

# STUDY OF VARIOUS NETWORK SIMULATORS

Dr. L. RAJA

*Assistant Professor, Dept. of Computer Science & Applications, Faculty of Science and Humanities, SRM Institute of Science & Technology, Vadapalani, Chennai - 26 – India.*

\*\*\*

**Abstract** - In the network research area, it is very costly to deploy a complete test bed containing multiple network computers, data links and routers to authorize a certain network protocol or a particular network algorithm. In this state of affairs, the network simulators save lot of money and time. Simulation is the process of designing real system model and conducting experiments for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system. With the simulator the network researcher are able to check whether the network has the ability to work in the real time. Also supports in testing new networking protocols or change the existing protocols in a prohibited manner. This paper presents an extensive survey on current network simulators introducing their characteristics, advantages and disadvantages.

**Keywords:** Network Simulator, NS2, NS3, OPNET, OMNeT++, QualNet, NetSim and J-Sim

## 1. Introduction

Simulation is a very important technology in modern time. Computer assisted simulation can model hypothetical and real-life objects or activities on a computer to study the well-designed structure. A network simulator is a system of implementing the network on the computer through which the performance of the network is calculated. The computer assisted simulation technologies are applied in the simulation of networking algorithms. The functional network field is narrower than general simulation and it is natural that more specific requirements will be placed on network simulations.

Network simulator allows the researchers to test the scenarios that are difficult or expensive to simulate in real world. Design of various network topologies using nodes, hosts, hubs, bridges, routers and mobile units etc. is possible. The network simulators are of various types which can be compared on the basis of: range (simple to the complex), specification of nodes, links and traffic between the nodes. Specifying about the protocols used to handle traffic in a network, user friendly applications (allow users to easily visualize the simulated environment.), text-based applications (permit more advanced forms of customization) and programming-oriented tools (providing a programming framework that customizes to create an application that simulates the networking environment to be tested).

Network simulators are used by people from different areas such as academic researchers, industrialized sectors and Quality Assurance (QA) to design, simulate and analyze the performance of different network protocols. They can also be used to evaluate the outcome of the different parameters of the protocols being studied. Normally a network simulator comprises of wide range of networking technologies and protocols that help users to build complex networks from basic building blocks like clusters of nodes and links. With their help, different network topologies can be designed using various types of nodes such as end-hosts, network bridges, routers, hubs, optical link-layer devices and mobile units. The following sections of the paper are organized as follows. Section 2, discusses about basic concepts in network simulation. Section 3, describes the various Simulators. In section 4, the preface of network simulators studied. In section 5, comparison of simulators is displayed and finally the conclusion in section 6.

## 2. Concepts in network simulation

In this section, we will introduce some basic concepts in the area of network simulation. Generally, network simulators try to represent the real world networks and it is a useful technique, given that the activities of a network can be modeled by calculating the interaction between the different network components (they can be end-host or network entities such as routers, packets or physical links) using mathematical formulas. They can also be modeled by actually or virtually capturing and playing back experimental observations from real networks. Upon receipt of the observation data from simulation experiments, the behavior of the network and protocols supported are analyzed in a series of offline test experiments. All types of attributes can also be modified in a controlled manner to assess how the network can behave under different parameter combinations. Another feature of network simulation worth noticing is that the simulation program can

be used and analyzed together with various strategy, links, applications etc. Typically, users can then adapt the simulator to fulfill their exact needs. Simulators support the most popular protocols and networks such as WLAN, TCP and WSN.

### 3. Simulators

Most of the commercial simulators are Graphical User Interface (GUI) driven, while some network simulators are Command-Line Interface (CLI). The design of the network describes the state of the network (nodes, routers, switches and links) and the events (data transfer, transmission delay, packet error etc.). The major output of simulation is the trace files which log every packet and event that occurred during simulation and is used for analysis. Also provides other tools to facilitate visual analysis of trends and potential trouble spots. Most of the network simulators are discrete event, in which the list of pending "events" are stored and processed in order. Some events triggers the future events (i.e.) the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node.

Simulation of networks is a very difficult task. For example, if blocking is high, then evaluation of the average occupancy is challenging because of high variance. To evaluate the probability of buffer overflow in a network, the time required for a precise answer can be enormously large. Techniques like "control variants" and "sampling" have been developed to speed simulation.

