



Network Simulators: Perspective View

Ashok Kumar¹, Arun Kumar²

¹D.A.V.College, Ambala City, Haryana (India)

²DCSA, Kurukshetra University, Kurukshetra, Haryana (India)

ABSTRACT

It is very costly to deploy a complete test bed containing multiple networked computers, routers and data links to validate and verify a certain network protocol or a specific network algorithm in the network research area. Under these circumstances the network simulators saves a lot of money and time in accomplishing this task. Network simulators are also particularly useful in allowing the network designers to change the existing protocols in a controlled and reproducible manner besides testing new networking protocols. The present paper presents a comprehensive survey of current network simulators. It is hoped that the present survey would serve as good reference source in selecting the appropriate network simulators for their research or practical requirements.

Keywords: Network Simulator, GloMoSim/QualNet, OPNET, NS-2, NS-3, OMNeT++

I. INTRODUCTION

As a part of modern technology simulation is of utmost importance. It can be applied to different science, engineering, or other application fields for different purposes. Computer assisted simulation can model hypothetical and real-life objects or activities on a computer which can further be studied to access how the system functions.

Simulation is the imitation of some real object, state of affairs, or process. Simulations are often used for modeling natural, machine or human systems so as to gain insight into their functioning. In addition to this, it is a very important mechanism which helps in understanding interactions between various systems, parts of which may be difficult to recreate or control in the real world. As a part a modern technology, simulations are used widely for testing, performance optimization and measurement, safety engineering besides their application in training and education. In light of the present study simulation is of vital significance, since it involves wireless networks. It is nearly impossible to reproduce the wireless propagation environment, difficult to use real radio-wave based transmission to test our concept, and non-trivial to use real wireless sensor/devices to present our research idea. Use of a simulator can solve all such challenges. For communication networks, Simulators can provide near accurate reproductions of most of the features in the environment, such as noise, probability of loss or alteration of data. Simulators often allow user implementations of protocols for transmission, propagation, reception or other communication aspects to work with their “ether”. Mostly, they provide a data-view model of wireless devices. Using simulators thus allows us to concentrate on the research idea instead of physical implementation of details. They enable us in building a valid model whose behavior and functioning can be repeated and measured, to sum, a statistically verifiable proof of concept. The present paper aims to define the

general requirements sought from a simulator. It discusses the specific characteristics pertaining to simulation of mobile ad-hoc networks. It compares, briefly, the currently existing popular simulator.

II. GENERAL SIMULATOR CHARACTERISTICS AND REQUIREMENTS

In this article a search is on for a simulator that closely replicates the wireless network and that too particularly mobile ad-hoc network system and its components. Before going into the details regarding the selection of a simulator, some characteristics have been presented as:

Basic features:

a) A simulator has to allow defining the characteristics of the system that one aims to model. The simulator must serve the general-purpose, enough for being able to provide the default behaviors of the system e.g. a communication network simulator must have features to model a few transmission media (i.e. propagation media like air, fiber-optics etc).

b) The simulator must be flexible enough to allow the user to define objectives by utilizing what the simulator offers; e.g. often, in communications network research, users create newer protocols, or approaches to a particular challenge. They expect the simulator to provide “pluggable” features which shall enable them to quickly execute the components they have designed using the other available features.

Features may be allowed in a particular format by a simulator. This format is important in the selection of a simulator.

Input-output from the real system: The next important feature is the format of input and output data from the system. Input data are necessary for setting the main parameters of the model e.g. transmission characteristics, inter-arrival rates of traffic, channel loss etc. Output data are necessary to validate functionality, and or calculate the importance of the obtained results. Equally important is the format of data used to specify input or collect output, since creating input data files or rendering and analyzing output results should not become a task of its own.

Design of the experiment and simulation runs: This involves designing the whole experiment. It includes various aspects such as details about each simulation, preferences for data to be collected, simulation runs, time duration of each simulation run, etc. It may also involve finding out what support the simulator has for synchronizing clocks, random number generators and such optional details.

III. POPULAR SIMULATORS

Many simulators exist, some of them are: GloMoSim/QualNet, OPNET, NS-2, NS-3 and OMNeT++.

