Qualitative Analysis of Wireless Sensor Network Simulators

D. P. Mishra
Scholar CSVTU, Bhilai

Ramesh Kumar
Prof. Computer Sc. & Engg. BITD

ABSTRACT

Wireless Sensor Networks is gaining importance day by day and they are more and more promising because of the potential and features available with them to revolutionize life and economy, with its wide range of applications. Wireless Sensor Networks (WSN) is formed by using large no of motes. Since analyticla modelling of WSN is very complex,itleads to simple analysis. Otherhand, preparing test-beds requires a huge effort. therefore, simulation tools are must to study WSN, and itrequires a suitable model based on solid assumptions and framework. Simulatorsease the way of implementation. In addition, simulation resultare totally dependant on the specific environment, scenario under study, physical layer, hardware consideration and assumptions madeare not equally correct to capture the real parameters and behavior of WSN. In order to work on WSN, simulators plays a very important role by providing simulation environment, that willmeet researchers requirement to conduct experiments in specific area. This paper would discuss the important features and classification of various simulators.

General Terms

Wireless Sensor Network Simulators

Keywords

WSN, Sensor network, simulation, motes, tools.

1. INTRODUCTION

The sensors nodes are having capability to transmit, receive and forwardsensed data to the sink node or aggregator. WSN works in bi-directional mode, they are havingcapability to establish bidirectional communication channel, and may forward sense parameter from sensor node to the gateway, and may execute commands given by base station to end sensor node. Motivation behind the development of WSNs was based on requirement of military applications such as battlefield surveillance; WSNs are widely used in industrial, residential and wildlife environments. Structure health monitoring, home automation, and animal tracking are some of the applications of Wireless sensor networks [1]. The sensor nodeswould act as digital sensing organ that senses the physical phenomena of interest

- Target tracking & battle damage assessment
- It may be used for nuclear, biological & chemical attack detection
- It can monitor the vehicle traffic, Pedestrian, intelligent transportation grid.
- It can report wildlife habitat monitoring forenvironment conservation.
- It helps in detecting forest fire.

2. SENSOR NETWORK PROTOCOL

The Wireless sensor network consists of various sensor nodes as shown in Fig.1 [2].Sensor nodes are having constraint of resources such as limited computing, battery and bandwidth. The protocol stack is having application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane as shown in Fig.2. Based on the requirement & sensing parameters different types of application software were built and they can be used on the application layer.

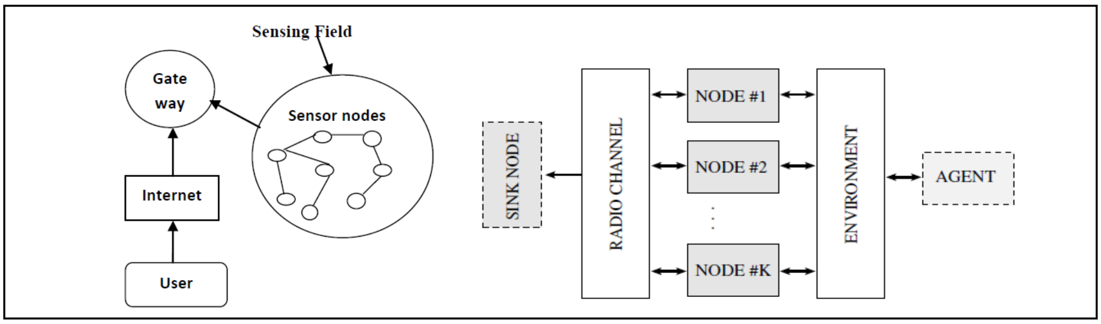

Fig 1: Basic structure of Wireless Sensor Network & its functional blocks

Simulationis essential tool for researching and experimenting with the WSN. It requires suitable model based on solid assumption and appropriate framework to ease implementation. Simulation is the process of designing&modelling of a real system and conducting experiments with the model for the purpose of either understanding the behaviour of the system or evaluating different strategies for the operation of WSN system. Detailed discussion of simulation methodology, in general, can be found in [3, 4].

