A Research Survey on Experimental Tools for Simulating Wireless Sensor Networks

V.Chandrasekaran
Asst.Professor (Sl.Gr)
Department of ECE
Velalar College of Engineering
and Technology,
Erode-638012
Tamil Nadu, India

S. Anitha
Asst.Professor
Department of IT,
Kongu Engineering College,
Erode-638 052
Tamil Nadu, India

A.Shanmugam, Ph.D
Principal and Research
Supervisor
Bannariamman Institute of
Technology
Sathyamangalam-638 401
Tamil Nadu, India

ABSTRACT
The success of Wireless Sensor Networks lies in the fact that the tools available for detecting the failures are demonstrated in many long term implementations. WSNs are scarce or limited resources, delicate, difficult in analysis and prone to mistakes easily by nature. With the ease of tools installations increases, nowadays the causes of failure identification in such a network becomes efficient. Due to the high cost and increased time consumption of deployment of WSNs, almost in all the fields, many experimental tools are used before real environment implementation. From 2005 onwards, these tools are rapidly increased in number and performance proportionately with the emerged new technologies. In this survey, many tools are discussed and compared with several features for implementation. Even though many tools are developed for different applications, still there exists open challenges' regarding its reachability, suppleness, simplicity, locality of nodes and protection. To avoid these issues, WSNs platforms and operating systems has to be matured by having improved diagnostic capability. The key objective of this work is to review various tools available for the emerging Wireless Sensor Network and its services suitable for different categories of applications by performing different functions thus paving the way for changes in Information and Communication field in the upcoming years.

Keywords
Wireless Sensor Networks, Experimental tools, Information and Communication field.

1. INTRODUCTION
Wireless Sensor Networks (WSNs) is a network which has no infrastructure and formed by a large number of sensor nodes that are miniature devices behaving in an autonomous manner. The sensor nodes in WSNs sense the environment and the information are processed in the transmitter then the result is transmitted to one or more base stations over wireless link. The researcher must have prior knowledge about the statistical performance of scenario, systems, protocols, gathered data and applications before enter into the research in WSNs. Presently, there are many tools are present to satisfy the above conditions but many of them used in theoretical part only.

A large WSN application should be implemented after checking the design specifications and requirements with the selected hardware. By using simulation or emulators the architectures, services and suitable protocols are developed, tested and evaluated. When the number of nodes is increased then it is not possible to prefer test-bed to evaluate certain parameters, because it will be more expensive and takes long time. But test-bed is the only option to develop reliable and portable network; in this case simulation becomes invalid. So the simulator or emulator is chosen based on the parameter concerned and application specific, that balancing both merits and demerits.

In Wireless Sensor Networks the system suitability is checked by using simulation before using it. The simulation can be used to test the scalability of algorithms used and it doesn't consider the hardware platform constraints. And it is also used to simplify the software development process for a particular WSN application. For example, TOSSIM is a WSN specific simulator that makes use of the TinyOS architecture for hardware resource in the abstraction layer and is used to connect hardware platform to the TOSSIM software without any interface.

For these reasons, Simulation is the most favorable tool for mobile network. However recent research papers publication in SENSORCOMM 2012 reporting the trend towards real test-beds, above 50% of researchers prefers simulation for their work. There are many specific simulators and emulators are available for WSN like TOSSIM, TOSSF, Dingo, Sensorsim, Avovra and Emstar, Atmeu respectively and many general simulators and emulators like NS-2, OPNET, QUALNET, Glomosim, J-Sim, OMNet++ are also available for WSNs. Because of the WSNs behavior such as mobility and adhoc nature simulators are extended or a special simulator is designed for adhoc network such as NS-3. This type of simulators doesn't support WSNs directly but only focus on protocols and algorithms supported by different layers of protocol stack. Though recently developed tools addresses various important issues like localization and security in WSNs, their common aim differs by design goals, architecture, and abstraction level of applications. The various network simulators can be compared based on the range, nodes specification and its connections, nodes traffic and the traffic handled by the protocols and its information, different applications and programming tools.

