SURVEY ON SIMULATION AND EMULATION TOOLS IN WIRELESS SENSOR NETWORK

K.BALAJI*
Assistant Professor, Department Of Computer Science And Engineering
SNS College Of Engineering
Coimbatore, Tamilnadu

B.JAI VIDHYA*
PG Scholar, Department Of Computer Science And Engineering
SNS College Of Engineering
Coimbatore, Tamilnadu
balajikothandaraman@ymail.com
vidhyabala.vidhya@gmail.com
Mobile no: 8056754909

Abstract: Sensor networks are dense wireless networks of small, low-cost sensors which collect and propagate environmental data. Wireless sensor networks (WSNs) assist monitoring and controlling of physical environments from remote location with better accuracy. They have applications in a variety of fields such as environmental monitoring, military applications, and water or waste water monitoring and health care applications. Sensor nodes have various energy and computational constraints because of their economical nature and adhoc method of deployment. The objective of this involvement is to present expositive review content on currently available experimental tools used for most emerging field. Currently due to high cost of sensor nodes, the most researches in wireless sensor networks area is performed by using the experimental tools in various institutes and research centre's before implementing real one. Also the facts gathered from these experimental tools can be realistic and convenient. So, the experimental tools provide the better option for studying the behavior of WSNs before and after implanting the physical one.

Keywords: Wireless Sensor Networks, Experimental tools, Sensor nodes, adhoc method

I. INTRODUCTION

A. WSN

Wireless Sensor Networks which consist of spatially distributed self-configurable sensors, entirely meet the requirement. A typical WSN is built of several hundreds or even thousands of sensor nodes. A sensor node may vary in size. It can be generally described as a network of nodes that cooperatively sense and may control the environment enabling interaction between persons or computers and the surrounding environment.

B. SIMULATORS AND EMULATORS

A large WSN application should be implemented after checking the design specifications and requirements with the selected hardware. By using simulators or emulators the architectures, services and suitable protocols are developed, tested and evaluated. When the number of nodes are increased then it is not possible to prefer test-bed to evaluate certain parameters, because it will be more expensive and takes long time. But test-bed is the only option to develop reliable and portable network; in this case simulation becomes invalid. So the simulator or emulator is chosen based on the parameter concerned and application specific, that balancing both merits and demerits. In Wireless Sensor Networks the system suitability is checked by using simulation before using it. The simulation can be used to test the scalability of algorithms used and it doesn't consider the hardware platform constraints. And it is also used to simplify the software development process for a particular WSN application. So, Simulation is the most favorable tool for mobile network.

An emulator is a special type of simulator that aims to evaluate WSN applications in a realistic way to have high performance. Emulator is a better choice for testing, debugging and performance evaluation of WSN applications directly compared with simulator. For implementation the emulator tools, cross-layer techniques are also taken into account in addition with individual lower layer functions. Emulators allow different software/hardware to be experienced on a single platform without the original system requirements; therefore, allowing cheaper alternatives in many digital level scenarios. Even though the initial development costs may be high, an emulator can be very cost efficient over a long term due to its versatility.

C. DIFFERENCE BETWEEN SIMULATORS AND EMULATORS

- Simulation can be used in task or situational training areas in order to allow humans to expect certain situations and be able to react properly; decision-making environments to test and select alternatives based on some criteria; scientific research contexts to analyze and interpret data; and understanding and behavior prediction of natural systems, such as in studies of stellar evolution or atmospheric conditions. And an emulator duplicates the functions of one system using a different system, so that the second system behaves like the first system. This focus on exact reproduction of external behavior in contrast to some other forms of computer simulation, which concern an abstract model of the system being simulated.
- Simulation involves modeling the core target. The end result of a good simulation is that the simulation model will emulate the target which it is simulating. And emulation is the process of mimicking the external behavior to match an existing target. The internal state of the emulation mechanism does not have to be reflecting accurate internal state of the target which it is emulating.