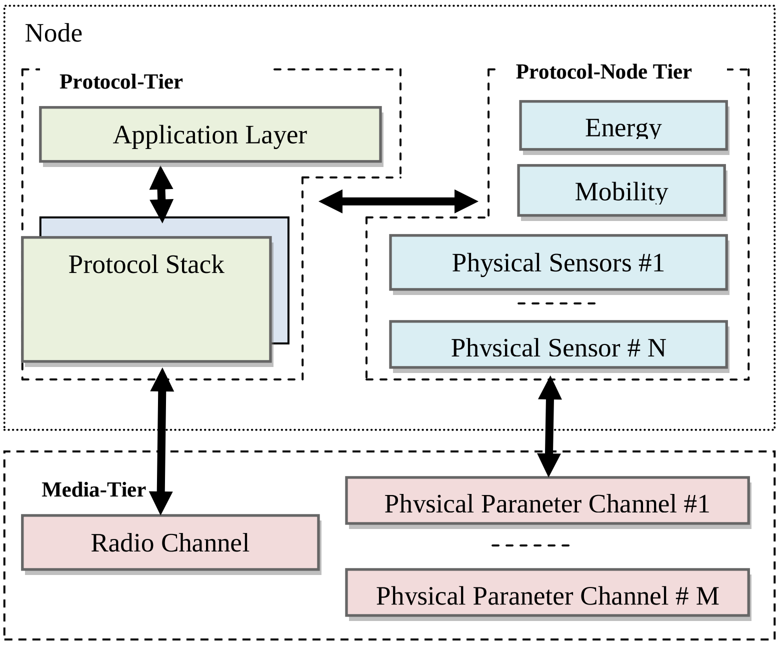

Fig 2: Layered Architecture of Wireless Sensor Network – with different Tiers

The simulation result is fully dependent on the environment and hardware/physical layer assumption which may not be so accurate to predict the real behaviour of wireless sensor network. Simulation is necessary to test the application and protocols specific to scenario or area specific study. The correctness of the model and Suitability of model for the implementation is one of the most important factors for WSN simulations. The key properties of good Simulator are

- Availability
- Scalability.
- Performance
- Support for rich-semantics & scripting languages
- GUI, debugging and tracing support.

3. TYPES OF SIMULATORS

There are three main types of simulators used for simulation

1. Monte Carlo Simulation
2. Trace Driven Simulation
3. Discrete Event Simulation

Trace driven and discrete event simulators are commonly used for simulating Wireless Sensor Networks

3.1 Monte Carlo Simulators

Monte Carlo Simulation is a computerized mathematical technique, that allows used to do quantitative analysis and decision making. This technique provide a wide range of possible outcome with probability, it shows all the possible outcome that will help to take most conservative decision making

This technique is introduced during World War II, first developed by scientist working on atom bomb; it was named for Monte Carlo, the Monaco resort town for its casinos.

3.2 Trace Driven Simulation

Trace-Driven simulator provides different services [5, 6], and are commonly used in real system. Results given by simulation is having more credibility. It provides more accurate workload; & detail information allow users to deeply study the simulation model. The input values provided in the simulation process is fixed or remain constant.

Trace based simulators are having some cons that Increases the complexity due to increase in information and the complexity of the simulation; Suspiciousness of results produced by simulator on varying workloads.

3.3 Discrete-event Simulation

Discrete-event simulation (DES) [6], models the operation of a system as a discrete sequence of events. Each event occurs at a particular instance of the time and marks a change of state in the system.

The simulation may list pending events that can be simulated by routines. Global variables, which describe the system state, simulation time, allow the scheduler to predict this time in advance. Simulation process includes input, output, initial, and trace routines. Addition to all this simulation process provides dynamic memory management that may add new entities and remove the old one from the model. For the ease of debugging, debugger breakpoints are provided in discrete-event simulation, thususers may verify the code without disrupting program in step mode.