The objective of this survey is to present the architecture, summary, environment, key features, merits, demerits and limitations of each experimental or diagnostic tool for WSNs. Then the overall goal is to the compare performance of these tools for suitability to different applications and comparing with certain features, parameters, scenario and criteria of concern that is useful for researchers in their work.

The rest of the paper is organized as follows: In Section II, basic concepts of simulators and emulators such as its definition & necessity, steps in the study and different kinds of simulation & emulation are outlined. In Section III,

categories of network simulators and emulators are listed with architecture diagram. In Section IV, comparison table is depicted with certain features. In Section V, review analysis is justified. In Section VI, paper is concluded.

2. SIMULATORS AND EMULATORS FOR WSN

A simulator is a software tool that is used for research and development for producing imitation of the reality. In many fields the concept of simulation is mostly used. There are many simulators are available to experiment the simulation concepts. The real-world entities of different field are modeled and imitated. Based on the level of abstraction of the parts those to be modeled and imitated different tools are used with different objective. Traditional simulators specific to WSNs only imitates wireless media with the constraints.

An emulator is a special type of simulator that aims to evaluate WSN applications in a realistic way to have high performance. Emulator is a better choice for testing, debugging and performance evaluation of WSN applications directly compared with simulator. For real implementation using emulator tools, cross-layer techniques are also taken into account in addition with individual lower layer functions.

3. CATEGORIES OF NETWORK SIMULATORS AND EMULATORS
3.1. NS-2

NS-2 is a discrete event simulator used in the networking area for research purpose. In 1989, NS-2 is designed as an alternative to REAL network simulator and significantly became popular over the past few years. Because of its modular design approach effective extensibility is possible.

At the starting stage NS-2 supports only simulation of TCP, routing and multicast protocols over wired and wireless networks and later it also used to simulate wireless LAN protocols, mobile ad-hoc networks and wireless sensor networks.

Key features: Because of the object-oriented feature of NS-2 design and use of new protocols is possible. It also has the key features that include sensor channels, battery models, lightweight protocol stacks, hybrid simulation support and scenario generation tools. It also supports the visualization tool called NAM (Network AniMator) and it can be generated either using GUI or through TCL script. Due to its status and straightforward protocol development there are increased numbers of various categories of protocols publicly available. It produces detailed results as the simulation run at the packet level supporting transport layer protocols.

Merits:

 1) It supports by protocol stack of all layer functions.

 2) The cost of simulation is less because of open source and it also allows modifying and customizing the codes easily by the researchers.

Demerits:

1) Energy consumed by different hardware, software and firmware components of a WSN node cannot be measured.

2) Addition of new protocol is tedious with the interdependence of modules in the object-oriented design.

Limitations:

 a) It is used to simulate only   <100 nodes.

