We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Network and System Simulation Tools for Next Generation Networks: A Case Study[1]

S. Mehta, Mst. Najnin Sulatan, H.Kabir, N.Ullah and K. S. Kwak
Inha University
Korea

1. Introduction

The modern information society will continue to emerge, and demand for wireless communication services will grow. Future generation wireless networks are considered necessary for the support of emerging services with their increasing requirements. Future generation wireless networks are characterized by a distributed, dynamic, self-organizing architecture (I. F. Akyildiz et al., 2006). These wireless networks are broadly categorized into different wireless networks according to their specific characteristics. Typical examples include Ad-Hoc/Mesh Networks, Sensor Networks, Cognitive Radio Networks, etc as shown in figure 1. These wireless networks could then constitute the infrastructure of numerous applications such as emergency and health-care systems, military, gaming, advertisements, customer-to-customer applications, etc. Not only their importance in military applications is growing, but also their impact on business is increasing. The emergence of these wireless networks created many open issues in network design too. More and more researchers are putting their efforts in designing the future generation wireless networks.
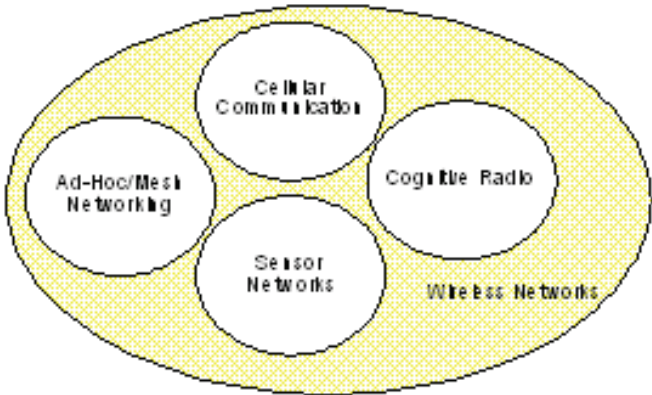

Fig. 1. Different kinds of wireless networks

---

[1] Some part of this chapter was published at AMS'09, Indonesia (S.Mehta, et. al, 2009).

There are three main traditional techniques for analyzing the performance of wired and wireless networks; analytical methods, computer simulation, and physical measurement or a testbed measurement. Traditionally, formal modeling of systems has been via a mathematical model, which attempts to find analytical solutions to problems and thereby enable the prediction of the behavior of the system from a set of parameters and initial conditions. However, it is widely known that comprehensive models for wireless ad hoc networks are mathematically intractable. On its own, each individual layer of the protocol stack may be complex enough to discourage attempts at mathematical analysis. Interactions between layers in the protocol stack magnify this complexity. The construction of real testbeds for any predefined scenario is usually an expensive or even impossible task, if factors like mobility, testing area, etc. come into account. Additionally, most measurements are not repeatable and require a high effort.

Simulation is, therefore, the most common approach to developing and testing new protocol for a wireless network. Simulation has proven to be a valuable tool in many areas where analytical methods aren't applicable and experimentation isn't feasible. Researchers generally use simulation to analyze system performance prior to physical design or to compare multiple alternatives over a wide range of conditions. In context with networks, and especially wireless networks, simulators are used for the development and validation of new algorithms, such as routing algorithms in wireless networks, or protocols. Improvements of existing algorithms, as well as testing a networks capacity and efficiency under specific scenarios is also a simulators task. Many publications typically include performance simulations and commonly compare routing protocols. Simulators model the real world in a specific way. Their purpose is to ease the understanding of it, to surge its behavior and especially research its reactions on particular events. There are a number of advantages to this approach: lower cost, ease of implementation, and practicality of testing large-scale networks (J. Lessmann et. al, 2008).

The goal of simulators is to achieve an "as real as possible" situation in order to make the simulation results realistic and therefore adaptable. Because it is impossible to collect and implement all the data and details playing a role within the real world, the simulators have to be trimmed. Now, the main difficulty is where to start cutting off details and where to end with it while dealing with simulation. The correct level of detail decides whether a simulation is useful or not, and therefore a difficult part in the development process. While less details in simulation could produce results which are deluding or in some cases even false, the effects of too many details can also make the simulation useless: Necessarily the implementation is more time- consuming and the simulation takes longer. When it comes to wireless network simulation, three main points are important: Firstly, the algorithms and protocols should be error free and have to be implemented in adequate detail, and secondly, the simulation environment, such as mobility schemes, must be realistic. Finally, a proper method is needed to analyze the collected data. Even though simulation is a powerful tool, it is still occupied with potential pitfalls (J. Heidemann et. al, 2001). To help overcome this, it is important to know the different tools available and their benefit and drawbacks there in associated. The goal of this paper is to give an over all short review to simulation system, especially discussion about commonly used simulation tools in system and network, and a cautionary guideline to avoid the pitfall associated with simulation for all who are using or will be using simulation tools for their research.
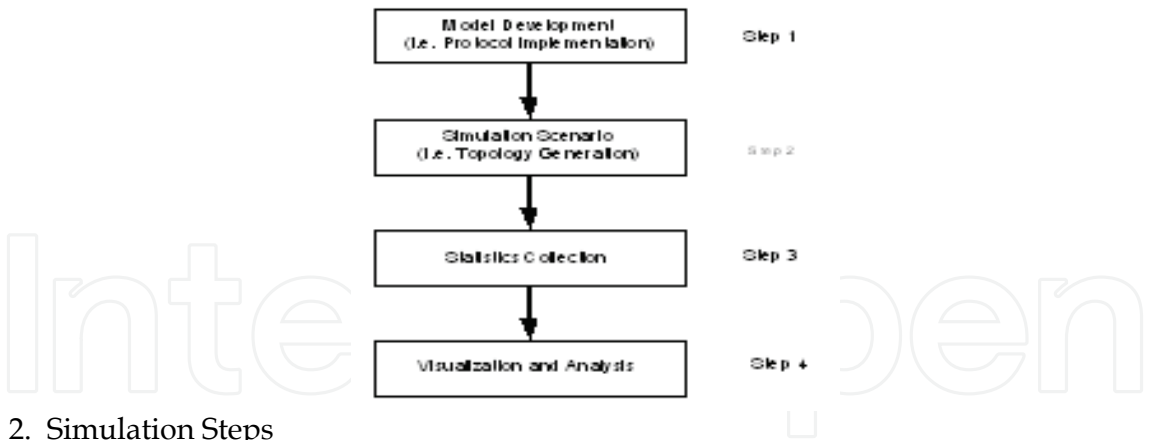
Fig. 2. Simulation Steps

