

# A Comprehensive Review of Simulation Tools for Wireless Sensor Networks (WSNs)

Anand Nayyar<sup>1,\*</sup>, Rajeshwar Singh<sup>2</sup>

<sup>1</sup>Desh Bhagat University, Mandi Gobindgarh, Punjab, India

<sup>2</sup>Doaba Group of Colleges, Nawanshahr, Punjab, India

**Abstract** The trend towards adoption of Wireless Sensor Networks is increasing in recent years because of its diverse use in various fields. Wireless Sensor Network is formed via interconnection of large number of sensor nodes. Each and every sensor node deployed in network monitors various parameters like Temperature, Humidity, Ambient Light, Gas etc. and send the data to the master node. Despite of several applications and diverse uses, sensor networks face various shortcomings like energy, localization, security, self-organization, fault tolerance and many more. So, the area of Wireless Sensor Network is under rigorous research and development by various researchers across the globe to develop new algorithms, protocols and techniques to make WSN network more efficient and reliable. Before live implementation, testing of the developed technique requires rigorous testing. But it is not always possible to have live sensor network environment. So, in that case, Simulation is the only way to test the research before moving towards live implementation. Large numbers of simulation tools are available for WSN network till date, out of which some are dedicated towards wireless sensor networks and some for both wireless and wired networks. The main objective behind this research paper is to do a comprehensive review of various simulation tools of Wireless Sensor Networks to enable researcher to select the most competent tool for simulating wireless sensor networks and testing the research proposed. A comprehensive review of 31 WSN simulators is being presented along with their respective features comparison to assist researchers in advanced WSN based research.

**Keywords** Wireless Sensor Networks, Simulation Tools, Comparison, Performance Evaluation, Network Simulator

## 1. Introduction

In recent years, tons of new and advanced research has been done and is also picking up in the area of Wireless Sensor Networks [1] and this area is also catching the viral attention of researchers from all parts of the world for developing wide applications and making use of WSN networks in varied fields. Wireless Sensor Network comprises of a large number of sensor nodes having sensing and computing capabilities and are deployed in random manner. Each of the sensing devices in WSN network is called MOTE [2].

The following figure shows a view of simple wireless sensor network. Wireless sensor network [4] consists of one or more base stations known as gateways, a number of sensor nodes and end user. The output generated by one node is wirelessly transmitted to the base station for data collection, analysis and logging. Each and every node in the Wireless Sensor Network acts as router for transmitting the information from source node to sink node [122] [123] [125].

The end users are facilitated with the data from the sensor via some website or some application in the console terminal.

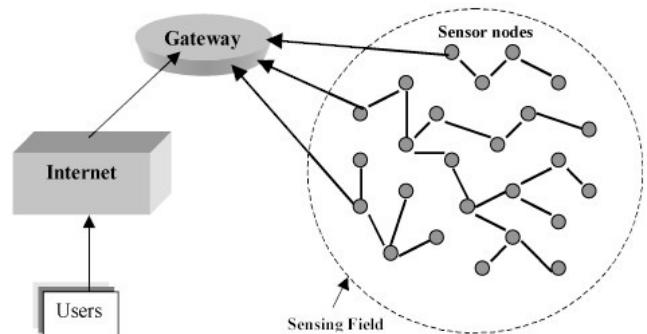


Figure 1. Simple Wireless Sensor Network

Sensor Networks face various kinds of issues which normally don't occur in other sorts of networks [1-4] like power constraints, resource constraints, hardware availability, low communication range, limited processing and storage, cost etc.

As the cost, time and complexity involved in the deployment as well as implementation in such networks is very high, so developers preferably like to get first-hand information on feasibility and reflectivity which is very important for implementation of system before hardware

\* Corresponding author:

anand\_nayyar@yahoo.co.in (Anand Nayyar)

Published online at <http://journal.sapub.org/jwnc>

Copyright © 2015 Scientific & Academic Publishing. All Rights Reserved

implementation.

So, nowadays various researchers are coming up with new protocols, new techniques and algorithms to overcome the challenges surrounding sensor networks to make it a reliable network and find more possibilities in deploying it in more areas of real world [124]. So, the evaluation and analysis of the techniques being proposed is utmost necessary and desirable. But testing and evaluating the protocols or theories proposed is not practically feasible through real experiments as it would be more complex, time consuming and even costly.

So, to overcome this problem, "SIMULATORS, EMULATORS and TESTBEDS [5] are effective tools to test and analyze the performance of protocols and algorithms proposed. With the help of simulation tools we can easily develop and test a new protocol for sensor networks. There are numerous advantages connected to this approach like easy implementation, low cost, analysis of real time results and even knowing the side effects and other effecting parameters on entire network.

The main objective of this research paper to provide comprehensive review of various Simulators (General Purpose Simulators and WSN Specific Simulators) which are available for researchers to do advanced research on Wireless Sensor Networks.

The Research Paper is organized as: Section II represents Wireless Sensor Network Simulator Architecture, Section III represents Classification criteria for evaluating sensor network simulators, Section IV highlights various Simulators for Wireless Sensor Networks, Section V represents the comparison of various simulators and Section VI presents conclusion and future scope.

## 2. Wireless Sensor Network (WSN) Simulator Architecture

There has been dramatic change in the architecture of WSN simulator from the traditional times to current years. Nowadays, the simulators include new and updated components in terms of energy, power consumption, graphical user interface, and inbuilt protocols cum results interpretation. WSN network is always application specific because the environments in which the sensors are deployed varies from condition to condition. A thorough test is being required so that to understand the conditions in which the sensor nodes are deployed in order to accurately estimate the node to node interaction [4].

WSN Simulator comprise of following modules: [4] [6]

1. Event
2. Medium
3. Environment
4. Node
5. Transceiver
6. Physical Protocol
7. MAC Protocol

### 8. Routing Protocol

### 9. Application Layer

**Event:** Event lays the primary foundation for functioning of all events and is regarded as abstract class. It comprise of: how event should work, provides procedures to compare events on the basis of their respective start time and whether there is some equality in events and abstract method to fire the event.

**Medium:** Medium defines the way of communication between the nodes. With medium, the working of nodes is determined. Medium must be informed regarding the presence of nodes, their respective position and other properties like transmitted power or receiver sensitivity. Medium contains the properties of bandwidth and wavelength of medium modelled and a reference to a propagation model at the time of construction. This model will lay the foundation to determine the signal strength transmitted and received by nodes.

**Environment:** Environment module has similar functionality like medium module. Environment determines the physical properties of model. In terms of WSN network, physical environment include: light, temperature, humidity, sound, coverage area, magnetic field etc.

**Node:** Node represent the single node in sensor network. As WSN network comprise of various set of nodes which are communicating among themselves via some routing protocol. Node module contains both hardware and software properties of node. Node component includes various properties like: processor, transceiver, sensor, energy source, routing protocol and applications.

**Transceiver:** This module basically covers the hardware transceiver of sensor node. It covers the transceiver states i.e. Sleep, Standby, Receive and Transmit along with behavior of nodes cum power consumption level of nodes. It generates events for the beginning and ending of every signal it transmits. All these events are exchanged with medium module instance.

**Physical Protocol:** Being the bottommost layer of network stack, it is mostly implemented in hardware of transceiver. The physical layer provides following services:

- Changing the state of transceiver;
- Carrier Sensing;
- Sending and Receiving Packets;
- Detecting energy on receipt of network packets;
- Channel changing on physical layer which support multiple channel.

**MAC Protocol:** MAC protocol layer lies between Physical Layer and Routing Layer of network stack and is mostly implemented on software operating in node's processor. In order to provide reliability and efficiency for WSN network the major work is done by MAC layer.

MAC layer is responsible for following services:

- Channel access policies
- Scheduling
- Buffer management

- Error management
- Sending and Receiving of packets between nodes.

In WSN, we need a MAC protocol to consider energy efficiency, reliability, low access delay and high throughput as major priorities to accommodate to sensor's limited resources and to avoid useless power consumption.

**Routing Protocol:** Routing Protocols comprise of various routing techniques for sending data between sensor nodes and base stations. The Routing Protocols in terms of Wireless Sensor Networks can be classified as follows:

- On the basis of Functioning and Applications: Proactive, Reactive and Hybrid Protocols
- On the basis of Participation Style: Direct Communication, Flat and Clustering Protocols
- On the basis of Network Structure: Hierarchical, Data-Centric and Location Based Protocols.

**Application Layer:** Application layer comprises of various application-layer protocols which perform various sensor network applications like query dissemination, node localization, time synchronization and network security. This layer is also responsible for sensor nodes traffic management and provides software support for various applications which translate the data in an understandable form or transmit queries to obtain certain information.

### 3. Classification Criteria for Evaluating Sensor Network Simulator

#### A. Types of Simulations Surrounding Wireless Sensor Networks

The following are three types of simulation:

- Monte Carlo Simulation
- Trace-Driven Simulation
- Discrete-Event Simulation

Wireless Sensor Network Simulators use Trace-Driven as well as Discrete Event Simulation.

**Discrete Event Simulation:** Discrete Event Simulation [7] [8] is mostly used in Wireless Sensor Network mainly because of its ease in simulating various jobs running on various sensor nodes. Discrete Event Simulation lists pending events which can be simulated by routines. The global variables lists the simulation time which enables the scheduler for advance time prediction. Simulation includes various routines like input, output, and initial as well as trace. The simulation provides the facility of Dynamic Memory management which facilities the addition and deletion of various entities in simulation along with facility of debugger to check the code side by side without interrupting the normal operation of simulation.

**Trace-Driven Simulation:** Trace-Driven Simulation [7] mostly plays a crucial role in real time applications. It provides more detailed information which lets the users to have in-depth detail of the simulation model. But side by side

trace-driven simulation contains various drawbacks: Sometimes in-depth details increases the complexity of simulation.

#### B. Evaluation Parameters for Simulators

As lots of simulators are available these days for simulating sensor network. To determine the suitable simulator as per user requirements is not an easy task. So, a criteria [9] [10] [4] to evaluate and compare the simulators is required. The following are some Evaluation Parameters:

1. Type of Simulator: [11] Simulators for Wireless Sensor Networks are classified in following three categories:
  - **Generic Simulator:** Generic Simulators are especially designed for simulating networking areas. In this simulator, the user creates code in High Level Language which is entirely different as found in real sensor network. Generic level simulator focuses on various aspects of WSN like networking, sensing and data processing. These types of simulators are also used for evaluating new communication protocols but regarded as less reliable due to mismatch in code which is written in simulator as compared to real sensor nodes.
  - **Code Level Simulator:** Code Level simulators use the same code in simulator as used in real sensor. The code is written and compiled in the machine running simulator which is regarded as faster operation as compared to sensor node. They are OS specific as driver code needs to be replaced when the same code is installed in sensor node. Code Level Simulators are very efficient in locating bugs with regard to timing and hardware architecture.
  - **Firmware Level Simulator:** Firmware Simulators are based on sensor nodes emulation and software which runs in simulator and is also the same in case of real sensor. Provides more details as compared to generic and code level simulator but side by side, firmware level simulators are also very slow in their operation with regard to code execution.
2. License: Depending on simulator to simulator, some are commercial while some are open source and are available free of cost to download and distribute.
3. Platform: Operating system on which the simulator operates. It can be Windows or Linux while there are some simulators which can operate on both windows and Linux distributions.
4. Usage Popularity: Top Page rank hits in various search engines by searching with "Simulator Name" along with the purpose for which we are looking.
5. WSN Platforms: Various sensors types and platforms which can be simulated in that simulator.
6. Ease of Code: It is also regarded as important criteria for evaluating depending how far the high level language programming is easy to program and how quickly the simulation scenarios can be created with

less code and less space. WSN is also bounded to programming languages effectiveness.

**Graphical, Debug and Trace Support:** (A): Graphical: Whether the simulator has graphical user interface or not; (B) Debugging Aid: How quickly the simulator can detect the error or bug in programming by the user. (C): Trace Support: Representing results in form of graphs. Example: XGraph tool in NS-2 Simulator.

## 4. Simulation Tools for Wireless Sensor Network

To assist researchers in choosing correct simulator to develop new algorithms, protocols and techniques in Wireless Sensor Network, a detailed explanation of various general purpose and specific purpose simulation tools is presented.

Simulation Tools for Wireless Sensor Network is as follows:

### 1. NS-2 (Network Simulator-2) [12] [13] [14]

Network Simulator 2 (NS-2) is regarded as discrete event simulation tool and has proved its worth in research of dynamic communication networks. NS-2 was developed in the year 1989 and since its inception various contributions are done that has brought various revolutions in the field of network research. The foundation of this simulator was based on REAL network simulator developed by University of California and Cornell University. Since 1995, DARPA also supported the NS-2 development via VINT (Virtual InterNetwork Testbed) and now National Science Foundation (NSF) is also contributing a lot to its development.

NS-2 is totally based on Object Oriented (OO) programming so it is also known as Object Oriented Discrete Event Simulator. It consists of two languages: C++ and Object oriented Tool Command Language (OTcl). C++ is primarily used for implementing various protocols and extending simulation libraries whereas OTcl scripts does the task of configuring simulator, network topology setting, creating network scenarios and displaying simulation results. C++ and OTcl are binded together using TclCL.

NS-2 comprise of 3,00,000 lines of code and is available free of cost and is used globally in academia. It can run on various operating systems like Linux, FreeBSD, MAC OS X, Solaris and even windows via use of third party software called Cygwin. The latest version is 2.36.

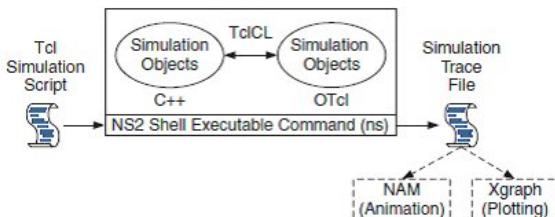


Figure 2. Basic Architecture of NS-2 Simulator [14]

With regard to Wireless Sensor Network, NS-2 provides support for various protocols like 802.11, 802.16, 802.15.4, IR-UWB etc. [9]. But even though of lots of contributions from varied researchers around the world, NS-2 faces serious drawbacks in terms of WSN simulations [5] like: Sensing model doesn't exist. The parameters which are used during simulation of nodes in WSN like energy model, packet formats, and MAC protocols are entirely different as we use in real world sensor network scenario. Another drawback surrounding NS-2 simulation results is that it also lacks application support which is required due to sensor network interaction between application level and protocol level.

NS-2 supports dual output which can be either text-based or graphical based. For graphical based simulation NS-2 has inbuilt tool i.e. NAM (Network Animator) which shows live movement of packets in the nodes, node position and live simulation scenario and also contains XGraphs which shows the graphical analysis of the results drawn at the end of simulation.

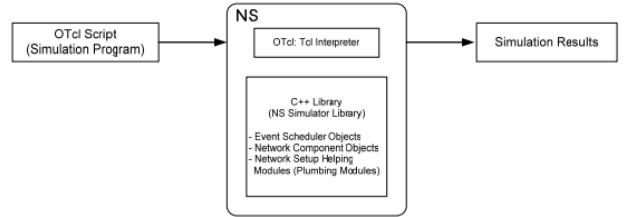


Figure 3. Basic Structure of NS-2 Simulation

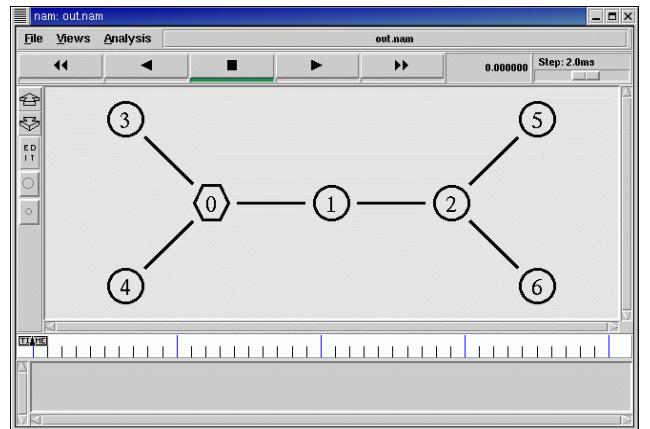


Figure 4. NAM (Network Animator in NS-2)

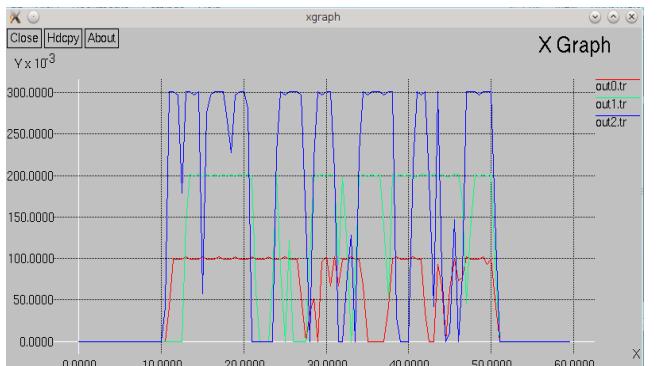


Figure 5. XGraph Tool in NS-2

## 2. NS-3 (Network Simulator-3) [15] [42]

Like NS-2, NS-3 is also regarded as discrete-event simulator. The objective behind its development was to enhance research in communication networks. NS-3 is open source simulator and was launched in June 2008. The latest version is 3.21 released in August 2014.