 b) It is not used for simulating the issues of bandwidth or the power consumption in WSNs.
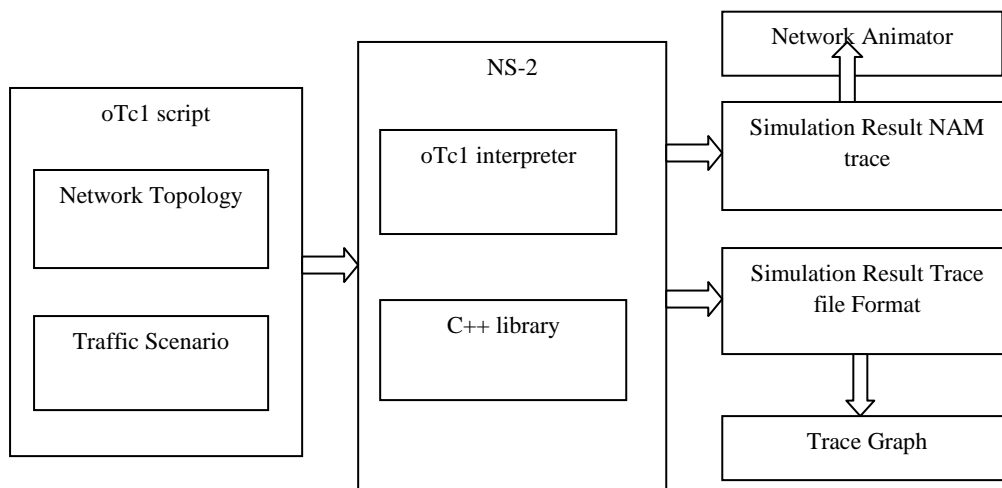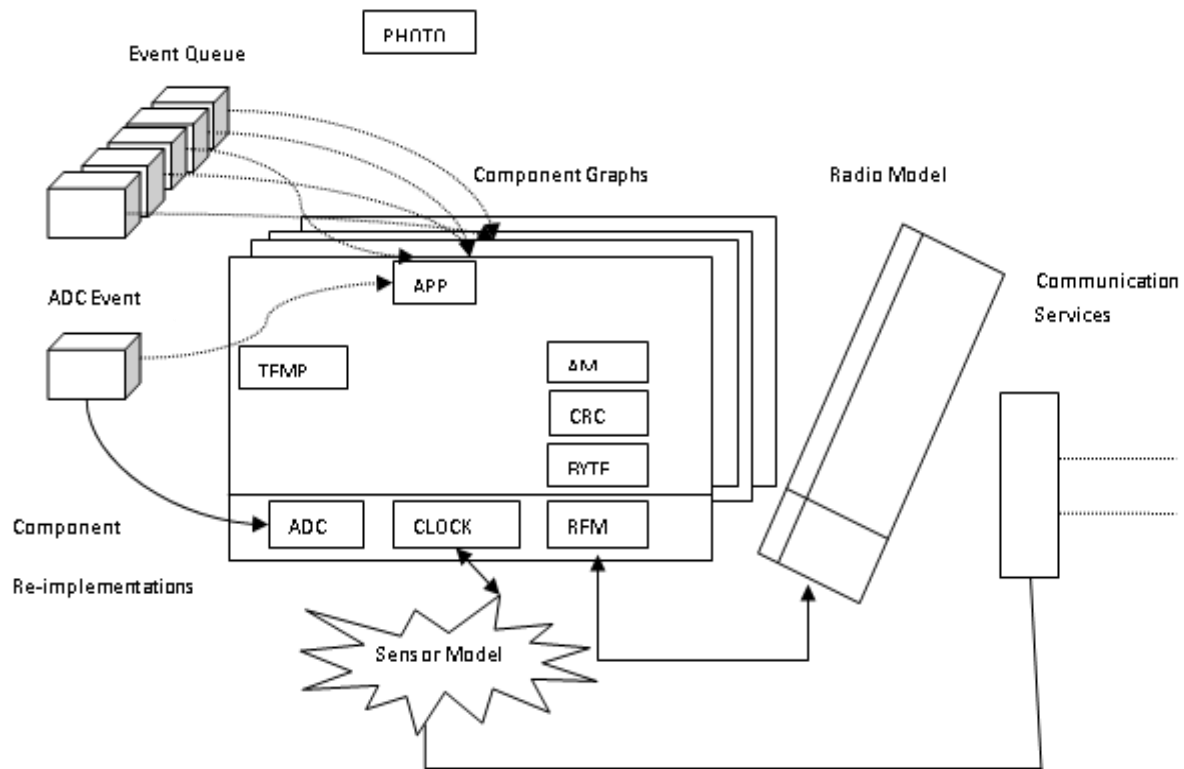

Fig 1 : Architecture of NS-2

3.2.TOSSIM

TOSSIM, a discrete event simulator is specifically developed at UC Berkeley under MICA Motes platform for TinyOS wireless sensor networks. TOSSIM supports a bit level in network layer functions rather than transport layer function at packet level unlike NS-2. TOSSIM is used to simulate entire TinyOS applications.

Environment: The simulation of TinyOS required accuracy and complexity of the radio model that is provided through its various mechanisms.

Key features: It provides mechanisms to interact with the network. It also has a key feature that the packet traffic can be monitored and injected statically or dynamically into the network and it simulated bit by bit basis experimenting higher applications too. It produces high performance because simulation occurs at bit level transmission.

Fig 2: Architecture of TOSSIM

Demerit:

1) TOSSIM is works on the assumption of each node running the same code so it has less flexible.

Limitation: Unsuitable for heterogeneous environments.