4. WSN SIMULATION SOFTWARE

Detailed study simulation software's used in the selected IEEE Journal and Conferences, Nurul I. Sarkar *et. al.* [7] surveyed all papers published in the IEEE Transactions on Communications (1071 papers), and in proceedings of IEEE GLOBECOM (2991papers), INFOCOM (817 papers), and ICC (3114 papers) between 2007 and 2009. A total of 8370 papers were surveyed, about the survey results were summarized 42.8% of 8370 papers surveyed have mentioned that they use ns-2 for network modelling and simulation tasks. About 36.8% of the total papers surveyed have used MATLAB whereas 7.6% used OPNET. The remaining 4.2%, 1.6% and 0.8% of the total papers surveyed have used QualNet, GlomoSim, OMNet++, respectively and around 6.2% of the papers surveyed did not mention the name of the simulators that they had used. That's why we have chosen the fifteen below mentioned tools on the basis of their widely accepted usage.

4.1 NS-2

NS-2 is the abbreviation of Network simulator version two, developed in year 1989 it uses the REAL network simulator [5][8]. NS-2 is supported by Defence Advanced Research Projects Agency and National Science Foundation. It is a discrete event network simulator built in Object- Oriented extension Tool Command Language and C++. One can run NS-2 simulator in Linux environment or in Cygwin, which is a UNIX-like environment that provides command-line interface running on Windows. NS-2 is a popular network simulator that can used in both wire and wireless field. This simulator is open source and will provide online documentation and support for TCL.

Features:

- Object oriented design allow creating and using of new protocol.
- It provide visualization tool-NAM (Network Animator)

Limitation:

- Needs to be familiarize with scripting language and modelling technique;
- Tool Command Language (TCL) is bit difficult to understand and write code in it.
- NS-2 is more complex and time-consuming than other simulators to model a scenario in specific cases.
- Graphical support is poor, no Graphical User Interface (GUI); users must have to write text commands of the electronic components nodes and channels.
- Frequent changes made in the code base may affect the consistency of outcome, or produces bugs.

4.2 NS-3

NS-3 simulator is an open-source and discrete-event network simulator specifically designed for research and educational use. It was first developed in 2006. Several version have been released so far after its first development. The latest version NS-3.22 was released in September 2014.

Features:

- NS-3 is not an extension of NS-2; it is a new simulator. NS2 & NS3 both are written in C++ but NS-3 is a new simulator that does not support the NS-2 APIs.
- Few models from ns-2 are ported from NS-2 to NS-3.
- Project maintain NS-2 while NS-3 is being built, and the study of transition and integration would be done.
- NS-3 is open-source, and mainfocus on to maintain an open environment for researchers to contribute and share their software.

Limitation:

- Python bindings do not work on Cygwin.
- Only IPv4 is supported.

4.3 TOSSIM

It is discrete event simulator for TinyOS, specially designed for embedded operating system. Developed at UC Berkeley it mainly emphasizing code readability, and C++. People can run TOSSIM on Linux Operating Systems or on Cygwin in Windows & will run on custom mote hardware. Supports Python, NesC, and C++ TOSSIM also provides open sources and online documents.

Features:

- Open Source and online documentation
- Graphical User Support (Tiny ViZ)
- Simple and powerful
- Support thousands of Nodes

Limitation:

- Specially designed for tinyOS, not specifically designed for simulation performance metrics of other new protocol [9].
- Therefore, TOSSIM cannot accurately simulate issues of the energy consumption in WSN;
- Power TOSSIM(Which another TinyOs Simulation extending power of TosSim) can be used for energy specific calculations,
- Every node must have to run on NesC code, a programming language that is event-driven, component-based and implemented on TinyOS,
- TOSSIM can only emulate the type of homogeneous applications.

.

4.4 OMNeT ++

It is modular discrete event simulator which is implemented in C++. It is very simple to work on it, due to its clean design. OMNET++ provides a powerful GUI library for animation, tracing and debugging.

It is very popular tool and its limitation specific to model is being overcome by recent contributions. As example, a mobility framework has recently been released for OMNET++, and it can be used as a starting point for WSN modelling. Additionally, several new proposals for localization and MAC protocols for WSN have been developed with OMNET++, under the Consensus project, and the software is publicly available.

