of the routing protocols to be simulated and the support of the under investigation problem are of great importance. Moreover, the number of nodes targeted also determines the choice of the simulation tool. Sequential simulators should not be expected to run more than 1,000 nodes. If larger scales are needed, then parallel simulators are a wise choice. Table 2 summarizes the properties of the discussed simulation tools. We can conclude the following points from the previous sections:

1. NS-2 is the most popular one in academia because of its open-source and plenty of components library. Also, the contribution from the research community in the component library increases its popularity. However, it is quite a complex task to install it and have it works properly. Even after installing it, it is difficult to be learned and used specially that it uses two languages C++ for data and OTCL for control. There is no clear separation between C++ and OTCL. Moreover, NS-2 acquires comparable execution time specially in scalable networks and the graphical presentations

of simulation output data is very lack.

2. NS-3 is designed to replace the current popular NS-2 by redesigning a lot of the mechanisms based on the feedback of the research community. However, NS-3 is not an updated version of NS2 since that NS3 is a new simulator and it is not backward-compatible with NS2. Since NS3 is a non-commercial software results in a larger and therefore more active community, which constantly helps to improve, extend and upgrade NS3.
3. Based on the available functionality, strong focus on wireless networks, GloMoSim is considered as the second famous simulator after NS-2 simulator. Its design is based on parallel/distributed computing environment which allows it to scale up to networks with thousands/millions of nodes.
4. NetSim has very good GUI support. This will help to design a good network and also help in the analysis of the network. The main strength of NetSim is that the package can be run on a variety of operating systems. However, the use of NetSim is limited to academic environments only.
5. QualNet is a commercial simulator that extends GloMoSim by bringing support, a complete documentation, a complete set of user-friendly tools for building scenarios and analyzing simulation output. Also, QualNet extends the set of models and protocols supported GloMoSim. QualNet supports thousands of nodes and run on a variety of machines and operating systems. It has a comprehensive network relevant parameter sets and allows verification of results through by inspection of code and configuration files. However, QualNet does not have any predefined model constructs.
6. OPNET is a commercial tool, purchase of the software and the model libraries are expensive. In addition, it suffers from many disadvantages such as complexity, time required to learn it, memory consumption, and difficulty to modify the library models. However, OPNET is a popular simulator used in industry for network research and development. The GUI interface and the programming tools are also useful to help the user to build the system they want. Also, it has a user friendly graphs, charts, statistics, and even animation can be generated by OPNET for users convenience.
7. OMNet++ is open sourced and widely acknowledged in academia. It has very powerful graphical interface and modular core design. OMNeT++ has generic and flexible architecture which makes it successful also in other areas like the IT systems, queuing networks, hardware architectures, or even business processes as well. OMNET++ and GloMoSiM most suitable for carrying out large scale network simulations. OMNET++ has superior performance than NS-2 and also has some merits over OPNET like free availability and graphical runtime environment.

## 5. Conclusion

Network simulators help the network designers to implement new networking protocols or to modify the existing protocols in a controlled and efficient manner. In this paper, we present an overview of different network simulators which can be used for simulating wired as well as wireless networks. Appropriate guidelines are also provided about the network simulators which will be beneficial in selecting a simulator to perform a particular task or to build project with specified requirements. Each simulator has its domain of relative strengths and weaknesses compared to other simulators, which makes the selection of the appropriate simulator depends on several factors including budget, type of network, size of the network, results analysis, user interface, programming language preferred by the researcher and availability of benchmark code.

**Acknowledgment**

We thank Palestine Technical University Kadoorie for their support of this research.

Table 2: Comparison of networks simulators

| Tool / Feature | NS2 | NS3 | GloMoSim | QualNet | OPNET | OMNeT | NETSIM |
|---|---|---|---|---|---|---|---|
| Interface | C++/OTCL | C++ (with an optional Python scripting API) | Parsec (C-based) | Parsec (C-based) | C/ C++ | C++ Has extensive GUI | Java |
| Parallelism | No | Supports both simulation and emulation | Yes | Yes | Yes | Yes | Yes |
| Popularity | High | High | Moderate | Low | Low | Low (MPI/PVM) | Moderate |

| License | Open source, some extensions require license | Open source, some extensions require license | Open source | Commercial version of GloMoSim (relatively expensive) | Free for academic and educational use/ Commercial (relatively expensive) | Free for academic/educational use/commercial | Commercial |
|---|---|---|---|---|---|---|---|
| Documents and user support | Excellent | Excellent | Poor | Excellent | Excellent | Good | Excellent |
| Required time to learn | Long | Long | Moderate | Easy to learn | Long | Moderate | Easy to learn |
| Scalability | Limited | Moderate | High | High | High | Limited | High |
| Extendibility | Excellent | Excellent | Excellent | Excellent | Excellent | Excellent | Excellent |
| Graphic interface support | Limited GUI | Limited GUI (comprehensible and easy to handle) | Limited GUI | Excellent GUI | Excellent GUI (complex) | Good | Excellent GUI |

## References


[1] Chen Y, Liestman A, Liu J. Clustering algorithms for ad hoc wireless networks. In: Pan YXaY, editor. Ad Hoc and Sensor Networks. vol. 28: Nova Science Publisher; 2004.

