Exploration of New Simulation Tools for Wireless Sensor Networks

B. Sai Chand[1], K. Raghava Rao[2], S. Sreedhar Babu[3]

[1]UG project Scholar, Department of EEE, K L University, Guntur, Andhra Pradesh-522502, India
saichand.bitra@gmail.com

[2]Professor, School of Computing, K L University, Guntur Andhra Pradesh-522502, India
krraocse@gmail.com

[3]Associate Professor, Department of ECE, K L University, Guntur, Andhra Pradesh-522502, India
seva.sreedhar@kluniversity.in

Abstract: ***Wireless Sensor Networks (WSN) are formed by a large number of networked sensing nodes. To model analytically a WSN and it usually leads to oversimplified analysis with limited confidence. Deploying sensor networks test-beds supposes a huge effort. Therefore, simulation is essential to study WSN. However, it requires a suitable model based on solid assumptions and an appropriate framework to ease implementation. In addition, simulation results rely on the particular scenario under study (environment), hardware and physical layer assumptions, which are not usually accurate enough to capture the real behavior of a WSN, thus threatening the credibility of results. However, detailed models yields to scalability and performance issues, due to the large number of nodes that depending on application, have to be simulated. Therefore, the tradeoff between scalability and accuracy becomes a major issue when simulating WSN. In this illustration survey a suitable model for WSN simulation is introduced, highlighting features and criteria for choosing an appropriate simulation tool, model and framework.***

Keywords: WSN, survey, scalability, framework, simulation, model

1. Introduction

Wireless Sensor Networks can be considered a Network, formed by hundreds or thousands of sensing devices communicating by means of wireless transmission. Technological improvements in sensing, processing and communication coupled with improvements in fabrication technology have made it possible to use "Wireless Sensor Networks (WSN)" in a wide variety of applications. They can range from military espionages, disaster management to weather monitoring. So this makes wireless sensor networks technology a new research area which has emerged to provide wireless solutions for various wired sensor applications. Due to the optimization of cost, energy and size of sensor nodes, they are highly application dependent. Operations on sensor nodes are highly concurrent and reactive. Application development is also complicated due to the small size of the nodes. Nodes, in wireless sensor networks, are also susceptible to failures like system crash, system freezing …etc. In such situations, it is needed to restart the system many times testing wireless sensor networks. It can be frustrating. Although reasonable tools exist for evaluating large sensor networks in simulation, only a real sensor network test bed can provide the realism exigent to understand resource limitations, communication loss, and energy constraints. In most of the applications, we need to go through a trial phase to understand the detailed needs of the application, before we can design an optimized system. This necessitates the need for a test bed for the development of wireless sensor network (WSN). We would like to design a test bed which is flexible, cost efficient, easy to use, power efficient, provides excellent debugging facilities and covers all major requirements of wireless sensors networks. In WSNs, main factor arise was about energy used. There exists an increasing concern about the methodology and assumptions of simulations [1], [2]. On the other hand, implementing a complete model requires a considerable effort. A tool that helps to build a model is needed, and the user faces the task of selecting the appropriate one. Simulation software commonly provides a framework to model and reproduce the behavior of real systems. However, actual implementation and "secondary goals" of each tool differ considerably, that is, some may be designed to achieve good performance and others to provide a simple and friendly graphical interface or emulation capabilities. The aim of this paper is to provide some insight on the building blocks of a general simulation model for WSN, introducing its specific issues. Also, to facilitate newcomers the selection of the most appropriate tool for their needs, the most extended WSN simulation environments are reviewed. Section 2 introduces a wide vision of WSN simulator architecture and section 3 describes the capabilities and particular features of general and specific simulation packages currently used to simulate WSNs.

1.1 Importance of Simulation in WSNs

Building a WSNs test bed is very costly. Running real experiments on a test bed is costly and difficulty. Besides, repeatability is largely compromised since many factors affect the experimental results at the same time. It is hard to isolate a single aspect. Moreover, running real experiments are always time consuming. Therefore, WSNs simulation is important for WSNs development. Protocols, schemes, even new ideas can be evaluated in a very large scale. So, simulation is essential to study WSNs, being the common way to test new applications and protocols in the field. This leads to the recent boom of simulator development. However, obtaining solid conclusions from a simulation study is not a trivial task. There are two key aspects in WSNs simulators: (1) The correctness of the simulation models and (2) the suitability of a particular tool to implement the model.