Major drawback is compatibility since most of the available model are developed by independent group i.e they doesn't share a common interface, which make it very difficult to combine the things. As an example, not even the localization and MAC protocols developed in the Consensus project are compatible.

Features:

- Powerful Graphical User Interface (making tracing and bugging easier)

- Simulate power Consumption problem

Limitation:

- Lack of available protocols in its library, compared to other simulators.
- Number of protocol is not large enough.
- Compatibility problem (not portable)

4.5 J-SIM

J-Sim is a component-based simulator, developed in Java. It provides real-time process based simulation. The main benefit of J-Sim is its considerable list of supported protocols, including a WSN simulation framework with a very detailed model of WSNs, and implementation of localization, routing and data diffusion WSN algorithms. J-Sim models are easily reusable and interchangeable and will provide the maximum flexibility. Additionally, it provides a GUI library for animation, tracing and debugging support.

Features:

- J-Sim models arereusable and interchangeable to in order to make simulation process simple.
- J-Sim supports large number of protocols; as well support data diffusions, routings and localization simulations in WSNs
- J-Sim can simulate radio channels and power consumptions in WSNs.
- GUI library provided help users to trace and debug programs in easier and effective way.
- Platform is independent and easy for users to choose specific components to solve the individual problem.
- As compared to NS-2, J-Sim can simulate larger number of sensor nodes more than 500, as well it can save memory space.

Limitation:

- J-Sim is relatively complex for use.
- The execution time is more than NS-2.

4.6 CASTALIA

It is WSN simulator, generally used for Body Area Networks (BAN) and networks of low-power embedded devices. It is based on the OMNeT++ platform and can be used by researchers and developers to test their distributed algorithms and protocols in realistic wireless channel and radio models, it can be used to evaluate different platform their characteristics for specific applications, since it is highly parametric, and can simulate a wide range of platforms [10, 11].

Features:

- Advanced channel model based on empirically measured data.
- Advanced radio model based on real radios for low-power communication.
- Extended sensing modelling provisions
- Node clock drift
- MAC and routing protocols available.
- Designed for adaptation and expansion.

Limitation:

- Castalia is not sensor-platform specific.

4.7 QUALNET

It is commercial version of GloMoSim based on C++, which is the product of Scalable Network Technology (SNT).GloMoSim is academic research version (an open source) but QualNet [12] is commercial version that supportsGUI for scenario design, simulation and analysis.

Features:

- Full GUI to perform all the jobs.
- Speed, scalability& portability
- Enables model fidelity (i.e. more accurate modelling of real world networks)
- Built in measurement on each layer.
- Modular layer stack design.
- Scalable for parallel execution.

4.8 ATEMU

ATEMU is an emulator of an AVR (Modified Harvard architecture 8/32-bit RISC single-chip microcontroller, which was developed by Atmel in 1996)processor for WSN built in C; AVRis used in MICA. ATEMU is a specific emulator for WSNs; that may provide support to run TinyOS on MICA2 platform. It can emulate the communication among the sensors along with every instruction implemented in every node. This emulator provides open sources and online documentation. ATEMU architecture is shown in **Fig 3.**

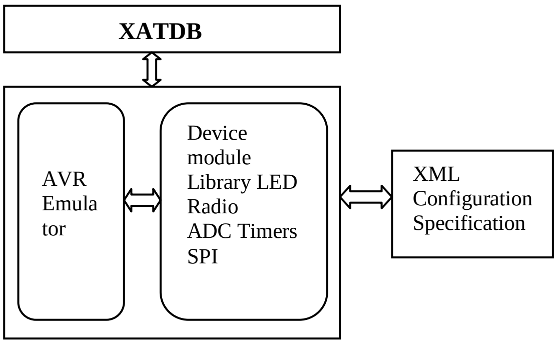
XATDB
AVR Emulator
Device module Library LED Radio ADC Timers SPI
XML Configuration Specification

Fig 3: ATEMU Architecture

Features:

- Simulate multiple Sensor nodes at a time and each sensor node may run different program
- Having large library of wide range of hard devices
- Can provide high level of detailed emulation in WSN
- GUI can help to edit and monitor the program
- Open source nature will save cost

Limitation:

- Simulation time too high
- Few function to simulate routing and clustering

4.9 AVRORA

Avrorais built on java platform& specially designed for WSN. It is similar to ATEMU and it can simulate AVR-based microcontroller and MICA2 sensor nodes. Avroracombines pros of TOSSIM and ATEMU in order to limit the cons of both the simulators.