[2] Mahmoud F. Reputed Authenticated Routing for Ad Hoc Networks Protocol (Reputed-ARAN). Mater Thesis. Cairo; 2005.

[3] Chhabra GS, Verma G, Patheja PS. Efficient fuzzy ant colony-based multipath QoS aware routing protocol in mobile ad hoc network. International Journal of Mobile Network Design and Innovation. 2018;8(4):225-34.

[4] Chlamtac I, Conti M, Liu J. Mobile ad hoc networking: imperatives and challenges. Elsevier, Ad Hoc Networks. 2003;1:13–64.

[5] Loo J, Mauri JL, Ortiz JH. Mobile ad hoc networks: current status and future trends: CRC Press; 2016.

[6] Yang SH, Bao L, editors. Scalable mobility management in large-scale wireless mesh networks. IEEE Wireless Communications and Networking Conference (WCNC 2011); 2011 28-31 March; Cancun, Mexico: IEEE.

[7] M.Bheemalingaiah, Naidu MM, Rao DS, Moorthy PS. Survey of Routing Protocols, Simulation Tools and Mobility Models in Mobile Ad Hoc Networks. International Journal of Innovations & Advancement in Computer Science. November 2017;6(11):201-2013.

[8] Hogie L, Bouvry P, Guinand F. An Overview of MANETs Simulation. Electronic notes in theoretical computer science. 2006;vol.150(no. 1):81-101.

[9] Nickelsen A, Jensen M. Evaluation of routing dependability in manets using a topology emulator [Master Thesis]. Denmark: Aalborg University; 2007.

[10] Hasan MS, Yu H, Griffiths A, Yang T, editors. Co-simulation framework for Networked Control Systems over multi-hop mobile ad-hoc networks. Proceedings of the International Federation of Automatic Control; 2008 6-11 July; Seoul, Korea.

[11] Newport C. Simulating mobile ad hoc networks: a quantitative evaluation of common MANET simulation models. Technical Report. Dartmouth College Computer Science; 2004. Report No.: TR2004-504.

[12] Jorge H, Juan-Carlos C, Carlos T. Testing Applications in MANET Environments through Emulation. EURASIP Journal on Wireless Communications and Networking. 2009;vol. 2009:20.

[13] Judd G, Steenkiste P. Repeatable and realistic wireless experimentation through physical emulation. ACM SIGCOMM Computer Communication Review. 2004;vol. 34(no. 1):63-8.

[14] Maltz DA, Broch J, Johnson DB. Experiences designing and building a multi-hop wireless ad hoc network testbed. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.; 1999 05 MAR Report No.: ADA368412.

[15] Jain R. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling. New York: Wiley-Interscience; 1991 May.

[16] Gayatry Borboruah, Gypsy Nandi, A Study on Large Scale Network Simulators, International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7318-7322

[17] Kurkowski S, Camp T, Colagrosso M. MANET simulation studies: the incredibles. ACM SIGMOBILE Mobile Computing and Communications Review. 2005;vol. 9(no. 4):50-61.

[18] Ronit L. Patel, Maharshi J. Pathak, Amit J. Nayak, Survey on Network Simulators, International Journal of Computer Applications, Vol. 182(no. 21), October 2018.

[19] Frey H, Görgen D, Lehnert J, Sturm P. A Java-Based Uniform Workbench for Simulating and Executing Distributed Mobile Applications Scientific Engineering of Distributed Java Applications. In: Guelfi N, Astesiano E, Reggio G, editors. Lecture Notes in Computer Science. 2952: Springer Berlin / Heidelberg; 2004. p. 116-27.

[20] Keshav S. REAL: A network simulator. Berkeley, Calif, USA: University of California, Berkeley. Department of Electrical Engineering Computer Sciences. Computer Science Division; 1988 December.

[21] Bajaj L, Takai M, Ahuja R, Tang K, Bagrodia R, Gerla M. Glomosim: A scalable network simulation environment. UCLA Computer Science Department Technical Report. 1999;990027:213.

[22] Riley GF, editor The georgia tech network simulator. ACM SIGCOMM workshop on Models, methods and tools for reproducible network research (MoMeTools ’03); 2003: ACM.

[23] McCanne S, Floyd S. Network Simulator-ns (version 2),Lawrence Berkeley Laboratory, Berkeley, California. 1997 [Available from: http://www.isi.edu/nsnam/ns/.