NS-3 is not regarded as extension of NS-2 simulator, NS-3 is a new simulator not supporting any API's belonging to NS-2.

As programs written in NS-2 are coded in OTcl and results can be visualized using NAM and XGraph but pure C++ code is not possible in NS-2. But in NS-3, all the programs are written in pure C++ with optional python bindings. There is no Graphical Tool which is available in NS-3, but still graphical results can be interpreted using NetAnim open source software.

For wired topology, NS-3 provides device model of a simple network of Ethernet which uses CSMA/CD as its protocol scheme with exponentially increasing back-off to contend for the shared transmission medium. With regard to Wireless Sensor Networks, various modules like 802.15.4, 6LoWPAN and RPL are already integrated in NS-3.

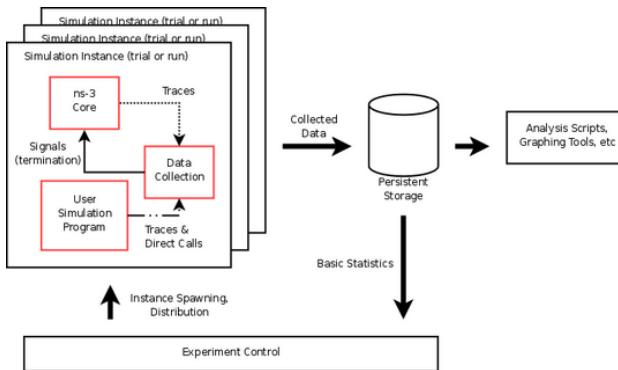


Figure 6. Architecture of NS-3 Simulator

NS-3 can also work in various operating system platforms like Linux and Windows via Cygwin.

## 3. OMNeT++ [16] [9] [5] [4] [17] [20]

OMNeT++ [16] is another powerful object oriented discrete event network simulator for wireless sensor networks. It is regarded as extensible, modular and component based C++ simulation library framework for building wireless network simulations. Basically OMNeT++ is not a simulator but it provides frameworks and tools for writing simulation scenarios. It is compatible to run on different operating systems like Windows, Linux and MAC OS X. The current version of OMNeT++ is 4.6 which was released in July 2014 with primarily focused on improving the usage of Graphical Runtime Environment (TKenv). TKenv GUI is re-engineered for single-window mode to improve the user experience and usage of simulator. The primary role of TKenv is to show the animation of network model, movement of nodes, displaying results, visualization changes etc. OMNeT++ is free to download for education and research purposes. Company has also got licensed

version called “OMNEST” and license is available via Simulcraft Inc.

The basic elements which make up OMNeT++ are called modules. Modules are of three types: (A): Simple Module: It is written in C++; (B): Compound Module: The module which links other modules via connections; (C): Network Module: Top level compound module is called network module.

OMNeT++ includes Eclipse based IDE environment which enables C++ programming and debugging of modules as well as graphical and textual editing of NED Files. NED (Network Description) lets the user declare simple modules and connect and assemble them into compound modules.

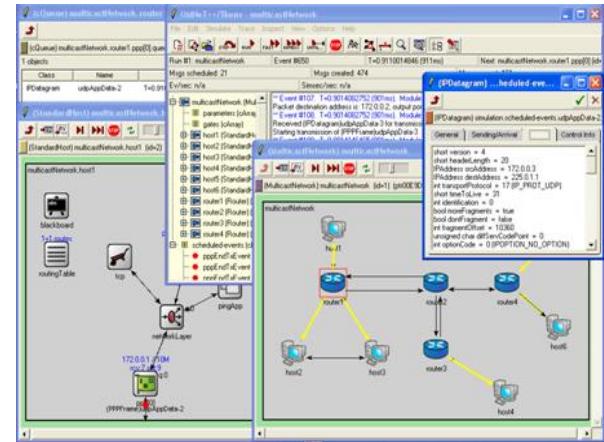


Figure 7. OMNeT++ Simulator IDE Interface

The following is the basic NED code which is being written to make the topology in OMNeT++:

```
//  
// A network  
//  
network Network  
{  
    submodules:  
        node1: Node;  
        node2: Node;  
        node3: Node;  
        ...  
    connections:  
        node1.port++ <-> {datarate=100Mbps;} <->  
        node2.port++;  
        node2.port++ <-> {datarate=100Mbps;} <->  
        node4.port++;  
        node4.port++ <-> {datarate=100Mbps;} <->  
        node6.port++;  
        ...  
}
```

NED Code for Topology Description

The Simulator which is built primarily for Wireless Sensor Network in OMNeT++ is CASTALIA.

CASTALIA [18] [19]

Castalia, a popular Wireless Sensor Network, Body Area

Networks (BAN) simulator which is made especially for low power embedded devices. The simulator is primarily being used by researchers to simulate algorithms and protocols in real time wireless channels and radio models and to see the real time node results and study interpretations.

Castalia is not regarded as sensor specific platform rather it was developed to provide real time results of the algorithm being developed by the researcher on any specific sensor platform. The latest version of Castalia is 3.2 which was released in March 2011.

Castalia provides the following:

- Channel Model to study mobility of nodes, study the path between the nodes and any interference level with regard to signal strength.
- Radio Model to interpret SINR, packet size, module type and carrier sensing.
- MAC and Routing Protocols study on nodes.
- Monitoring CPU power consumption.

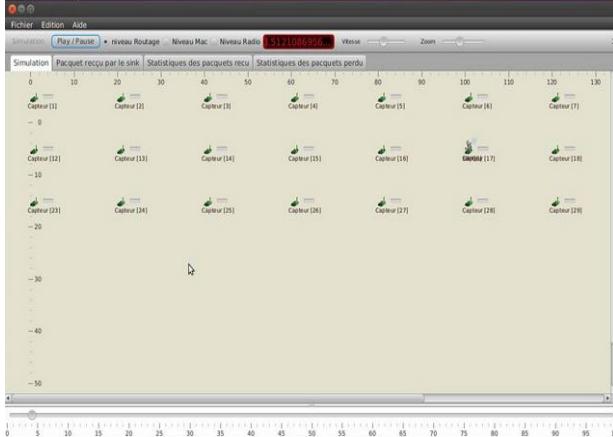


Figure 8. Castalia GUI for WSN Simulation

#### 4. J-Sim [21] [22] [23] [24] [25] [5] [17]

J-Sim (JavaSim) Simulator is regarded as general purpose simulator and is based on component based software architecture, Autonomous component architecture (ACA). J-Sim has been developed in Java and is tightly coupled with ACA making J-Sim a platform neutral, extensible and reusable environment. J-Sim has a Script Interface which facilitates execution of various script based languages like Perl, TCL or Python. The latest version of J-Sim is 1.3 + patch 4 which facilitates J-Sim with a Java implementation of TCL interpreter with TCL/Java extension known as Jacl.

J-Sim has a sharp edge over NS-2 simulator as unlike in NS-2, classes/methods/fields in Java need not be explicitly exported in order to be accessed in TCL environment; rather all the public classes/methods/fields in Java can be accessed in TCL environment.

J-Sim consists of the following packages:

- Base Package
- .NET Package
- INET Package
- Diffserv Package

- Intserv Package
- RUV System
- Common Class Library
- J-Sim Total
- Third Party which consists of: Jacl; Jython; PtPlot

The following figure shows the GUI of J-Sim Simulator:

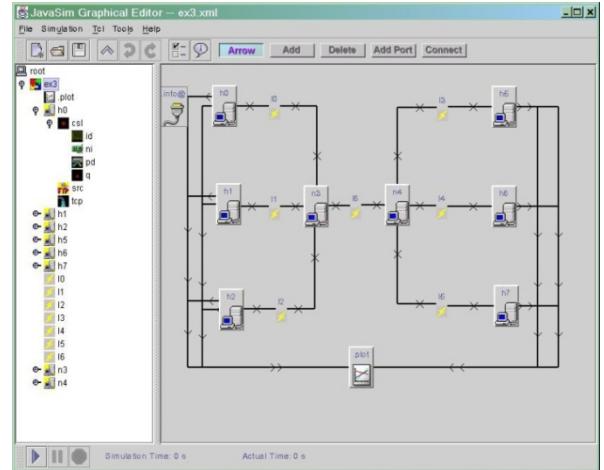


Figure 9. J-Sim GUI

J-Sim provides a framework for Wireless Sensor Network Simulation which has been built upon ACA and INET and Wireless Protocol stack and has following components:

- a. Target, Sensor and Sink Nodes.
- b. Sensor Channels and Wireless Communication channels
- c. Physical media such as Seismic channels, mobility models and power models.

Considering simulation of Wireless Sensor Networks, J-Sim has used to implement three protocols of WSNs: Localization, Geographic Routing and Directed Diffusion [24]. As compared to NS-2, J-Sim simulator has various advantages in terms of execution time, memory allocation, and scalability. Moreover it provides good platform to carry out large-scale WSN simulations with more than 1000 nodes which is somewhat not possible in NS-2. J-Sim provides framework for several classes for simulating sensor network in terms of coverage, time indexing and power management.

#### 5. Mannasim [26] [27]

Mannasim Simulator is basically a module for WSN simulation based on Network Simulator 2 (NS-2).

Mannasim, a wireless sensor network simulator is comprised of two components:

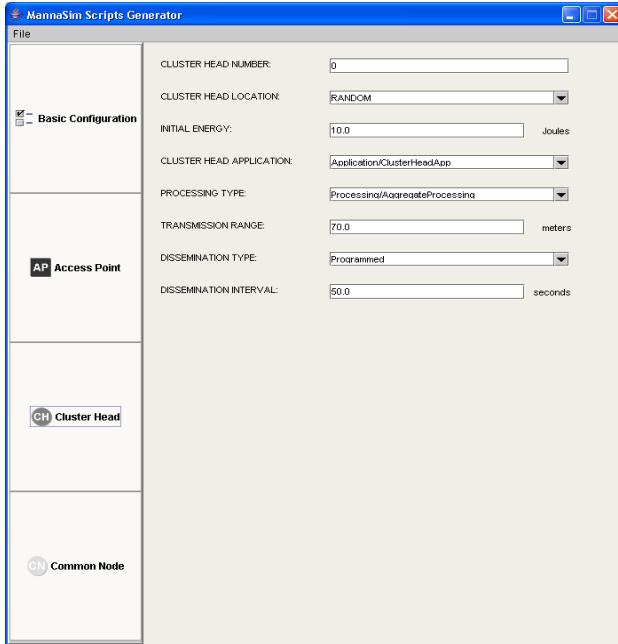
- a. Mannasim Framework
- b. Script Generator Tool

**Mannasim Framework:** Mannasim framework is basically an extension of NS-2 simulator and provides new modules for design, development and analysis of various Sensor network applications.

**Script Generator Tool (SGT):** Mannasim Script Generator Tool provides easy to use front end for creation of TCL simulation scripts. It is part of Mannasim framework and is

fully written in Java Language making it platform independent.

The following figure highlights SGT tool of Mannasim:



**Figure 10.** Mannasim Script Generator Tool

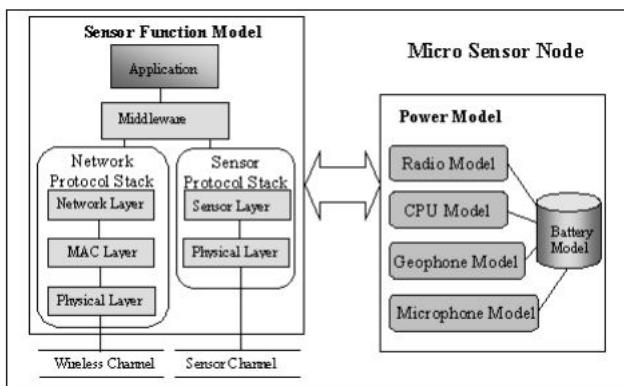
In case of Wireless Sensor Network scenario, Mannasim provides a detailed simulation framework for accurate modeling of sensor nodes and comprehensive testbed for simulating various algorithms and protocols of Wireless Sensor Networks.

It is regarded as highly signified tool for WSN to simulate the nodes and analyze the performance of nodes on various algorithms. Various WSN routing protocols like AODV, DD, DSDV, DSR, LEACH, and TORA can be simulated.

Mannasim can be installed along with ns-2.29 patch and is open source.

## 6. SensorSim [5] [28] [29] [30]

SensorSim, an another framework for simulating wireless sensor network is based on NS-2 simulator with various add on features like: (a) Sensing channels and Sensor Models (b) Battery Models (c) Lightweight protocol stacks for wireless microsensors (d) Scenario Generation (e) Hybrid Simulation.



**Figure 11.** MicroSensor Node Model

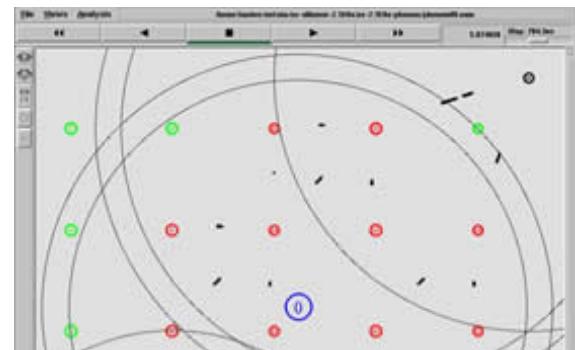
The basic objective behind the development of SensorSim is to provide deeper understanding of sensor networks and to provide a deep framework for development of new protocols cum applications and performance evaluation techniques for sensor networks. SensorSim has inherited various features from other event driven simulators having hybrid simulation mode which facilitates new features like interaction of real and simulated nodes, new communication protocols and real time user interaction with GUI interface.

Apart from various advantages and new features for simulating WSN network, SensorSim has scalability problem. And due to inability of the developers to provide support for public release, SensorSim today is not currently available to the public.

## 7. NRL Sensorsim [13] [31] [32] [27] [33]

NRL's Sensorsim is basically an extension to NS-2 simulator for simulating Wireless Sensor Network and was developed by The Naval Research Laboratory group. NRL Sensorsim supports simulation and detection of Carbon Monoxide concentration, Seismic activity and audible sound. NRL Sensorsim also has the feature to support energy consumption model in sensor network. NRL Sensorsim is now no longer in development and no support is being provided to public.

The following diagram shows the GUI of NRL Sensorsim:



**Figure 12.** NRL Sensorsim GUI

## 8. NCTUns 6.0 [6] [34] [35] [36] [37] [38]

NCTUns- (National Chiao Tung University Network Simulator) - An Open Source Discrete event network simulator cum emulator software runs on Linux and provides easy to use integrated GUI environment for network planning and research.

The main components of NCTUns Simulator are: GUI, job dispatcher, simulation engine, patched Linux kernel, applications and daemons.

- GUI provides user interface to create and control the simulation in an interactive manner.
- Job dispatcher monitors the status of various machines which are participating by receiving status reports sent from coordinator programs running on them.
- Coordinator Program monitors the machine status that whether the machine is busy or idle and in turn send the report to Job dispatcher.

The following figure shows the architecture of complete NCTUns Simulator:

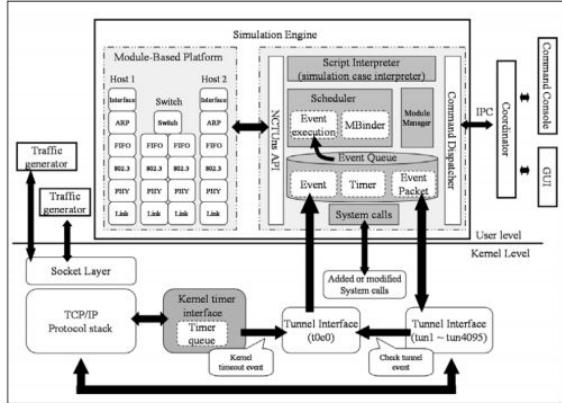


Figure 13. NCTUns Simulator Architecture

The latest version release of NCTUns is 6.0 which was released in May 2010.

The following are the features of NCTUns Simulator:

- Uses real world Linux TCP/IP protocol stack to generate high end simulation results.
- Can use any existing or in-development UNIX program as Traffic Generator program and needs no modification.
- Can use real world UNIX network configuration and monitoring tools.
- Simulation environment is almost the same as Real world IP networks.
- Facilitates simulation of various networks like WSN, MANETS, Optical networks, GPRS, DVB/RCS Satellite network, WiMAX network, IEEE 802.11p/1609 VANETS.
- Can simulate almost every network device as per IEEE standards may be Wired or Wireless device.
- Has ability to simulate large quantity network protocols like CSMA/CS, HTTP, FTP, RIP, OSPF, 802.11 etc. to name a few.
- High simulation speed

The following figure shows the GUI interface of NCTUns Simulator:

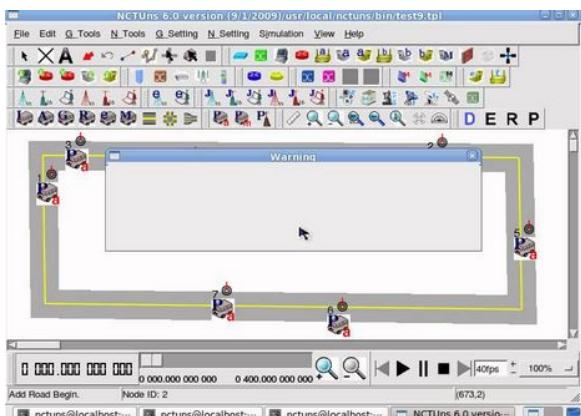


Figure 14. GUI interface of NCTUns Simulator

With regard to WSN Simulation, NCTUns Simulator efficiently simulates various routing protocols like AODV, DSR, and DSDV.