#### 3.1 List of Network Simulators

There are many both free/open-source and proprietary simulators. Examples of notable simulation software are ordered based on how frequently they are mentioned in research papers.

1. NS2 (Network Simulator 2)
2. NS3 (Network Simulator 3)
3. OPNET.
4. OMNeT++.
5. NetSim.
6. QualNet.
7. J-Sim.)

#### 3.2 Uses of Network Simulators

Network simulators serve a variety of requirements. Simulators are relatively fast and economical when compared to the cost and time involved in setting up an entire bed containing multiple network computers, data links and routers. They authorize researchers to test scenarios that might be particularly difficult or expensive to emulate using a real hardware - for instance simulating a scenario with several nodes or experimenting with a new protocol in the network. Simulators are mainly useful in allowing researchers to test new networking protocols or changes to existing protocols in controlled and reproducible surroundings.

A typical simulator encompasses a wide range of networking technologies and can help the users to build complex networks from basic building blocks such as selection of nodes and links. Various types of nodes in Hierarchical networks resembling computers, hubs, bridges, routers, links, switches mobile units etc can be designed with the help of simulators.

Various types of Wide Area Network (WAN) technologies like TCP, ATM, IP etc. and Local Area Network (LAN) technologies like Ethernet, token rings etc., can be imitated with a simulator and the user can examine various standard results apart from devising some novel protocol or routing strategy. Network simulators are widely used to simulate battlefield networks in Network-centric warfare.

There are ample varieties of simulators, ranging from simple to complex. A simple simulator must enable a user to represent network topology, to specify nodes on the network, the links and the traffic between the nodes. More complex systems may permit the user to specify everything about the protocols used to handle traffic in a network. User friendly applications permit users to envision easily the working mechanism of their simulated situation. Text-based applications offer a less sensitive interface, but permits more advanced forms of customization.

## 4. Overview of Network Simulators

Currently there are many network simulators that have different features in different aspects. Short lists of the current network simulators include NS-2, NS-3, OPNET, OMNeT++, NETSIM, QualNet, and J-Sim. These network simulators are selected for discussion regarding their features, advantages and restrictions in this paper.

### 4.1 NS2

The Ns2 is a discrete event simulator targeted at packet level networking research and provides substantial support to simulate group of protocols like TCP, UDP, FTP and HTTP. It comprises of two simulation tools. Ns-2 is primarily UNIX based and fully simulates a layered wire or wireless network from the physical radio transmission channel to high-level applications. The simulator is written in C++ and a script language called OTcl.

**C++:** C++ is fast to run but slower to change, making it suitable for detailed protocol implementation.

**OTcl:** OTcl runs much slower but can be changed very quickly (and interactively), making it ideal for simulation configuration. Ns provides glue to make objects and variables appear on both languages.

NS2 uses an OTcl interpreter by which the user writes an OTcl script that defines the network, (number of nodes and links) the transaction in the network (sources destinations, type of traffic) and the type of protocols used. The outcome of the simulation is a trace file that can be used for data processing (calculate delay, throughput etc). To visualize the simulation, a program called Network Animator (NAM) is used. It visualizes the packets as they propagate throughout the network. The ns-2 simulator has numerous features that make it suitable for our simulations.

- A network environment for ad-hoc networks,
- Wireless channel modules (e.g.802.11),
- Routing along multiple paths,
- Mobile hosts for wireless cellular networks.
- Download of ns-2 source code is possible and can be compiled for multiple platforms.

#### Advantages:

1. NS2 has large number of available models, realistic mobility models, powerful and flexible scripting and simulation setup, large user community and ongoing development.
2. NS2 provides an easy traffic and movement pattern by including an efficient energy model.
3. It provides a set of randomized mobility model [29] and there are several projects to bring advanced mobility models to the simulators.
4. Complex scenarios can be easily tested.
5. Popular for its modularity.

#### Limitations:

1. NS2 needs to be recompilation every time if there is a change in the user code.
2. Real system is too complex to model i.e. complicated infrastructure.

### 4.2 NS3

The ns-3 simulator is a discrete-event network simulator for Internet systems, targeted primarily for research and learning purpose. The ns-3 project, started in 2006, is open-source free software, licensed under the GNU GPLv2 license. It will rely on the current contributions of the community to develop new models, debug or maintain the existing ones, and share the results. Ns3 is mainly used on LINUX systems and not limited to internet based systems alone.

