

# Simulation vs. Emulation: Evaluating Mobile Ad Hoc Network Routing Protocols

Furqan Haq and Thomas Kunz  
Systems and Computer Engineering  
Carleton University  
Ottawa, Ont., Canada K1S 5B  
haq\_furqan@hotmail.com, tkunz@sce.carleton.ca

## Abstract

*In order for simulation studies to be useful, it is very important that the simulation results match as closely as possible with the testbed results. This paper compares emulated testbed results with simulation results from NS2 and GloMoSim. OLSR was used as a routing protocol and NRL Mobile Network Emulator (MNE) for dynamic topology control and manipulation. Five Linux based laptops, equipped with IEEE 802.11b wireless network cards were used for testbed implementation. At low traffic rates, testbed results matched closely with the simulation results, at higher traffic rates, testbed results not only differed from the simulation results both qualitatively and quantitatively but the simulation results from both the simulators were barely comparable in some scenarios.*

## Introduction

Network engineers use simulation tools to debug and test the reliability and QoS of network protocols as well as hardware equipment. This makes simulation a very important step towards the deployment of wireless communication networks. A simulation is only useful if the simulation results match as closely as possible with the testbed results.

With the popularity of commercial wireless devices such as IEEE 802.11 and Bluetooth, a transition is taking place from traditional wired networks to wireless networks. In the wireless domain, Mobile Ad-hoc networks are getting more and more popular due to their flexible, dynamic, self-healing and self-configuring nature.

In spite all the technological achievements and cutting edge research occurring in the field of mobile wireless networks, there are growing concerns regarding the reliability of results generated by wireless network simulators.

## Related Work

There are very few papers discussing the accuracy of simulation results and the variation of the results among different wireless network simulators.

An interesting paper published by Distributed Systems Laboratory, Ecole Polytechnique Fédérale de Lausanne, compares the simulation results of OPNET, NS2 and GloMoSim. IEEE 802.11 MAC implementations and a very simple flooding algorithm were used to simulate wireless ad-hoc networks using 50 mobile nodes. According to the paper, there was huge diversion between results from different simulators and in some cases results were barely comparable. The paper also mentioned that there is a scarcity of real experiments that demonstrate the correctness of wireless network simulators [1]. This paper gave us our primary motivation to perform simulations using various simulation tools available and compare the simulation results with the testbed results.

Kunz compared MANET simulation results with testbed results using the BCAST protocol, a multicast routing protocol. According to his findings, in a stationary environment, all results were close to each other, and a mobile emulator introduces negligible additional overhead. However, in mobile scenarios, his measurements in the emulated environment differed both qualitatively and quantitatively from the simulation results. According to the report, tools like MNE (Mobile Network Emulator) can help in testing and debugging protocols for real testbed but it is not clear if these tools are suitable for the performance evaluation of the protocol [2].

Based on the findings of the two above research projects, a need for detailed and carefully planned simulation and emulation experiments was realized. To ensure the reliability of the results, it was decided to use different traffic models and a highly dynamic network topology.

The primary goal of the project was to find out where simulation and emulation yield drastically different results and come up with guidelines for how/whether to use simulation as an appropriate tool for performance studies of protocol.

### OLSR Protocol

The Optimized Link State Routing protocol (RFC 3626) is a proactive routing protocol, which uses a link-state scheme in an optimized manner to diffuse topology information. OLSR message flooding is optimized to run in a multi-hop wireless network to preserve bandwidth. “Optimization is acquired through a technique called *MultiPoint Relaying*. The idea of MultiPoint Relaying (MPR) is to minimize the overhead of flooding messages by restricting the control messages to selected nodes.” [3]

### MNE

MNE is designed to work under Linux-based distribution environments with IPTABLES network filtering capabilities installed. A packet filter looks at the header of packets as they pass through and based on user-defined rules decides whether the packet should be accepted for further processing or should be dropped. Packet filtering is used for security, control and watchfulness. MNE takes advantage of these control features of IPTABLES to block packets from a particular node. The blocking is done based on the wireless link range. In MNE, packets from the nodes that are deemed out of range are blocked with the IPTABLES MAC Filtering component [4].

The control module provides a communication back channel for transmitting the node’s location information. The location update information is constantly transferred to all the nodes in the network using a broadcast channel (also known as back channel or controlled network). Either the wired or the wireless interface can be used to transmit location updates; however transmitting both traffics (dynamic topology information and user traffic) on the same physical channel causes congestion. In our work, the wired interface was used to broadcast location update to avoid congestion on the wireless channel.