## 9. SSFNet [6] [39] [40] [41] [42]

SSFNet-Scalable Simulation Framework Network Models is an open source simulation software for simulating networks especially Wireless Sensor Networks. It is basically designed for simulating various network scenarios like network topology, protocols and traffic and also enables simulation of Wide Area Network like Internet.

SSFNet is a complete Java based implementation of SSF engine along with various highly scalable Internet protocols like IP, TCP, UDP, BGP, OSPF etc. and large network elements like Routers, Switches, Links, LAN's etc.

SSFNet models are regarded as "Self-Configuring" which means each SSFNet class instance can configure itself separately by querying the parameters in the database which may be local or over the web.

SSFNet models provide components for simulating networks at IP level and above and include models for hosts, router, links and a framework for modeling protocols.

The following figure shows the Simulation Layers of SSFNet Simulator:

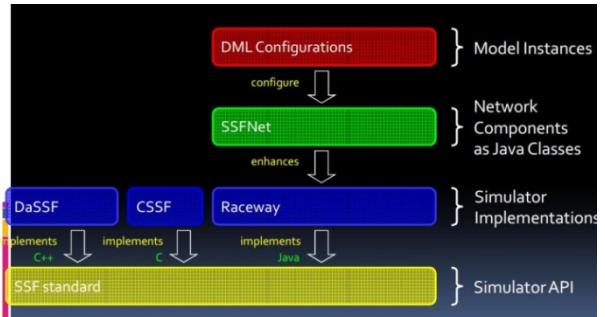


Figure 15. Simulation Layers of SSFNet

SSFNet is basically divided into the following four packages:

- SSF.OS- It contains the framework for protocols models that are related to operating system.
- SSF.Net- It includes the hardware for modelling network objects such as hosts, routers, links etc.
- SSF.Util.Random- It contains additional classes for generation random stream.
- SSF.Util. Streams- It contains the support for efficient multiple-point network, monitor the infrastructure for collecting stream data.

The current release of SSFNet simulator is 2.0 was done on January 15, 2004.

## 10. GloMoSim [42] [43] [44] [17] [4] [45] [5]

GloMoSim (Global Mobile Information System Simulator) is discrete event scalable simulation software that simulates wireless and wired network systems and is built using PARSEC simulation environment. GloMoSim as well as PARSEC were developed by Parallel Computing Lab at UCLA.

A layered approach is being used by GloMoSim to carry out simulation between different layers and a Standard API is being used so that new models and layers can be exchanged and integration can be facilitated.

Two different node mobility models: Random Waypoint Model and Random Drunken Model is being supported by GloMoSim. When the nodes in the network move according to a model, it is known as Random Waypoint model.

When a node in the simulation environment chooses random destination and moves to the location as per the speed specified in configuration file it is known as Random Drunken Model.

The following Table 1 [44] shows GloMoSim models at major layers of OSI model:

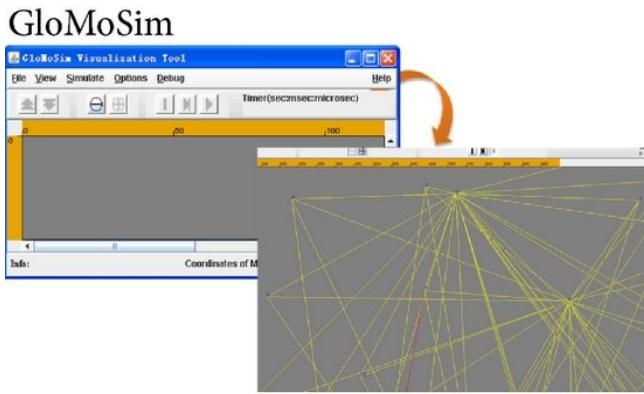
**Table 1.** Models in GloMoSim Simulator

Layer	Model
Physical Layer	Free Space, Rayleigh, Ricean, SIRCIM
Data Link Layer	CSMA, MACA, MACAW, FAMA, 802.11
Network Layer	OSPF, DSR, WRP
Transport	TCP, UDP
Application	Telnet, FTP

GloMoSim was non-commercial simulator and the current version which is available these days is version 2.03. But now a Commercial version of GloMoSim has been evolved with name “*QualNet*”.

With regard to Wireless Sensor Networks, basic routing protocols like AODV, DSR is available in it. But the current version of GloMoSim doesn't offer any sensor network features. But GloMoSim provides good platform for simulating Mobile IP Networks.

The following figure shows the GUI interface of GloMoSim simulator:



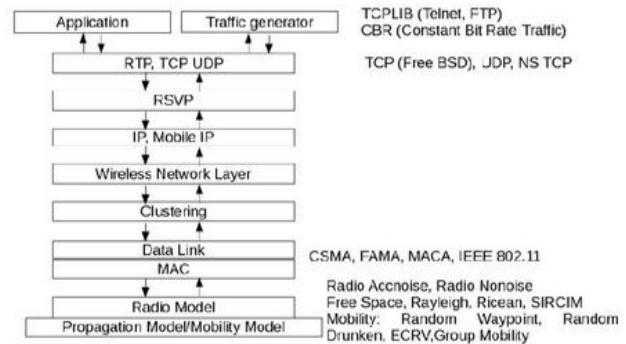
**Figure 16.** GUI Interface of GloMoSim Simulator

The following figure highlights the architecture of GloMoSim Simulator:

PARSEC (PARallel Simulation Environment for Complex Systems) [46] [44].

PARSEC (PARallel Simulation Environment for Complex Systems) was designed by Parallel Computing Laboratory, UCLA. It is basically a C-Based simulation

language for sequential and parallel execution of discrete-event simulation models.

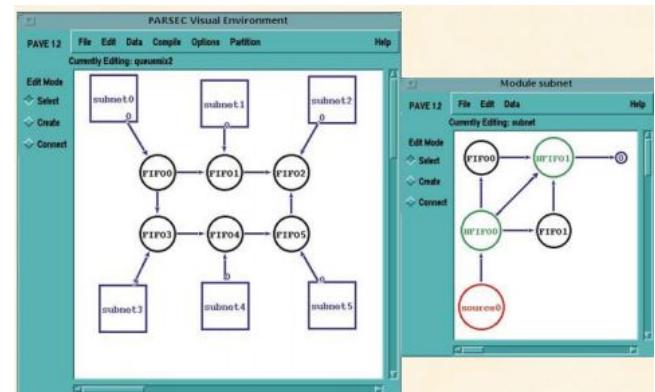


**Figure 17.** GloMoSim Architecture

PARSEC supports both Windows and UNIX platforms. PARSEC is based on Maisie Simulation language with following major modifications:

- Simple Syntax.
- Usage of modified language for posting code from simulation model to operational software.
- Robust and Efficient runtime kernel.
- Integration of new protocols for parallel performance prediction.

The following figure gives a visual representation of PARSEC Environment:



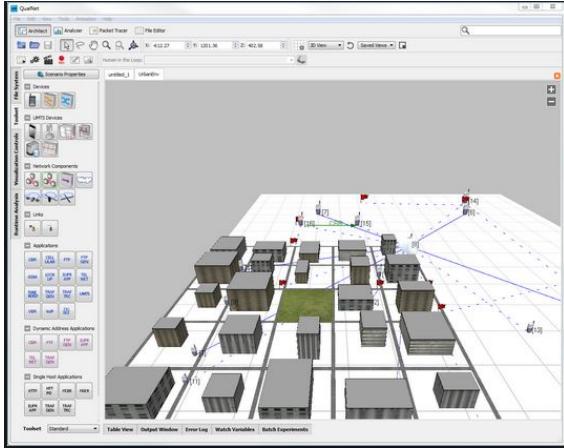
**Figure 18.** PARSEC Visual Environment GUI

## 11. QualNet 7.0 and EXata 5 [42] [17] [48]

QualNet-(Quality Networking) a commercial network simulation software from Scalable Technologies is basically derivative product from GloMoSim Simulator. QualNet enables high fidelity “Virtual” model of network which consists of routers, switches, servers, access points, radios, antennas, computers and any other equipment's along with various protocols to enable packet movements across the network scenario.

QualNet 7 and EXata network simulator is supported across various platforms like Windows 7/8/8.1, Centos 5.9, RHEL 6/7 and Ubuntu 12.04 version onwards and was released in October 2013.

The following figure shows the GUI interface of QualNet 7 Simulator:



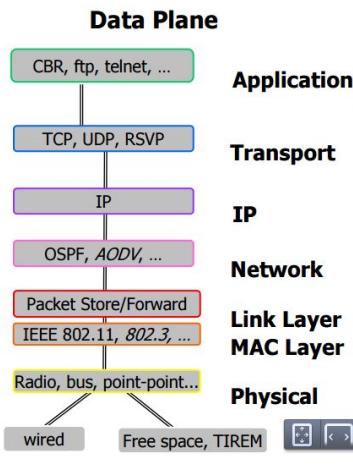
**Figure 19.** GUI Interface of QualNet 7 Simulator from Scalable-Techologies

The following Table 2 highlights the components of QualNet Simulator:

**Table 2.** Components of QualNet Simulator

Component	Description
1. Architect	A GUI Design and Visualization tool to setup network scenarios, connections, users and even nodes. Architect tool enables in-depth visualization and network topology analysis. It also enables showing of real-time statistics via graphs depending on the simulation running.
2. Analyzer	A Graphical Statistical tool used for displaying various parameters of simulation undergoing on. All the statistics can also be exported to MS-Excel via CSV format.
3. Packet Tracer	GUI Tool for showing visual representation of packet trace files which are generated during network simulation. Basically text files in XML form and shows the details of packet movements across various layers of OSI model.
4. File Editor	Tool basically used for text editing.
5. Command Line Interface	Enables CUI interface to the simulator.

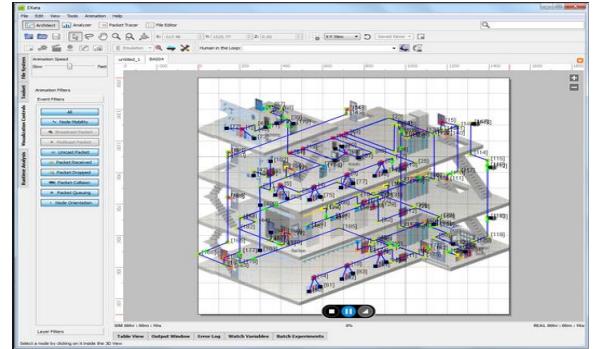
The following figure shows QualNet Layer Model:



**Figure 20.** QualNet Layer Model

**EXata:** A simulation platform which contains all functionality of QualNet with a real-time network emulation interface which enables the users to integrate hardware and run applications on the simulation environment. EXata enables live simulation of cyber vulnerabilities and threats so that research can be undertaken by users to prevent network security against outside attacks.

The following figure demonstrates the GUI interface of EXata Simulator:



**Figure 21.** GUI Interface of EXata Simulator

The following are the features which are included in QualNet 7 and EXata 5 making them as the most advanced simulator platform these days:

- IPv6 Emulation Support
- IGMPv3 Support
- DHCP Model
- DNS Model
- FQDN (Fully Qualified Domain Name) in Applications
- IPv6 Auto configuration Model
- Channel Names
- Federate Extractor
- HLA 1516
- RPR FOM 2.0017

## 12. sQualNet Simulator [49-52]

sQualNet-Scalable Quality Networking a scalable network simulator based on QualNet. sQualnet has wide suite of sensor network specific models i.e. Sensing and Radio Channels, Sensor Protocols (MAC, Routing), battery and power consumption models along with support for multi-tiered sensor network evaluation. sQualnet has an added edge over various other simulators like NS-2, OPNET, QualNet and other sensor specific simulators is that it has ability for real-code simulation. It also has advantages over emulators like TOSSIM, ATEMU in terms of hybrid simulation which provides assistance in large scale system development and evaluation at low cost.

It is also regarded as kernel efficient protocol which allows experiments to run at higher scalability in both contexts i.e. Spatial and Temporal.

The latest version of sQualnet available currently is 1.0 Beta. It was a project funded by National Science Foundation (NSF) and Defense Advanced Research Projects Agency (DARPA) with joint collaboration with Network and

Embedded Systems Laboratory (NESL) and Parallel Computing Laboratory at University of California, Los Angeles (UCLA).

The following figure shows the Capabilities of sQualnet Simulator:

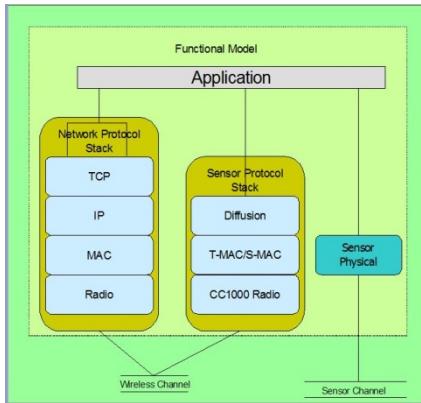


Figure 22. sQualnet Simulator Capabilities

### 13. OPNET Modeler Suite [53-60] [42] [17]

OPNET (Optimized Network Engineering Tools) was created by OPNET technologies in 1986. The Latest version of OPNET Modeler Suite is 18.0.3 which was recently released in April 2015. Now, OPNET is part of Riverbed Application and Network Performance Management Solutions.

OPNET tool is a powerful collection of tool sets to create and test large network environments because of its coverage

to following components of communication networks:

- Application Performance Management
- Planning
- Engineering
- Operations
- Research and Development

OPNET provides an interactive GUI interface to build network scenarios and enables its testing. Basically users can make use of drag and drop actions to create high end network environments of any sort in matter of seconds. OPNET Modeler and IT Guru are basically the packages used for network simulation.

OPNET Modeler is basically a commercial version available but IT Guru Academic Edition is basically a free software license given by OPNET as part of University Program throughout the world.

The following figure highlights the GUI interface of OPNET Modeler 18.0.3:

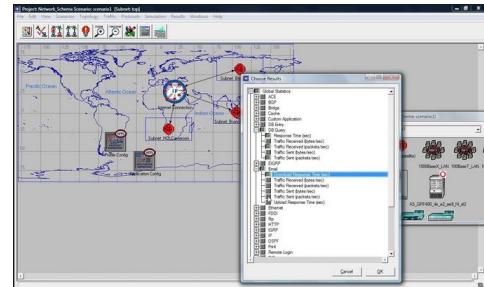


Figure 23. OPNET Modeler Interface

Table 3. List of Editors in OPNET Simulator

Editor Name	Description
Project Editor	Used for main staging for network simulation. Basically network model is built here using models from library, statistics choosing about network, simulation running and results viewing. We can create and process nodes, build packet formats, create filters and parameters in Project Editor.
Node Editor	Basically helps the user in defining the behavior of each network object. Network Object comprise of multiple modules which defines the behavior of node. Modules comprise of data creation, data storage etc.
Process Model Editor	It helps to create process models which are represented via Finite State Machines (FSMs). In this operations are being performed via C/C++ code.
Link Model Editor	It creates new types of Link objects which can have different attributes and representations.
Path Editor	It is used to create new path objects which define a traffic route. Logical Connections like MPLS, ATM or Frame relay are being used as path for traffic routing.
Demand Editor	It is used to define demand models and it determines attributes interfaces, presentation and behavior.
Probe Editor	It generates statistics of the simulation.
Simulation Sequence Editor	It helps to specify additional simulation constraints and contains a set of various attributes which control runtime simulation.
Antenna Pattern Editor	It controls the direction-dependent gain properties of antenna. This editor is only used in Wireless Simulations.
Filter Editor	It creates additional filters to build new filter models.
ICI (Interface Control Information) Editor	It defines the internal structure and used for interrupt-based inter-process communication.
Modulation Curve Editor	It plots BER and SNR.
Packet Format Editor	It defines internal structure of packet as set of fields.
PDF (Probability Density Function) Editor	It defines overall results and any sort of transmission errors.
Project Editor Window	Basically a GUI window for defining simulation and user can drag and drop components of various network devices on this window.

OPNET is a discrete event simulator and is available both in 32-bit as well as 64-bit for Windows and Linux Operating System. OPNET Riverbed modeler makes use of Object Oriented technology to create mapping from the graphical design to the implementation of the real system.

OPNET models the network in a hierarchical approach which somewhat matches to hierarchical structure of Internet: Networks, Satellites and Nodes. OPNET basically supports three types of Links: Point-to-Point, Bus and Wireless. Wireless Link is basically used in Wireless, Mobile or Satellite Network. Although, OPNET has various libraries for Wireless and Wired networks, it also supports Wireless Sensor Networks. OPNET basically comprise of three main tools: NetDoctor, ACE and MVI which enables the analysis of networks and proper future implementations.