3.1 GloMoSim/QualNet [1]: This is a public domain simulator developed by UCLA (UCLA Computer Science Department is well known for its research). The authors state that it is being designed using the parallel discrete-event simulation capability provided by Parsec, which is a C-based simulation language, developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete-event simulation models. GloMoSim currently supports protocols for purely wireless networks. It is built using a layered approach that is similar to the OSI seven layers network architecture. Standard APIs (Applied Programming Interface) are used between the different simulation layers, to allow integration of models developed at different layers by different people. GloMoSim has drifted away from creating each of the OSI layers as a separate entity



to representing each node as a single entity, with each layer being represented only by standard APIs to initialize, finalize etc. They claim that this not only allows sharing of memory areas that all OSI layers need to access (viz CPU), but also allows for better performance, scalability and ease of programming use. GloMoSim is thus perceived to be modular, easy to use and flexible, besides maintaining a high degree of detail. GloMoSim needs the Parsec compiler, and coding knowledge of C and Parsec (to a lower extent). Qualnet is the commercial flavor of GloMoSim, and has additional implementations of layers/modules and features like GUI based analysis tools.

3.2.OPNET [2]: It is the registered commercial trademark and the name of product presented by OPNET Technologies incorporation. It became one of the most famous and popular commercial network simulators by the end of 2008. Because of its use for a long time in the industry, it has become mature and has occupied a big market share. OPNET claims to be the fastest simulation engine among leading industry solutions. It has a wide variety of niche simulators for the wired/wireless areas. It also has many of wired/wireless protocol and vendor device models with source code, and allows Object-oriented modeling of components. Modeling environment is a hierarchical one and has a slightly more complex method of definition of nodes as finite state machines. They also have an optional System-in-the-Loop to interface simulations with live systems. The simulator being flexible allows integration with other libraries and simulators. With help from a rich suite of integrated, GUI-based debuggers and analyzers the setup, configuration can be done. OPNET is available at special rates for educational institutions. OPNET inherently has three main functions: modeling, simulating, and analysis. In modeling, it provides intuitive graphical environment to create all kinds of models of protocols for modeling. For simulating, it uses different advanced simulations technologies and can be used to address a wide range of studies. While for analysis, the emulation results and data can be analyzed and displayed very easily for convince of its users. User friendly graphs, charts, statistics, and even animation can be generated by OPNET.

3.3NS-2 [3]: One of the most widely used in the research community, NS-2 has grown via contributions from the research community as well as DARPA (Defense Advanced Research Projects Agency) and is available as freeware. Being an object-oriented discrete event simulator that follows the layered approach, NS-2 is accompanied by a rich set of protocols. Beside this NS-2 is also an emulator, and can talk to real networks. However, to its disadvantage, it has a large footprint, and is not very scalable. NS-2 also ranks low on the flexibility and ease of use fronts. Also, complex the process to include/implement new protocols is complex.

- NS-2 is object-oriented
- It is discrete event driven network simulator
- The programming it uses is C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT).
- NS-2 separates control path implementations from the data path implementation.
- The scenes can be changed easily by programming in the OTcl script. When a user wants to make a new network object, he can either write the new object or assemble a compound object from the existing object library, and plumb the data path through the object. This plumbing makes NS-2 very powerful.
- Another feature of NS-2 is the event scheduler. In NS-2, the event scheduler keeps track of simulation time and release all the events in the event queue by invoking appropriate network components.

3.4 NS-3 [4]: Similar to NS-2, NS-3 is also an open sourced discrete-event network simulator which targets primarily for research and educational use. NS-3 is licensed under the GNU, GPLv2 license, and is available for research and development. NS-3 is designed to replace the current popular NS-2. However, NS-3 is not an updated version of NS-2 since that NS-3 is a new simulator and it is not backward-compatible with NS-2.

The basic idea of NS-3 comes from several different network simulators including NS-2, YANS [6], and GTNetS [7]. The major differences lying between NS-3 and NS-2 include:

- a) Different software core:** The core of NS-3 is written in C++ and with Python scripting interface.
- b) Attention to realism:** protocol entities are designed to be closer to real computers.
- c) Software integration:** support the incorporation of more open-source networking software and reduce the need to rewrite models for simulation.
- d) Support for virtualization:** lightweight virtual machines are used.
- e) Tracing architecture:** NS-3 is developing a tracing and statistics gathering framework trying to enable customization of the output without rebuilding the simulation core.