There are four basic steps to run a simulation as shown in figure 2. First step is to develop a model (e.g. implementation of a protocol); second step is to create a simulation scenario (e.g. designing a network topology and traffic scenario); third step is to choose and collection of statistics, and finally fourth step is to visualize and analysis of simulation results which may be carried out after (or during, in some cases) the simulation execution. The problem with such approach is that one cannot guess in advance how many replications is needed for securing small errors of estimates, and if the errors are found to be too large, simulations need to be repeated. This is referred to as offline sequential analysis of simulation output data. Of course, this is not a very efficient way of data analysis. It is generally required that final results from any simulation are to analyze output data on-line, during simulation. Then, the simulation can be stopped when the statistical errors of the estimates become sufficiently small.
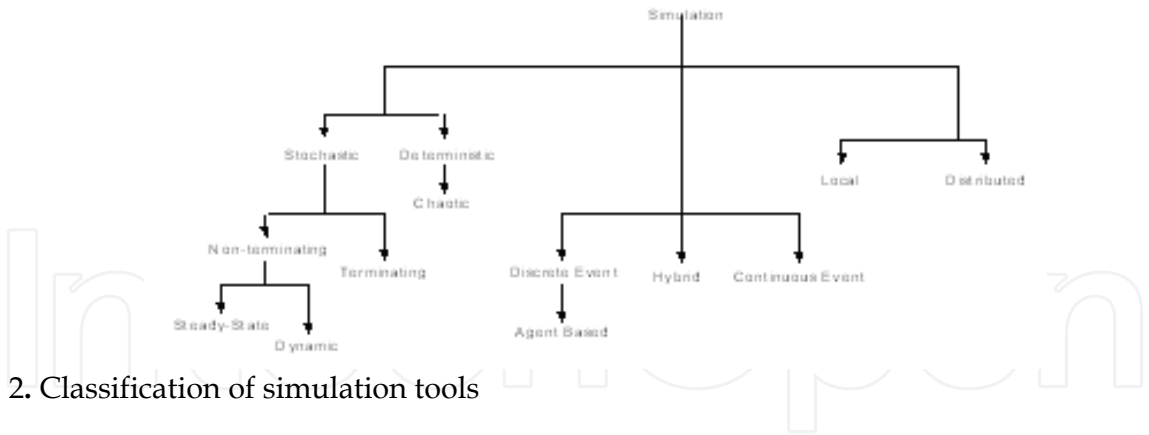

Fig. 2. Classification of simulation tools

Before going further, we present the classification of simulation tools which could be a good summary for those with little previous exposure to the topic (S. M. Sanchez, 1999). As depicted in figure 3 simulation tools can be classified according to several criteria including:

- Stochastic or deterministic
- Steady State or dynamic
- Terminating or non terminating
- Discrete or continuous or hybrid
- Local or distributed

Stochastic simulation: Most of the realistic simulation tools The world is full of uncertainty, and most (if not all) realistic simulation models will incorporate some randomness as well as some element of time elapsing. Such tools can be used to examine a diverse set of applications. For example, the simulation may have been designed to model the operation of a customer service center, traffic patterns over a particular location grid, hospital facilities utilization, waiting times for customers arriving at a service center, the number of cars passing through an intersection during a 5 minute period, the efficacy of various strategies in combat warfare, the impact of changes in layout and equipment on production throughput, and more.

Deterministic simulation: Deterministic simulations use fixed, non-random values to specify the model and particular variant of the system under investigation. Because there is no randomness, the output is also fixed for any specific set of inputs. Chaotic model is the special case of deterministic model.

Non-terminating simulation: In a non-terminating system, the duration of the system is not finite. The Internet exemplifies a non-terminating system. Non-terminating simulations are used to simulate non-terminating systems. In a non-terminating simulation, there is no event to signal the end of a simulation, and such simulations are typically used to investigate the long-term behavior of a system. Non-terminating simulations must, of course, stop at some point, and it is a non-trivial problem to determine the proper duration of a non-terminating simulation. If the behavior of the system becomes fairly stable at some point, then there are techniques for analyzing the steady-state behavior of the system using non-terminating simulations.

Terminating simulation: Terminating systems have fixed starting condition and a naturally occurring event that marks the end of the system. An example of a terminating system is a work day that starts at 9 am and ends at 6 pm at office. For terminating systems the initial conditions of the system generally affect the desired measures of performance. The purpose of simulating terminating systems is to understand transient behaviour of the system.

Steady-State simulation: Steady-state models use equations defining the relationships between elements of the modeled system and attempt to find a state in which the system is in equilibrium. Such models are often used in simulating physical systems, as a simpler modeling case before dynamic simulation is attempted.

Dynamic simulation: Dynamic simulations model changes in a system in response to (usually changing) input signals.

Discrete event simulation: A discrete event simulation manages events in time. Most computer, logic-test and fault-tree simulations are of this type. In this type of simulation, the simulator maintains a queue of events sorted by the simulated time they should occur. The simulator reads the queue and triggers new events as each event is processed. It is not important to execute the simulation in real time. It's often more important to be able to access the data produced by the simulation, to discover logic defects in the design, or the sequence of events. Most of the network simulation tools fall under this category.

Agent-Based Simulators: This is a special class of discrete event simulator in which the mobile entities are known as agents. Whereas in a traditional discrete event model the entities only have attributes, agents have both attributes and methods (e.g., rules for interacting with other agents). An agent-based model could, for example, simulate the behavior of a population of animals that are interacting with each other.

Continuous Simulators: These Simulators are opposite in nature to discrete simulators. This class of tools solves differential equations that describe the evolution of a system using continuous equations. These types of simulators are most appropriate for any information or material which can be described as evolving or moving smoothly and continuously, rather than in infrequent discrete steps or packets. For example, simulation of the movement of water through a series of reservoirs and pipes can most appropriately be represented using a continuous simulator.

Hybrid Simulators: These tools combine the features of continuous simulators and discrete simulators. That is, they solve differential equations, but can superimpose discrete events on the continuously varying system.

Distributed simulator: Distributed models run on a network of interconnected computers, possibly through the Internet. Simulations dispersed across multiple host computers like this are often referred to as distributed simulations.

Local simulator: Local simulator models run on an individual machine or within an interconnected cluster.

2. Related Work

There are several surveys, comparisons, and also some case studies about wireless network and system simulators. They all differ with respect to the selection of evaluated simulators, the intention of the work, the focus of the potential comparison and the level of detail. Table 1 summarizes the previous related works.

Reference	Type of Study	Simulator Tools	Scope of Study
(B. Schilling, 05')	Comparison	Opnet, ns-2	Initialization, accuracy
(S. Duflos et. al, 06')	Comparison	Opnet,ns-2, QualNet, OMNeT++, JSim, SSFNet	For critical infrastructure
(M.Karl, 05')	Comparison	ns-2, TOSSIM	Models, visualization, architecture, components
(L. Hogie et al, 05')	Description	GloMoSim, ns-2, DIANEmu, GTNetS, J-Sim, Jane, NAB, PDNS, OMNeT++, Opnet, QualNet, SWANS	Overview
(E.E.lopez et al, 06')	Comparison	SSF, SWANS, J-Sim, NCTUns, ns-2, OMNeT++, Ptolemy, ATEMU, Em-Star, SNAP, TOSSIM	Models, type of visualization
(D.Curren, 07')	Description	ns-2, GloMoSim, Opnet, SensorSim, J-Sim, Sense, OMNeT++, Sidh, Sens, TOSSIM, ATEMU, Avrora, EmStar	Overview
(L.Begg et al, 06')	Comparison	Opnet, ns-2, OMNeT++, SSFNet, QualNet, J-Sim, Totem	availability/credibility of models, usability
(G.F. Lucio et al, 03')	Case Study	Opnet, ns-2, testbed	Accuracy of results
(K. Pawlikowsk, 02')	Survey	In general simulation study	Credibility , accuracy