MNE supports several different ways of emulating node motion, which includes file, random, ns2, static, line and circle. In MNE, NS2 motion files can be used directly. The MOVE module of MNE provides location information represented in terms of longitude and latitude and this information is then transmitted to all the nodes through the controlled network described above.

Each node in the network broadcasts its location information with the MAC address of the wireless network card over the back channel. The DYNAMICS

module uses this information for calculating the distance between the nodes and performing filtering. The MAC address is used to block packets from a particular node.

### Dynamic Network Topology

Dynamic network topology scenarios were carefully selected to simulate 1 hop, 2 hop and multi-hop connections in different scenarios. An area of 600x600 meters was selected as the basis for our dynamic topology. During the simulation of 400 seconds, the nodes acquired four different topologies.

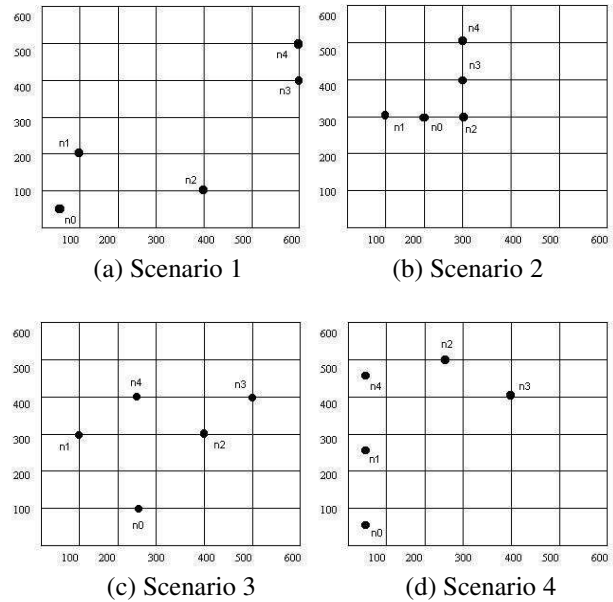


Figure 1: Network Topologies

The above figure shows the orientation of nodes at time  $t = 0, 180, 300$  and 400 seconds. In first scenario at time  $t = 0$ , nodes  $n_0$  and  $n_1$  are within each other’s direct communication range,  $n_4$  and  $n_3$  can also send and receive packets but  $n_2$  is out of range. Shortly after  $t = 0$  sec, all the nodes start moving and at  $t = 180$  sec they acquire positions shown in scenario 2. In scenario 2 node  $n_1$  cannot communicate with  $n_4$  directly and has to go through an intermediate node and vice versa, hence establishing a 2-hop communication network. In this scenario any node ( $n_0, n_2, n_3$ ) can be chosen as a MPR to relay all the packets. Scenario 3 and 4 were designed to simulate multi hop communication.

### Simulation Model

NS2 [5] is a discrete event simulator for, among others, Linux that can be used to define different scenarios. NS2 depends on several external components such as Tcl, otc and TclCL. A simulation is defined by a TCL program and a topology is defined using ns commands. NS2 can be used to simulate wired or wireless network. NS2

includes a model for IEEE 802.11 and the protocol stack includes implementations for ad-hoc routing protocols which makes it an ideal simulator for mobile ad hoc networks.

GloMoSim is a scalable simulation environment for wireless and wired network systems designed by UCLA's parallel computing laboratory [6]. GloMoSim can be used to simulate a variety of wireless network protocols. The protocol stack includes models for the channel, radio, MAC, network, transport, and higher layers.

For both simulators, different research groups have developed OLSR models and made them publicly available. We integrated these models into NS2 and GloMoSim and ensured that we used consistent values for all protocol parameters (such as the periodicity of HELLO messages).

Five mobile nodes were used in the simulation model. The MAC and the Logic Link Layer implemented in NS2 and GloMoSim for IEEE 802.11 was used in the simulation. A Drop Tail Queue of size 100 packets was selected as an interface queue. An omni-direction antenna having unity gain was used. A two-ray ground reflection model was used for radio propagation.

### Traffic Model

CBR agents were used to generate traffic in the network. The following table shows the packet size and number of packets sent per second for all the experiments.

Experiment #	Packet Size	Duration
1	128 bytes	1 packet/sec
2	128 bytes	10 packets/sec
3	512 bytes	10 packets/sec

Table 1: Traffic Rates

Each node's CBR was connected to every other nodes receiver agent hence forming a meshed network. All the links were bi-directional. User Datagram Protocol (UDP) was used as Transport Layer protocol to minimise the overhead of establishing a connection.