A "correct" model based on solid assumption is mandatory to derive trustful results. In the rest of this survey several main-stream WSNs simulators are described and compared in more detail.

2. Sensor Network Simulations Models

With the development of simulation tools for WSN, their corresponding models have been introduced. By emulation, we mean the ability to combine simulated and real systems. Classical network simulators are detailed power and energy consumption models or environment models. This section describes a general component model, derived from [3], [4] for WSN simulation tools. This model is suitable for most of the evaluation tools employed in on-going research on WSN.

2.1 Network Model

Figure 1 depicts the general model at a network-wide scale. The following components are considered:

1) Nodes: Each node is a physical device monitoring a set of physical variables. Nodes communicate with each other via a common radio channel. Internally, a protocol stack controls communications. Unlike classical network models, sensor modes include a second group of components: The physical node tier, which is connected to the environment. Nodes are usually positioned in a two or three dimensional world. An additional "topology" component, not showed in figure 1 may control node coordinates. Depending on the application and deployment scenario, a WSN can contain from a few to several thousands of nodes. Section 2.2 describes the structure of a node.
2) Environment: The main difference between classical and WSN models are the additional "environment" component. This component models the generation and propagation of events that are sensed by the nodes, and trigger sensor actions, i.e. communication among nodes in the network. The events of interest are generally a physical magnitude as sound or seismic waves or temperature.
3) Transceiver: It characterizes the propagation of radio signals among the nodes in the network. Very detailed models use a "terrain" component, connected to the environment and radio channel components. The terrain component is taken into consideration to compute the propagation as part of the radio channel, and also influences the physical magnitude.
4) Micro controller: It has sink nodes, these are special nodes that, if present, receive data from the net, and process it. They may interrogate sensors about an event of interest. The use of sinks depends on the application and the tests performed by the simulator. And it has Agents; it is a generator of events of interest for the nodes. The agent may cause a variation in a physical magnitude, which propagates through the environment and stimulates the sensor. This component is useful when its behavior can be implemented independently from the environment, e.g., a mobile vehicle.

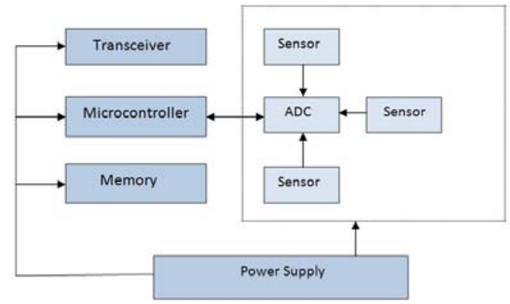

Figure.1 Wireless sensor network model

2.2. Node Model

Node behavior depends on interacting factors that cause cross-layer interdependencies. A convenient way to describe it is to divide a node into abstract tiers, as represented in Figure2.

- The 'Protocol-tier' comprises all the communication protocols. Typically, three layers coexist at this tier: A MAC layer, a routing layer and a specific application layer. Note that the operation of the protocol tier usually depends on the state of the physical tier described below, e.g. a routing layer can consider battery constraints to decide on packet route. Hence, an efficient method to interchange tier information must be developed.
- The 'physical-node' tier represents the hardware platform and its effects on the performance of the equipment. Actual composition of this tier may change depending on the specific application. The common elements of this tier are the set of physical sensors, the energy module and the mobility module. Physical Sensors describe the behavior of the monitoring hardware. Energy module simulates power consumption in the component hardware, a critical issue in WSN evaluation. Mobility module controls sensor position.
- The ' media-tier' is the link of the node with the "real world". A node is connected with the environment through: (1) A radio channel, and (2) through one or more physical channels. Physical channels receive environmental events as described in section 2.1