Table 1. Related works on Simulator comparison

All of the works listed in table 1 consider different simulators or differ in their aim from this paper. The works parented in (S. Duflos et. al, 06', L. Hogie et al, 05', E.E.lopez et al, 06', D.Curren, 07', L.Begg et al, 06', K. Pawlikowsk, 02') are the close to our work as they include some common simulators J-Sim, OMNeT++, and ns-2, which we also consider for our study. However, (S. Duflos et. al, 2006') examines their suitability for simulating the failure of critical infrastructures like electricity or telecommunication networks. This is very unrelated to what we present here. A huge list of simulators is presented in (L. Hogie et al, 2005, D.Curren, 2007)however, they do not give a comparative study. Rather, their works consists of more or less description of each simulator tools independently. In (E.E.lopez et al, 2006)authors give an overview about the different issues in wireless networks on a general basis. Only at the end of their work they presented a table comparing the considered simulation tools according to different features such as their language, the available modules, and GUI support, etc. the most detailed comparison is presented in (L.Begg et al, 2006). However, they consider all the simulators from an industrial research point of view, which are less relevant for academic researchers. They also miss several practical issues regarding the credibility and reliability of the tools. In (K. Pawlikowsk, 2002) authors presented a survey study of more then 2200 research papers in the field of network simulation studies and point out several systematic flaws in that. We follow the similar kind of work line of (K. Pawlikowsk, 2002) but with different aims. Our goal in this paper is to make a basic contribution to the wireless network community by a) Giving overall short overview of some widely used system and network simulation tools, b) comparing simulation tools on the basis of several features and a survey report of more then 800 research papers in the field of system and networks in recent years (2000~2008), c) listing our recommendations for the designers of protocols, models, and simulators.

The remainder of this paper is organized as follows. In Section 3, we provide a brief overview on "widely used" network and system simulation tools, and their comparisons and results from our survey. Finally, conclusions are presented in section 4.

3. System and Network Simulation Tools

For network protocol designers, it is often difficult to decide which simulator to choose for a particular task, especially for NGNs. Therefore, we conduct a survey to find a wireless system/network simulator that provides a good balance between availability of ready to use models, scripting and language support, extendibility, graphical support, easiness of use, etc. The survey is based on a collection of a number of criteria including published results, interesting characteristics and features. From our survey results, we broadly categories system and network simulators as: "Widely Used" simulators and "Other" simulators. We discuss more about these two categories in the later sections of this paper. The network simulators taken into consideration as "Widely Used"are Ns-2, GloMoSim, J-Sim, OMNet++, OPNet, and QualNet. While for physical layer[2,] very few simulator tools are used. The theoretical, numerical, statistical analyses are vastly used for physical layer. As a simulator, NS2, TOSSIM, GloMoSim, Qualnet, OPNET, OMNET++ are used for the implementation of the error model at the physical layer (M. Lacage et al, 2006). However,

---

[2] In rest of the paper we keep using terms "Physical Layer and "System" interchangeably, unless and otherwise specified.

none of them appears to implement sub-channels or to finely model an explicit packet detection and timing synchronization phase (G. F. Riley, 2003). MATLAB and Monte Carlo based simulations are "Widely Used" simulators in case of system. In this section, we will give a short overview and a comparative study about the eight "widely used" network and system simulators, respectively.

Ns-2: Ns2 is a discrete event simulator for networks. It began as ns (Network Simulator) in 1989 with the purpose of general network simulation. The core of core of the simulator and most of the network protocol models are written in C ++, and the rest is in OTcl. In general, C++ is used for implementing protocols and extending the ns-2 library. OTcl is used to create and control the simulation environment itself, including the selection of output data. Simulation is run at the packet level, allowing for detailed results. Ns2 provides OSI layers excluding presentation and session layers. It has a huge pool of available features, offering a large number of external protocols already implemented. Ns-2 does not scale well for sensor networks. This is in part due to its object-oriented design. While this is beneficial in terms of extensibility and organization, it is a hindrance on performance in environments with large numbers of nodes. Another drawback to ns-2 is the lack of customization available. Packet formats, energy models, MAC protocols, and the sensing hardware models all differ from those found in most wireless devices (Ns2, web link).

GloMoSim: GloMoSim was developed in 1986 fro mobile wireless networks at UCLA (California, USA). GloMoSim is written in Persec, which is an extension of C for parallel programming. New protocols and modules for GloMoSim must be written in Parsec too. GloMoSim respects the OSI standard. The ability to use GloMoSim in a parallel environment distinguishes it from most other wireless network simulators. Like ns-2, GloMoSim is designed to be extensible, with all protocols implemented as modules in the GloMoSim library. GloMoSim still contains a number of problems. While effective for simulating IP networks, it is not capable of simulating any other type of network. This effectively ensures that many wireless networks can not be simulated accurately. Additionally, GloMoSim does not support phenomena occurring outside of the simulation environment, all events must be generated from another node in the network. Finally, GloMoSim stopped releasing updates in 2000. Instead, it is now updated as a commercial product called QualNet (Glomosim, web link).

J-SIM: J-Sim is a general purpose java based simulator developed by a team at the Distributed Realtime Computing Laboratory (DECL) of the Ohio State University. It is built according to the component-based software paradigm and written in Java. Everything in J-Sim is a component: a node, a link, a protocol. Each component can be atomic or composed of other components. Connection between components is done through ports. Actually, there are three possible ways to connect ports: one-to-one, one-to-many, and many-to-many. On a more abstract level, J-Sim distinguishes two layers. The lower layer Core Service Layer (CSL) comprises every OSI layer from network to physical, the higher layer comprises the remaining OSI layers. Initially designed for wired network simulation, its Wireless extension proposes an implementation of the IEEE 802.11 MAC—which is the only MAC supported so far. This extension turns J-Sim to a viable MANETs simulator. J-Sim also features a set of components which facilitates basic studies of wireless/mobile networks, including three distinct radio propagation models and two stochastic mobility models. J-Sim works on any operating system that turns Sun's Java SDK 1.5 or later and is open source (J-Sim, web link).

OMNet++ : OMNet++ (Object Modular network Testbed in C++) is well designed discrete event simulation environment written in C++. OMNET++ is actually a general-purpose simulator capable of simulating any system composed of devices interacting with each others. The mobility extension for OMNeT++ is intended to support wireless and mobile simulations within OMNeT++. This support is said to be fairly incomplete. OMNet++ is for academic and educational use. Modules are connected in a hierarchical nested fashion, where each module can contain several other modules. Modules can be defined as being either simple or compound. Simple modules are used to define algorithms, and make up the bottom of the hierarchy. Compound modules are a collection of simple modules that interact with one another, using messages. OMNeT++ provides a component-based, hierarchical, modular and extensible architecture. Components, or modules, are programmed in C++ and new ones are developed using the C++ class library which consists of the simulation kernel and utility classes for random number generation, statistics collection, topology discovery etc. OMNeT++ has a number of advantages over the other simulators. OMNeT++ accurately models most hardware and includes the modeling of physical phenomena. All layers of the protocol stack can be modified. Despite its apparent advantages, OMNeT++ has remained relatively obscure. The original implementation does not offer a great variety of protocols, and very few have been implemented, leaving users with significant background work if they want to test their own protocol in different environments. OMNet++ works on Linux, Unix-like systems and windows XP/2K (OMnet, web link).