Merits:

1) By using this emulator each node can be emulated under ideal transmission conditions and Hidden Terminal problems can also be identified with this emulator.

2) Snce TOSSIM supports more number of nodes for simulation, real-world problems are solved with accurateness.

3.3.GLoMoSim

GLoMoSim is used for solving complex problems in both wired and wireless communication networks. An aggregation technique is introduced in this tool to improve the performance of the simulation. A node is identified by its physical position that is used to represent the particular entity of an application since each node in GLoMoSim is denoted by the geographical area in the simulation.

Environment: GLoMoSim is used for implementation of radio propagation effects, MAC protocols (including 802.11), mobile wireless routing protocol, UDP and TCP and is also used for simulation of mobile IP networks.

Key features: GLoMoSim supports protocols which are used only for wireless networks as it is designed using layered approach. To provide interaction between layers, it uses standard APIs.

Merits:
GLoMoSim simulates IP networks in an effective manner.

Demerits:
a) When the sensor networks want to simulated precisely GLoMoSim becomes fails.

b) If the events occurring only in the neighboring node within the network GLoMoSim reacts otherwise it doesn't react to any events that are occurring outside the simulation environment.
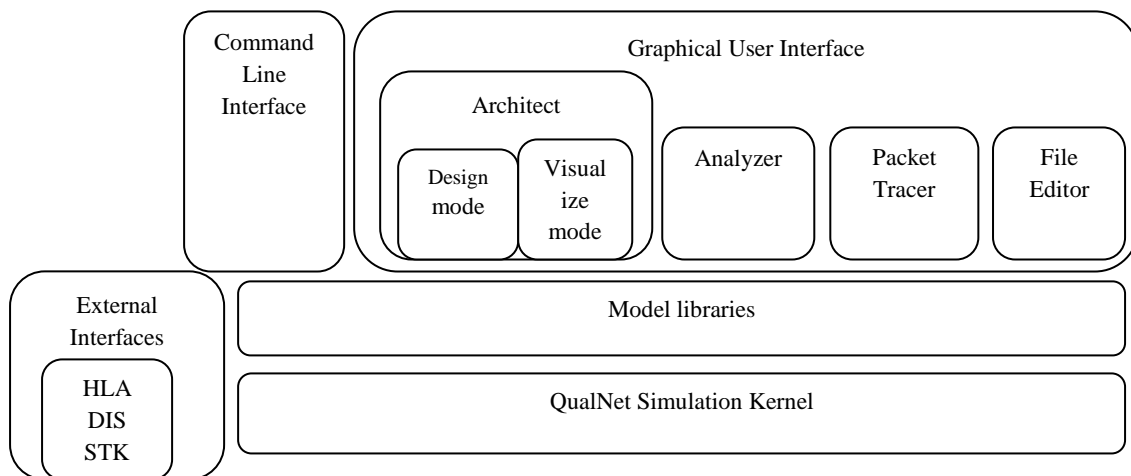
Limitations:
a) Scalability of the network is limited as memory required for initializing each entity is very high.

b) GLoMoSim released a commercial product called QualNet and stopped releasing updates in 2000.
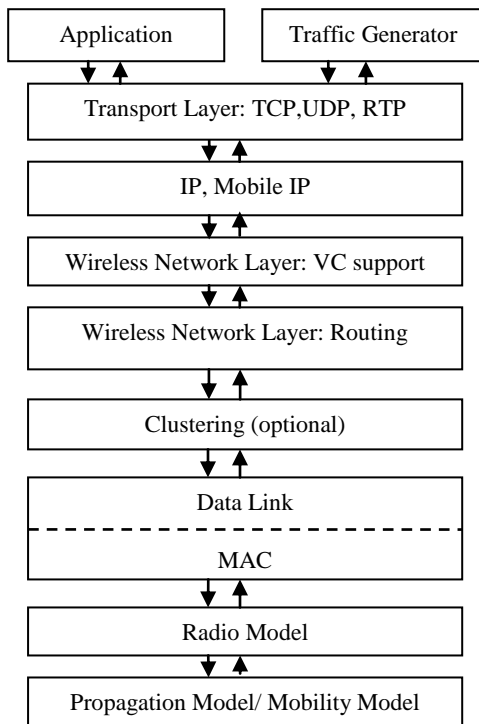
Fig 3 : Architecture of GLoMoSim

Fig 4 : Architecture of Qualnet

3.4. Qualnet

Qualnet is introduced in 2000-2001 as a commercial network by Scalable Network Technologies. It is ultra high fidelity network simulation software. It is used to predict wireless, wired and mixed-platform network and networking device performance. It is used to simulate large, heterogeneous networks and the distributed applications run execute on such networks.