OPNET 18.0.3 Modeler provides various editors to facilitate easy network simulation.

The following Table 3 shows the list as well as description of various editors in OPNET.

#### 14. SENSE [5] [24] [8] [27] [17] [61] [62] [63] [71]

SENSE: Sensor Network Simulator and Emulator a powerful sensor network simulator developed in 2004. The latest version available for SENSE is 3.0.3.

SENSE is a discrete event simulator written in C++ and its working is similar to J-Sim and runs on the top of COST, written in CompC++ (A component based extension to C++). The design of COST was basically influenced by mutual understanding of Component-based software architecture and component-based simulation.

With regard to Wireless Sensor Networks, SENSE has battery models, MAC Layer protocols like NullMAC, IEEE 802.11 with DCF, application, network and physical layer models. SENSE also includes a series of SSR Protocols like: Self-Selective Routing (SSR), Self-Healing Routing (SHR) and Self-Selective Reliable Path (SRP).

For results visualization and simulation animation, SENSE has iNSpect tool. The latest version of iNSpect tool is: 3.5.

The most important critical factors of SENSE are:

- Extensibility: Network simulation architecture is based on component based simulation. Component-Port model removes interdependency in simulation models as in OOPS architecture and then a simulation component classification is being used to solve the problem of managing simulation time.
- Reusability: Removal of interdependency between models lays foundation of reusability. In SENSE simulator, there is another level of reusability which was made possible by extensive use of C++ template: a component is usually declared as a template class so that it can handle different type of data.
- Scalability: Like SSFNet and GloMoSim, SENSE also includes the support for parallelization.

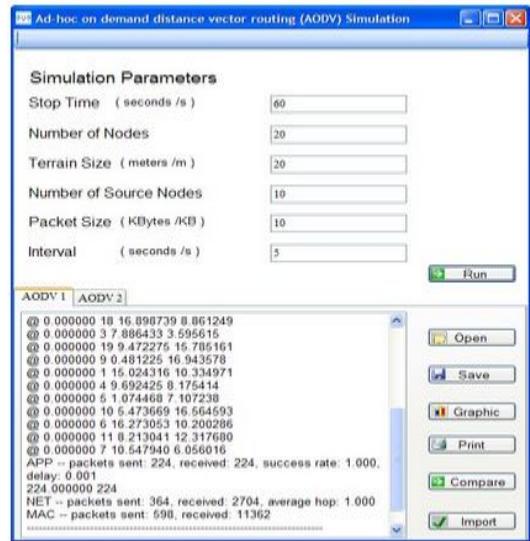
The following Table 4 shows the list of various components and simulation engines available right now in

SENSE Simulator:

**Table 4.** Components of SENSE Simulator

Component Name	Short Description/Ingredients
G-Sense [64] [65]	Guide User Interface for SENSE simulator especially for WSN simulation and includes various simulation features like parameter input, stop time, Number of nodes, Terrain Size, Number of Source Nodes, Packet Size and Interval.
Battery Model	Linear Battery; Discharge Rate Dependent
Application Layer	Random Neighbor, Constant Bit Rate
Network Layer	Simple Flooding, AODV, DSR, SSR, SHR
MAC Layer	NullMAC, IEEE 802.11 with DCF
Physical Layer	Duplex Transceiver
Wireless Channel	Free Space, Adjacency Matrix
Simulation Engine	CostSimEng (Sequential)

The following figure shows the GUI interface of G-Sense Visualization Tool for Simulating AODV Protocol:



**Figure 24.** AODV Simulation in G-Sense Simulator

SENSE, is still in development phase and developers are releasing new patches for its stability but still lots of new models and templates are to be developed with regard to Wireless Sensor Networks.

#### 15. DRMSim [42] [66] [67] [68] [69] [70] [71]

DRMSim-Dynamic Routing Model Simulator- A Discrete event simulation software, a joint research project venture by Alcatel-Lucent Bell, Université de Bordeaux (LaBri) and INRIA at Sophia Antipolis (Mascotte project) and is supported by Euler fire project whose main is to research and develop a new routing scheme suitable for future Internet.

As basic routing protocol i.e. BGP (Border Gateway Protocol) which drives today's Internet shows various limitations in terms of routing table entries which are stored locally, updated in timely fashion and dynamic exchange. It is right now somewhat practically impossible to deploy

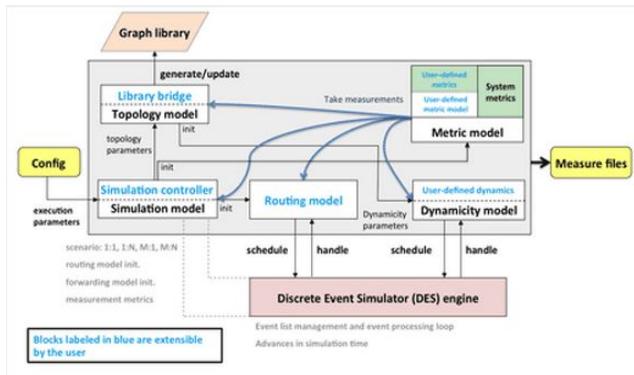
new-designed protocols on Internet at a very high-scale so before deployment, a valid simulation is prime requirement. So, the idea behind the development of DRMSim was to simulate routing models on large-scale networks. DRMSim enables simulation of large-scale network with minimum of 10000 nodes which in turn requires careful analysis of data structures that will be used on the network model as well on the granularity and time management of the simulation model.

DRMSim simulator is basically based on Java software so that quality of code can have extensibility and re-usability.

Till date, RIP, BGP and NSR routing protocols are being implemented in DRMSim simulator.

The latest version of DRMSim is 1.3.1 which was released on March, 2013.

The following figure shows DRMSim Architecture:



**Figure 25.** DRMSim Architecture

DRMSim has the following features cum properties:

- DRMSim is a discrete event simulator written in Java Language.
- DRMSim has modular design which means new topologies, routing protocols and dynamic models can easily be deployed and simulated in this simulator.
- DRMSim monitors each and every node on the network which means the data structure wise, this simulator is highly efficient.

DRMSim comprise of following components:

- Topology Generators like Brite, Inet, GLP etc.
- Import external topologies. E.g. CAIDA maps.
- Supports high end routing protocols like BGP, DSDV up to 16000 nodes.
- Metric Computation Module- Calculates performance metrics like routing path, routing table size, communication cost etc.

DRMSim is an active simulation project under three year EULER exploratory research project via Grant No: 258307 and currently the progress is going on Geometric Routing, Greedy Routing Models.

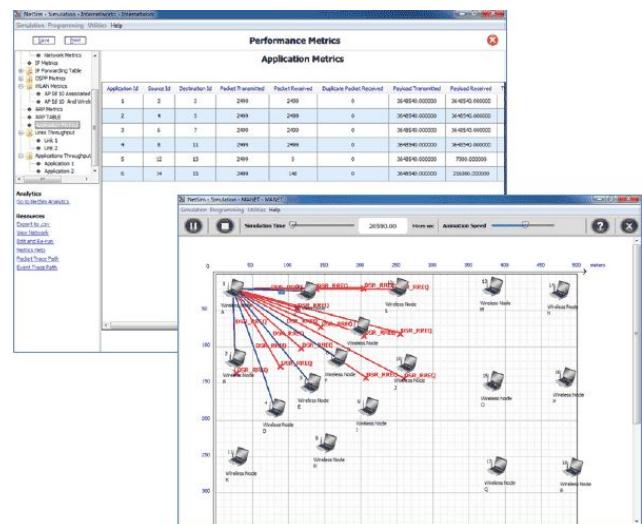
## 16. NetSim [42] [72] [73]

NetSim- Network Simulator is a Stochastic Discrete Event Simulator used for protocol modeling, Network Research

and Development and Defense Applications. NetSim provides a unique platform for simulating various types of networks like MANET, WSN, Cognitive Radio, Cellular-GSM, CDAM, Wi-Fi: 802.11 a/b/g/e/n/ac and even more. It plays a crucial role in developing and simulating new application models- Voice, Video, FTP, HTTP, Email and more. It also features detailed packet animation and information cum packet and event trace. It enables the users to do custom code development and debugging. NetSim provides both Command Level and Graphic Interface to enable users to deploy, simulate and analyze various network scenarios.

The Latest version of NetSim from Boson is version 7.2.

The following figure shows the screenshot of NetSim GUI along with Performance Metric:



**Figure 26.** NetSim GUI

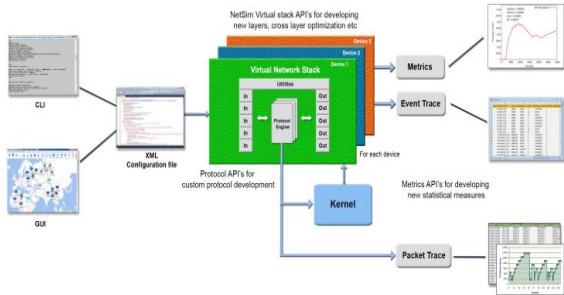
NetSim is available as Standard Version which is also known as Academic Version along with Advanced Version.

The following Table 5 demonstrates latest protocols cum functionalities which are available in NetSim v8 for various networks:

**Table 5.** Protocols Supported by NetSim Simulator

Type of Network	Protocols/Facilities
MANET	DSR, AODV, Mobility via Random Waypoint and Random Walk Models; Codes for various sorts of attacks like Node Failure, Sinkhole, Blackhole etc.
Wireless Sensor Network (WSN)	MAC Layer covering GTS, CCA, Slotted CSMA/CA, Unslotted CSMA/CA, Beacon transmission, LEACH Protocol, PHY Models, Output via Routing Overhead, Power, Lifetime, Packet Delivery Ratio etc.
Cognitive Radio	MAC Layer- DS-MAP, US-MAP, BW Request, Quiet Period, Normal Mode Operation, PHY, Connection Establishment, Others like Spectrum Manager, Sensing Function, Sensing automation and channel model
LTE	Network Layer, MAC Layer, PHY Layer, MIMO Model, HARQ Model
CCNA/CCNP Courses	Advanced functionalities- EIGRP, HSRP, VRRP etc.

The following figure highlights the architecture of NetSim 8 Simulator- Simulation Scenario:



**Figure 27.** NetSim 8 Simulation Architecture

### 17. UWSim [4] [74] [75] [76] [77]

UMSim- UnderWater Simulator, an open source tool with prime focus towards marine robotics research and development. UMSim was a joint venture development of RAUVI and TRIDENT research projects. As most of the current available network simulators mainly focus on ground-based sensor systems and Adhoc networks but underwater simulation is usually ignored.

UWSim primarily focus on handling research and simulation scenarios of underwater sensor environments e.g. Low bandwidth, low frequency, high transmission power and limited memory. UWSim is basically designed to handle network scenarios of underwater communication.

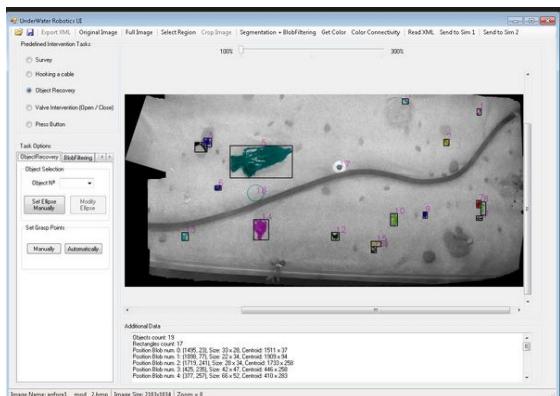
UWSim simulator has been implemented in C++ and makes use of OpenSceneGraph (OSG) and osgOcean. OSG is open source 3D graphics application programming interface used by developers in various applications like Virtual Reality, Gaming, Scientific Visualization, Visual Simulation etc. osgOcean is basically used for rendering the ocean surface and special effects.

Windows XP was the prime operating system taken for development of UWSim along with .net Framework 2.0.

The latest version of UWSim is 1.1 and only supports Ubuntu Operating system version 11.04 onwards.

UWSim can only simulate underwater sensor network and no other sensor network can be simulated using this software.

The following figure shows the GUI interface of UWSim Simulator:



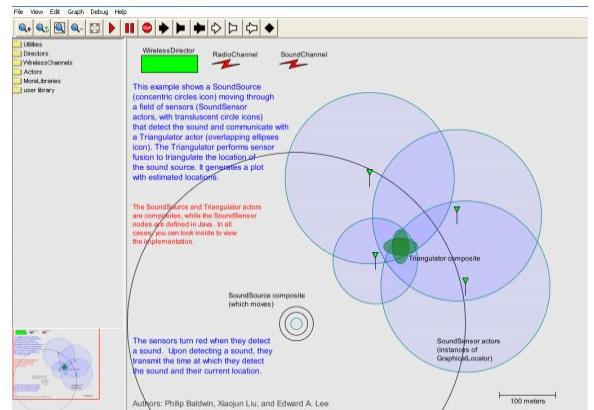
**Figure 28.** GUI Interface of UWSim

### 18. Visual Sense [78] [4] [27] [28] [71] [79]

Visual Sense- Visual Sense is a modeling cum simulation software for wireless and wireless sensor networks which was built on Ptolemy II. It supports component based modeling, representation and analysis of communication channels, sensors, ad-hoc networking protocols, localization strategies, media access control protocols and even energy consumption in sensor nodes etc. As an extension to Ptolemy it consists of new Java classes and XML files. It supports actor-oriented definition of network nodes, wireless communication channels, physical media such as acoustic channels and wired subsystems. The software architecture consists of set of base classes for defining channels and sensor nodes, library of sub classes which provide channel modes and node models along with visualization framework.

Visual Sense simulator is part of Ptolemy II and the latest version of Visual sense is 8.0.1 which is part of Ptolemy 8.0.1 release and was released on October 28, 2010.

The following figure shows the GUI interface of Visual Sense Simulator:



**Figure 29.** GUI Interface of Visual Sense

Visual supports sensor networks, antenna networks, terrain models and battery models. But it doesn't provide any protocols above wireless medium or any sensor or any network other than sound.

### 19. Viptos [28] [27] [80] [81] [82] [83]

Viptos (Visual Ptolemy and TinyOS) is an integrated graphical development and simulation environment for TinyOS based Wireless Sensor Network.

Viptos is basically an interface/bridge between TinyOS and Ptolemy II and provides a GUI interface for TinyOS developers.

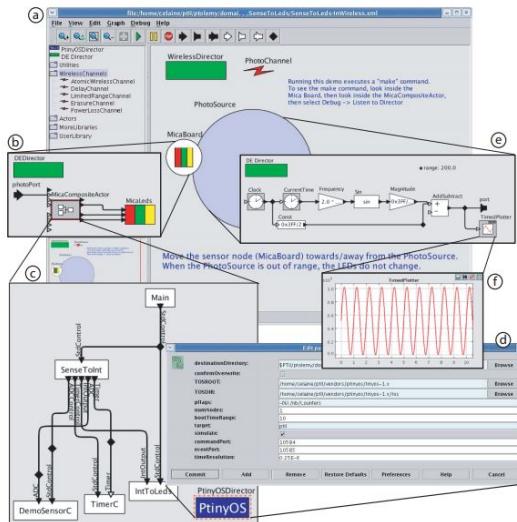
The following figure shows Wireless Communication Simulation in Viptos:

With the help of Viptos, the embedded system developers can construct block and arrow diagrams to create TinyOS programs written in nesC, C based programming language. The main advantage of Viptos simulator is that it can transform nesC program into TinyOS supported platform.

The Latest version of Viptos is 1.0.2 which was released in 2006 along with Ptolemy II 6.0.2 with following new features:

- Compatibility with Cygwin
- Multihop routing using TinyOS Surge Protocol
- Java exception handling.
- Improved User Interface.

Viptos has full capability of Visual Sense simulator which is also based on Ptolemy II to model communication channels, networks and non-TinyOS nodes.



**Figure 30.** GUI Interface of Viptos-Wireless Simulation

## 20. PTOLEMY II [6] [84] [71]

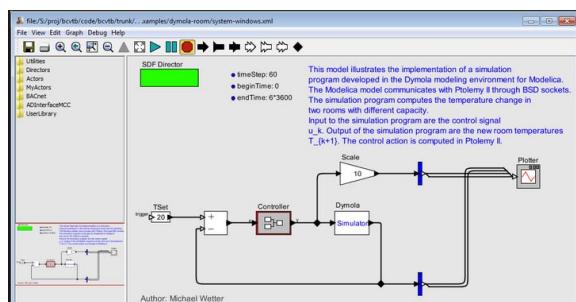
Ptolemy II is an open source simulation software with actor-oriented design. Actors are defined as software components which execute concurrent and communicate through messages sent via interconnected ports. Ptolemy II is under rigorous development since 1996 and is a collection of Java classes and packages. It addresses the modeling, simulation and design of concurrent, real-time, embedded systems.

Ptolemy doesn't support network emulation but is regarded as a good simulator having strong features and libraries for simulating both wired and wireless networks.

It comprises of graphical editor Vergil to support editing of syntax. Various specialized simulation tools like HyVisual (Hybrid Systems Modeling), Kepler, Visual Sense, Viptos has been created via Ptolemy.