**C++:** implementation of simulation and core model.Ns-3 is built as a library which may be statically or dynamically linked to a C++ main program. These libraries describe the beginning of simulation and their topology.

**Python:** C++ wrapped by Python. Python programs to import an “ns3” module. The features of NS3 simulator are given below.

1. Modular, documented core
2. C++ programs and Python scripting
3. Alignment with real systems
4. Software integrations
5. Virtualization and test bed integration
6. Attribute system
7. Updated models

#### **Advantages:**

1. High modularity than its ancestor NS2.
2. Support simulation for TCP, UDP, ICMP, IPv4, multicast routing, P2P and CSMA protocols.
3. Support for ported code should make model validation easier and more credible.
4. Much more flexible than any other simulators.
5. Wide range of use in both optimization and expansion of the existing networks.

#### **Limitations:**

1. NS3 still suffers from lack of credibility.
2. NS3 is intended to replicate the successful mode of NS2 in which various organizations contributed to the models and components based on the framework of NS2.
3. NS3 needs a lot of specialized maintainers in order to avail the merits of NS3 as the commercial OPNET network simulators.
4. Active maintainers are required to respond to the user questions, bug reports and help to Test & validate the system.

### **4.3 OPNET**

OPNET is extensive and powerful simulation software which enables to simulate heterogeneous networks with a range of protocols. Modeler is a commercial network simulation environment for network modeling and simulation [SAR12]. It allows the users to design communication networks, procedure, protocols, and applications with flexibility and scalability. The network is simulated graphically and the graphical editors mirror the structure of actual networks and their mechanism. An object-oriented system approach is used in the modeler.

**C (C++):** The programming language in OPNET is C (recent releases support C++ development). The initial configuration (topology setup, parameter setting) is usually achieved using Graphical User Interface (GUI), a set of XML files or through C library calls. Simulation scenarios (e.g., parameter change after some time, topology update, etc.) usually require writing C or C++ code; although in simpler cases one can use special “scenario” parameters (e.g., link fail/restore time).

#### **OPNET’s detailed features include:**

1. Fast discrete event simulation engine
2. Lot of component library with source code
3. Object-oriented modeling
4. Hierarchical modeling environment
5. Scalable wireless simulations support
6. 32-bit and 64-bit graphical user interface
7. Customizable wireless modeling
8. Discrete Event, Hybrid, and Analytical simulation
9. 32-bit and 64-bit parallel simulation kernel
10. Grid computing support
11. Integrated, GUI-based debugging and analysis
12. Open interface for integrating external component libraries

**Advantages:**

1. Leverage three different simulation technologies to efficiently tradeoff simulations detail and speed.
2. Fast discrete event simulation engine.
3. Customizable wireless modeling.
4. Integrated GUI based debugging and analysis [28].

**Limitations:**

1. Complex GUI operation.
2. It does not allow much number of nodes within a single connected device.
3. Accuracy of results is limited by the sampling resolution.
4. Simulation is inefficient if nothing happens for long periods

**4.4 OMNET++**

It is a component-based, modular and open architecture discrete event simulator framework. The most common use of OMNeT++ is for simulation of networks, but it is also used for queuing network simulations and other areas as well. It is licensed under its own Academic Public License, which permits GNU Public License like freedom but only in noncommercial settings. It provides component architecture for models.

**C++:** The *C++ class library* comprises of simulation kernel and utility classes (for random number generation, statistics collection, topology discovery etc) -- this one is used to create simulation components (*simple modules* and channels); infrastructure to assemble simulations from these components and configure (*NED language, ini files*); runtime user interfaces or *environments* for simulations (*Tkenv, Cmdenv*); an Eclipse-based simulation IDE for designing, running and evaluating simulations; extension interfaces for real-time simulation, emulation, MRIP, parallel distributed simulation, database connectivity and so on.

**The OMNeT++ components include:**

1. Simulation kernel library
2. Compiler for the NED topology description language (nedc)
3. Graphical network editor for NED files (GNED)
4. GUI for simulation execution, links into simulation executable (Tkenv)
5. Command-line user interface for simulation execution (Cmdenv)
6. Graphical output vector plotting tool (Plove)
7. Graphical output scalars visualization tool (Scalars)
8. Model documentation tool (opp\_neddoc)
9. Utilities (random number seed generation tool, make file creation tool, etc.)
10. Documentation, sample simulations, etc.