Environment: Qualnet uses C/C++ for implementing new protocols and follows a procedural paradigm. It uses the parallel simulation environment for complex systems (PARSEC) to perform basic operations; hence it can run on distributed machines.

Key features: The key feature of Qualnet is that it is a discrete-event simulator, event driven and time aware. It uses a layered architecture that is run by each node. If a protocol resides in a particular layer at one node, the packets are passed down crossing the remaining layers at the sending node across the network; and then up to the protocol stack at the receiving node.

Merits:
Due to its modular design and GUI facility Qualnet is easy to learn and modify.
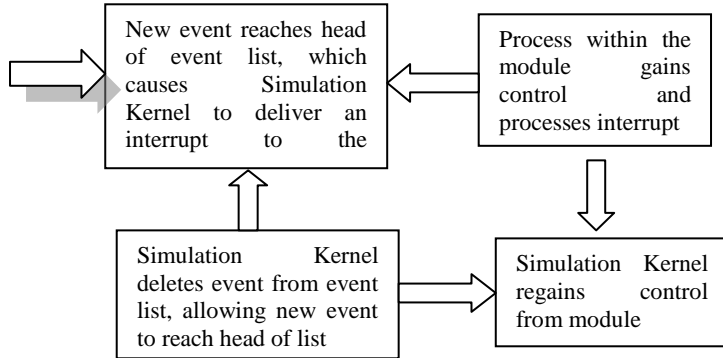
Demerits:
It will be available only for commercial usage.

Limitations: Since there are many in – built functions are available, user-friendly codes are limited.

3.5.OPNET

OPNet is an object oriented network simulator written in C++, widely used in commercial and general purpose and it was well-established (1986). It is used to support discrete event, hybrid

and analytical simulation. It provides a very rich set of modules for all layers of protocol stacks including the IEEE 802.11 family, IEEE 802.15.4, and routing protocols like AODV and DSR. A state machine is used to implement the each level of the protocol stack but it is difficult to abstract such a state machine from a pseudo-coded algorithm.

Fig.5 Architecture of OPNET

Merits:
Model parameters can be changed.

Demerits:
1) OPNET is not open source software.
2) Scalability problems are introduced due to Object-oriented design of OPNET.

Limitations: The number of protocols publicly available for OPNET is limited because of source code licensing constraints.

3.6.J-Sim

After Ns-2, J-Sim is a general purpose simulator modeled developed at the University of Washington by the National Simulation Resource. Unlike Ns-2, yet, J-Sim uses the concept of components, replacing the notion that each node should be represented as an object. In this model three top level components are used: the target node (which produces stimuli), the sensor node (that reacts to the stimuli), and the sink node (the ultimate destination for stimuli reporting). In simulator each component is broken into different parts and modeled differently. It uses different protocols in different simulation runs due to the breakdown of each component.

Environment: J-Sim provides support for sensors and physical phenomena. Energy modeling, with the exception of radio energy consumption, is also appropriately provided for sensor networks.

Key features: J-Sim has a improved feature than NS-2 and other simulators. Most importantly, the scalability of J-Sim is better as it uses component based architecture while NS-2 and other simulators uses object oriented model. It also has features of improved energy model and an ability to simulate the use of sensors for phenomena detection. It is used to simulate applications like SensorSim and the real hardware sensors is connected to the simulator with the support of 8.4 connection.

Merits: It is more scalable than any other simulators.

Demerits:
a) Next to NS-2, J-Sim is complicated to use.

b) b) As Java is used, it is less efficient and unnecessary overhead occurs in intercommunication model.

Limitations: It supports only MAC protocol that can be used is 802.11.

