ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

Simulators and Emulators used for Wireless Sensor Network

Dr. V. Vasanthi[1]

Assistant .Professor, Department of Computer Science, Rathinam College of Arts and Science, Coimbatore, India

Abstract: Wireless Sensor Networks (WSNs) have been gaining growing interest in the past years; WSN tasks of a each sensor network work include not only monitoring and measuring certain phenomena, but also delivering gathered data. Sensor network require applying different techniques used in wired and wireless network. Simulation tools for wireless sensor networks are increasingly being used to study sensor webs and to test new applications and protocols in this evolving research field. It is useful to researcher to verify new ideas and analyze with proposed algorithms in virtual environment and help to avoid expensive hardware usage and time consumption also. This paper provides a comprehensive survey and comparisons of various popular sensor network simulators with a view to help researchers to choose the best simulator available for a particular application environment.

Keywords: Wireless sensor networks, Simulator, Emulator, Comparison, Survey Performance evaluation

I. INTRODUCTION

With the development of embedded system and network technology, there has been growing interest using low power devices. Wireless Sensor Networks (WSNs), which consist of spatially distributed self-configurable sensors, perfectly meet the requirement. The sensors provide the ability to monitor physical or environmental conditions, such as temperature, humidity, vibration, pressure, sound, motion and etc, with very low energy consumption.

The sensors have ability to transmit the information sensing data to the base station. Most modern WSNs are bi-directional, enabling two-way communication, which could collect sensing data from sensors to the base station as well as disseminate commands from base station to end sensors. The first WSNs is used and motivated by military applications such as battlefield surveillance; WSNs are widely used in industrial environments, residential environments and wildlife environments. Structure health monitoring, healthcare applications, home automation, and animal tracking become representative WSNs applications.

A typical Wireless Sensor Network (WSN) is built using thousands of “sensor nodes”. Each node has the ability to communication with every other node wirelessly, thus a typical sensor node has several components: a radio transceiver with an antenna which has the ability to send or receive packets, a microcontroller which could process the data and schedule relative tasks, several kinds of sensors sensing the environment data, and batteries providing energy supply. Power constraints, limited resources and hardware and reliability are major problems in sensor network and the nodes than those found in conventional networks and these are considered while developing protocols for use in sensor networks. The main reason is more number of base stations is involved and many nodes are intersected and then the end user is involved. The output of these sensor nodes are transmitted to the base station for analysis, logging and data collection.

Simulating Motivation

Sensor network is developed and deployed in different fields and applications. Networking is an important concept and it plays a vital role in everyday activities. The number of network users is highly increased and new software and protocols are used in recent trends. The Virtual Internetwork Testbed project is first project is used to analyze the protocol operations in both high and low level .The simulator requires both non-functional and specific requirements are tested and evaluated.

Non-functional requirements are it is open source, platform independence and visualization module.

Functional requirements are provided like hardware simulation, battery and power models, propagation modeling, protocol modeling, physical environment and emulation.

II. TYPES OF SIMULATIONS

There are three types of simulation: Monte Carlo Simulation, Trace-Driven Simulation and Discrete-Event Simulations [11].

Discrete-event simulation [11] is widely used in WSNs, because it can easily and simulate on different sensor nodes. It can list the pending events, which can be simulated by routines. The global variables, which describe the system state, can represent the simulation time, which allow the scheduler to predict this time in advance. This simulation includes input routines, output routines, initial routines, and trace routines. In addition to

DOI 10.17148/IJARCCE

that it provides dynamic memory management, which can add or drop the entities in the model. Debugger breakpoints are provided in discrete-event simulation, thus users can check the code step by step without disrupting the program operation.

In the Trace-Driven Simulation [Jain91] different services are available. This kind of simulation is commonly used in real system. The simulation results have more realistic and it provides more accurate workload; these detail information allow users to deeply study the simulation model. Usually, input values in this simulation constant unchanged. However, this simulation also contains some drawbacks. The tool, which is using firmware as well as hardware to perform the simulation, is called emulator [Imran10]. Emulation can combine both software and hardware implementation. Emulator implements in real nodes, thus it may provide more precision performance. Usually emulator has highly scalability, which can emulate numerous sensor nodes at the same time. A study on simulation tools and their advantage and disadvantage as follows.

III. SIMULATION TOOLS

This section illustrates main-stream simulation tools used in WSNs are Glomosim, NS-2, Sensor sim, prowler, omnet++, TOSSIM, EmStar, OMNeT++, J-Sim, ATEMU, and Avrora, and analyzes the advantage and disadvantage of each simulation tool. The network simulator is discrete event packet level simulator. The network simulator covers a very large number of application of different kind of protocols of different network types consisting of different network elements and traffic models. Network simulator is a package of tools that simulates behavior of networks such as creating network topologies, log events that happen under any load,analyze the events and understand the network.