The current version of Ptolemy II is 10.0.1 and was released on December 17, 2014.

The following figure highlights the GUI based interface of Ptolemy II simulator for simulation of Network Models:



**Figure 31.** Ptolemy II GUI Interface

The following Table 6 will highlight some of the features of Ptolemy II Simulator:

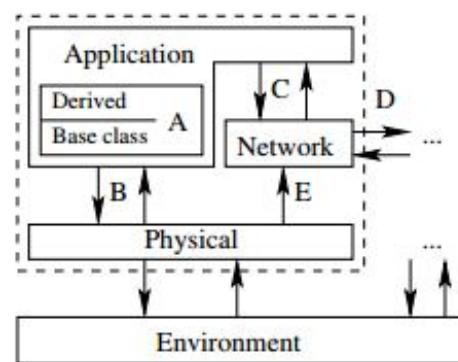
**Table 6.** Features of PTOLEMY II Simulator

Feature Name	Description
Java Language Development	As Ptolemy is based on Java language, so it is platform independent, threaded and network aware software.
Modularity	Comprise of various software's which can be used independently.
Abstract Syntax and Semantics Separation	It defines a software kernel which comprise of various classes which lays the foundation of hierarchical graph.
Heterogeneity	Combination of data flow and non-dataflow is easier.
Mutable Systems	Design can be modified even when the simulator is running.
Software Architecture	It is designed via taking object modeling and design patterns and test suite is implemented in TCL Blend via TCL/Java Interface.
User Interface	Modular interface using Diva utility
Improved Design	Implementation and coding part is simple to design and interpret.

## 21. SENS [27] [5] [28] [17] [4] [85] [86] [87]

SENS-Sensor Network Simulator and Emulator is a component based application level wireless sensor network simulator and has feature rich ability of simulation via extensibility and reusability. It has modular and layered architecture and has four main components: (1) Application; (2) Network; (3) Physical; (4) Environment.

The following figure demonstrates the components of SENS Simulator:



**Figure 32.** Components of SENS Simulator

SENS comprise of several simulated sensor nodes interacting with environment and each node has three main components: Application, Network and Physical. SENS is a customizable sensor network simulator which facilitates users to assemble application-specific environments and such environments are modeled in SENS by various signal propagation characteristics. The same source code which is being deployed on sensors in simulation can be as such used

in real sensors which makes this simulator portable because the programming language used is C++.

In light of Wireless Sensor Networks, SENS has novel mechanism for modeling physical environment because of tight integration with real sensors in terms of computation, communication and interaction.

As compared to other WSN simulators, SENS has less ability for customization, so no change can be brought to MAC protocol along with other low level network protocols. SENS does appear to use one of the most sophisticated environmental models and facilitate the use of sensor as well. But only phenomenon which is detected is Sound. But still lots to be done in this simulator and question still arise that whether it is a better simulator or not.

The Latest version can run with GCC 3.3 and was launched on Jan 31, 2005.

The Following Table 7 highlights the network components of SENS Simulator:

**Table 7.** Network Components of SENS Simulator

Component Name	Component Description
Simple Network	immediate, guaranteed delivery to all neighbours within range
Prob Lossy Network	Probabilistic delivery and delay; delivery probabilities can optionally decrease under heavy traffic.
Collision Lossy Network	calculates collisions at receiving end based on message overlap and relative signal strengths; selectable interval size

## 22. SHAWN [4] [17] [27] [88] [89] [90]

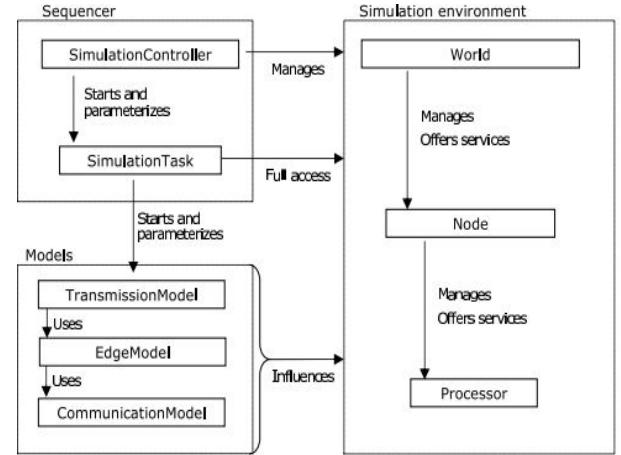
SHAWN- A Discrete event Open Source Simulator for wireless sensor network is written in C++ language. The basic objective behind the development of SHAWN Simulator to make it fully customizable. One central approach of Shawn is to simulate the effect caused by a phenomenon, not the phenomenon itself. Example: Instead of simulating a complete MAC layer which includes radio propagation model, its effects i.e. Packet Loss and Corruption are only modeled in Shawn.

Shawn provides support for development and testing of high level algorithms cum distributed protocols and has the ability to simulate large scale networks with ability to accommodate a network scenario of more than 100k of nodes.

The following points highlights Shawn Protocol Development Cycle:

- Centralized Implementation: In this phase, the first results and performance are being noted down.
- Simplified Decentralized Protocol: In this phase, proving has to be done that protocol runs in decentralized manner and produce valid results.
- Fully Distributed Protocol: The protocol developed is finalized by seeing the parameters- number of messages, consumption of energy, runtime and resilience to message loss.

The following figure highlights the Architecture of SHAWN Simulator:



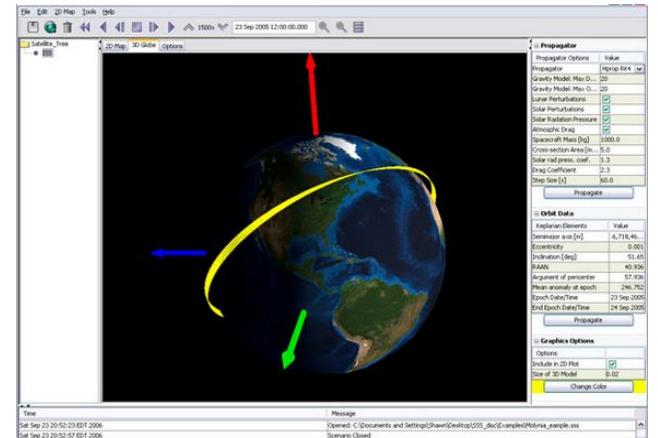
**Figure 33.** Architecture of Shawn Simulator

The following three models lay strong foundation of Shawn Simulator:

- Communication Model: It determines if two nodes can communicate by means of exchanging messages.
- Edge Model: It is responsible for different ways of calculating and maintaining graph representations (connection maps) of the network.
- Transmission Model: Transmission models are used to decide whether the message will get delayed, corrupted or dropped.

Shawn simulator can be extended to support WISEBED APIs and standards and can read and write WiseML files and provide an iWSN and Session Management implementation to be even connected to federated testbeds, and supports the Wiselib to run an algorithm in simulation before bringing it to a real testbed.

The following figure shows the GUI interface of Satellite Simulation in SHAWN Simulator:



**Figure 34.** GUI Interface of Satellite Simulation in Shawn Simulator

The Latest version of Shawn in 1.1.6 and is supports both Windows via Cygwin and Linux Operating System. But there is no stable release of this simulator. And in addition to

the Shawn Simulator a graphical visualization tool called "Viz" is provided with the simulation package.

### 23. SIDnet-SWANS [27] [91] [92] [93] [94]

SIDnet-SWANS is a Java-based visualization tool for simulating various application of Wireless Sensor Networks. SIDnet-Swans is based on JiST-SWANS simulator architecture and is used to implement IEEE 802.15.4 MAC-layer protocol. SID-net-SWANS is an efficient simulator to enable end-users to observe the behavior of new algorithms in various different phenomenon's like sudden loss of service both at an individual node as well as collection of nodes in network scenario, unexpected increase/decrease in measurement values, change in network load or increase in network nodes. The simulator also enables users to define various other parameters like temperature, humidity and dynamic movement of nodes in the network.

The following are the features of SIDnet-SWAN:

- Flexible GUI interface for users to interact with Wireless sensor network.
- Real-time data and visual analysis of sensor nodes.
- Explicit control over simulation speed.
- Efficiency towards modeling and management towards energy area of wireless sensor networks.
- Various parameters like Temperature, Humidity and dynamic movements can be monitored because of modular design.

The following figure highlights the SIDnet-SHAWN Simulator architecture:

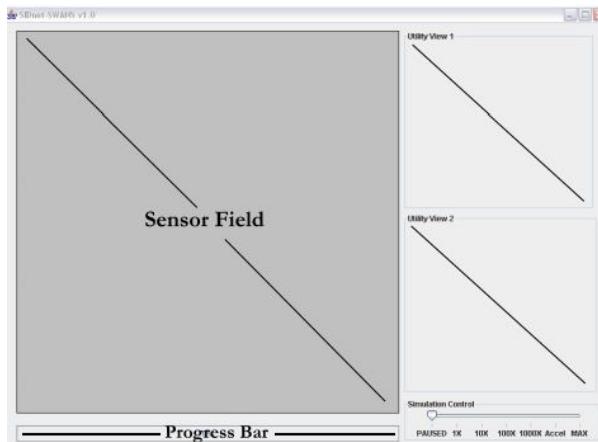


Figure 35. SIDnet-SHAWN Architecture

The components of architecture are listed as follows:

- Sensor Field: It is the area where sensor nodes are placed and sensor field can be setup by the user.
- Utility Views: Utility views are basically user-defined tools such as displaying the energy map and other types of statistical information of the sensor nodes simulating in the environment.
- Simulation Control Interface: It allows the user to control the speed of the simulator.
- Progress Bar: It is used to set the visual feedback

regarding any operation of the simulation being performed on the simulator.

The following figure highlights the Sensor Network Real-Time Simulation being performed at SIDnet-SWANS Simulator:

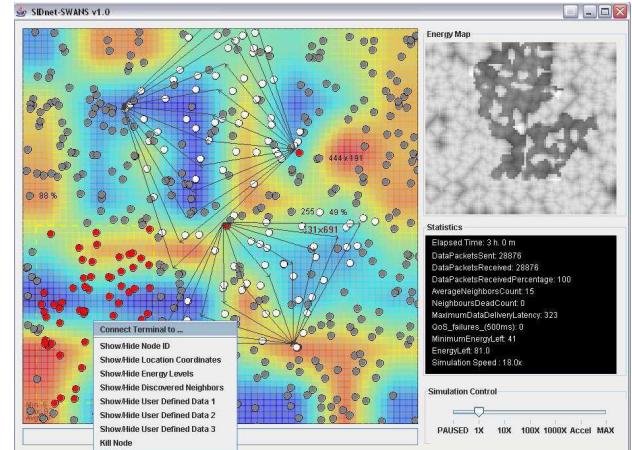


Figure 36. Wireless Sensor Network Simulation being performed at SIDnet-SHAWN Simulator

The latest version of SIDnet-SWANS is version 1.5.6 which was released in June 2011 and also contains JiST-SWANS version 1.0.6. SIDnet-SWANS simulator project.

### 24. WSim/Worldsens Simulator/WSNet Simulator [27] [95-98]

WSim/Worldsens-An Event Driven wireless sensor network simulator provides an integrated environment for development cum rapid prototyping of sensor network applications. WSNet simulator models applications, protocols as well as radio medium communication with high amount of efficiency cum accuracy. The objectives behind the development of WSim development are debugging, profiling and performance evaluation for deeply embedded Wireless Sensor Network applications.

WSim simulator can be used any sort of embedded operating system like TinyOS, ContikiOS, FreeRTOS, MANTIS OS etc. and can simulate various real-time sensor nodes like Teas Instruments MSP430, Tiez 430 USB or Olimex Easyweb2.

The following figure highlights the architecture of WSNet Simulator:

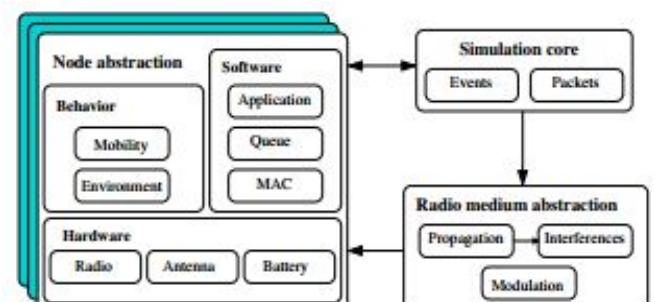


Figure 37. WSNet Simulator Architecture

The latest stable release of WSNet Simulator is version 9.07 and can be installed across different platforms like Windows, Linux, Solaris and MAC operating systems.

## 25. WSN Localization Simulator [27] [99]

Localization is one of the serious issues surrounding proper deployment of sensor nodes in the real world. If the deployment of the sensor nodes is not correct, the results and data will not lay any efficiency. So, how the sensor nodes would be deployed is basically the task of WSN Localization simulator.

The latest version of WSN Localization Simulator is 2.1 [99] and is available open source.

WSN Localization simulator comes bundle with 8 Localization Algorithms: HiRLoc, SeRLoc, Centroid, PTA, ADLA, HADLA, DIL, RAL. Various network parameters can be set like: Number of Random Locators, Number of Static Sensor Nodes, Number of Mobile Sensors, Number of Beacon Nodes and it provides almost all parameters to the researchers to research on the area of localization and has easy and neat and clean GUI for interface and output can be stored in .xls file also.

The following figure highlights GUI interface of WSN Localization Algorithm:

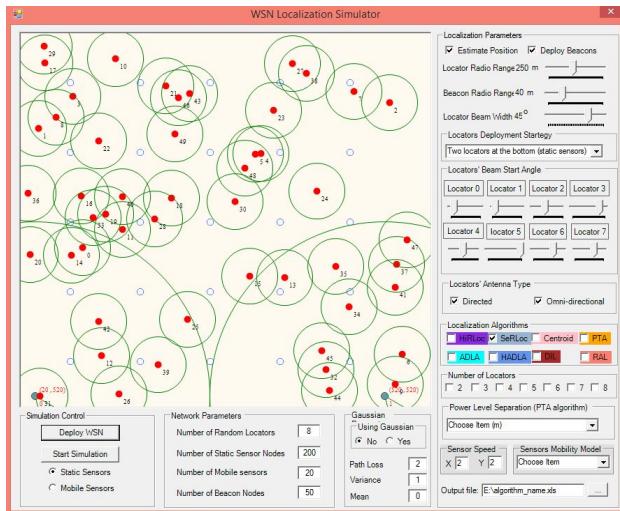


Figure 38. WSN Localization Simulator GUI Interface

## 26. NetTopo Simulator [100-105]

NetTopo is regarded as extensible cum integrated framework for simulating and visualizing wireless sensor networks. The basic objective behind design and development of NetTopo Simulator is to analyze various algorithms in WSN. With regard to simulation, users can define large number of parameters over sensor nodes like energy consumption, bandwidth management and NetTopo efficiently takes into consideration large-scale heterogeneous networks. NetTopo is an open source simulator and has been designed in Java language and has more than 80 Java classes and 11000 lines of code.

Two routing protocols i.e. TPGF and GPSR are already been implemented in this simulator. In addition to simulation, NetTopo has a visualization tool to present the data gathered

on actual WSN testbed.