MML: JSim's Mathematical modeling language

RTML: JSim model hypertext markup
BCL: JSim biological component libraries
XSIM: Legacy modeling system
I4: PET image analysis
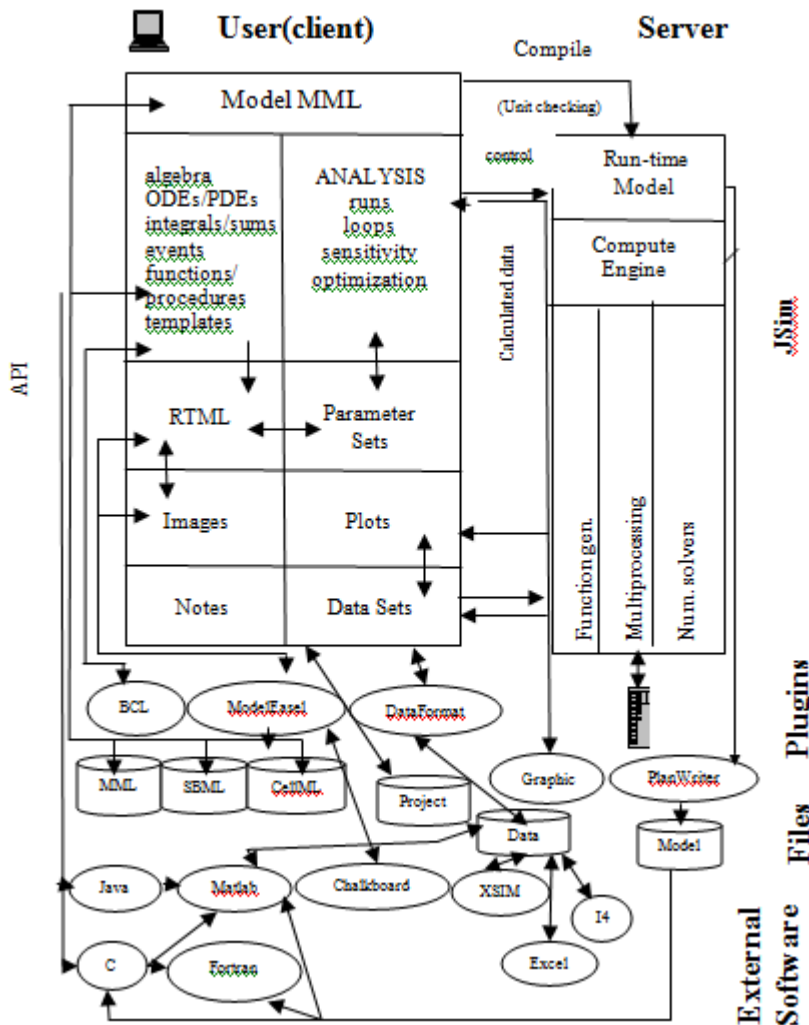Chalkboard: Chem. network builder GUI

Fig 6 : Architecture of J-Sim

3.7. OMNeT++

OMNeT++ is a discrete event simulator framework that is component-based, modular and open architecture. It is mostly used to simulate the computer 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 allows GNU Public License like freedom but only in noncommercial settings. It provides component architecture for models.

Environment: A components (simple modules and channels) is created by using C++ class library which consists of the simulation kernel and utility classes (for random number generation, statistics collection, topology discovery etc). C++ is also used to assemble simulations from these components and configure them (NED language, ini files). It provide 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.
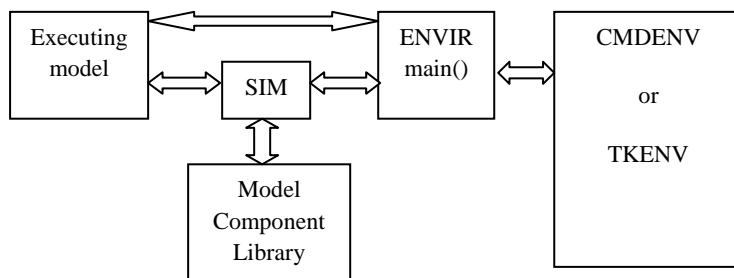

Fig.7 : Architecture of OMNET++

Key features

OMNeT++ offers graphical runtime environment and Eclipse-based IDE, and a host of other tools. It has a features for real-time simulation, network emulation, alternative programming languages (Java, C#), database integration, SystemC integration, and several other functions. For animation and debugging the simulator also provides a strong GUI support.