3.1 NS-2

NS-2[12] is the abbreviation of Network simulator version two, which first been developed by 1989 using as the REAL network simulator. Now, NS-2 is supported by (DARPA) Defense Advanced Research Projects Agency and (NSF) National Science Foundation. NS-2 is a discrete event network simulator built in Object-Oriented extension of Tool Command Language and C++ [C++]. People can run NS-2 simulator on Linux Operating Systems or on Cygwin. This simulator is open source and provides online document. It helps to use network simulator and to get acquainted with the simulated objects and understand the operations of network simulation and we also need to analyze the behavior of the simulation object using network simulation. Steps followed to write a program

- Initialization and termination aspects of network simulator.
- Defining the network nodes, links, queues and topology as well.
- Defining the agents and their applications
- Network Animator(NAM)
- Tracing

Advantage

- Non-specific network simulator
- Open source it saves time and money and easy to modify and improve the code.
- It helps to increase the efficiency of simulation.
- It is used to provide details of the protocols and their operation.
- It is used to reduce packet and event processing time.
- OTcl helps in the following way:
- With the help of OTcl we can describe different network topologies
- It helps us to specify the protocols and their applications
- It allows fast development
- Tcl is compatible with many platforms and it is flexible for integration
- Tcl is very easy to use and it is available in free

Disadvantage

- Need to know Scripting language and modeling techniques
- TCL(Tool Command Language)
- More Complex sometimes to do desired job and time consuming than other.
- No GUI
- Continuous change of code base the result is not consistent.

3.2 TOSSIM

Tossim[8] is an emulator specifically designed for WSN running on TinyOS, which is an open source operating system targeting embedded operating system. In 2003, TOSSIM was first developed by UC Berkeley's TinyOS project team. TOSSIM is a bit-level discrete event network emulator built in Python [Python], a high-level programming language emphasizing code readability, and C++. People can run TOSSIM on Linux Operating Systems or on Cygwin on Windows. TOSSIM also provides open sources and online documents.

Advantage

- open source model free online document save the emulation cost
- support thousands of nodes simulation
- more accurately simulate the real world situation

Disadvantage

- TOSSIM's run-instantly execution model does not capture CPU time.
- It does not model energy consumption, though there is an add-on

3.3 EmStar

EmStar [11] is an emulator specifically designed in C for WSN, and it was first developed by University of

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

California, Los Angeles. EmStar 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.

Advantage

- allows the users to run each module separately without sacrificing the reusability of the software
- Environment is flexible for deployment and simulation among sensors.
- It is GUI and it is helpful for users to control electronic devices.
- many online documents to facilities the widely use of this emulator

Disadvantage

- Only applicable to iPAQ-class sensor nodes and MICA2 motes
- Limited Scalability
- Does not support parallel simulation
- Not efficient as other frameworks

3.4 OMNeT++

OMNeT++ [14] is a discrete event network simulator built in C++. OMNeT++ provides both a noncommercial license, used at academic institutions or non-profit research organizations, and a commercial license, used at "for-profit" environments. This simulator supports module programming model. Users can run OMNeT++ simulator on Linux Operating Systems, Unix-like system and Windows. OMNeT++ is a popular non-specific network simulator, which can be used in both wire and wireless area. Most of frameworks and simulation models in OMNeT++ are open sources.

Advantage

- GUI
- support MAC protocols and localized protocol
- Channel control

3.5 J-Sim

J-Sim[6] 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.

Advantage

- Reusability, interchangeability
- support MAC protocols and localized protocol
- supports around 500 nodes
- Independently platform
- Saves memory size

Disadvantage

- Low efficiency of simulation
- Unnecessary run-time overhead

3.6 ATEMU

ATEMU [7] is an emulator of an AVR processor for WSN built in C; AVR is a single chip microcontroller commonly used in the MICA platform. ATEMU provides GUI, Xatdb; people can use this GUI to run codes on sensor nodes, debug codes and monitor program executions. People can run ATEMU on Solaris and Linux operating system. ATEMU is a specific emulator for WSNs; it can support users to run TinyOS on MICA2 hardware. ATEMU can emulate not only the communication among the sensors, but also every instruction implemented in each sensor. This emulator provides open sources and online documents.

Advantage

- Multiple nodes at the same time and each node can run different programs
- Large library with hard devices

Disadvantage

- Simulation time is much longer than other simulation tools
- Only few functions are used to simulate routing and clustering problems

3.7 Avrora

Avrora [2] 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.

Advantage

- It is a combination of TOSSIM and ATEMU simulators
- It supports for energy consumption
- It provides open source and online documents
- Supports faster speed and better scalability
- More accuracy than Tossim

Disadvantage

- Not have GUI
- It doesn’t support network communication tools

3.8 SENS

A Sensor Environment and Network Simulator. SENS is a wireless sensor network simulator with modular, layered architecture with customizable components which model an application, network communication, and the physical environment. It enables realistic simulations, by using values from real sensors to represent the behavior of component implementations. Such behavior includes sound and radio signal strength characteristics and power usage.