**Advantages:**

1. Provides a powerful GUI environment.
2. Tracing and debugging are much easier than other simulators.
3. Accurately models most hardware and include the modeling of physical phenomena.

**Limitations:**

1. It does not offer a great variety of protocols and very few protocols have been implemented, leaving users with significant background work.
2. Poor analysis and management of typical performance.
3. The mobility extension is relatively incomplete.

## 4.5 NETSIM

NetSim is a discrete event simulator developed by Tetcos in 1997, in association with Indian Institute of Science. It has also been featured with Computer Networks and Internets V edition by Dr. Douglas Comer, published by Prentice Hall. It has an object-oriented system simulating environment to support simulation and analysis of voice and data communication scenarios for High Frequency Global Communication Systems (HFGCS).

**Java:** It creating fast, platform independent software that could be used in simple, consumer electronic products. Java designed for simple, efficient, platform-independent program for creating WWW-based programs. Using Java one can create small programs called applets that are entrenched into an HTML document and viewable on any Java-compatible browser. Java applets are compiled into a set of byte-codes, or machine-independent processing instructions.

### Features:

- NetSim modeling and simulation are supported for Aloha, Slotted Aloha, Token Ring/Bus, Ethernet CSMA/CD, Fast Ethernet, WLAN - IEEE 802.11 a/b/g/n and e, X.25, Frame Relay, TCP, UDP, IPv4 and IPv6, Routing - RIP, OSPF, BGP, MPLS, MANET, GSM, CDMA, Wire-less Sensor Network, Zigbee, Cognitive radio)[5].
- It simulates a wide variety of Cisco routers, including 2500 series, 2600 series, 2800 series, and 3600 series, as well as the Cisco Catalyst 1900 series, 2900 series, and 3500 series switches. Protocol libraries are available as open C code for user modification. This can help to avoid the time consuming process such as encoding, customization and configuring commercial simulators to meet customer specific needs. Along with the Boson Virtual Packet Technology engine NetSim utilizes Boson's proprietary Router & Simulator EROUTER software technologies, to produce individual packets. These packets are routed and switched through the simulated network, allowing the simulator to build an appropriate virtual routing table and simulate proper networking. Other simulation products on the market do not support this level of functionality.
- It can be used to create a simulation of the topology of corporate network and help practice trouble-shooting without using devices on the production network.

### Advantages:

1. NetSim has a GUI which features drag and drop functionality for devices, links etc. i.e. Modeling in NetSim is simple and user friendly.
2. It has a built in analysis framework that provides intra and inter-protocol [19] performance comparison with graphical options.
3. Data packet and control packet flow can be visual-ized through NetSim built-in packet animator.
4. It is easy to learn all about NetSim.

### Limitations:

1. NetSim is a single process discrete event simulator. A single event queue is used for the simulation which at any given time contains one entry for each station on the network.
2. Free version of NetSim is not available.

## 4.6 QualNet

It is a commercial network simulator from Scalable Network Technologies, Inc in 2000-2001. It is an ultra high fidelity network simulation software that predicts wired or wireless platform network and their device performance. For a large, heterogeneous network and distributed applications such networks are executed.

**C++:** For implementing new protocols, Qualnet uses C/C++ and follows a procedural paradigm. It uses the parallel simulation environment for the basic operations of complex systems (PARSEC). Hence it can run on distributed machines.

### Features:

- QualNet can support real-time speed to enable network developers and designers to run multiple analysis by varying model, network, and traffic parameters in a short time.

- It can model thousands of nodes by taking advantage of the latest hardware and parallel computing techniques. A very powerful simulation tool that can support simulation of 500 to 20,000 nodes.
- QualNet can run on cluster, multi-core, and multi-processor systems to model large networks with high fidelity.
- It provides high fidelity commercial protocol and device models to enable more accurate modeling of real-world networks.
- It enables the designer to design large wired and wireless networks using pre-configured or user-designed models.
- It also facilitates to design new protocol models and to optimize new and existing models.
- QualNet can connect to other hardware and soft-ware applications, such as OTB, real networks, and a third party graphical software in order to enhance the value of the network model.