Merits:

1. It can support MAC protocols and some localized protocols in WSN.

2. It simulates power consumptions and channel controls.

Demerits

As individual researching groups developed the models separately, the compatible problem will rise. This makes the combination of models difficult and programs may have high probability of report bug.

Limitations

Only limited number of protocols are available.

Table 1. Comparison of WSN simulating Tools

Simulators	Programming Language	Simulator or Emulator	Discrete-EventSimulation or Trace-Driven Simulation	GUI	General or Specific Simulator	Open Source	Latest Version	Released year	Availability (URL)
NS-2 (Network Simulator version 2)	C++,C,Tcl, Otcl	Simulator	Discrete-Event Simulation	Yes	general	Yes	2.35	1989	http://www.isi.edu/nsnam/ns/ns-build.html
TOSSIM (TinyOS Simulator)	C/C++,Python, nesC	Emulator	Discrete-Event Simulation	Yes	specifically designed for wsns	Yes	2.1.1	2010	http://www.tinyos.net
GloMoSim (Global Mobile Information System Simulator)	C/parsec	Simulator	Discrete-Event Simulation	Yes	General	Yes	2.03	1999	http://pcl.cs.ucla.edu/projects/*glomosim*
Qualnet	C/C++	Simulator	Discrete-Event Simulation	Yes	General	Commercial	5.1	2000	http://www.it.iitb.ac.in/~qualnet/
Opnet (Optimized Network Engineering Tools)	C/C++	Simulator	Discrete-Event Simulation	Yes	General	Commercial	9.1	1987	http://www.opnet.com/university_program/itguru_academic_edition/
J-Sim (Java-based simulation)	Java	Simulator	Discrete-Event Simulation	Yes	General	Yes	1.3 + patch4	2006	http://sites.google.com/site/jsimofficial/
Omnet++ (Objective Modular Network Testbed in C++)	C,C++,Java, C#,nesC	Simulator	Discrete-Event Simulation	Yes	General	Yes	4.3	2005	http://www.omnetpp.org/component/docman/cat_view/17-downloads/1-omnet-releases

4. REVIEW ANALYSIS

In the previous sections, the background of various network simulators that are mostly used is discussed and also compared with each other based on some significant features which can be generic or specific. With this, the review also used to analyses the tools that are mostly used by the researchers and makes them aware of the simulator issues and pitfalls for choosing the best simulator for their application.
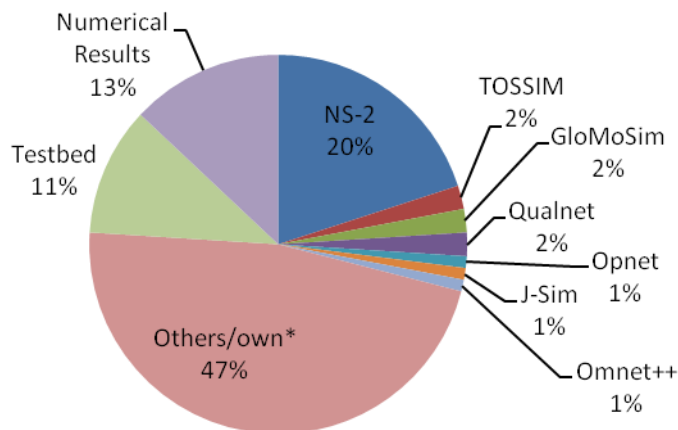
* includes MATLAB, Monte Carlo method and rarely used simulators like Prowler, Emstar, JiST/SWANS,ROSS, GTNetS

Fig 8: Simulator usage in Routing Layer

The fig.8 shows the list of widely used simulators and others in routing layer are denoted in percentage by referring the papers from 2000 to till date. From the above chart it is confirmed that the NS-2 simulator is widely used by the wireless networks community compared to others. Various simulator tools are widely used among the researchers because of an addressable issue like scalability, cost, area etc, even though experimental setup or test bed shows more accurate results. After experimental analysis and evaluation recent papers in conferences, journals focus that simulation as the most feasible approach for quantitative analysis of wireless networks. even though many applications can even produce good results with numerical/analytical methods, the trend is to generate precise results at varying conditions over time. Simulation is the only method to achieving this requirement.