II. SIMULATORS FOR WIRELESS SENSOR NETWORK

A simulator is software that imitates selected parts of the behavior of the real world and is normally used as a tool for research and development. Depending on the intended usage of the simulator, different parts of the real-world system are modeled and imitated.

A.TYPES OF SIMULATION TOOLS

The Simulation tools are of two types. And they are of

- *SYNCHRONOUS SIMULATION*

The synchronous simulation is based on rounds. At the beginning of each round, the simulators increment by one unit.

- ASYNCHRONOUS SIMULATION

The asynchronous simulation is merely event based. The simulator holds a list of events, which is sorted by the time when these events should happen (arrival of message, execution of timer-handler).

B. CATEGORISATION OF SIMULATORS

- GENERIC NETWORK SIMULATOR

The focus of the simulation is on networking the simulator typically provides detailed simulation of the radio medium, but less detailed simulation of the nodes. The user of the simulator typically writes the simulation application in a high level language different from the one used for the real sensor network.

- CODE LEVEL SIMULATOR

Code level simulators use the code in simulation as in real sensor network nodes. The code is compiled for the machine that is running the simulator, typically a PC workstation that is magnitudes faster than the sensor node.

- FIRMWARE LEVEL SIMULATOR

These simulators are based on emulation of the sensor nodes and the software that runs in the simulator is the definite firmware that can be deployed in the real sensor network.

- ALGORITHM LEVEL SIMULATOR

  These simulators focus on the logic, data structure and presentation of the algorithms.

- PACKET LEVEL SIMULATOR

These simulators implement the Data Link and Physical Layers in a typical OSI network stack.

- INSTRUCTION LEVEL SIMULATOR

These simulators model the CPU execution at the level of instructions or even cycles. They are often regarded as emulators.

III. SIMULATION TOOLS

A. NS-3

The first release, ns-3.1 was made in June 2008, and the project continued making periodical software releases, and more recently has moved to three releases per year. ns-3 made its eighteenth release (ns-3.18) in the third quarter of 2013. NS-3 is built using C++ and Python with scripting capability. Its library is wrapped to python to the pybindgen library which delegates the parsing of the ns-3 C++ headers to gccxml and pygccxml to generate automatically the corresponding C++ binding glue. These automatically generated C++ files are finally compiled into the ns-3 python module to allow users to interact with the C++ ns-3 models and core through python scripts. The ns-3 simulator features an integrated attribute-based system to manage and per-

instance values for simulation parameters. All of the configurable default values for parameters are managed by this system, integrated with command-line argument processing, Doxygen documentation, and an XML-based and optional GTK-based configuration subsystem.

B.GLoMoSim

Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wired communication networks. The node aggregation technique is introduced into GloMoSim to give significant benefits to the simulation performance. In GloMoSim, each node represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined by the physical position of the nodes.

C.MiXiM

MiXiM (Mixed simulator), MiXiM has been introduced as a very powerful extension to simulate wireless and mobile networks using the discrete event simulator OMNeT++. 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.

D.Castalia

Castalia is a simulator for Wireless Sensor Networks (WSN), Body Area Networks and generally networks of low-power embedded devices. 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 uses the lognormal shadowing model as one of the ways to model average path loss, which has been shown to explain empirical data in WSN.[1] It also models temporal variation of path loss in an effort to capture fading phenomena in changing environments (i.e., the nodes or parts of the environment are moving). Castalia's temporal variation modeling is designed to be fitted to measured data instead of making specific assumptions on the creation of fast fading. Other features of Castalia include: physical process modeling, sensing device bias and noise, node clock drift, and several MAC and routing protocols implemented. Castalia is developed at the National ICT Australia since 2006. Since 2007 it is made public as an open source project under the Academic Public License. The current release version is 3.3.

IV. EMULATION TOOLS

A.MSPSim

It is a Java-based instruction level emulator of the MSP430 series microprocessor and emulation of some sensor networking platforms. Supports loading of IHEX and ELF firmware files, and has some tools for monitoring stack, setting breakpoints, and profiling.