**Advantages:**

1. QualNet supports multiprocessor systems and distributed computing.
2. It can simulate a mixture of both wired and wireless networks.
3. It provides GUI that is convenient and easy-to-use.
4. It facilitates sophisticated animation capabilities.
5. It can run on cluster, multi-core, and multi-processor systems.

**Limitations:**

1. The simulation tool QualNet is an extension of GloMoSim which is being commercialized.
2. Installation of QualNet on Linux is difficult.
3. The java based user interface provided by this simulation software is slow.

**4.7 JSIM**

J-Sim has been developed by a team at the Distributed Real-time Computing Laboratory (DRCL). The project has been sponsored by the National Science Foundation (NSF), DARPA's, Air Force Office of Scientific Research's Multidisciplinary University Research Initiative, the Ohio State University and University of Illinois at Urbana-Champaign. J-Sim is free and available with source code.

**Java:** Java is easy to learn and easy to use. In case of any problems, source texts provided with J-Sim can be used to create fresh code, compiled in the target environment, thus 100% compatible with JVM used. Java provides a class called Thread whose instances run parallel with other such instances. It is an object-oriented language, providing the concepts of classes, instances, encapsulation, inheritance and polymorphism. J-Sim provides basic classes for simulation, process and queue which can be either directly used or extended according to specific user's requirements.

**Tcl:** Scripting is an essential part of J-Sim, used to "glue" all the components and define how the operation of the system takes place. It makes possible to manipulate Java objects in the Tcl environment, such as creating an object, invoking a method or accessing a field variable of a Java object.

**Features:**

- It is loosely-coupled, component-based programming model. The ability to handle data independently contexts along with the fact that components are loosely coupled and only bound to one another at system integration time. This is the key reason that a component can be reused in other software systems with the same context. It is as the same fashion as IC chips used in hardware design.
- It provides dynamic thread execution framework for real-time process-driven simulation. In J-SIM, the simulation engine extends the worker Pool class and monitors the activities of all worker threads. It maintains a globally-observed virtual system time that is proportional to the real time.
- Implementation of a complete suite of Internet Integrated/Differentiated/Best Effort Services protocols. The inference of the same is three fold with all the Internet protocol classes available, the abstract classes and by virtue of the component hierarchy.
- A dual-language environment that allows auto-configuration and on-line monitoring. This closely mimics the IC debugging and testing process [25].

- Representation of generic interface classes for trace-driven simulation, with which (i) the network of interest can be emulated based on real network traffic loads and (ii) fidelity of *J-SIM* can be validated (and the *(in)accuracy* quantitatively characterized) through analysis/comparison of real network performance data and simulation data.

**Advantages:**

- A simple and well-defined component-based soft-ware architecture with object oriented paradigm. This facilitates hierarchical modeling of complex systems
- Simulation engine is built in the runtime and is transparent to components.
- It provides Process-oriented modeling and wait () methods, synchronization methods to further ex-tend programming flexibility.
- It can work with both discrete event simulation and real-time process-based simulation.
- It will implement a parallel simulation in the autonomous component architecture.

**Limitations:**

- Java has some security restrictions. So forth JSIM can be prevented from persistence in data.
- Operation of JSIM required clear concept of queuing algorithm.
- The graphical model designer, which has limited capabilities as of now (can only be used to design a model), is intended to be a GUI-based model builder that would do much of the code generation that has to be done manually.

**5. Comparison of Simulators**

MANET simulators exhibit different features and models. The choice of a simulator should be driven by the requirements and the level of details necessary. If high-precision PHY layers are desirable, then ns-2 is the wisest option. On contrary, if the wireless tools has no impact on the targeted protocol, modern simulators (like NAB or Jane) which propose high-level abstractions and polished object-oriented designs will be adapted. These targeted nodes determine the choice of the simulation tool. Sequential simulators should not be anticipated to run more than 1,000 nodes. To end with, most non-commercial simulators suffer from lack of good certification and support. Using a commercial one, may help in case of difficulty. In addition, commercial simulators feature extensive list of supporting protocols, while open source solutions give complete empowerment. These are listed in the table below.