### Testbed Implementation

NRL MNE was used for dynamic topology control and manipulation. The Ethernet wired interface was used to broadcast location update to avoid congestion on the wireless channel. An implementation of OLSR (RFC 3626) from the PROTOcol Engineering Advanced Networking (PROTEAN) Research Group was used for the testbed experiments.

Multi-Generator (MGEN version 4) (an open source software from NRL) was used to generate traffic over the wireless channel to obtain packet delivery statistics. MGEN provides the ability to perform IP network performance tests and measurements using UDP/IP traffic. To ensure the reliability of the results, great care was taken while selecting network settings that will match as closely as possible with the settings defined in the network simulators for the simulation model.

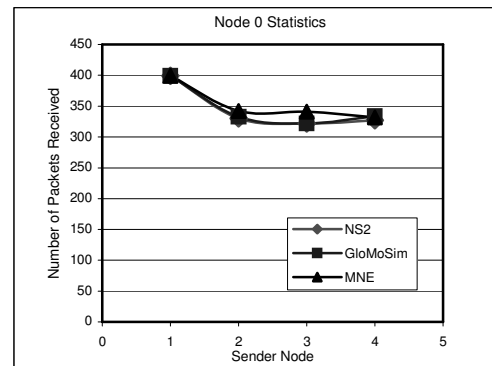
### Performance Comparison

In total three simulation and testbed experiments were performed.

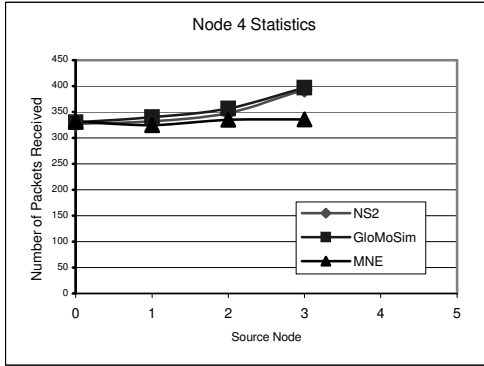
- Low bandwidth (1 packet/s of 128 bytes)
- Medium bandwidth (10 packet/s of 128 bytes)
- High bandwidth (10 packet/s of 512 bytes)

In this experiment we focused on the packet delivery ratio and compared the number of packets received by the emulated testbed nodes in all three scenarios with the simulation results from NS2 and GloMoSim. As the clocks on the laptops were not closely synchronized, we did not measure packet latencies. And the number of control messages propagated in the network was comparable in all experiments because the network topology was unchanged and protocol parameters were all set to the same values.

In the first scenario, the accumulated channel bandwidth utilized by the CBR agents of all the nodes in the network nodes was 20Kb/s (excluding the bandwidth consumed by the OLSR). The maximum bandwidth available was 2Mb/s. Simulation and emulated testbed results of the first scenario were very close to each other. The average percentage difference between the number of packets received by the testbed nodes and the nodes in the simulation model was less than 5%.



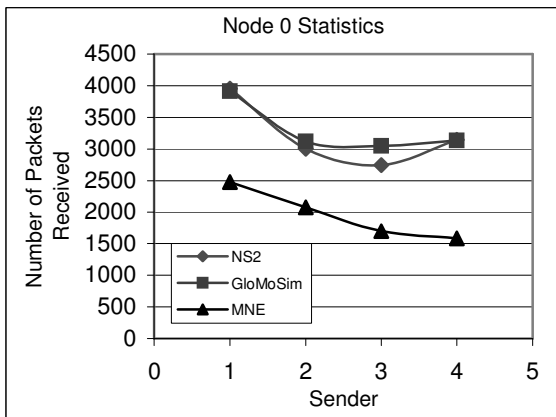
Graph 1: Node 0 Statistics (128 bytes packets 1/sec)



Graph 2: Node 4 Statistics (128 bytes packets 1/sec)

Graphs 1 and 2 show the number of packets received by node 0 and 4 respectively. The trend holds for all the other nodes and simulation results matched emulated testbed results both qualitatively and quantitatively.