B. EmStar

EmStar is an emulator specifically designed for WSN built in C. It is a trace-driven emulator [Girod04] running in real-time. People can run this emulator on Linux operating system. This emulator supports to develop WSN application on better hardware sensors. Besides libraries, tools and services, an extension of Linux microkernel is included in EmStar emulator.

C. EmPro

It is environment/energy emulation and profiling system for WSNs. It accurately outputs electrical signals to emulate not only digital and analog inputs to the sensors but also the power sources as well as RF attenuation according to pre-programmed sequences. This emulation approach enables researchers to run the networked sensors in real-time in a realistic manner with full controllability and reproducibility. EmPro in profiling mode can also capture the observable behavior of WSNs for detailed analysis. Experimental results on the Eco and Mica-2 WSN platforms show that EmPro can drive these hardware systems in real-time with high accuracy.

D. Freemote Emulator

It is a lightweight and distributed Java based emulation tool for developing WSN softwares. The objective of this platform is to support the emerging Java based Motes based on optimized JVM and platforms. The Free- mote emulator focuses on behavior credibility by mixing emulated nodes and real nodes reachable through a specialized bridge rather than on time based performance evaluation accuracy. This emulator splits the

Software architecture of a Mote in three independent layers connected through well defined interfaces (Application, Routing and Data Link and Physical). Freemote is a fully configurable WSNs emulator. It can easily be used to develop new algorithms for WSNs but is also capable to support large scale experiments (up to 10,000 nodes) including all kind of real nodes based on the IEEE 802.15.4 communication standard. It also allows the developer following the behavior of WSNs and debugging tricky implementation problems.

V. COMPARISON TABLE FOR SIMULATION/EMULATION TOOLS

S.NO	Tools	Key Features	Limitations
1	NS-3	It performs better than NS-2. It supports visualization.	It still need contributed codes and need to develop more.
2	GloMoSim	It is used in a parallel environment. It is designed to be extensible.	It is effectively limited to IP networks. It has problems in models and MAC protocols.
3	MiXiM	It provides rich toolbox to enable simulation and facilitate performance.	It needs to be developed in several areas.
4	Castalia	It includes physical process modeling, sensing device bias and noise. The routing protocols are implemented.	It is not a sensor platform specific. It is not used for testing the code.
5	MSPSim	It has tools for monitoring stack and profiling.	It needs to be accessed via command lines.
6	EmStar	It provides an interface with hardware. It allows use of actual communication and sensors.	It uses simple environment model and network medium. It lacks in algorithm and does not support parallel simulations.
7	EmPro	It is used to reproduce and can be controlled.	It needs to improve the response time.
8	Freemote Emulator	It provides a useful visualization tool.	It needs to extend in Java motes.

VI. CONCLUSION

This paper brings out latest simulation tools in building and deploying new sensor networks applications models and best utilization of tools. But latest tools are good and close to reality in finding results of Senor Network's simulation. In Future, more tools come into this area of research automatically generated model as per application oriented and based on domain.

REFERENCES

[1] Becker.M, Timm-Giel.A., Murray, K., Lynch, C., Görg C. Pesch.D." Comparative Simulations of WSN" Cunningham, P., Cunningham, and M. (eds.): ICT-Mobile Summit 2008.

[2] Mrs. Poonam Chhimwal, Dhajvir Singh Rai, Deepesh Rawat "Comparison between Different Wireless Sensor Simulation Tools", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834, p- ISSN: 2278-8735. Volume 5, Issue 2 (Mar. - Apr. 2013), PP 54-60

[3] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/, 2002.

[4] Anu Maria "Introduction to Modeling and Simulation" proceedings of the 1997 Winter Simulation conference Ed S.Andradottir, K.J.Healy, D.H.Withers, and B.L.Nelson.

[5] Suraj G. Gupta, Mangesh M. Ghonge, Parag D. Thakare, Dr. P. M. Jawandhiya " Open-Source Network Simulation Tools: An Overview" ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 4, April 2013.