OPNet: OPNet (Optimized Network Engineering Tools) Modeler is a discrete-event network simulator first proposed by MIT in 1986 and is written in C++. It is a well established and professional commercial suite for network simulation. It is actually the most widely used commercial simulation environment. However, it can be used free of charge by researchers applying to University Program of the product. Unlike ns-2 and GloMoSim, OPNET supports the use of modeling different network-specific hardware, such as physical-link transceivers and antennas. OPNET Modeler features an interactive development environment allowing the design and study of networks, devices, protocols, and applications. For this, an extensive list of protocols is supported. Particularly, MAC protocols include IEEE 802.11a/b/g and Bluetooth ones.OPNET can also be used to define custom packet formats. The simulator aids users in developing the various models through a graphical interface. The interface can also be used to model, graph, and animate the resulting output. One of the most interesting features of OPNet is its ability to execute and monitor several scenarios in a concurrent manner. However, OPNET also suffers from the same object-oriented scalability problems as ns-2. OPNet modeler runs on Windows XP/2K, Linux and Solaris platforms (Opnet, web link).

QualNet: QualNet network simulation software has been developed and marketed by Scalable Network technologies.. It is a commercial ad hoc network simulator based on the GloMoSim. It provides a comprehensive set of tools with many components for custom network modeling and simulation. Models in source code form provide developers with a solid foundation from which to build new functionality or to modify exiting functionalities. QualNet does have a range of wired as well as wireless models but its main strength is in the wireless area. QualNet also largely extends the set of models and protocols supported by the initial GloMoSim distribution. As it is built on top of GloMoSim, QualNet is written in Parsec (Qualnet, web link).

MATLAB: MATLAB is an interactive software environment and programming language from The MathWorks which has been founded in 1984. MATLAB was written in C. It supports cross-platform operating system. It is used to make measurements, analyze and visualize data, generate arbitrary waveforms, control instruments, and build test systems. It provides tools and command-line functions for data analysis tasks such as signal processing, signal modulation, digital filtering, and curve fitting. MATLAB and companion toolboxes provide engineers, scientists, mathematicians, and educators with an environment for technical computing applications. With MATLAB and Simulink, one can (a) develop digital signal processing (DSP) algorithms, (b) model and simulate systems, (c) automatically generate code for embedded DSPs, MCUs, GPPs, FPGAs, and ASICs, and (d) verify and validate the hardware and software implementations. But the drawback of this package is to deploy MATLAB functions as library files, which can be used with .NET or Java application building environment it is needed that the computer where the application has to be deployed needs MCR (MATLAB Component Runtime) for the MATLAB files to function normally. Another drawback is that M-code written for a specific release of MATLAB often does not run with earlier releases as it may use some of the newer features (Matlab-Mathwork, web link).

Monte Carlo based Simulatiors[3]: This method solves a problem by generating suitable random numbers and observing that fraction of the numbers obeying some property or properties. The name and the systematic development of Monte Carlo methods date from about 1944. Monte Carlo simulation methods are especially useful for modeling phenomena with significant uncertainty in inputs and in studying systems with a large number of coupled degrees of freedom. The method is useful for obtaining numerical solutions to problems which are too complicated to solve analytically. In the science and engineering communities, Monte Carlo simulation is often used for uncertainty analysis, optimization, and reliability-based design. But it avoids higher order statistics of the output sequences. So, approximate method of it is the main disadvantage of Monte Carlo method. By increasing the number of iterations by the costs of simulation time, any degree of precision can be easily achieved. Another limitation is the number of random numbers that can be produced by random number generating algorithm. To use it one can develop codes in MATLAB or C/C++ or Visual Basic or Java. The popularity of Monte Carlo methods has led to a number of superb commercial tools (Monte Carlo Simulation, web link).

3.1 Comparison

In this sub section we summarize the most interesting capabilities, advantages, and drawbacks of existing tools for wireless networks in table 2. Table 2 has all simulators considered in the previous section listed in the consecutive columns and special features/capabilities in the context of all simulators in the consecutive rows, respectively.

Sr.N.	Tools Features	NS2	GloMo-Sim	J-Sim	OMNet++	OPNet	QualNet	MATLAB
1	Applicability	Net./Sys.	Net./Sys.	Network	Net./Sys.	Net./Sys	Net./Sys.	System

---

[3] "Monet Carlo based simulation" is representing a category of simulators rather then an individual simulation tool.

2	Interface	C++/OTcl	Parsec (C-Based)	Java/Jacl	C++/NED	C or C++	Parsec (C-Based)	C++
3	Available Modules	T/W/Ad/WSNA	T/W/Ad	T/W/Ad/WSNA	T/W/Ad	T/W/Ad/WSN	T/W/Ad/WSNA	Data Acquisition Toolbox, Instrument Control Toolbox , Image Acquisition Toolbox
4	Mobility	Support	Support	Support	No	Support	Support	Support
5	Graphical Support	No or very limited visual aid	Limited Visual aid	Good visualization and debug facility	Good visualization and excellent facility for debug	Excellent graphical support, Excellent facility for debug.	Good graphical support, Excellent for debug.	Excellent graphical support, Excellent facility for debug.
6	Parallelism	No	SMP/Beowulf	RMI-based	MPI/PVM	Yes	SMP/Beowulf	Yes
7	License	Open Source	Open Source	Open Source	Free for academic and educational use	Free academic License for limited use	Commercial	Commercial
8	Scalability*	Small	Large	Small	Large	Medium	Very Large	Very Large
9	Documentation and user support	Excellent	Poor	Poor	Good	Excellent	Good	Excellent
10	Extendibility*	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent
11	Emulation	Limited	Not Direct	Yes	Limited	Not Direct	Yes	Yes

T: Traditional Models (eg. TCP/IP, Ethernet)
W: Wireless Support (eg. Propagation model, IEEE 802.11)
Ad: Ad-Hoc Support (eg. AODV, DSR)
WSN: Wireless Sensor Networks Support (eg. S-MAC, Direct Diffusion)
WSNA: Advance Wireless Sensor Networks Support (eg. Zigbee, Energy Model)
  *Concerning table 2, as no exact metrics are available for scalability and extendibility, we define Very Large > Large> Medium > Small, and Excellent > Good > Poor, respectively.

Table 2. Comparison of Different Network and System Simulation Tools

Wireless Networks simulators exhibit different features and models. Each has advantages and disadvantages, and each is appropriate in different situations. In choosing a simulator from the available tools, the choice of a simulator should be driven by the requirements. Developers must consider the pros and cons of different programming languages, the means in which simulation is driven (event vs. time based), component-based or objectoriented architecture, the level of complexity of the simulator, features to include and not include, use of parallel execution, ability to interact with real nodes, and other design choices. While design language choices are outside of the scope of this paper, there are some guidelines that appear upon looking at a number of already existing simulators. Most

simulators use a discrete event engine for efficiency. Component-based architectures scale significantly better than object-oriented architectures, but may be more difficult to implement in a modularized way.

Defining each wired/wireless node as its own object ensures independence amongst the nodes. The ease of swapping in new algorithms for different protocols also appears to be easier in object-oriented designs. However, with careful programming, component based architectures perform better and are more effective. Generally, the level of complexity built into the simulator has a lot to do with the goals of the developers and the time constraints imposed. Using a simple MAC protocol may suffice in most instances, and only providing one saves significant amounts of time. If high-precision PHY layers are needed, then ns-2 (coupled with the highly-accurate PHY) is clearly the wisest choice. The number of nodes targeted also determines the choice of the simulation tool. Sequential simulators should not be expected to run more that 1,000 nodes. If larger scales are needed, then parallel simulators are a wise choice. Finally, most non-commercial simulators suffer from a lack of good documentation (NS2 is an exceptional case here) and support. Using a commercial one might help in case of troubles. Moreover, commercial simulators usually feature extensive lists of supported protocols, while open source solutions give full empowerment.