The NS-3's features are as follows:

- Modular, documented core
- C++ programs and Python scripting
- Alignment with real systems
- Software integration
- Virtualization and testbed integration
- Attribute system
- Updated models

3.5 OMNet++ [5] : Being on account of its an open source, open architecture simulator, the components of OMNET++ are defined by nested hierarchical modules in a simple text based language which is easy to learn, while being very expressive. The behavior of these components can then be elaborated in C++. OMNet++ offers an easy to use GUI for graphical network editing, animation and configuring simulation runs. OMNet++ has a basic output analyzer, which can display collected statistics in graphical formats. It is well documented, and has discussion forums and is scalable too. However, not many OSI/mobility related models are implemented. Nevertheless, its base infrastructure is very extensible, and it is easy to modify. This offsets the lack of implemented models this to a certain extent. Thus, OMNet++ is perceived to be a better choice when a lot of customization or development is expected. The models or modules of OMNet++ are assembled from reusable components as OMNet++ is designed to provide a component-based architecture. Modules are reusable and can be combined in various ways which is one of the main features of OMNet++.

The OMNet++ components [5] include:

- Simulation kernel library
- Compiler for the NED topology description language (nedc)
- Graphical network editor for NED files (GNED)
- GUI for simulation execution, links into simulation executable (Tkenv)
- Command-line user interface for simulation execution (Cmdenv)

- Graphical output vector plotting tool (Plove)
- Graphical output scalars visualization tool (Scalars)
- Model documentation tool (opp_neddoc)
- Utilities (random number seed generation tool, makefile creation tool etc.)
- Documentation, sample simulations, etc.

IV. CONCLUSION

In this paper, a comparative study has been done in which the wired/wireless network simulators MANETs GloMoSim/QualNet, OPNET, NS-2, NS-3 and OMNeT++ have been studied. Each of them also showed areas of relative weakness compared to the other candidates. OMNeT++ and NS-2 are the most mature ones. While OMNeT++ shined at GUI support, NS-2 profits from the large number of available models. Though, both support productivity. Selection of simulator is more of research area oriented and many times user specific. Simulators are tools for researchers and they use them as per their own convenience and availability.

V. ACKNOWLEDGMENTS:

We wish to express our heartiest thanks to Dr. Ashwani Kush, Associate Professor & Head department of Computer Science, University College, Kurukshetra University, Kurukshetra for his valuable suggestions in the preparation of this manuscript.

--

REFERENCES

- [1] Available at QualNet official site, <http://www.scalable-networks.com/products/>
- [2] OPNET Modeler, available at <http://www.opnet.com/>
- [3] NS2 official website, available at <http://www.isi.edu/nsnam/ns/>
- [4] NS3 official website, available at <http://www.nsnam.org/documents.html>
- [5] OMNeT++ official website, available at <http://www.omnetpp.org/>
- [6] Yet another Network Simulator, available at <http://yans.inria.fr/code/yans/?summary>
- [7] The Georgia Tech Network Simulator (GTNetS), available at <http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/>
- [8] D. Bertsekas and R. Gallager, "Data Networks" Prentice Hall Publ., New Jersey, 2002.



Traffic generation	Radio propagation Models	Mobility models	Experiment setup	Input/output definition	Topology definition Language/Model	Component/topology Definition	Internal architecture
Unknown	Yes	Yes	Flat files (GUI in Qual-Net)	Flat files	Parsec, C	Skeleton Interface files	Layered
Available	Yes	Yes	GUI customizable	GUI based editor, Proto-C	Proto-C, OO models	Finite State Models	Hierarchical
Available	Yes	Yes	OTcl interfaces and script files	OTcl based files	C++ and OTcl	OTcl object model	Layered
					C++ and Python scripting		
Not available	Yes, basic	Yes, basic	Flat files, GUI	Flat files	Flat files, C++	Text files	Hierarchical, nested

Factor	Scientific acceptance	Licensing	Scalability	Extensibility	Modifiability	Usability	Traffic profiling
Glomosim/QualNet	Good	Free for universities (QualNet commercial)	Yes	Yes	Not so easy	Moderate. Less documentation (Qualnet is better)	Java-based trace, monitoring and statistics
Opnet	Reasonable	Commercial (discounted for universities)	Yes	Yes	Moderate	Good	Available, GUI based
NS-2	Very good	Public domain	Not good	Not easy	Complex	Sparse GUI, tough to debug, large footprint	Tcl/Tk based analyzer for Trace files
NS3	Very good	Public domain	good	Yes	Good		
OMNet++	Reasonable	Public domain	good	Easy	Good	Good	GUI based basic inspectors