Features:

- Instruction level simulator, that fills the gap of TOSSIM &ATEMU
- High speed&scalability
- Simulate different programming code projects.

Limitation:

- GUI is not available

4.10 EMSIM

EmSim will simulate networks of 32-bit nodes by using the Linux supported Emstar [14]. For simulation a separate EmStar process tree correspond to each node in the web, will enable tested application to get done implemented in production environment.

Features:

- EmTOS enables EmSim to usenesCapplications as module of the simulation.
- EmView is visualizationand analysis tool, used to present the topology and current status of the network.

4.11 FREE EMULATOR

It is used for developing software nodes based on the Javaand IEEE 802.15.4 standard [15]. The software is a part of the free mote Environmentthat also includes the free moteTest bed [16]. System architecture defines: Physical, Data Link (MAC), Routing and Application layer. There are three node typesintroduced: Emulated Nodes run by thesimulator in separated threads. Bridges are the gateeways to the real world and Real Nodes that compose the test bed.

Features:

- The framework will executes the same code on emulated as well as Java-based devices and supports communication with MEMSIC TelosB and MICAz hardware platforms running TinyOS[17].
- The Emulator may be installed either as a standalone application or as a web service, will provides remote access to the environment.
- One can run simulator in two ways one from command line loading a prepared configuration XML file. Second by using a Graphical User Interface allowing to set-up parameters and visualize the experiment.

4.12 MSPSIM

It is an emulator for WSN nodes and is based on TexasInstruments MSP430 microcontroller. Program simulates and displays a visual representation of the whole sensor board equipped with elements such as sensors, interfaces andLEDs. Proper design and java implementation would increase the number of supported node [18]. MSPsim is a partof the Contiki Operating System [19] and can be used in cross-level simulations conducted with the COOJA platform [20].

Features:

- Instruction level simulator
- It is capable of loading unmodifiedtarget platform ELF and IHEX firmware files.
- Supports monitoring of the executed code.

4.13WSIM

WSim (A Sensor node platform emulator) supports running, debuggingand analysis of application, it will the exact firmware the target node or hardware. WSim reflects hardware details specific to no. of Texas Instruments MSP430 microcontroller-based sensor platforms with flash memory, chipset for communication and peripherals.

Features:

- Not depend on any embedded operating system,
- May run the target platform code without modification or recompilation.
- Timing, interrupt data, memory and estimation for power consumption is possible
- Bigger simulation scenarios can be carried out on distributed computers exchanging information by means of IP multicast.

4.14 PROWLER

It is probabilistic WSN simulator powered by MATLAB environment [21].Supports radio propagation model Gaussian and Rayleigh, also supports collision detection schemes for MICA2 MAC-layer communicationprotocol.

At the application level, a set of basic events and actions related to thesimulation is defined. JProwler[22] is similarto Prowler, developed in Java and available inTinyOS CVS repository

Features:

- Simulators may simulate behaviour of thedevice, but they won’t simulate the effect of thecommunication channels. Imperfect wireless channels greatly affect the applications performance, so it incorporate the changes in simulators for providing accurate result.
- It probabilistic wireless for distributed systems, capable to simulate from application to physical communication layer.
- Provides an easy way of application prototyping with nice visualization capabilities

4.15 LOCALIZATION SIMULATOR

WSN localization simulator is designed for determination of the location of sensor nodes. The program comes with eight localizationalgorithms. Several parametersthat define network topology, strategy locators, antennatype, path loss and node mobility.

Features:

- Supports almost all the mobility models.
- Supports shadowing and propagation models.
- Events are recorded and written into an external trace file.
- Will support almost all the attack and defence mechanisms like wormhole, Sybil, Spoofing and Replay attacks
- MICA2 and TelosB is supported

5. COMPARISONS BETWEEN DIFFERENT SIMULATORS

The key features and limitations of each of these simulators are summarized in Table 1.