3.2 Analysis

In the previous section we provide the background on a number of different network simulators and present the comparison of some important features of each. In continuation of our research work, we present our survey results to take up on the credibility issues of simulation studies in wireless networks, and to alert the researchers on some common simulation issues and pitfalls. We conducted a survey on wireless networks, especially on Ad-hoc/Mesh/Sensor/Cognitive Radio networks studies published in some of the premiere conferences of the wireless networks from years 2000 to 2008. Table 3 lists the name of all conferences that we considered in our survey. We only included the full papers on PHY, MAC and Routing layers in our survey, not the poster and demonstration papers. We reviewed each paper individually avoiding word searches or other means of automatically gathering results. For consistency, the same person reviewed all of the papers; to validate the results and to correct the few inconsistencies we had a second person review all of the papers again.

Sr.No.	Confernce Name	Applicable Area	Average Acceptance Ratio*	Specialized Area	Years
1	ACM MobiCom	Network	≈13%	Ad-hoc/Mesh/Sensor Networks Tracks	2000~2008
2	ACM/IEEE MobiHoc	Network/System	≈15%		2000~2008
3	ACM Sigcomm	Network/System	≈15%		2000~2008
4	IEEE Infocom	Network	≈20%		2000~2008
5	IEEE Percom	Network	≈13%		2003~2008
6	IEEE GlobeCom	System	≈40%		2000~2008
7	IEEE WCNC	System	≈42%		2000~2008
8	IEEE ICC	System	≈34%		2007~2008
9	ACM SenSys	Networks/System	≈17%	Sensor Networks	2003~2008
10	EWSN	Networks/System	≈16%		2004~2008
11	IEEE CrownCom	Networks/System	≈35%	Cognitive Radio Track	2006~2008

12	IEEE DySpan	Networks/System	≈25%		2005~2008
13	IEEE CogART	Network	N/A		2008
14	IEEE MilCom	Networks/System	N/A		2005~2008
15	Mobile Networks and Applications	System	≈18%	Networks and Applications	2000~2008
16	MASCOTS	System	≈33%	Measurement, modelling and performance analysis of computer systems and communication networks	2000~2008
17	ACM SigMobile	System	≈27%	Mobility of systems, users, data, and computing	2000~2008

*Average Acceptance ratio is calculated over mentioned years, unless otherwise specified.
N/A: we couldn't provide the average acceptance ratio, as exact figure about the acceptance ratio was not mention on the respective conference sites.

Table 3. Name of conferences

Table 4 shows the detailed database of survey data; here, we categorized our data into mainly three categories: MAC layer, Routing layer and PHY layer (especially, for system simulators). Our database includes all related fields papers from above listed conferences. From our survey, we come across many simulator tools, and we broadly classified them into two main categories: "Widely Used" network simulators and "Other" network simulators, they are summarized in table 5.

Sr. No.	Conference Name	Years	Ad-Hoc/ Mesh Networks			Sensor Networks			Cognitive Radio Networks			Total
			Routing	MAC	PHY	Routing	MAC	PHY	Routing	MAC	PHY	
1	ACM MobiCom	2000~2008	25	19	-	10	5	-	-	-	-	59
2	ACM/IEEE MobiHoc	2000~2008	38	32	10	11	13	-	-	-	-	104
3	ACM Sigcomm	2000~2008	27	3	5	4	17	-	-	-	-	56
4	IEEE Infocom	2000~2008	45	26	-	9	6	-	-	-	-	86
5	IEEE Percom	2003~2008	5	10	-	5	10	-	-	-	-	30
6	IEEE GlobeCom	2000~2008	-	-	45	-	-	49	-	-	23	117
7	IEEE WCNC	2000~2008	-	-	-	-	-	21	-	-	7	28
8	IEEE ICC	2007~2008	-	-	-	-	-	9	-	-	18	27
9	ACM SenSys	2003~2008	-	-	-	24	-	7	-	-	-	31
10	EWSN	2004~2008	3	-	-	12	24	4	-	-	-	43
11	IEEE CrownCom	2006~2008	-	-	-	-	-	-	4	41	50	95
12	IEEE DySpan	2005~2008	-	-	-	-	-	-	1	34	26	61
13	IEEE CogART	2008	-	-	-	-	-	-	1	4	-	5

14	IEEE MilCom	2005~2008	-	-	-	-	-	14	-	17	8	39
15	Mobile Networks and Applications	2000~2008	-	-	12	-	-	-	-	-	-	12
16	MASCOTS	2000~2008	-	-	7	-	-	-	-	-	-	7
17	ACM SigMobile	2000~2008	-	-	8	-	-	-	-	-	-	8
		Total	143	90	87	75	75	104	6	96	132	808

Table 4. Survey Data

Sr.No.	"Widely Used" Network Simulators	"Other" Network Simulators	"Widely Used" System Simulators	"Other" System Simulators
1	NS2	Matlab	NS2Matlab	TOSSIM
2	GloMoSim	TOSSIM*	Monte Carlo	Own Simulators
3	J-Sim	Monte Carlo	-	MICA2
4	OMNet++	Own Simulators	-	Spectrum Analyzer
5	OPNet	Simulation Package not mentioned	-	Simulation Package not mentioned
6	QualNet	Rarely used simulators+	-	Rarely used simulators&

*TOSSIM falls under the category of Emulators.
+Rarely used simulators: Includes ROSS, JiST/SWAN, Prowler, Emstar, and EmSim, just to name a few. These simulators are not cited for more then 2/3 research papers in our survey so we put them under the tag named "Rarely used simulators".
& Rarely used simulators (System) : Includes Bayesian Estimator, MFC Coding, MDL/AIC, DUALFOIL, SWEET, BLUE-BCH Estimator, Microwave Studio, BELLHOP, QT Based Simulator, etc. , these simulators are not cited for more then 2/3 research papers in our survey so we put them under the tag named " Rarely used simulators".