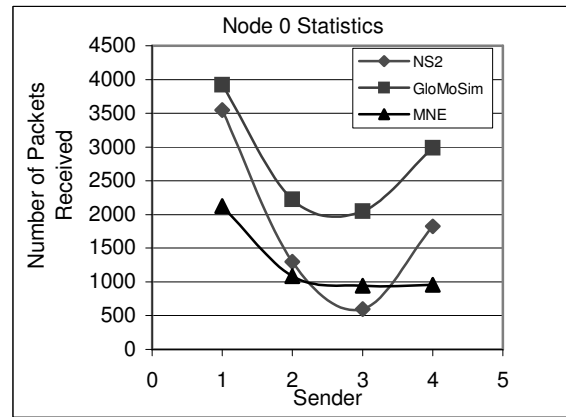
In the second scenario (medium bandwidth), accumulated bandwidth used by the CBR traffic agents was 200Kb/s plus the bandwidth required by the routing protocol. When we increased the number of packets sent per second by the factor of 10, the simulation results differed quantitatively from the emulated testbed results. The percentage difference between the simulation results and testbed results was also increased by a factor of 10 as compared to the results acquired in the first scenario. Hence testbed nodes received 10 times less packets as compare to nodes in the simulation model. The results among different simulators were pretty consistent and the percentage difference was less than 2 percent. Graph 3 shows the packet delivery ration of node 0 in the second scenario. The results are similar for rest of the nodes.



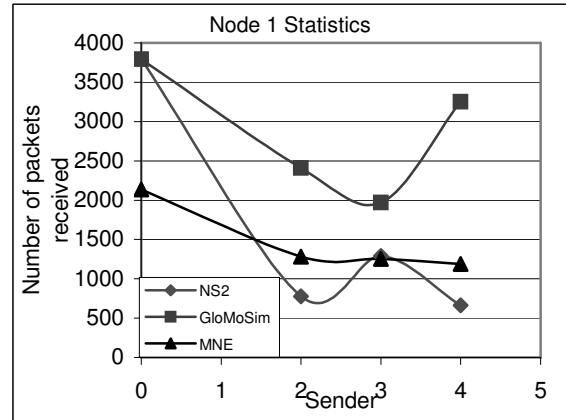
Graph 3: Node 0 Statistics (128 bytes packets 10/sec)

After analyzing the results from the first and second scenario, we decided to increase the size of the packet from 128 bytes to 512 bytes. We wanted to see if there is

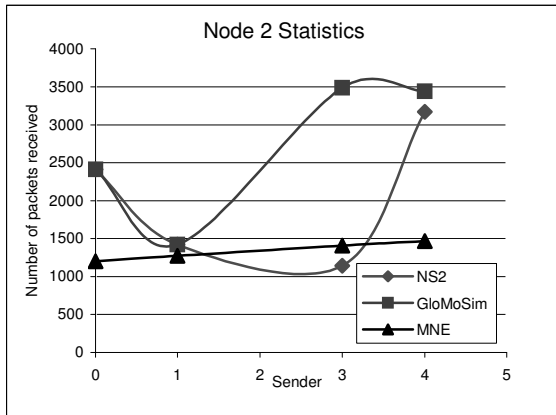
a pattern in the inconsistency of the results between the testbed and simulation results when we increase the traffic load. By increasing the size of the packet by a factor of 4 the accumulated bandwidth used by the nodes in the network reached 800Kbytes/s. We were hoping to see another quantitative shift in the results between testbed and simulation but the results we acquired were somewhat surprising. Testbed results of the third scenario differed both quantitatively and qualitatively from the simulation results. Furthermore the results from NS2 and GloMoSim were barely comparable in some scenarios. However the bandwidth utilized by the nodes was far less than the maximum allowed bandwidth (2Mb/s). Overall, the results were very inconsistent.



Graph 4: Node 0 Statistics (512 bytes packets 10/sec)

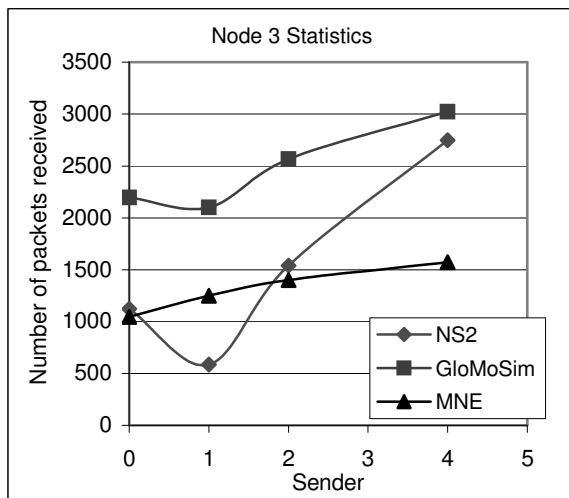


Graph 4: Node 1 Statistics (512 bytes packets 10/sec)



Graph 5: Node 2 Statistics (512 bytes packets 10/sec)

Graphs 4-6 shows the results obtained from the third round of the testbed and simulation experiments. GloMoSim nodes consistently received more packets than NS2 and emulated testbed nodes. In some scenario NS2 received more packets than MNE and vice versa.