*Table: Comparisons of Various Simulators*

Name	License	Programming Language	Supported Operating System
NS2	Open source	C++, TCL	GNU/Linux, FreeBSD, Mac OS X, Win-dows XP, Windows Vista and Win. 7.
NS3	Open source	C++, Python	GNU/Linux, FreeBSD, Mac OS X, Win-dows XP, Windows Vista & Windows 7.
OPNet	Commercial	C	Windows XP, Vista, 7 & Windows NT 4.0.
OMNET++	Open source	C++	Windows XP or Lat-er, Linux, Mac OS X,
NETSIM	Open source	Java	Windows (7, Vista) and windows XP
QUALNET	Commercial	C++	UNIX, Window- MAC, Linux
JSIM	Open source	JAVA, Tcl	Windows XP, Vista & 7, MAC OS X, Linux.

## 6. Conclusion

In this paper, we have presented a general overview of an open source network simulators for those who are not familiar with it. A brief introduction of Open-Source Technology key concepts, Network Simulation and different open source network simulator. We analyzed and discussed advantages, disadvantages and future scope of Open-Source Technology. Its uses are more efficient for academics, industries, new researchers and student. It is easier for all the users to learn and use, then we introduced four different open source network simulators NS2, NS3, OMNeT++ and J-Sim. We also discussed about current feature, merits & demerits, challenges, future scope of all simulators. Currently NS2 is the best option simulator, for GUI interested OMNeT++, for large network simulation Opnet, Qualnet, Netsim are better than ns2, Jsim is best for wireless sensor.

## References

- 1) [OPNET] OPNET Modeler, <http://www.opnet.com/>
- 2) [NS2] NS2 official website, <http://www.isi.edu/nsnam/ns/>
- 3) [NS3] NS3 official website, <http://www.nsnam.org/documents.html>
- 4) [OMNeT] OMNeT++ official website, <http://www.omnetpp.org/>
- 5) [REAL] REAL 5.0 simulator overview, <http://www.cs.cornell.edu/skeshav/real/overview.html>
- 6) [J-Sim] J-Sim homepage, <http://www.j-sim.org/>
- 7) [QualNet] QualNet official site, <http://www.scalable-networks.com/products/>
- 8) Mohammed Humayun Kabir, Syful Islam, Md. Javed Hossain, Sazzad Hossain, "Detail Comparison of Network Simulators", International Journal of Scientific & Engineering Research, Volume 5, Issue 10, October-2014.
- 9) Jianli Pan, jp10@cse.wustl.edu (A project report written under the guidance of Prof. Raj Jain) A Survey of Network Simulation Tools: Current Status and Future Developments.
- 10) Sujata V. Mallapur, Siddarama . R. Patil, "Survey on Simulation Tools for Mobile Ad-Hoc Networks", IRACST International Journal of Computer Networks and Wireless Communications(IJCNWC), ISSN: 2250-3501 Vol.2, No.2, April 2012.
- 11) Mrs. Saba Siraj, Mr. Ajay Kumar Gupta, Mrs Rinku-Badgujar, "Network Simulation Tools Survey", International Journal of Advanced Research in Computer and Communication Engineering, ISSN : 2278-1021, Vol.1, Issue4, June 2012.
- 12) Luc Hogi , Pascal Bouvry, Frédéric Guinand, "An Over-view of MANETs Simulation", Available at: [www.researchgate.net](http://www.researchgate.net)
- 13) J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. "Comparative Study of Wireless Network Simulator", The Seventh International Conference on Networking, pages 517-523, 2008.
- 14) Mrs. Saba Siraj, Mr. Ajay Kumar Gupta, Mrs Rinku-Badgujar, "Network Simulation Tools Survey", International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, June 2012.

- 15) András Varga, Rudolf Hornig, AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT, SIMUTools, March 03 – 07, 2008, Marseille, France.
- 16) Marek Małowidzki, “Network Simulator: A Developer’s perspective”, Proc. of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS’04), 2004, page no 73-83.
- 17) Camp, T., J. Boleng and V. Davies, A Survey of Mobility Models for Ad Hoc Network Research, Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications (2002), pp. 483–502.
- 18) Cavin, D., Y. Sasson and A. Schiper, On the accuracy of manet simulators, in: POMC ’02: Proceedings of the second ACM international workshop on Principles of mobile computing (2002), pp. 38–43.
- 19) Conti, M., S. Giordano, G. Maselli and G. Turi, Mobileman: Mobile metropolitan ad hoc networks, in: Proceedings of the 8th International IFIP-TC6 Conference, Lecture Notes in Computer Science LNCS 2775, 2003, pp. 194–205.
- 20) 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.