Table 1: Key features, Pros and Cons of Sensor Networks

Software	Simulator/ Emulator	Discrete-Event/Trace Driven	Programming Language	Open Source	General/ Specific	GUI	Pros	Cons
NS2	Simulator	Discerete	C++	Open	General	NO	Easy to add new	Supports only two

	& also possible to use it as emulator supported by Free BSD	event		Source			protocols. Supports large number of protocols Visualization tool.	wireless MAC protocols, 802.11, and a single-hop TDMA protocol. Familiarization with scripting
NS3	Simulator/ Emulator	Discerete event	C++	Open Source	General	NO	NS-3 is not an extension of NS-2; it is a new simulator. NS-3 is open-source	- Python bindings do not work on Cygwin. - Only IPv4 is supported.
Tossim	More precise simulator can emulate	Discerete event	Nesc	Open Source	Specific to WSN	YES	Highly accurate Visualization tools are available.	Compilation steps loose the fine grained timing & interrupt properties
J-Sim	Simulator	Discerete event	Java	Open Source	General	YES	Supports energy modelling, with the exception of radio energy consumption Component-centric architecture.	Low efficiency of simulation. Unnecessary run-time overhead
Omnet ++	Simulator	Discerete event	C++	Licensed(Commercial/ Non Commercial)	General	YES	Powerful GUI Simulate power Consumption problem	Number of protocol is not large enough. Compatibility issues
Castalia	Simulator	Discrete event	C++	Open Source	Specific to WSN/BAN	YES	Advanced channel, radio modelling MAC and routing protocols available. Highly tuneable	- Not a sensor specific platform. - Not useful if one would like to test code compiled for a specific sensor node platform.
Qualnet	Simulator/ Emulator with Exata	Discerete event	C++	Commercial	General	YES	Easy-to-use and clear User Interface. Sophisticated animation Support formultiprocessor systems anddistributed computing.	Difficult installation on Linux. Slow Java-based UI. - Very expensive.
ATEMU	Emulator	Discerete event	C	Open Source	Specific for WSN	YES	Can simulate multiple Sensor nodes at thesametime with differentprogram	Simulation time is much longer Fewer option to simulate routing and clustering
Emsim	Simulator	Trace Driven	C	Open	Specific to Embedded OS	YES	Linux based Supports EmTos&nesC	Supports network of 32-bit nodes running Linux-based Emstar software environment
AVRORA	Simulator	Discrete Event	Java	Open Source	Specific to WSN	NO	It is instruction level simulator Faster speed and better scalability	GUI is not available

							Can simulate different programming code	
Free Emulator	Emulator	Discrete/Trace Event Driven	Java	Open Source	Specific	YES	Supports Micaz&TelosB platform Can be configured with XML Free Enviornment	-
MPSim	Simulator/ Emulator	Discrete Event	Java	Open Source	Specific	YES	Instruction level simulator Monitoring & profiling of the code. Cross level simulation is possible	-
WSim	Simulator/ Emulator	Discrete Event	C		General	YES	Does not depend on any embedded OS WSim can run the target platform code without modification or recompilation. Large simulations can be carried out on distributed computers	-
Prowler	Simulator/ Emulator	Discrete Event	Matlab/Java based	Commercial	Specific	YES	Powered by Matlab Supports Mica Supports collision detection scheme	-
WSN Localization Simulator	Simulator/ Emulator	Discrete Even	C#	Commercial	Specific	YES	Used for localization ofnodes Uses localization algorithm Supports defence algorithms for attack	-

6. CONCLUSION

Simulation is well known tool for experimenting and studying Wireless Sensor Networks, due to the feasibility analysis and the difficulties of setting up real experiments or test beds simulation is must. Qualitative analysis will provide basic guideline to help implementer & researcher for selecting a desired simulation environment for WSN. Goal of the paper is to provide detailed study and background of different Wireless Sensor Network simulators and present the features and limitations for each of them. Knowledge of the pros and cons of different simulators is valuable asset, since it allows users to select the appropriate simulator.