The following figure shows the GUI Interface of NetTopo Simulator:

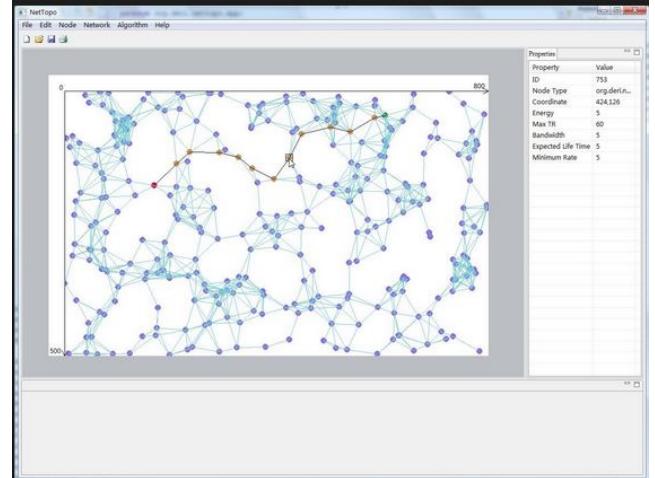


Figure 39. GUI Interface of NetTopo Simulator

The following figure highlights Visualization Tool of NetTopo Simulator:

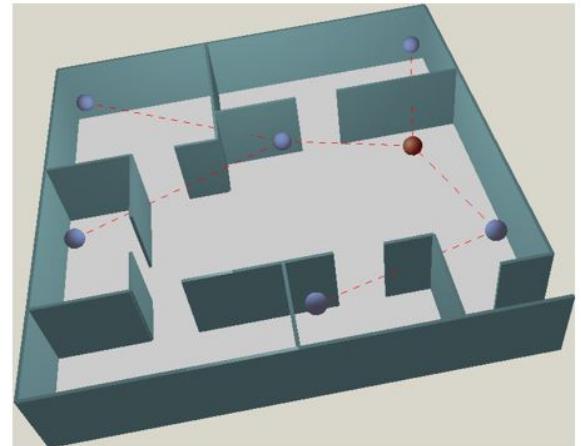


Figure 40. Visualization Tool of NetTopo Simulator

The following figure highlights the Architecture of NetTopo Simulator:

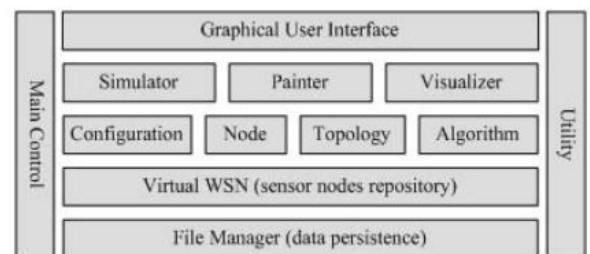


Figure 41. NetTopo Architecture

The following Table 8 explains the various modules/components of NetTopo Architecture:

*Main Control and Utility and two main components involved in Architecture of NetTopo Simulator*

**Table 8.** Components of NetTopo Simulator

Component Name	Description
Main Control	Acts as coordinator of all the interactions of other components. Works as bridge between input and output interfaces so that simulation operations can be carried out in an efficient manner.
Utility	Performs the basic services: defines application exceptions, format verification, number transforms and dialogue wrappers.
File Manager	It performs various duties like: maintain log of runtime simulation, statistical results recording, and records of all sensor nodes.
Virtual WSN	It does the storage of references of virtual sensor nodes in serialized format so that recovery and reusability tasks can be carried out when necessary.
Node	It is basically a virtual sensor node deployed in the GUI interface for WSN simulation. In NetTopo, various properties of sensor nodes can be configured like temperature, humidity, vibration and pressure etc.
Topology	It is basically the structure of the WSN network which is being designed by user for research and carrying out simulation tasks.
Algorithm	It represents the algorithm to be applied in Sensor network. NetTopo supports various types of algorithms of routing, clustering, scheduling, controlling etc.
GUI	GUI is basically the interface provided to the user to interact with the simulator. It comprise of three main components: (a) Display Canvas- Used for viewing WSN network. (b) Painter- It is an abstract interface for defining various 2D or 3D painting requirements. (c) Visualizer and Simulator: Both required to add sensor nodes on Display Canvas to carry out simulation tasks.

The following is the list of features of NetTopo Simulator:

- Easy customization of WSN network topology
- Easy customization of properties of sensor nodes from user-defined attributed.
- Extensibility to simulate user-customized algorithms.
- File-based function facilitating users to save and retrieve their simulating process.
- GUI Interface and support for XML file import and export
- 2D and 3D Easy to use Graphical Visualization Tool.

The NetTopo simulator is supported on Windows and Linux operating system.

## 27. SIDH [106] [5]

SIDH- A Component-based discrete event simulator written in Java is especially designed for wireless sensor network. The aim behind development of SIDH simulator is to simulate thousands of sensor nodes even faster than real-time or any other available WSN simulator. SIDH is easily to reconfigure and adaptable to different levels of simulation detail and accuracy; various communication

media; sensors and actuators; environmental conditions and various protocols and applications.

The following Table 9 gives the detailed description of various modules which SIDH simulator is being comprised of:

**Table 9.** Modules of SIDH Simulator

Module Name	Description
Simulator	It is the foundation module of the simulator. It is in this module various other modules interact with each other.
Event	It is an abstract base class which provides basic functionality to all events. Events tell the time at which an event should fire.
Medium	Medium plays the role of how the sensor nodes will broadcast the signals. Medium lays a strong foundation for the communication model as when the node is activates, what are the changes in the node, node transmission power and other factors like bandwidth and wavelength of medium.
Propagation Model	It defines the signal strength from receiver to transmitter. Various propagation models are used for signal propagation.
Environment	It has somewhat same functionalities as medium module which comprise of physical properties under which sensor operates. Environment also contains physical phenomenon's such as temperature, light, humidity, pressure etc.
Node	It is basically a sensor node which makes up the sensor network. Every node has various components like Processor, transceiver, sensor, actuator, energy source, network protocol and applications.
Transceiver	It models the hardware of sensor node that whether the sensor node is in sleep or active or receive or transmit state.
Protocol	It defines the various rules and regulations of packet movement among the nodes. Protocol can be physical, MAC, Routing or any other protocol.
Application Layer	It is the top layer of the network stack. It interface with other lower layers to implement a WSN application.

## 28. PROWLER [107-109]

PROWLER- Probabilistic Wireless Sensor Network Simulator- An event driven simulator for wireless sensor network runs under MATLAB and can operate either in deterministic or probabilistic mode. The objective behind development of Prowler simulator was to simulate Berkeley MICA Mote operating on TinyOS and other general systems.

Prowler was basically designed in MATLAB so it provides easy interface to prototype various applications and nice visualizations.

Prowler Simulator provides an accurate Radio Model but only provides a single MAC Protocol which is in Tiny OS.

The current version of Prowler v 1.24 which was tested on Matlab 6.5.

The following figure shows GUI Interface of Prowler Simulator:

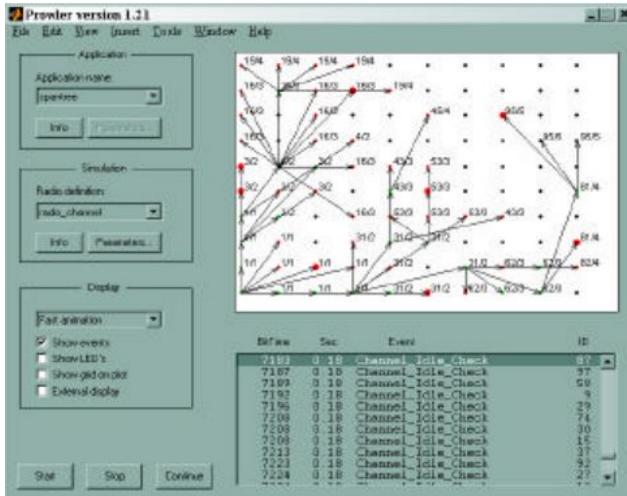


Figure 42. Prowler GUI

An extension to Prowler is JProWler. JProWler is written in Java Language.

## 29. MATLAB/Simulink [110-113]

Matlab- Matrix Laboratory- A very high performance, varied feature rich computation and visualization tool comprising of thousands of features was built by Mathworks Inc. The Latest version currently available for download or purchase for Commercial purposes is MATLAB 2015A.

The most important feature of MATLAB is its easy programming capability and easy platform for users to develop their own custom functions. MATLAB has various toolboxes like Aerospace, Control System Design, Fuzzy Logic, Symbolic Computations, Statistics, Communication and many others.

The following figure shows basic GUI of MATLAB 2015a software:

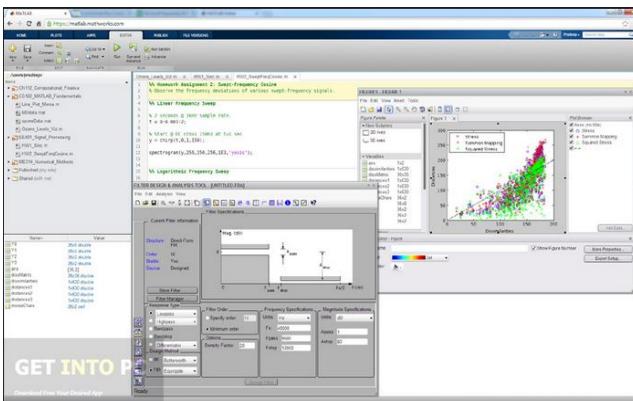


Figure 43. MATLAB 2015a GUI

Matlab is supported by another important software at its backend known as Simulink. Simulink supports both Linear and Non-Linear systems, modelled in continuous time, sampled time or mix of both. Simulink provides GUI for building block diagrams, drag and drop feature to design diagrams and its components.

The following figure shows the GUI interface of Simulink and its library cum components:

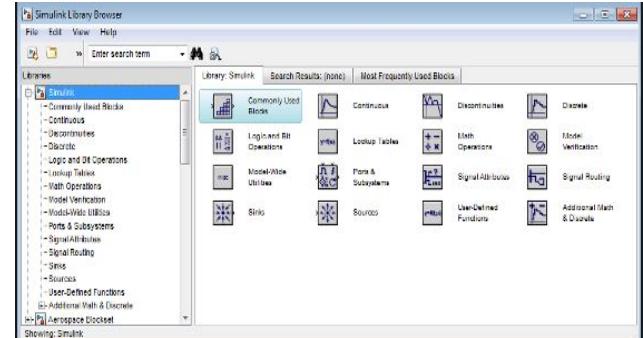


Figure 44. Simulink GUI

With regard to Simulating Wireless Sensor Network, MATLAB and SIMULINK provides communication toolbox set to build a complete WSN model system. Simulation procedure includes building the hardware architecture of transmitting nodes, modeling the communication channel and receiving node architecture. IEEE 802.15.4 was chosen as base standard for physical layer communication and various parameters like SNR (Signal to Noise Ratio, Attenuation and Interference) are considered.

The following figure shows comprehensive detail of WSN Model Architecture developed in Matlab and Simulink:

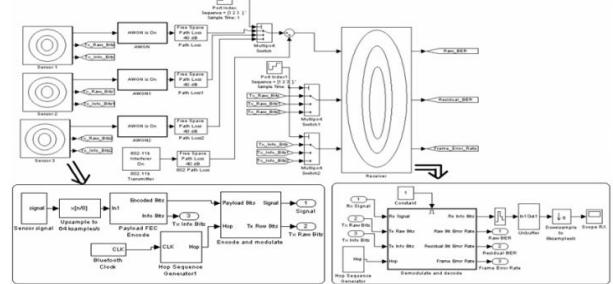


Figure 45. Wireless Sensor Network Model built in Matlab and Simulink

The following Table 10 explains the architecture components of Sensor Node.

Table 10. Architectural Components of Sensor Nodes

Architecture Component	Comprised of:
Transmitter	<p>Operates on Bluetooth at 2.4 GHz and consists of following:</p> <ul style="list-style-type: none"> <li>Sensor Signal stage</li> <li>Up-Sampling to 64ksample/s</li> <li>Payload FEC encode</li> <li>Bluetooth clock</li> <li>Hop Sequence Generator</li> <li>Encoder and Modulator</li> </ul>
Medium	<ul style="list-style-type: none"> <li>AWGN Channel</li> <li>Path Loss</li> <li>802.11b interferer</li> <li>Multiport Switch</li> </ul>
Receiver	<ul style="list-style-type: none"> <li>Hop Sequence Generator</li> <li>Demodulation and Decoding</li> <li>Zero-Order Hold</li> <li>Un-Buffer</li> <li>Down-sampling to 8ksamples/s</li> <li>Scope RX</li> </ul>

### 30. PiccSIM [115] [116] [117]

PiccSIM- Platform for Integrated Communications and Control design, Simulation, Implementation and Modeling was developed by Wireless Sensor Network Research Group as joint venture between Department of Automation and Systems Technology and Department of Communications and Networking at Aalto University, Espoo, Finland.

The idea behind the development of PiccSIM simulator was to make a complete toolset for design, simulation and implementation of Wireless Networked Control Systems (WiNCS) along with Network Controlled System (NCS).

PiccSIM simulator comprise of Simulink (Used for Simulating Dynamic System) and NS-2 (Used for Network Simulation). In addition to this, a GUI interface for modeling and designing the network and control system.

With combination of Simulink and NS-2 PiccSIM can be known as PiccSIM Toolchain which can be used for designing and modeling real wireless nodes.

The following figure highlights PiccSIM Architecture:

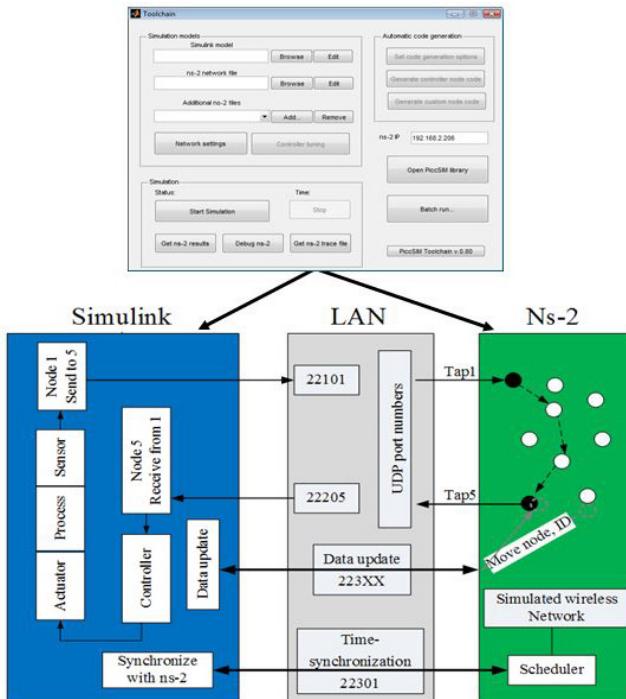


Figure 46. PiccSIM Architecture

PiccSIM makes use of Simulink and NS-2 simulator for NCS simulation. The toolchain uses MATLAB as well as Simulink for controlling the model.

PiccSIM Library comprise of various network node blocks to create wireless communications and Network Setting window provides a GUI interface for running network scripts.

The simulator requires MATLAB and NS-2 to be installed on single machine whether based on Linux or Windows Operating System or might be running on VMware.

The following are the various features of PiccSIM Simulator:

- Wireless Network and Network Controlled Simulation and are combined via Simulink.
- GUI interface for generating network scenario and generating NS-2 TCL scripts.
- Unique data exchange mechanism between Simulink and NS-2
- Complete Automatic code generation of network nodes.
- Remote User Interface which enables simulation without PiccSIM platform.

The latest version of PiccSIM is version 1.16 and now it has been enhanced and a new software of simulation has evolved as True Time-PiccSIM Simulator.

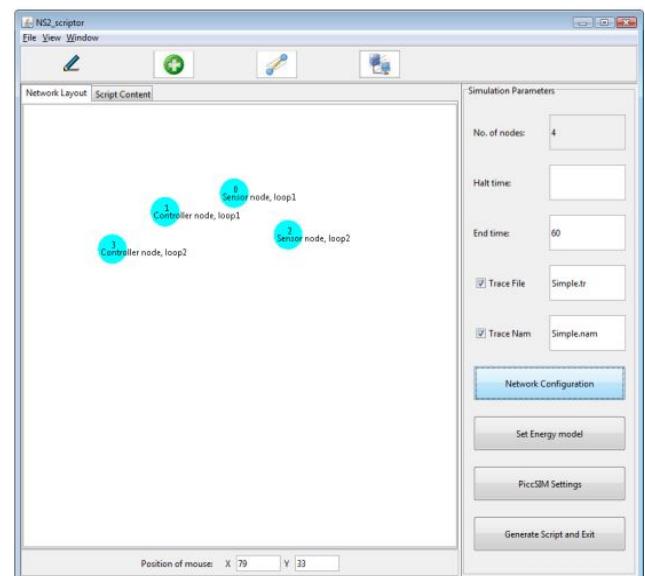


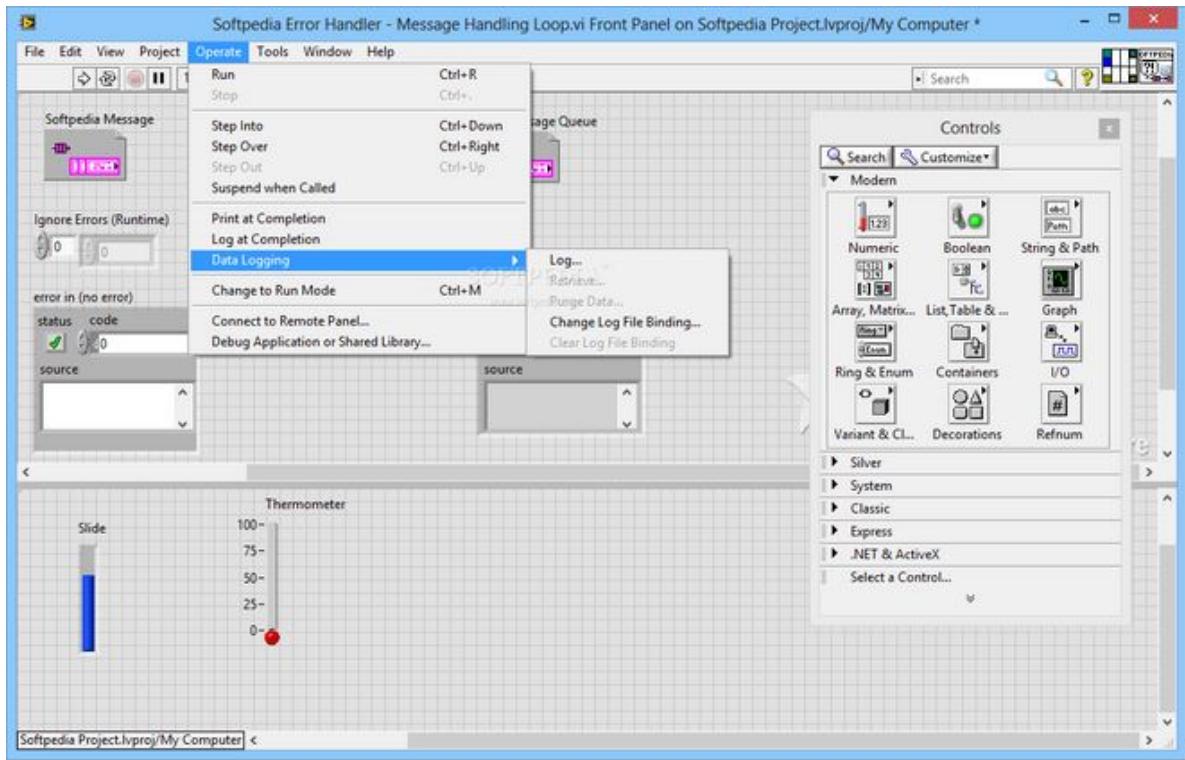
Figure 47. NS-2 Script Generator for PiccSIM Simulator

### 31. LabVIEW [118-121]

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) provides a unique platform and development environment for visual programming language. It is product from National Instruments Inc. LabVIEW these days is most commonly used for data acquisition, instrument control and industrial automation. It supports various operating systems like Windows, Linux, UNIX and even Mac OS X.