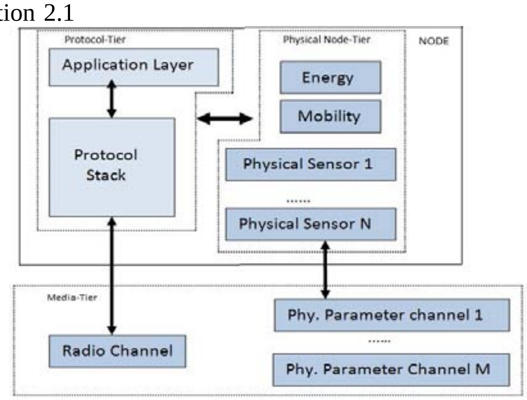

Figure. 2. Tier-based node model

Usually, the key properties to select suitable simulation environment are:

1) Reusability and availability.
2) Performance and scalability.
3) Support for rich-semantics scripting languages to define experiments and process results.
4) Graphical, debug and trace support.

In below we highlighted features and usage in the context of the WSN.

Reusability and availability: Simulation is used to test novel techniques in realistic and controlled scenarios. Researchers are usually interested in comparing the performance of a new technique against existing proposals. Therefore, two key aspects are: Does the simulation tool include implementations of common models? How easy is to modify or integrate a new model with the existing ones? The first question mainly depends on how long a framework has been used for, and how many people use it. Early and widely adopted frameworks have many available models and it is very likely that the new successful proposals will be added to next releases. All the general purpose packages include a more or less complete TCP/IP suite, which can be considered the minimum standard support. In addition, typical requirements for WSN simulators are: Ad-hoc routing support plus wireless MAC protocols, and propagation and mobility models to synthesize the physical node distribution. For specific tools the question is different: All the specific frameworks are able to execute native sensor code. Hence, every application, protocol or component developed for the actual sensor platform can be simulated or emulated. Only some specific parts are purely simulated, e.g. the radio channel or the physical media environment. Summing up, in this case protocols availability depends on the real availability of them for the target platform, and vice versa.

Performance and scalability: Performance and scalability is a major concern when facing WSN simulation. The former is usually bounded to the programming language effectiveness. The latter is constrained to the memory, processor and storage size requirements.

The configuration of a WSN typical trial requires to answer (at least) questions like: How many nodes are there in the test?, where is each node placed?, do nodes move?, how many of them?, how they move?, which energy model is used?, how many physical environments are?, how they generate events?, which physical magnitudes should measure each node?, which statistics must be measured in the experiment?, which are the parameters of the radio model? The vast amount of variables involved in the definition of a WSN experiment requires the use of specific input scripting languages, with high-level semantics. Additionally, it is likely that large quantities of output data will also be generated through many replicas of the experiments. Therefore, a suitable output scripting language that helps to obtain the results from the experiments quickly and precisely is desirable. Graphical, debug and trace support.

3. Sensor Networks Simulations Software Tools

In this section the most relevant simulation environments used to study WSN are introduced, and their main features and implementation issues described and discussed. The simulators are as fallows

3.1 NS-3

Network Simulator-3 (NS-3) is free, open source software. Like its predecessor, NS-3 relies on C++ for the implementation of the simulation models. However, NS-3 no longer uses oTcl [5] scripts to control the simulation, thus abandoning the problems which were introduced by the combination of C++ and oTcl in NS-2. Instead, network simulations in ns-3 can be implemented in pure C++, while parts of the simulation optionally can be realized using Python as well. Moreover, NS-3 integrates architectural concepts. It is a simulator with good scalability characteristics. These design decisions were made at expense of compatibility. In fact, NS-2 models need to be ported to NS-3 in a manual way. Besides performance improvements, the feature set of the simulator is also about to be extended. NS-3 advantages over NS-2 of being its modularity, flexible tracing and multi-technology (from the beginning). NS-3 performance for large scale simulations, capable to simulate large amount of nodes (20000 nodes or more). NS-3 allows researchers to study Internet protocols and large-scale systems in a controlled environment. NS-3 lacks an integrated development/visualization environment (IDE).