Graph 6: Node 3 Statistics (512 bytes packets 10/sec)

## Conclusion

Based on the dynamic topology and traffic model used in this experiment, the performance results from the simulation tools NS2 and GloMoSim and the emulated testbed results differ both qualitatively and quantitatively at higher traffic rates. The variation in the results kept on increasing with an increase in the load in network. However at lower traffic rates (1 packet/s 128bytes) simulation results were very close to testbed results.

Based on the results of this experiment, wireless network simulation tools should be used with great care when evaluating performance of MANET routing protocols. At higher bandwidth wireless network simulators results

may not evaluate the routing protocols correctly. However, lower bandwidth may result in realistic simulation results. The testbed used in this experiment was an emulated testbed (implements using MNE) therefore; the emulated testbed results may differ from the real testbed results. In the emulated environment, wireless nodes were placed close to each other and therefore physically share the wireless channel. In certain simulation scenarios, where the nodes were out of each others range and were able to access the physical channel concurrently, they were unable to access the channel concurrently in our emulated environment. Between NS2 and GloMoSim, results from NS2 were much closer to the testbed results as compared to the GloMoSim, especially at higher bandwidth. Using GloMoSim to evaluate the routing protocol performance did, in general, result in higher packet delivery ratios, compared to both NS2 and the emulated testbed, sometimes by a factor of more than 100%

## Future Work

In this experiment, we only looked at the packet delivery ratio. We did not consider packet latency because in Ad Hoc mode it is difficult to synchronize the clocks. Several algorithms have been proposed to synchronize the clock. Kunz and Carlos proposed a clock-sampling mutual network synchronization algorithm for wireless ad hoc networks [7]. In future work, clock synchronization protocols can be implemented (in NS2, GloMoSim and testbed) to compare the packet latency.

The testbed used in this experiment was an emulated testbed. We are currently working on a real multihop wireless testbed on our campus and plan to perform performance studies on that testbed in the future. These results can then be compared with both the emulated testbed results and the simulator results, varying the traffic model. One disadvantage of a real testbed will be the difficulty in exploring a number of different dynamic topologies, so we would like to determine which tool (NS2, GloMoSim, or emulator) will more accurately predict the actual testbed results. Finally, in this experiment we used OLSR as a routing protocol, future work can compare the testbed results with the simulation results using AODV or DSR protocol. Various traffic models can also be tested such as FTP, TELNET, VoIP.

## Reference

- [1] D. Cavin, Y. Sasson, A Schiper, "On the Accuracy of MANET Simulators", Distributed Systems Laboratory Ecole Polytechnique Fédérale de Lausanne, in *Workshop on Principles of Mobile Computing 2002*, Toulouse, France
- [2] Thomas Kunz, "Implementation vs. simulation: Evaluating a MANET multicast protocol", in *Proceedings of the Global Mobile Congress 2004*, Shanghai, China, pages 129-134, October 2004, Delson Group 2004
- [3] Network Working Group, The Internet Engineering Task Force, Optimized Link State Routing Protocol, January 2005,  
<http://www.ietf.org/rfc/rfc3626.txt?number=3626>
- [4] W. Chao, J. Macker, J. Weston, "Mobile Network Emulator," Communication Systems Branch Information Technology Division Navel Research Laboratories, Washington, DC, USA, January 24, 2003
- [5] Information Sciences Institute USA Viterbi School of Engineering, "The Network Simulator - ns-2," September 2004  
<http://www.isi.edu/nsnam/ns/>
- [6] UCLA Parallel computing laboratory, University of California "About GloMoSim," September 2004  
<http://pcl.cs.ucla.edu/projects/glomosim/>
- [7] Carlos H. Rentel and Thomas Kunz, "A clock-sampling mutual network synchronization algorithm for wireless ad hoc networks", *Proceedings of the IEEE Wireless Communications and Networking Conference 2005*, New Orleans, USA, March 2005