5. CONCLUSION

Simulation tools are broadly used in checking the algorithms and protocols for the tentative analysis due to the rapid prototyping and tackling of large scale systems. Many simulators perform better by using computer but it becomes unsuitable in real environments in terms of completeness, complexity, correctness and faithfulness. Emulators can be used by researchers after simulation to verify that the intended application working as per the specifications and requirements. These emulators embed hardware with the software applications and simulates in real-time environment. But emulator approach is considered as relatively difficult because it uses the complete profiling requirements of components and protocols involved in the application. Due to these drawbacks of simulators and emulators, the performance can be evaluated by using WSN test beds before applying them into real world environments. To enhance the lifetime of the WSNs the other experimental tools for data visualization, debugging, code updating /reprogramming and network monitoring are also used before or after real implementation.

The conclusion of this review depicts the fact that many experiments have to be conducted to find the appropriate tool for the application of interest and to get the accurate result both in real and virtual environments.

6. ACKNOWLEDGEMENT

7. REFERENCES

[1] A. K. Dwivedi1, O. P. Vyas2, "An Exploratory study of Experimental Tools for Wireless Sensor Networks", doi:10.4236/wsn.2011.370, July 2011 (http://www.SciRP.org/journal/wsn), pp.215–240.

[2] Abdelrahman A, Fayez Al-Fayez, Tariq A, Mohammad H, Andrew N, "Simulation Issues in Wireless Sensor Networks: A Survey", SENSORCOMM 2012, The Sixth International Conference on Sensor Technologies and Applications, IARIA, 2012,ISBN: 978-1-61208-207-3, pp..222-228.

[3] Fei Yu, "A Survey of Wireless Sesor Network Simulation Tools", http://www1.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html, April 2011, pp. 1-10.

[4] Andre´Rodrigues,Tiago Camilo,Jorge Sa´ Silva,Fernando Boavida,"Diagnostic Tools for Wireless Sensor Networks:A Comparative Survey",Springer Science+Business Media, LLC 2012,DOI 10.1007/s10922-012-9240-6,pp.1-45.

[5] B. Musznicki and P. Zwierzykowski, "Survey of Simulators for Wireless Sensor Networks", International Journal of Grid and Distributed Computing, ISSN: 2005-4262, Vol. 5, No. 3, September 2012, pp. 23-50.

[6] Qutaiba I. Ali, "Simulation Framework of Wireless Sensor Network (WSN) Using MATLAB/SIMULINK Software",

[7] D.S.Hira,"System simulation",published by S.Chand & Company Ltd.,Reprint 2009,ISBN 81-219-2059-0,pp.1-296.

[8] Saba S, Ajay Kumar G, Rinku-B, "Network Simulation Tools Survey", International Journal of Advanced Research in Computer and Communication Engineering, June 2012,ISSN : 2278 – 1021,Vol. 1, pp.201-210.

[9] S. Mehta, Niamat Ullah, Md. Humaun Kabir, Mst. Najnin Sultana, and Kyung Sup Kwak, "A Case Study of Networks Simulation Tools for Wireless Networks",

Third Asia International Conference on Modelling & Simulation,IEEEComputerSociety,2009,pp.661-666.

[10] Xiang,Rajive,Mario, "GloMoSim:A Library for Parallel Simulation of Large-scale Wireless Networks",pp.1-8.

[11] P.Levis,N.Lee,M.Welsh,D.Culler,"TOSSIM:Accurate and Scalable Simulation of Entire TinyOS Applications",SenSys 2003,pp.1-50.

[12] http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

[13] Punit Rathod, QualNet Tutorial, 10th Sep 2005.

[14] http://j-sim.cs.uiuc.edu/drcl.inet/ex_echoer.html

[15] http://www.omnetpp.org/pmwiki/index.php?n=Main.Om netppInNutshell

9