3.2 OPNET

Optimum Network Performance (OPNET), OPNET Modeler in particular, is a research oriented package, and it is one of the leading simulators used in the industry. OPNET [6] provides relatively much powerful visual or graphical modeling approach (Graphical user interface).It enables graphical editors to us to edit our own devices, configure your own networks, design your own protocols, define your own packet formats, etc. OPNET is specialized for network research and development. It can be flexibly used to study communication networks, devices, protocols, and various applications. Because of the fact of being a commercial software provider, OPNET has Object-oriented programming technique is used to create the mapping from the graphical design to the implementation of the real systems. We can see all the topology configuration and simulation results. The parameters can also be adjusted and the experiments can be repeated easily through easy operation through the GUI. OPNET workflow goes like this first we need to "create network models" and then "choose statistics" then "run simulation" and finally "View and analyze results". For modeling, it provides intuitive graphical environment to create all kinds of models of protocols. For simulating, it uses three different advanced simulations technologies and can be used to address a wide range of studies. For analysis, the simulation results and data can be analyzed and displayed very easily. User friendly graphs, charts, statistics, and even animation can be generated by OPNET for user's convenience.

3.3 GloMoSim

Global Mobile Information System Simulator (GloMoSim) is network protocol simulation software that simulates wireless and wired network systems. GloMoSim [7] uses the Parsec compiler to compile the simulation protocols. Parsec is a C-based simulation language, developed by the Parallel Computing Laboratory. GloMoSim is specifically used for mobile wireless networks. Its libraries are built in Parsec. It is layered architecture with easy plug-in capability. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications using includes multicast, large scalability support. For multiprocessor simulation millions of sensor nodes, for single machine, around several handed nodes, main advantages with use of GloMoSim are achievement of large scalability, good mobility models specify for wireless simulation, no specific routing protocols for sensor network, no energy consumption models, has transport layer and IP address support. Hard for user to simulate large sensor network so no hardware environment, demonstrated scalability using very high fidelity models asymmetric communications using direct satellite broadcasts. Several proposals for WSN protocols have been tested with it. Recently, a development kit for WSN has been released, sQualnet [8].

3.4 MiXiM

MiXiM (Mixed simulator), MiXiM [8] has been introduced as a very powerful extension to simulate wireless and mobile networks using the discrete event simulator OMNeT++ [9]. Its framework is based on OMNeT++. MiXiM aims to provide the developer with a powerful and feature-rich toolbox to enable and facilitate the simulation and performance analysis of wireless networks. At the same time the structure and design of MiXiM is such, that it tries to hide the complexity of such simulations and provides the developer with a clean and easy to use interface. It was created for mobile and wireless networks (wireless sensor networks, body area networks, ad-hoc networks, vehicular networks, etc.).MiXiM enables to model both 2D and 3D environment. Apart from the communicating devices, several other objects like houses or walls to simulate radio propagation of signals can be placed in the environment. The positions of all nodes can be managed by Connection Manager.

3.5 Castalia

Castalia is an open-source simulator for wireless sensor networks. Castalia is based on OMNeT++ platform and is developed for networks of low power embedded devices such as wireless sensor nodes. It is based on the OMNeT++ platform and used by researchers and developers to test their distributed algorithms and/or protocols in a realistic wireless channel and radio model, with a realistic node behavior especially relating to access of the radio. Castalia is a “tunable” simulator where many parameters may be adjusted using input to simulate real environment. Several examples of simulation applications are provided. In Castalia, nodes are OMNeT++ modules that do not connect to each other directly but through the wireless channel module. When a node has a packet to send, it goes through the wireless channel which then decides which nodes should receive the packet. Due to its accurate modeling and relative ease of use, Castalia has gained wide acceptance in the WSN research Community with a number of citations in the literature. Despite its acceptance though, there has not been a rigorous evaluation study on the performance capabilities and scalability of Castalia. Researchers can easily import their algorithms or protocols into Castalia, and then merge with other provided features. Castalia also support researchers and developers to build their own protocols.