Table 5. Network and System Simulator Tools
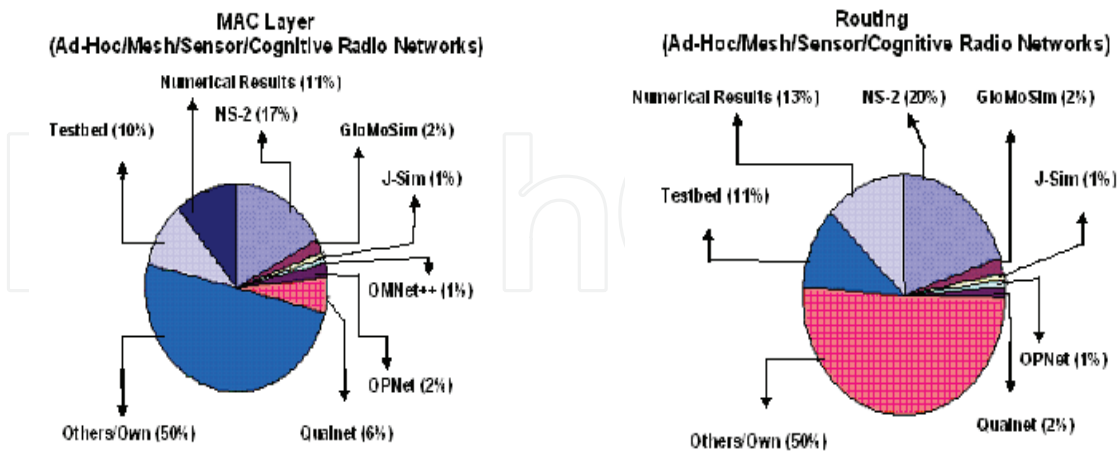
Fig. 3. Simulator Usage in MAC and Routing layers

3.2.1 Analysis: Network
Figure 4 shows the simulator usage results of our survey at different levels of protocol stack, especially at Routing and MAC layers. From figure 4, it is also interesting to know that

testbed or experimental studies are also gaining popularity in recent years, and their usage ratio is almost same in Routing and MAC layers. It also shows a good start from the wireless networks community to present more realistic, practical, and sound research results. But still there are many issues such as scalability, cost, area, etc., need to be addressed to make testbed or experimental setup widely accepted among the community. As of current research practice, simulation is currently the most feasible approach to the quantitative analysis of wireless networks. As we can see from figure 4 NS-2 is the most popular/used simulator among the "Widely Used" network simulator tools. To our surprise we find numerical/mathematical results are more dominating than "Widely Used" tools (NS2 is an exceptional case), as general trend is to present rigorous simulation results than mathematically sound results in wireless networks community. It is worth to note that these numerical results also include the theoretical aspect of the field. From figure 4 we can find a very interesting observation that in both the layers other/own category is at the top. To know the reason we further expand survey results on other/ own category as shown in figure 5.
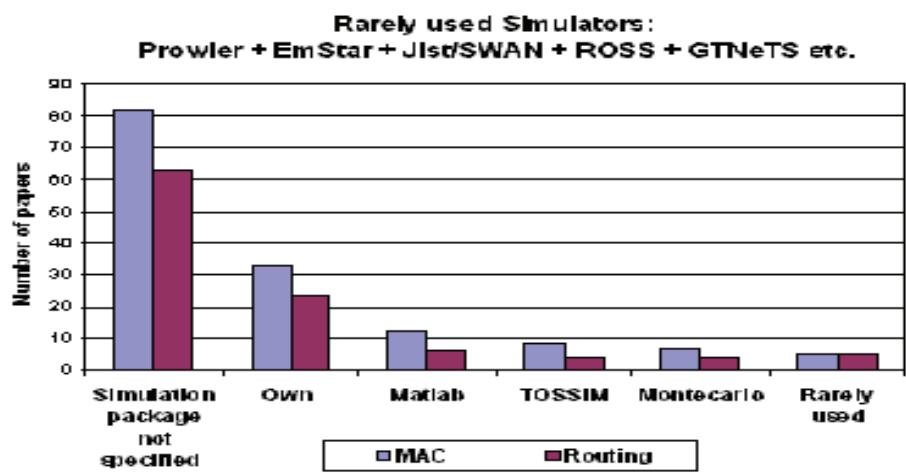
Fig. 4. Survey results on "Other/Own" Category

As we can see from figure 5 major part of other/own category is occupied by the simulator tools which are not specified in the papers. When the simulator used is not specified within a published paper, the repeatability and credibility of the simulation study are questionable. Second topmost category is "own" where researchers have used their self-developed or custom made simulation tools. It is also difficult, if not possible, to repeat a simulation study when the simulation is self developed and code is not available. Rest of the simulator tools, especially Matlab, TOSSIM, Montecarlo, and "rarely used" simulator tools, have a small portion of participation in wireless networks research. One very important fact come out in our survey is that a very few papers (hardly 3/4 papers) cited about the code availability, for whatever reasons but this issue really need an attention from the community. Further more, we obtained some interesting observations from our survey as shown in figure 6.

The execution and analysis of any experiment/simulation study must be based on mathematical principles and need to be statistically sound. For any experimental/ simulation study to be statistically sound must present the number of times simulation runs, confidence levels that exist in the results, and a list of any statistical assumption made. To our surprise, the large numbers of papers don't even bother to present this basic information

regarding their research results. As we see from figure 6 nearly 150 papers aren't independently repeatable because of the lack of simulation's information. Additionally, the papers often omitted simulation input parameters such as traffic model or type. As shown in figure 6 nearly 250 papers didn't specify any traffic model or type they have used. So, this lack of basic information raises many questions on the reliability and repeatability of wireless networks research.
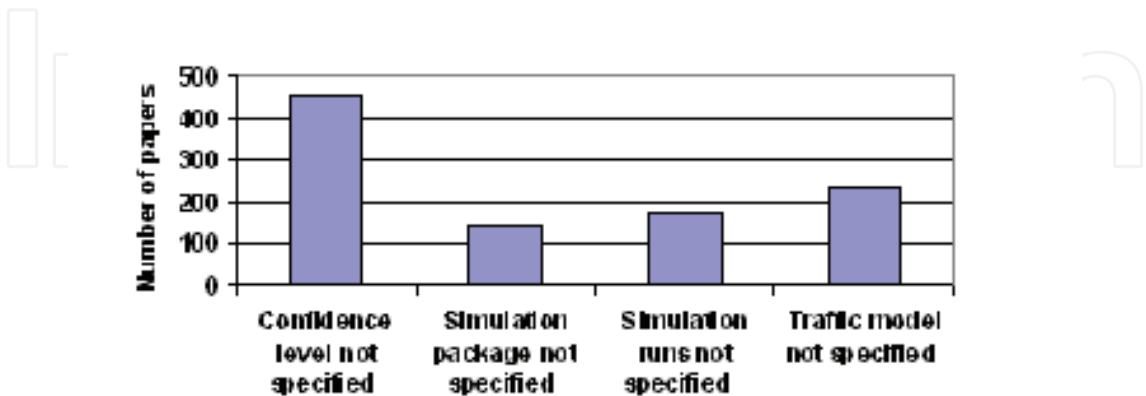

Fig. 5. Simulation issues

3.2.2 Analysis: System

Figure 7 shows the percentage of usage of various simulators in Physical layer. In this layer, sometimes it may not be possible to set up simulation environment. Theoretical analysis would be the best option for this type of situation. A large number of researchers adopt theoretical and numerical analysis in Physical Layer. In this survey, it is noticeable that testbed has approximately same popularity as in the case of Routing and MAC layers. By experiments it can be get more pragmatic and acceptable results. But experimental set up is not always feasible in many cases due to monetary and other limitations. In this regard, MATLAB is more prominent and widely used simulator in Physical layer as compared with Routing and MAC layers. It has many intelligent tools for various simulation purposes. Most of the communication systems can be simulated by using MATLAB. It has little practice in Network simulation. NS2 also plays a significant role in Physical Layer simulation. In addition, Monte Carlo is also an extensively used simulator in Physical Layer as opposed to Routing and MAC Layer. Some researchers also have interest with GloMoSim, QualNet and OPNet but in a little portion same as other two layers. Again, "other/ own" category is in the surprising top position. We would like to find out the reasons behind it.