LabVIEW provides easy creation of user interfaces known as front panels into development. LabVIEW programs are called Virtual Instruments (Vis). Every VI has three components: Block Diagram, Front panel and Connector Panel.

As LabVIEW comprise of various toolboxes like MATLAB. In LabVIEW 2014 special attention has been given towards integration of Wireless Sensor Network Module. With LabVIEW WSN module, one can add intelligence to NI wireless sensor network nodes via same graphical programming approach to acquire and process data among various sensor nodes. With LabVIEW one can easily incorporate C code in the node for making routing and other all sorts of algorithms.



**Figure 48.** GUI Interface of LabVIEW 2015

The following points highlight some of the features of NI WSN Kit:

- Contains everything to evaluate NI wireless sensor networks and LabVIEW.
- Contains WSN Gateway and programmable  $\pm 10$  v analog input and thermocouple input nodes.

The following are the main features of WSN module by LabVIEW software:

1. Take advantage of Rapid development with easy to integrate graphical programming.
2. Research done on extension of battery life.
3. Perform custom analysis.
4. Reduce response time with embedded decision making.
5. Reduce costs and increase in performance

The latest version of LabVIEW is LabVIEW 2015/ 2014 SP1.

## 5. Conclusions and Future Scope

Simulation is regarded as integral part performed by each

and every researcher in various fields to test the research being carried out. In this research paper all the major WSN simulation tools are highlighted which are available till date along with their comparison to guide researchers in choosing the appropriate simulation platform to carry out all sorts of WSN based research.

Considering various parameters taken into consideration for performance comparison of various WSN simulators, NS-2, NS-3 and QualNet Simulator can be considered best simulators to carry out all sort of research in area of Wireless Sensor Networks.

## Future Scope

In near future, various emulators and testbeds would be evaluated and compared which will give vision to the researcher to test their novel work being done on physical testbeds and on real hardware sensors. A protocol based development would be performed and tested using NS-2/NS-3 simulator.

## Appendix

The following Table 11 highlights various general parameters based performance comparison of Various Simulation Tools being highlighted in this paper for Wireless Networks

**Table 11.** General Parameters Comparison of Various Simulation Tools of Wireless Sensor Networks

Sr. No	Name of Simulator	Simulator Type: (Discrete-Event/Trace-Driven)	Programming Language	License Type: Open Source or Commercial	Documentation Available (Yes/No)	Current Version	Developed by: (Company Name/Research Group)	Official Website
1.	NS-2	Discrete-Event	C++ & OTCL	Open Source	Yes	NS-2 version 2.35/ OTCL Version 1.14	DARPA. But currently development being done with SAMAN and through NSF with CONSER with other researchers including ACIRI	<a href="http://www.isi.edu/nsnam/ns/">http://www.isi.edu/nsnam/ns/</a>
2.	NS-3	Discrete-Event	C++	Open Source	Yes	NS-3.22	NS-3 Consortium	<a href="https://www.nsnam.org/">https://www.nsnam.org/</a>
3.	OMNeT++	Discrete-Event	C++	Open Source; Commercial	Yes	OMNeT++ 4.6	OMNeT++ Developers Group	<a href="http://www.omnetpp.org/">http://www.omnetpp.org/</a>
4.	JSim	Discrete-Event	Java; configuration Tcl/Java	Open Source	Yes	JSim 1.3 + patch 4	JSim has been sponsored by NSF, DARPA/IPTO, MURI/AFOSR, CISCO, Ohio State University, University of Illinois	<a href="http://jsim.cs.uiuc.edu/">http://jsim.cs.uiuc.edu/</a> or <a href="https://sites.google.com/site/jsimofficial/">https://sites.google.com/site/jsimofficial/</a>
5.	MANNASIM	Discrete-Event	Java	Open Source	Yes	Mannasim framework ns 2.29 patch	Researchers from Manna Research Group and from Sensorsnet Project	<a href="http://www.mannasim.dcc.ufmg.br/">http://www.mannasim.dcc.ufmg.br/</a>
6.	SensorSim	Discrete-Event	C++ and built on NS-2 sim.	Open Source	No	Not Available Now	Researchers from Networked & Embedded Systems Laboratory (NESL) at University of California, Los Angles, USA	<a href="http://nestee.ucla.edu/projects/sensorsim/">http://nestee.ucla.edu/projects/sensorsim/</a>
7.	NRLSensorSim	Discrete-Event	C++; Extension to NS-2 simulator	Open Source	Yes	NRLSensorSim 2.27 patch	U.S. Naval Research Laboratory-Networks and Communications System Branch	<a href="http://www.nrl.navy.mil/rd/ncs/products/sensorsim/">http://www.nrl.navy.mil/rd/ncs/products/sensorsim/</a>
8.	NCTUns	Discrete-Event	C++	Open Source	Yes	6.0	Network and Systems Laboratory, Dept. of Comp. Sc. And Info. Eng., National Chiao Tung University, Hsinchu, Taiwan	<a href="http://nsle.ee.nctu.edu.tw/networks.html">http://nsle.ee.nctu.edu.tw/networks.html</a>

9.	SSFnet	Discrete-Event	Java and C++	Open Source	Yes	2.0	DARPA, Institute for Security Technology Studies at Dartmouth and Renesys Corporation	<a href="http://www.ssfnet.org/homePage.html">http://www.ssfnet.org/homePage.html</a>
10.	GloMoSim	Discrete-Event	C and Parsec	Open Source	No	No Longer under Development	Scalable-Techologies	N/A
11.	QualNet	Discrete-Event	C++ and Parsec	Commercial with Separate Licenses for Industry and Academic	Yes	7.0	Scalable-Techologies	<a href="http://web.scalable-networks.com/content/qualnet">http://web.scalable-networks.com/content/qualnet</a>
12.	sQualNet	Discrete-Event	C++	Open Source	Yes	1.0	Funded by NSF and DARPA. Developed in Collaboration between NESL and Parallel Computing Laboratory at Univ. of California (UCLA)	<a href="http://nest.ee.ucla.edu/projects/squalnet/default.htm">http://nest.ee.ucla.edu/projects/squalnet/default.htm</a>
13.	OPNET	Discrete-Event	C and C++	Commercial	Yes	18.0.3	OPNET Inc. Now it is being taken over by Riverbed	<a href="http://www.riverbed.com/products/performance-management-control/opnet.html">http://www.riverbed.com/products/performance-management-control/opnet.html</a>
14.	SENSE	Discrete-Event	C++	Open Source	Yes	3.0.3	Developed by Center of Pervasive Computing and Networking, Comp. Sc. Department, Rensselaer Polytechnic Institute, New York	<a href="http://www.itacs.rpi.edu/">http://www.itacs.rpi.edu/</a>
15.	DRMSim	Discrete-Event	Java	Open Source	Yes	1.3.1	Joint research project is conducted by Alcatel-Lucent Bell, Université de Bordeaux (LaBri) and INRIA at Sophia Antipolis (Mascotte project) and is supported by the Euler Fire Project	<a href="http://drmsim.gforge.inria.fr/">http://drmsim.gforge.inria.fr/</a>
16.	NetSim	Event Trace	C and Java	Commercial (Different for Standard and Academic)	Yes	7.2	Tetcos	<a href="http://www.tetcos.com/net_sim_gen.html">http://www.tetcos.com/net_sim_gen.html</a>
17.	UWSim	Discrete Event	C++	Open Source	Yes	1.1	Developed by Researchers at IRSLab, Jaume-I University, Catellon	<a href="http://www.irs.ujj.es/uwsim/">http://www.irs.ujj.es/uwsim/</a>
18.	VisualSense	Discrete-Event	Java & XML	Open Source	Yes	8.0.1	Developed by EECS Department of UC Berkeley	<a href="http://ptolemy.eecs.berkeley.edu/visualsense/visualsense/ensedoc/">http://ptolemy.eecs.berkeley.edu/visualsense/visualsense/ensedoc/</a>

19.	Viptos	Event-Driven	nesC	Open Source	Yes	1.0.2	Developed by EECS Department of UC Berkeley under Ptolemy II Project	<a href="http://ptolemy.eecs.berkeley.edu/viptos/">http://ptolemy.eecs.berkeley.edu/viptos/</a>
20.	Ptolemy II	Discrete-Event	Java	Open Source	Yes	10.0.1	Developed by EECS Department of UC Berkeley under Ptolemy II Project	<a href="http://ptolemy.eecs.berkeley.edu/ptolemyII/">http://ptolemy.eecs.berkeley.edu/ptolemyII/</a>
21.	SENS	Discrete-Event	C++	Open Source	No	Version can run with GCC 3.3	Developed by UCLA	<a href="http://osl.cs.illinois.edu/se/ns/">http://osl.cs.illinois.edu/se/ns/</a>
22.	SHAWN	Discrete-Event	C++	Open Source	No	1.1.6	Developed by Uni Lubeck and TU Braunschweig	<a href="http://www.wisebed.eu/#appdev_shawn">http://www.wisebed.eu/#appdev_shawn</a> ; <a href="https://github.com/imtn/shawn/">https://github.com/imtn/shawn/</a>
23.	SIDnet-SWANS	Discret-Event	Java	Open Source	Yes	1.5.6	Developed by Joint Lab of Samsung, Advanced Institute of Technology and City University of New York	<a href="http://www.ece.northwestern.edu/~ocg47/SIDnet.html">http://www.ece.northwestern.edu/~ocg47/SIDnet.html</a>
24.	WSim/WorldSense/WSNet	Event-Driven	C/C++	Open Source	Yes	9.07	Developed by Guillaume Chelium, INRIA Research Scientist	<a href="http://forge.inria.fr/projects/wsnet/">http://forge.inria.fr/projects/wsnet/</a>
25.	WSN Localization Simulator	Discrete Event	Java	Open Source	No	2.1	Developed by Abdelhady Mohammad Naguib in June 2013	<a href="http://www.codeproject.com/Articles/606364/Wireless-Sensor-Network-Localization-Simulator-v">http://www.codeproject.com/Articles/606364/Wireless-Sensor-Network-Localization-Simulator-v</a>
26.	NetTopo	Discrete Event	Java	Open Source	Yes	1.0	Developed by Lei Shu	<a href="https://sites.google.com/site/leishuderif/nettopo">https://sites.google.com/site/leishuderif/nettopo</a>
27.	SIDH	Discrete Event	Java	Open Source	Yes	N/A	N/A	N/A
28.	Prowler	Probabilistic Discrete Event	Matlab/Java	Open Source	Yes	1.24	Developed by NEST (Network Embedded Systems Technology)	<a href="http://www.isis.vanderbilt.edu/projects/nest/prowler">http://www.isis.vanderbilt.edu/projects/nest/prowler</a>
29.	MATLAB	Discrete-Event	C/Java	Commercial	Yes	Matlab 2015a	Mathworks Inc.	<a href="http://www.mathworks.com">www.mathworks.com</a>
30.	PiccSIM	Discrete-Event	Matlab + NS-2	Open Source	Yes	1.6	Department of Automation and Systems Technology and Department of Communications and Networking at Aalto University	<a href="http://wsn.aalto.fi/en/tools/piccsim/">http://wsn.aalto.fi/en/tools/piccsim/</a>
31.	LabVIEW	Discrete Event	C/C++/Java and others	Commercial	Yes	LabVIEW 2015	National Instruments Inc.	<a href="http://www.ni.com/labview">www.ni.com/labview</a>

## REFERENCES

- [1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4), 393-422.
- [2] Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12), 2292-2330.
- [3] Imran, M., Said, A. M., & Hasbullah, H. (2010, June). A survey of simulators, emulators and testbeds for wireless sensor networks. In *Information Technology (ITSim), 2010 International Symposium in* (Vol. 2, pp. 897-902). IEEE.
- [4] Sundani, H., Li, H., Devabhaktuni, V. K., Alam, M., & Bhattacharya, P. (2011). Wireless sensor network simulators a survey and comparisons. *International Journal of Computer Networks (IJCN)*, 2(5), 249-265.
- [5] Singh, C. P., Vyas, O. P., & Tiwari, M. K. (2008, December). A survey of simulation in sensor networks. In *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on* (pp. 867-872). IEEE.
- [6] Egea-Lopez, E., Vales-Alonso, J., Martinez-Sala, A. S., Pavon-Marino, P., & Garcia-Haro, J. (2005, July). Simulation tools for wireless sensor networks. In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS05)* (p. 24).
- [7] Jain, R. (2008). The art of computer systems performance analysis. John Wiley & Sons.
- [8] Discrete Event Simulation: URL [http://en.wikipedia.org/wiki/Discrete\\_event\\_simulation#Network\\_simulators](http://en.wikipedia.org/wiki/Discrete_event_simulation#Network_simulators). Accessed on 17 August, 2014.
- [9] Jevtić, M., Zogović, N., & Dimić, G. (2009, November). Evaluation of wireless sensor network simulators. In *Proceedings of the 17th Telecommunications Forum (TELFOR 2009)*, Belgrade, Serbia (pp. 1303-1306).
- [10] Mekni, M., & Moulin, B. (2008, August). A survey on sensor webs simulation tools. In *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on* (pp. 574-579). IEEE.
- [11] Eriksson, J. (2009). *Detailed simulation of heterogeneous wireless sensor networks*. Department of Information Technology, Uppsala University.
- [12] Fall, K., & Varadhan, K. (2005). The ns Manual (formerly ns Notes and Documentation). *The VINT project*, 47.
- [13] The Network Simulator-ns2-Available from: <http://www.isi.edu/nsnam/ns>
- [14] Issariyakul, Teerawat, and Ekram Hossain. *Introduction to network simulator NS2*. Springer, 2011.
- [15] <http://www.nsnam.org/docs/release/3.14/tutorial/singlehtml/index.html> (Accessed on April 1, 2015)
- [16] <http://www.omnetpp.org/> (Accessed on April 1, 2015)
- [17] Kellner, A., Behrends, K., & Hogrefe, D. (2010). Simulation environments for wireless sensor networks. no. June, 2.
- [18] <https://castalia.forge.nicta.com.au/index.php/en/> (Accessed on April 2, 2015)
- [19] Boulis, A. (2009). Castalia User Manual. *Online: http://castalia.npc.nicta.com.au/pdfs/Castalia-User Manual.pdf*.
- [20] Varga, A., & Hornig, R. (2008, March). An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops* (p. 60). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [21] <http://j-sim.cs.uiuc.edu/> (Accessed on April 1, 2015)
- [22] <http://www.physiome.org/jsim/> (Accessed on April 1, 2015)
- [23] <https://sites.google.com/site/jsimofficial/j-sim-tutorial> (Accessed on April 1, 2015)
- [24] Sobeih, A., Chen, W. P., Hou, J. C., Kung, L. C., Li, N., Lim, H., ... & Zhang, H. (2005, April). J-sim: A simulation environment for wireless sensor networks. In *Proceedings of the 38th annual Symposium on Simulation* (pp. 175-187). IEEE Computer Society.
- [25] Sobeih, A., Viswanathan, M., Marinov, D., & Hou, J. C. (2007, March). J-Sim: An Integrated Environment for Simulation and Model Checking of Network Protocols. In *IPDPS* (pp. 1-6).
- [26] <http://www.mannasim.dcc.ufmg.br/> (Accessed on April 1, 2015)
- [27] Musznicki, B., & Zwierzykowski, P. (2012). Survey of simulators for wireless sensor networks. *International Journal of Grid and Distributed Computing*, 5(3), 23-50.
- [28] Steyn, L. P., & Hancke, G. P. (2011, September). A survey of wireless sensor network testbeds. In *AFRICON, 2011* (pp. 1-6). IEEE.
- [29] Park, S., Savvides, A., & Srivastava, M. B. (2000, August). SensorSim: A simulation framework for sensor networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems* (pp. 104-111). ACM.
- [30] <http://nesl.ee.ucla.edu/projects/sensorsim/> (Accessed on April 2, 2015)
- [31] <http://www.nrl.navy.mil/itd/ncs/products/sensorsim> (Accessed on April 2, 2015)
- [32] Donward, I. (2004). NRL's sensor network extension to NS-2. Available on line at <http://pf.itd.nrl.navy.mil/nrlsensorim>.
- [33] Jaichandran, R., & Irudhayaraj, A. A. (2010). A Futuristic Analysis on the Application Problems of Wireless Sensor Networks (WSN) with Effective Strategies and Optimal Solutions. *International Journal of Computer Applications*, 1(8), 10-16.
- [34] <http://nsl.csie.nctu.edu.tw/nctuns.html> (Accessed on April 2, 2015)
- [35] Wang, S. Y., & Huang, Y. M. (2012). NCTUns distributed network emulator. *Internet Journal*, 4(2), 61-94.
- [36] Wang, S. Y., & Huang, Y. M. (2009). NCTUns Tool for Innovative Network Emulations. *Computer-Aided Design*