3.6 J-Sim

J-Sim is a discrete event network simulator built in Java. This simulator provides GUI library, which facilities users to model or compile the Mathematical Modeling Language, a “text-based language” written to J-Sim models. J-Sim provides open source models and online documents. This simulator is commonly used in physiology and biomedicine areas, but it also can be used in WSN simulation. In addition, J-Sim can simulate real-time processes. J-Sim contains both merits and limitations when people use it to simulate WSNs. To the merits, firstly, models in J-Sim have good reusability and interchangeability, which facilities easily simulation. Secondly, J-Sim contains large number of protocols; this simulator can also support data diffusions, routings and localization simulations in WSNs by detail models in the protocols of J-Sim. J-Sim can simulate radio channels and power consumptions in WSNs. GUI library, which can help users to trace and debug programs. The independent platform is easy for users to choose specific components to solve the individual problem. Comparing with NS-2, J-Sim can simulate larger number of sensor nodes, around 500, and J-Sim can save lots of memory sizes. However, this simulator has some limitations. The execution time is much longer. Because J-Sim was not originally designed to simulate WSNs, the inherently design of J-Sim makes users hardly add new protocols or node components.

3.7 AVRORA

Avrora is a simulator specifically designed for WSNs built in Java. Similar to ATEMU, Avrora can also simulate AVR based microcontroller MICA2 sensor nodes. This simulator was developed by University of California, Los Angeles, Compilers Group. Avrora provides a wide range of tools that can be used in simulating WSNs. This simulator combines the merits of TOSSIM [10] and ATEMU, and limits their drawbacks. Avrora also supports energy consumption simulation. This simulator provides open sources and online documents. Avrora is an instruction-level simulator, which removes the gap between TOSSIM and ATEMU. The codes in Avrora run instruction by instruction, which provides faster speed and better scalability. Avrora can support thousands of nodes simulation, and can save much more execution time with similar accuracy. Coming to the AVRORA limitations, It does not have GUI. In addition, Avrora cannot simulate network management algorithms because it does not provide network communication tools.

4. Conclusion

This paper brings out latest simulation tools in building and deploying new sensor networks applications models and best utilization of tools. Simulation is not enough itself to have accurate results and need a real time test bed to have real time results. But latest tools are good and close to reality in finding results of Senor Network’s simulation.

Future more tools come into this area of research automatically generated model as per application oriented and based on domain.

References

[1] K. Pawlikowski, H. D. Joshua Jeong, J. S. Ruth Lee. "On Credibility of Simulation Studies of Telecommunication Networks", IEEE Communications Magazine, vol. 40, no.1, pp. 132–139, January 2002.

[2] D. Kotz, C. Newport, B. Gray, J. Liu, Y. Yuan, C. Elliot. "Experimental Evaluation of Wireless Simulation Assumptions", In Proc. of the 7th ACM/IEEE Int. Symposium

[3] on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04), Venice, Italy, pp. 78–82, October 2004. S. Park, A. Savvides, M. B. Srivastava. "SensorSim: A Simulation Framework for Sensor Networks", In Proc. ACM Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2000), Boston, MA, pp.104– 111, August 2000

[4] 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, April 2005.

[5] MIT Object Tcl. [Online].Available: http://bmrc.berkeley.edu/research/cmt/cmtdoc/otcl [6]http://opnet.com/products/modeler/

[6] Global Mobile Information Systems Simulation Library Available: http://pcl.cs.ucla.edu/projects/glomosim/

[7] SQualnet: A Scalable Simulation Framework for Sensor

[8] Networks [online]

[9] Available: htpp://nesl.ee.ucla.edu/projects/squalnet/MiXiM simulator for wireless and mobile networks using OMNeT++ [online] Available: http://mixim.sourceforge.net/

[10] A.Varga. OMNeT++ Discrete Event Simulation System Available: http://www.omnetpp.org.

[11] *TOSSIM* - TinyOS Documentation Wiki Available:# http://docs.tinyos.net/tinywiki/index.php/TOSSIM

Author Profile

B. Sai Chand is doing his IIIrd year of B.Tech degree in Electrical and Electronics Engineering in K L University. He was joint-secretary for IEEE collage chapter. His area of interest includes Energy Systems, Wireless Sensor networks and power quality control.

K. Raghava Rao Currently, he is a professor in the School of Computing, K L University, A.P, India. His areas of interest include Sensor Networks, Data Mining and Sensor Web Services. He published several papers in national and international journals. Currently, he is carrying DST Funded project on Sensor Web enablement. He published a text titled "Core Java".

S. Sreedhar Babu Currently, he is Associate professor in the Department of Electronics and Communication Engineering, K L University, A.P, India. His research interests are in the field of Sensor Networks, Image processing and Embedded systems.