By expanding this other /own category in figure 8, we can see that major parts of it didn't mention the simulator name. So, it makes same difficulties as in Routing and MAC layer. Second major category is "own" simulators those are developed by researchers themselves or custom made. With these kinds of self developed simulators and if codes are also not available, it is not possible to study these simulators again. Some simulators like M-AFC Coding, Bayesian State Estimator, MDL/ AIC, DUALFOIL, SWEET, BLUE_BCH Estimator, BELLHOP etc., are also used in Physical Layer simulations infrequently, we named them under 'rarely used" category. MICA2, Spectrum Analyzer and TOSSIM have some usage also. Except TOSSIM any other simulators among these are not used in Wireless Network Simulations. These are specially designed for Physical Layer simulations. As other two

layers, we also give our attention on system parameters which play important role to get reliable and sound
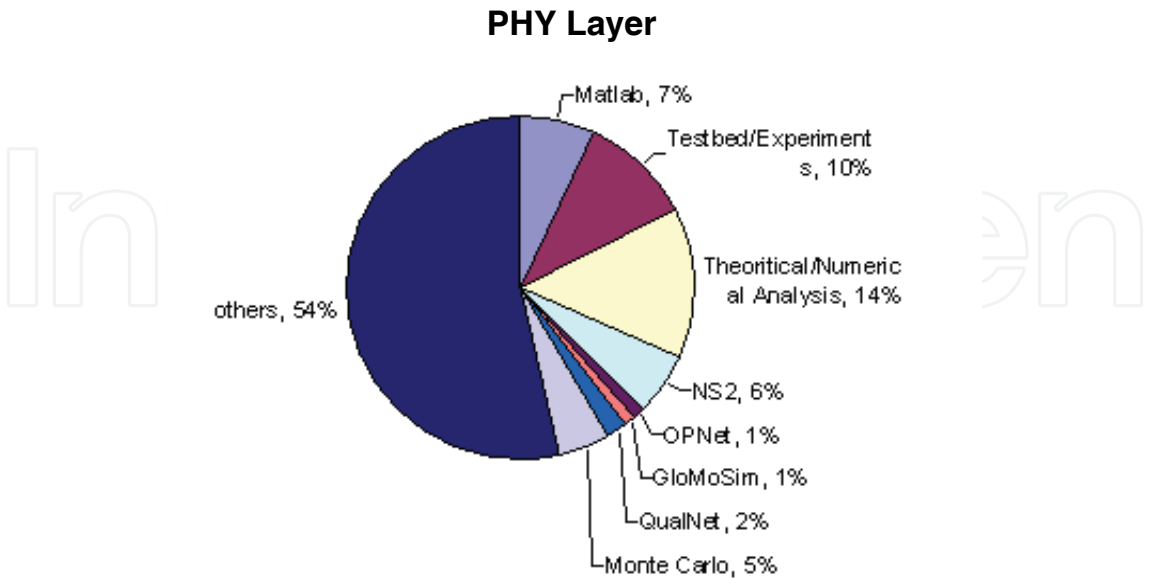
PHY Layer

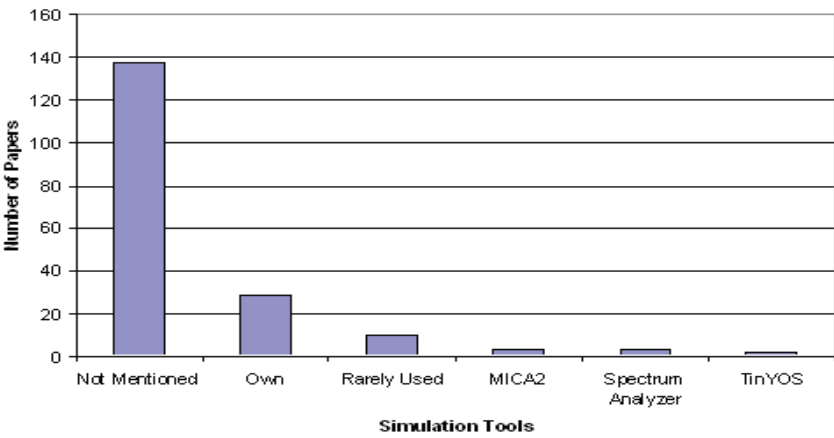Fig. 6. Simulator usage in PHY Layer

Fig. 7. Survey results on "Other/Own" Category (PHY)

It is already mentioned that results and analysis getting from a simulation study should be reliable and acceptable in a range. In general, a model of a physical system has error associated with its predictions due to the dependence of the physical system's output on uncontrollable or unobservable quantities. Confidence level is an important parameter for the simulation reliability. But from Figure 3 we can see that as like as Routing and MAC layers, most of the papers didn't mentioned the traffic models, acceptance levels, and other statistical parameters explicitly in Physical Layer. It is seen that around 225 papers didn't mention any confidence level in their simulation. Again, in more than 236 papers didn't mention what traffic model have been used. But in next generation networks traffic modeling will have to deal with two main issues: the radio resource management scheme and the effect of the user mobility in the traffic volume per cell. So, information about traffic model is necessary to further repeatability of a simulation and for reliable output. Number of independent run of any simulation has also an impact on accurate result. But a

large portion of researchers didn't state any information about that. So, questions may be raised about the credibility of the simulation based analysis.
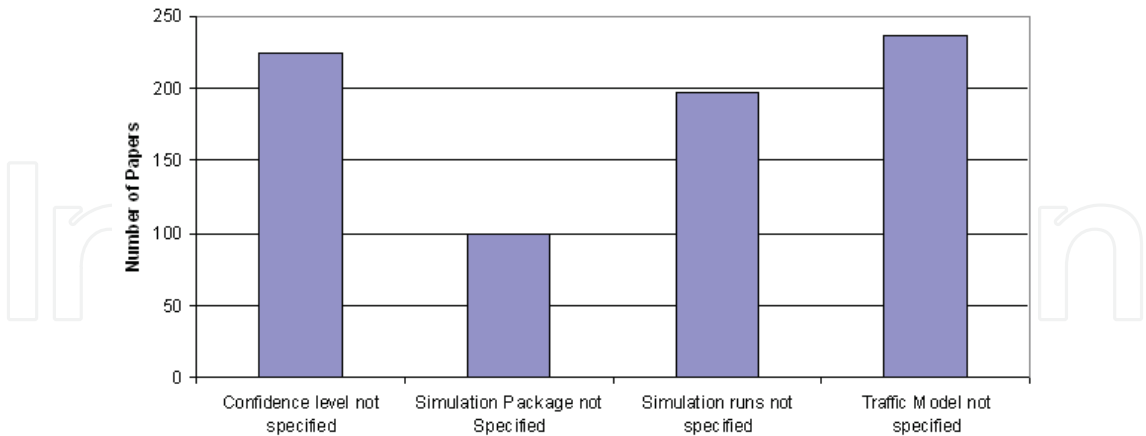

Fig. 8. Simulation issues (PHY)

To raise the awareness on the lack of reliability, repeatability, and credibility of simulation based studies we have developed a list of common issues and pitfalls as the starting point for improvement. We have written the list from our own experiences with simulations as well as the experience of others in the field. Some common issues and pitfalls are identified from our survey. We summarize these issues and pitfalls into the following categories: simulation setup and initial assumptions, simulation execution, and output analysis. They are summarized with our recommendations in table 6.