[24] Riley GF, Fujimoto RM, Ammar MH, editors. A generic framework for parallelization of network simulations. 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS ’99); 1999 October: Published by the IEEE Computer Society.

[25] Desbrandes F, Bertolotti S, Dunand L, editors. Opnet 2.4: an environment for communication network modeling and

simulation. European Simulation Symposium; 1993 October; Delft, Netherlands.
[26] Varga A, Hornig R, editors. An overview of the OMNeT++ simulation environment. First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008'),; 2008 March: ICST.
[27] Barr R. An efficient, unifying approach to simulation using virtual machines [Ph.D. Thesis]: Cornell University; 2004.
[28] Bagrodia R, Meyer R, Takai M, Chen Y, Zeng X, Martin J, et al. Parsec: A parallel simulation environment for complex systems. Computer. 1998;vol. 31(no. 10):77-85.
[29] Bagrodia RL, Liao WT. Maisie: A language for the design of efficient discrete-event simulations. IEEE Transactions on Software Engineering. 1994;vol. 20(no. 4):225-38.
[30] Kalhor S, Anisi M, Haghighat A, editors. A new position-based routing protocol for reducing the number of exchanged route request messages in Mobile Ad-hoc Networks. Second International Conference on Systems and Networks Communication (ICSNC) 2007 25-31 August; Cap Esterel, French Riviera, France.
[31] Schilling B. Qualitative comparison of network simulation tools. Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart; 2005.
[32] Estrin D, Handley M, Heidemann J, McCanne S, Xu Y, Yu H. Network visualization with the VINT network animator nam. IEEE Computer Magazine. 2000 vol. 33(no. 11):63 - 8.
[33] Cavin D, Sasson Y, Schiper A, editors. On the accuracy of MANET simulators. Second ACM international workshop on Principles of mobile computing; 2002; Toulouse, France: ACM.
[34] Simon Duque Anton DF, Dennis Krummacker, Christoph Fischer, Michael Karrenbauer, Hans D. Schotten, editor The Dos and Don'ts of Industrial Network Simulation: A Field Report. ISCSIC '18 Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control, September 21 - 23, 2018; Stockholm, Sweden ACM.
[35] Siraj S, Gupta A, Badgujar R. Network simulation tools survey. International Journal of Advanced Research in omputer and Communication Engineering. 2012;1(4):199-206.
[36] Zeng X, Bagrodia R, Gerla M. GloMoSim: a library for parallel simulation of large-scale wireless networks. ACM SIGSIM Simulation Digest. 1998;vol. 28(no. 1):154-61.
[37] Singh G, Singh H. Performance Evaluation of MANET Routing Protocols using NETSIM. i-Manager's Journal on Information Technology. 2016;6(1):8.
[38] Simulator QualNet. Scalable Network Technologies Inc. [Available from: www. qualnet. com.
[39] Di Caro GA. Analysis of simulation environments for mobile ad hoc networks. Technical Report No. IDSIA-24-03. Manno,Switzerland: DalleMolle Institute for Artificial Intelligence; 2003 December.
[40] Christiansen H, Kuijpers G, Yomo H, Fathi H. Simulator requirements-a comparative evaluation of tools 2003 [Technical University of Denmark, CNTK.:[Available from: http://nsl.csie.nctu.edu.tw/NCTUnsReferences/Simulation%20tool%20evaluation_v21.doc.
[41] Imre S, Kujbus C, Hollos D, Barta P, Kujbus C, editors. Simulation Environment for Ad-Hoc Networks in OMNeT++. European Information Society Technologies (IST) Mobile Summit conference; 2001 9-12 September; Sitges, Barcelona, Spain.

**Mohammad M. Qabajeh** received his B.Sc. in Computer Engineering from Palestine Polytechnic University, Palestine in 2000 and M.Sc. in Computer Engineering from Jordan University of Science and Technology, Jordan in 2006 and his PhD in Computer Engineering from International Islamic University Malaysia (IIUM) in 2012. Currently he is Assistant Professor at Palestine Technical University Kadorie (PTUK). His current research interests include Data Communication and Computer Networking, Wireless Networks and Routing Protocols. He published more than 30 papers in international journals and conferences. He is a reviewer for many ISI journals.

**Liana K. Qabajeh** received her B.Sc. in Computer Engineering from Palestine Polytechnic University, Palestine in 2000 and M.Sc. in Computer Engineering from Jordan University of Science and Technology, Jordan in 2005 and her PhD in Computer Sience from University Malaya (UM) in 2012. She is now working towards her Ph.D in Computer Science at University of Malaya, Malaysia. Currently she is Assistant Professor at Palestine Polytechinc University (PPU). Her current research interests are Distributed Systems and Ad-Hoc Networks.