7. REFERENCES

[1] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, “Next century challenges: scalable coordination in sensor networks”, ACM MobiCom’99, Washington, USA, 1999, pp. 263–270.

[2] Albeseder and M. Fuegger. Small PC-Network Simulation - a comprehensive performance case study. Research Report 77/2005, TU Wien, Institutf ̈urTechnischeInformatik, 2005.

[3] Nurul I. Sarkar and Syafnidar A. Halim,” A Review of Simulation of Telecommunication Networks: Simulators, Classification, Comparison, Methodologies, and Recommendations”, Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), March Edition, 2011

[4] “User Manual and documentation of Castalia” https://castalia.forge.nicta.com.au/index.php/en/documen

tation.html://castalia.npc.nicta.com.au/documentation.php. Accessed March 2015

[5] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Marino, J. Garcia-Haro; "Simulation Tools for Wireless Sensor Networks", Summer Simulation Multiconference, SPECTS, 2005, pp.2-9

[6] A Survey of Wireless Sensor Network Simulation Tools http://www1.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html

[7] C Perkins, EM Royer, S Das, Ad-Hoc On Demand Distance Vector Routing (AODV), IETF draft, 2000.

[8] Sourendra Sinha, ZenonChaczko, RyszardKlempous, "SNIPER: A Wireless Sensor Network Simulator", Computer Aided Systems Theory- EUROCAST , 2009, Volume 5717/2009, pp. 913-920

[9] Sobeih, W. Chen, J. C. Hou, L. Kung, N. Li, H. Lim, H. Tyan, H. Zhang, "J-Sim: A simulation and emulation environment for wireless sensor networks." In Proc. Annual Simulation Symposium (ANSS 2005), San Diego, CA, pp. 175–187, April2005.

[10] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, "EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks," in *Proceedings of General Track: 2004 USENIX Annual Technical Conference*, (Boston, USA), 27 June – 2 July 2004.

[11] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms," in *Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks*, vol. 10, (Hingham, MA, USA), pp. 563–579, Kluwer Academic Publishers, August 2005.

[12] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," in *IEEE Standard 802.15.4-2006 Part 15.4*, (New York, USA), 8 September 2006.

[13] J. S. Carson II, "Introduction to Modelling and Simulation," Winter Simulation Conference, December, 2004, pp. 1283-1289.Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, ANSI/IEEE Std. 802.11, 1999.

[14] M. Law and W. D. Kelton, "Simulation modelling and analysis", third Edition. New York: McGraw-Hill, 2000.

[15] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin, "EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks," in *Proceedings of General Track: 2004 USENIX Annual Technical Conference*, (Boston, USA), 27 June – 2 July 2004.

[16] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," in *IEEE Standard 802.15.4-2006 Part 15.4*, (New York, USA), 8 September 2006.

[17] T. Maret, "Overview of the Freemote Project." http://www.assembla.com/wiki/show/freemote/ Documentation. accessed March 2015.

[18] R. Kummer, T. Maret, P. Kropf, and J. F. Wagen, "FREEMOTE: A Wireless Sensor Networks Emulation System," in *Proceedings of 7th MINEMA workshop*, (Lappeenranta, Finland), 21 August 2008.

[19] J. Eriksson, A. Dunkels, N. Finne, F. ¨ Osterlind, and T. Voigt, "Mspsim – an extensible simulator for msp430-equipped sensor boards," in *Proceedings of EWSN, European Conference on Wireless Sensor Networks, Poster/Demo session*, (Delft, The Netherlands), 29–31 January 2007.

[20] "The Contiki OS." http://www.contiki-os.org. Accessed March 2015.

[21] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms," in *Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks*, vol. 10, (Hingham, MA, USA), pp. 563–579, Kluwer Academic Publishers, August 2005.

[22] S. Park, A. Savvides, M. B. Srivastava. "SensorSim: A Simulation Framework for Sensor Networks." In Proc. ACM Modeling, Analysis and Simulation of Wireles and Mobile Systems (MSWiM 2000), Boston, MA, pp. 104–111, August 2000