Category	Issues/Pitfalls	Recommendations
Simulation setup and initial assumptions	Network area, number of nodes, mobility Models, node distribution, traffic model, transmission range, bidirectional communication, capturing effect, simulation type: terminating vs. steady state, protocol stack model, RF propagation model, and proper variable definitions.	• Most of these issues can be easily solved by proper documentation. • Due to space limitation, sometimes publications can include only major settings. In this case authors can provide the external links or references, which include all the needed information. • Try to tune setting some parameters against an actual implementation if possible or improve the abstraction level of used models.
Simulation execution	Protocol model validation, PRNG validation, scenario initialization: empty caches, queues, and table; and proper statistics collection.	• Validating protocol models against analytical models or protocol specifications • Determining the number of independent runs required. • Proper setting and address of random number generators

		• Collecting data only after deleting transient values or eliminating it by proper preloading routing cache, queues, and tables.
Output analysis	Single set of data, Statistical analysis: autocorrelation, averages, aggregation, mean , and variance; confidence level	• Experiment should be run for some minimum number of times • Analysis should be based on sound mathematical principles • Provide proper confident interval for a given experiment.

Table 6. Important issues and recommendations

This paper summarizes the current state of practice, and identified some of the difficult issues that must be resolved to increase the reliability and credibility of simulation based studies. Further more, wireless community should take some concrete steps such as standardization of simulation tools and creating some universal virtual testbeds to resolve the points of consensus as mentioned above. Universal virtual tesbed could be a very useful for all the research groups around the globe and can also be used as standard measuring tool for wireless networks community.

4. Conclusions

In this paper, eight most "widely used" network and system simulators and their strengths and weaknesses were discussed based on a couple of papers and a survey. Then, the results of a survey of recent research publications on performance evaluation of networks were used to show that the majority of results of simulation studies of wireless networks published in technical literature have many pitfalls/issues. With this paper we documented these pitfalls and some important issues with some recommendations to increase the reliability and repeatedly of simulation studies. Finally, we hope, the results presented in this paper will motivate the researches to put their efforts in thorough descriptions of the simulation scenarios and taking care of pitfalls in simulation studies of wireless networks.

5. References

I. F. Akyildiz, W. Y. Lee, M.C.Vuran, and S. Mohanty. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. Computer Networks 50 (2006) 2127–2159, May 2006.

J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. Comparative study of wireless network simulators. in proceeding of ICN, 2008, pp. 517-523.

J. Heidemann, K. Mills, and S. Kumar. Expanding Confidence in Network Simulations. IEEE Network, vol. 15, no. 5, 2001, pp. 58-63.

 S. M. Sanchez. ABC's of output analysis. in proceeding of the 1999 winter simulator conference, 1999, pp. 24-32.

B. Schilling. Qualitative comparison of network simulation tools. Technical report, Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart, January 2005.

S. Duflos, G. L. Grand, A. A. Diallo, C. Chaudet, A. Hecker, C. Balducelli, F. Flentge, C. Schwaegerl, and O. Seifert. Deliverable d 1.3.2: List of available and suitable simulation components. Technical report, Ecole Nationale Supieure des Tommunications (ENST), September 2006.

M. Karl. A comparison of the architecture of network simulators ns-2 and tossim. In Proceedings of Performance Simulation of Algorithms and Protocols Seminar. Institut fr Parallele und Verteilte Systeme, Abteilung Verteilte Systeme, Universit Stuttgart, 2005.

L. Hogie, P. Bouvry, and F. Guinand. An overview of manets simulation. In Electronic Notes in Theoretical Computer Science, Proc. of 1st International Workshop on Methods and Tools for Coordinating Concurrent, Distributed and Mobile Systems (MTCoord 2005), LNCS, pages 81–101, Namur, Belgium, April 2005. Elsevier.

E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Mari, and J. Garcia-Haro. Simulation scalability issues in wireless sensor networks. IEEE Communications Magazine, 44(7):64–73, July 2006.

D. Curren. A survey of simulation in sensor networks. Student project, www.cs.binghamton.edu/˜kang/ teaching/cs580s/david.pdf, 2007.

L. Begg,W. Liu, K. Pawlikowski, S. Perera, and H. Sirisena. Survey of simulators of next generation networks for studying service availability and resilience. Technical Report TRCOSC 05/06, Department of Computer Science & Software Engineering, University of Canterbury, Christchurch, New Zealand, February 2006.

G. F. Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M. J. Reed. Opnet modeler and ns-2 - comparing the accuracy of network simulators for packet-level analysis using a network testbed. WSEAS Transactions on Computers, 2(3):700–707, July 2003.

K. Pawlikowski, H.-D. Jeong, and J.-S. Lee. On credibility of simulation studies of telecommunication networks. IEEE Communications, 40(1):132–139, January 2002.

M. Lacage and T. R. Henderson. Yet another network simulator. In WNS2, The workshop on ns-2: the IP network simulator, 2006.

G. F. Riley. The georgia tech network simulator. In MoMeTools ’03, pages 5–12, 2003.

http://www.isi.edu/nsnam/ns/

http://pcl.cs.ucla.edu/projects/glomosim/

http://www.j-sim.org/

http://www.omnetpp.org/

http://www.opnet.com/

http://www.scalable-networks.com/

http://www.mathworks.com/

http://www.solver.com/simulation/monte-carlo-simulation.

S.Mehta,Md. H. Kabir, Mst. N. Sultana, N. Ullah, and K.S. Kwak, “ A Case Study of Networks Simulation Tools for Wireless Networrks,” in procceding of AMS’09, May,pp. 661-666.
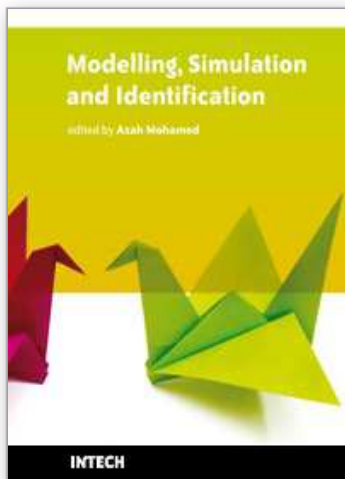
Modelling, Simulation and Identification

Edited by Azah Mohamed

Modeling, simulation and identification has been actively researched in solving practical engineering problems. This book presents the wide applications of modeling, simulation and identification in the fields of electrical engineering, mechanical engineering, civil engineering, computer science and information technology. The book consists of 17 chapters arranged in an order reflecting multidimensionality of applications related to power system, wireless communication, image and video processing, control systems, robotics, soil mechanics, road engineering, mechanical structures and workforce capacity planning. New techniques in signal processing, adaptive control, non-linear system identification, multi-agent simulation, eigenvalue analysis, risk assessment, modeling of dynamic systems, finite difference time domain modeling and visual feedback are also presented. We hope that readers will find the book useful and inspiring by examining the recent developments in the applications of modeling, simulation and identification.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

S. Mehta, Najnin Sulatan and K.S. Kwak (2010). Network and System Simulation Tools for Next Generation Networks: a Case Study, Modelling, Simulation and Identification, Azah Mohamed (Ed.), ISBN: 978-953-307-136-7, InTech, Available from: http://www.intechopen.com/books/modelling--simulation-and-identification/network-and-system-simulation-tools-for-wireless-networks-a-case-study

INTECH
open science | open minds