- and Other Computing Research Developments.*
- [37] Wang, S. Y., & Lin, C. C. (2010, May). NCTUns 6.0: a simulator for advanced wireless vehicular network research. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st* (pp. 1-2). IEEE.
- [38] Wang, S. Y., Chou, C. L., Lin, C. C., & Huang, C. H. (2010). The Protocol Developer Manual for the NCTUns 6.0 Network Simulator and Emulator. *National Chiao Tung University, Tajwan.*
- [39] Yoon, S., & Kim, Y. B. (2009, September). A design of network simulation environment using ssfnet. In *Advances in System Simulation, 2009. SIMUL'09. First International Conference on* (pp. 73-78). IEEE.
- [40] <http://www.ssfnet.org/homePage.html> (Accessed on April 2, 2015)
- [41] Cowie, J. H. (1999). Scalable simulation framework API reference manual. *S3 Consortium, March.*
- [42] Kabir, M. H., Islam, S., Hossain, M. J., & Hossain, S. Detail Comparison of Network Simulators.
- [43] Zeng, X., Bagrodia, R., & Gerla, M. (1998, May). GloMoSim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on* (pp. 154-161). IEEE.
- [44] Bajaj, L., Takai, M., Ahuja, R., Tang, K., Bagrodia, R., & Gerla, M. (1999). Glomosim: A scalable network simulation environment. *UCLA Computer Science Department Technical Report, 990027*, 213.
- [45] GAO, Z. G., ZHAO, Y. L., LI, X., CAI, S. B., & WANG, C. S. (2006). Analysis of GloMoSim Wireless Network Simulator. *Journal of System Simulation, S2.*
- [46] Bagrodia, R., Meyer, R., Takai, M., Chen, Y. A., Zeng, X., Martin, J., & Song, H. Y. (1998). Parsec: A parallel simulation environment for complex systems. *Computer*, 31(10), 77-85.
- [47] <http://web.scalable-networks.com/content/qualnet> (Accessed on April 3, 2015)
- [48] [http://web.scalable-networks.com/sites/default/files/PR1310\\_23-SCALABLE-Releases-QualNet-and-EXata-Upgrades.pdf](http://web.scalable-networks.com/sites/default/files/PR1310_23-SCALABLE-Releases-QualNet-and-EXata-Upgrades.pdf) (Accessed on April 3, 2015)
- [49] Vasu, B., Varshney, M., Rengaswamy, R., Marina, M., Dixit, A., Aghera, P., ... & Bagrodia, R. (2005, November). Squlnet: a scalable simulation framework for sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems* (pp. 322-322). ACM.
- [50] Varshney, M., Xu, D., Srivastava, M., & Bagrodia, R. (2007, April). sQualNet: A scalable simulation and emulation environment for sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks, New York, NY, USA. Citeseer* (p. 24).
- [51] Varshney, M., Bagrodia, R., Dixit, A., Vasu, B., Aghera, P., & Srivastava, M. sQualNet version 1.0 beta: Tutorial and Programmers Manual.
- [52] <http://nesl.ee.ucla.edu/projects/squalnet/> (Accessed on April 3, 2015)
- [53] Sethi, A. S., & Hnatyshin, V. Y. (2012). *The Practical OPNET User Guide for Computer Network Simulation*. CRC Press.
- [54] Lu, Z., & Yang, H. (2012). *Unlocking the power of OPNET modeler*. Cambridge University Press.
- [55] García-Hernando, A. B., Martínez-Ortega, J. F., López-Navarro, J. M., Prayati, A., & Redondo-López, L. (2008). *Software Technologies in WSNs* (pp. 1-49). Springer London.
- [56] Chang, X. (1999, December). Network simulations with OPNET. In *Proceedings of the 31st conference on Winter simulation: Simulation---a bridge to the future-Volume 1* (pp. 307-314). ACM.
- [57] Katzela, I. (1998). *Modeling and simulating communication networks: a hands-on approach using OPNET*. Prentice Hall PTR.
- [58] [http://www.opnet.com/university\\_program/itguru\\_academic\\_edition/](http://www.opnet.com/university_program/itguru_academic_edition/) (Accessed on April 4, 2015)
- [59] <http://www.riverbed.com/products/performance-management/opnet.html> (Accessed on April 4, 2015)
- [60] [https://support.riverbed.com/bin/support/static//doc/opnet/OnlineHTML/Modeler/modeler\\_17.5\\_PL6/wwhelp/wwimpl/j\\_s/html/wwhelp.htm](https://support.riverbed.com/bin/support/static//doc/opnet/OnlineHTML/Modeler/modeler_17.5_PL6/wwhelp/wwimpl/j_s/html/wwhelp.htm) (Accessed on April 4, 2015)
- [61] Chen, G., Branch, J., Pflug, M., Zhu, L., & Szymanski, B. (2005). SENSE: a wireless sensor network simulator. In *Advances in pervasive computing and networking* (pp. 249-267). Springer US.
- [62] Szymanski, B. K., & Chen, G. G. (2007). Sensor network component based simulator. *Handbook of Dynamic System Modeling*, 35, 1-16.
- [63] <http://www.ita.cs.rpi.edu/> (Accessed on April 4, 2015)
- [64] Rosa, P. M., Neves, P. A., Vaidya, B., & Rodrigues, J. J. (2009, September). G-Sense-A Graphical Interface for SENSE Simulator. In *Advances in System Simulation, 2009. SIMUL'09. First International Conference on* (pp. 88-93). IEEE.
- [65] <http://netgna.it.ubi.pt/Downloads/G-SENSE.php> (Accessed on April 4, 2015)
- [66] <http://drmsim.gforge.inria.fr/index.php?page=intro> (Accessed on April 4, 2015)
- [67] Hogie, L., Tahiri, I., Papadimitriou, D., & Majoczyk, F. (2010). DRMSim: a network simulator for the investigation of routing schemes User manual.
- [68] Lancin, A., & Papadimitriou, D. (2013). DRMSim: A Routing-Model Simulator for Large-Scale Networks. *ERCIM News*, 94, 31-32.
- [69] Hogie, L., Papadimitriou, D., Tahiri, I., & Majoczyk, F. (2010, May). Simulating routing schemes on large-scale topologies. In *Principles of Advanced and Distributed Simulation (PADS), 2010 IEEE Workshop on* (pp. 1-10). IEEE.
- [70] Chéour, R., Jmal, M. W., Lay-Ekuakille, A., Derbel, F., Kanoun, O., & Abid, M. (2013, October). Choice of efficient simulator tool for wireless sensor networks. In *Measurements*

- and Networking Proceedings (M&N), 2013 IEEE International Workshop on* (pp. 210-213). IEEE.
- [71] Redondo-López, L., Prayati, A., López-Navarro, J. M., Martínez-Ortega, J. F., & García-Hernando, A. B. (2008). Problem Solving for Wireless Sensor Networks. *Problem Solving for Wireless Sensor Networks:, Computer Communications and Networks*. ISBN 978-1-84800-203-6. Springer-Verlag London, 2008, 1.
- [72] NetSim 8 User Manual: Available at: <http://www.boson.com/files/support/NetSim-8-User-Manual.pdf> (Accessed on April 4, 2015)
- [73] [http://www.tetcos.com/netsim\\_gen.html](http://www.tetcos.com/netsim_gen.html) (Accessed on April 4, 2015)
- [74] Dhurandher, S. K., Misra, S., Obaidat, M. S., & Khairwal, S. (2008). UWSim: A simulator for underwater sensor networks. *Simulation*, 84(7), 327-338.
- [75] Prats, M., Pérez, J., Fernández, J. J., & Sanz, P. J. (2012, October). An open source tool for simulation and supervision of underwater intervention missions. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (pp. 2577-2582). IEEE.
- [76] Dhviya, V. P., & Arthi, R. Analysis of Simulation Tools for Underwater Wireless Sensor Networks.
- [77] <http://www.irs.uji.es/uwsim/> (Accessed on April 4, 2015)
- [78] Kohli, S., Lee, E. A., Liu, X., Zhao, Y., Ee, C. T., Brooks, C., ... & Zhou, R. (2005). Visualsense: Visual modeling for wireless and sensor network systems.
- [79] <http://ptolemy.eecs.berkeley.edu/visualsense/> (Accessed on April 4, 2015)
- [80] Cheong, E., Lee, E. A., & Zhao, Y. (2005, November). Viptos: a graphical development and simulation environment for tinyos-based wireless sensor networks. In *Sensys* (Vol. 5, pp. 302-302).
- [81] Teng, G., Zheng, K., & Dong, W. (2008, October). A survey of available tools for developing wireless sensor networks. In *Systems and Networks Communications, 2008. ICSNC'08. 3rd International Conference on* (pp. 139-144). IEEE.
- [82] <http://ptolemy.eecs.berkeley.edu/viptos/> (Accessed on April 4, 2015)
- [83] Cheong, E., Lee, E. A., & Zhao, Y. (2006). Joint modeling and design of wireless networks and sensor node software. *EECS Department, University of California, Berkeley, UCB/EECS-2006-150*.
- [84] <http://ptolemy.eecs.berkeley.edu/ptolemyII/> (Accessed on April 4, 2015)
- [85] Sundresh, S., Kim, W., & Agha, G. (2004, April). SENS: A sensor, environment and network simulator. In *Proceedings of the 37th annual symposium on Simulation* (p. 221). IEEE Computer Society.
- [86] <http://osl.cs.illinois.edu/sens> (Accessed on April 4, 2015)
- [87] Peterson, J. A. (2005). SENS: A simple, extensible sensor network simulator for exploring approximate aggregation techniques. *Master's thesis, Boston University*.
- [88] Kröller, A., Pfisterer, D., Buschmann, C., Fekete, S. P., & Fischer, S. (2005). Shawn: A new approach to simulating wireless sensor networks. *arXiv preprint cs/0502003*.
- [89] [http://www.wisebed.eu/#appdev\\_shawn](http://www.wisebed.eu/#appdev_shawn) (Accessed on April 4, 2015)
- [90] Fekete, S. P., Kröller, A., Fischer, S., & Pfisterer, D. (2007, June). Shawn: The fast, highly customizable sensor network simulator. In *Proceedings of the Fourth International Conference on Networked Sensing Systems (INSS 2007)* (Vol. 32).
- [91] <http://www.ece.northwestern.edu/~ocg474/SIDnet.html> (Accessed on April 4, 2015)
- [92] Ghica, O. C., Trajcevski, G., Scheuermann, P., Bischof, Z., & Valtchanov, N. (2008, November). Sidnet-swans: A simulator and integrated development platform for sensor networks applications. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (pp. 385-386). ACM.
- [93] Pathan, A. S. K., Monowar, M. M., & Khan, S. (Eds.). (2014). *Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test*. CRC Press.
- [94] Ghica, O. C. (2010). SIDnet-SWANS manual.
- [95] <http://perso.ens-lyon.fr/eric.fleury/Upload/wsim-docbook/tools.html#id4743825> (Accessed on April 5, 2015)
- [96] <http://wsim.gforge.inria.fr/overview.html> (Accessed on April 5, 2015)
- [97] Fraboulet, A., Chelius, G., & Fleury, E. (2007, April). Worldsens: development and prototyping tools for application specific wireless sensors networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* (pp. 176-185). IEEE.
- [98] Chelius, G., Fraboulet, A., & Fleury, E. (2007, March). Worldsens: a fast and accurate development framework for sensor network applications. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 222-226). ACM.
- [99] <http://www.codeproject.com/Articles/606364/Wireless-Sensor-Network-Localization-Simulator-v> (Accessed on April 5, 2015)
- [100] Shu, L., Wu, C., Zhang, Y., Chen, J., Wang, L., & Hauswirth, M. (2008). NetTopo: beyond simulator and visualizer for wireless sensor networks. *ACM SIGBED Review*, 5(3), 2.
- [101] Shu, L., Hauswirth, M., Chao, H. C., Chen, M., & Zhang, Y. (2011). NetTopo: A framework of simulation and visualization for wireless sensor networks. *Ad Hoc Networks*, 9(5), 799-820.
- [102] Shu, L., Wu, C., & Hauswirth, M. *NetTopo: a framework of simulation and visualization for wireless sensor networks, Digital Enterprise Research Institute (DERI), National University of Ireland. Galway, Technical Report, October 30, 2008. Available at: http://library.deri.ie/resource/jbug5jlj*.
- [103] Shu, L., Hauswirth, M., Wang, L., Yuan, Z., & Chen, Y. (2009, September). Visualizing simulation and testbed of wireless sensor networks with NetTopo. In *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization* (pp. 99-100). ACM.

- [104] Parbat, B., Dwivedi, A. K., & Vyas, O. P. (2010). Data visualization tools for WSNs: a glimpse. *International Journal of Computer Applications*, 2(1), 14-20.
- [105] <https://sites.google.com/site/leishuderi/nettopo> (Accessed on April 5, 2015)
- [106] Carley, T. W. (2005). Sidh: A wireless sensor network simulator.
- [107] Simon, G., Volgyesi, P., Maróti, M., & Lédeczi, Á. (2003, March). Simulation-based optimization of communication protocols for large-scale wireless sensor networks. In *IEEE aerospace conference* (Vol. 3).
- [108] Simon, G. (2003). Prowler: Probabilistic wireless network simulator. *Institute for Software Integrated Systems, Nashville*.
- [109] <http://www.isis.vanderbilt.edu/projects/nest/prowler/> (Accessed on April 5, 2015)
- [110] Simek, M., Moravek, P., & sa Silva, J. (2011). Wireless Sensor Networking in Matlab: Step-by-Step. In *proceedings of International Conference on Teleinformatics* (pp. 185-190).
- [111] Ali, Q. I. (2012). Simulation Framework of Wireless Sensor Network (WSN) Using MATLAB/SIMULINK Software. Edited by Vasilios N. Katsikis, 263.
- [112] Ali, Q. I., Abdulmaowjod, A., & Mohammed, H. M. (2010, November). Simulation & performance study of wireless sensor network (WSN) using MATLAB. In *Energy, Power and Control (EPC-IQ), 2010 1st International Conference on* (pp. 307-314). IEEE.
- [113] Matlab website: [www.mathworks.com](http://www.mathworks.com)
- [114] <http://wislab.cz/our-work/wireless-sensor-network-simulation-tutorial-for-matlab> (Accessed on April 5, 2015)
- [115] Kohtamaki, T., Pohjola, M., Brand, J., & Eriksson, L. M. (2009, March). Piccsim toolchain-design, simulation and automatic implementation of wireless networked control systems. In *Networking, Sensing and Control, 2009. ICNSC'09. International Conference on* (pp. 49-54). IEEE.
- [116] <http://wsn.aalto.fi/en/tools/piccsim/> (Accessed on April 5, 2015)
- [117] PiccSim Manual: Available at [http://wsn.aalto.fi/en/tools/piccsim/piccsim\\_manual\\_1.1.pdf](http://wsn.aalto.fi/en/tools/piccsim/piccsim_manual_1.1.pdf) (Accessed on April 15, 2015)
- [118] Labview Website: [www.ni.com/labview](http://www.ni.com/labview) (Accessed on April 5, 2015)
- [119] <http://www.ni.com/white-paper/8981/en/> (Accessed on April 5, 2015)
- [120] <http://www.ni.com/labview/wsn/> (Accessed on April 5, 2015)
- [121] <http://en.wikipedia.org/wiki/LabVIEW> (Accessed on April 6, 2015)
- [122] Nayyar, A., & Gupta, A. (2014). A Comprehensive Review of Cluster-Based Energy Efficient Routing Protocols in Wireless Sensor Networks. *IJRCCCT*, 3(1), 104-110.
- [123] Nayyar, A., & Sharma, S. (2014). A Survey on Coverage and Connectivity Issues Surrounding Wireless Sensor Network. *IJRCCCT*, 3(1), 111-118.
- [124] Nayyar, A. (2012, January). Simulation Based Evaluation of Reactive Routing Protocol for MANET. In *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on* (pp. 561-568). IEEE.
- [125] Nayyar, A., & Singh, R. (2014). A Comprehensive Review of Ant Colony Optimization (ACO) based Energy-Efficient Routing Protocols for Wireless Sensor Networks. *International Journal of Wireless Networks and Broadband Technologies (IJWNBT)*, 3(3), 33-55.