Advantage

- It is writtern in C++
- Platform independent
- WSN applications feature tight integration of computation, communication
- Interaction with the physical environment

Disadvantage

- Less customizable

3.9 COOJA (COntiki Os JAva)

COOJA is a simulator for the Contiki sensor node operating system. MSP Sim can be integrated into COOJA, forming COOJA/MSP Sim. It allows simultaneous cross-level simulation at application, operating system and machine code instruction set level. COOJA combines low level simulation of sensor node hardware and simulation of high-level behavior in a single simulation.

Advantage

- COOJA is flexible and extensible in that all levels of the system can be changed
- Sensor node platforms, operating system software, radio transceivers, and radio propagation models.
- COOJA is a Java application, all interaction with compiled Contiki code is done through Java Native Interface (JNI).

Disadvantage

- It is extensible and it is low efficiency
- GUI is absent and making extensible and time dependent difficult.

3.10. Castalia

Castalia is an application-level simulator for Wireless Sensor Network based on OMNeT++. It can be used to evaluate different platform characteristics for specific applications, since it is highly parametric, and can simulate a wide range of platforms. In Castalia, sensor nodes are implemented as compound modules, consisting of sub-modules that represent, for instance, network stack layers, application, and sensor.

Advantage

- sensing device noise, bias and node clock drift

Disadvantage

- Reliable and realistic framework for the first order validation of an algorithm before moving to implementation on a specific sensor platform.

IV. CONCLUSION

The purpose of this survey is to give a general picture of main-stream simulation tools using in WSNs, and help people to choose different simulation tools according to different needs. In the beginning part, this survey illustrates what is WSNs, why they need simulation, and what specific features should be considered when simulating WSNs. Then, this survey analyzes of main-stream simulators: NS-2, TOSSIM, EmStar, OMNeT++, J-Sim, ATEMU, and Avrora, SENS, COOJA, Castalia and compares their merits and limitations, shown in Table 1. Both general simulators and specific simulators are evaluated in this survey. The general simulators usually lack some functions to provide specific simulations in WSNs, however specific simulators with more comprehensive functions may perform better. According to different targets to choose different simulation tools in WSNs will be more efficient and effective. Emulation environment is special type of simulator is to enable realistic and sensible performance evaluation for WSN applications. The performance of WSN specific application depends on the factors in addition to its implementation using tools. The future research should in the simulation for understanding real world WSN specific scenarios as a good challenge. It proved that more realistic implementation on energy monitoring, open source, platform independent. Simulator developer concentrates on implementation of application using cross layer approach as challenge.

REFERENCES

[1] Sourendra Sinha, Zenon Chaczko, Ryszard Klempous, “SNIPER: A Wireless Sensor Network Simulator”, Computer Aided Systems Theory- EUROCAST , 2009, Volume 5717/2009, pp. 913-920.

[2] 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,

[3] Sangho Yi, Hong Min, Yookun Cho, Jiman Hong, “SensorMaker: A Wireless Sensor Network Simulator for Scalable and Fine-Grained Instrumentation”, computational science and its application-ICCSA, 2008, Volume 5072/2008, pp. 800-810

[4] Lei Shu, Chun Wu, Yan Zhang, Jiming Chen, Lei Wang, Manfred Hauswirth, “NetTopo: Beyond Simulator and Visualizer for Wireless Sensor Networks”, Future Generation Communication and Networking - FGCN , 2008, Volume1, pp. 17-20, ISBN: 978-0-7695-3431-2,

[5] Yunjiao Xue, Ho Sung Lee, Ming Yang, Kumarawadu, P., Ghenniwa, H.H., Weiming Shen, “Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks”, Electrical and Computer Engineering, CCECE Canadian Conference on, 22-26 April 2007, pp.1372 – 1375, ISBN: 1-4244-1020-7,

[6] Muhammad Imran, Abas Md Said, Halabi Hasbullah, “A Survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks”, Information Techonology(ITSim), 2010 International Symposium in, June 2010, pp. 897 – 902. ISBN: 978-1-4244-6715-0

[7] J. Polley, D. Blazakis, J. McGee , D. Rusk, J.S. Baras, “ATEMU: A Fine-grained Sensor Network Simulator”, First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, October 4-7, 2004.

[8] Clay Stevens, Colin Lyons, Ronny Hendrych, Ricardo Simon Carbajo, Meriel Huggard, Ciaran Mc Goldrick, “Simulating Mobility in WSNs: Bridging the gap between ns-2 and TOSSIM 2.x”, 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, 2009, ISBN: 978-0-7695-3868-6.

[9] Philip Levis, Nelson Lee, Matt Welsh, David Culler, “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, SenSys, 2003, ISBN:1-58113-707-9

[10] Raj Jain, "Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling", Wiley Computer Publishing, John Wiley & Sons, Inc, 1991, ISBN: 0471503363.

[11] Discrete_event_simulation, , Description: an introduction of discrete-event-simulation in wiki webpage

[12] J. Elson, S. Bien, N. Busek, V. Bychkovskiy, A. Cerpa, D. Ganesan, L. Girod, B. Greenstein, T. Schoellhammer, T. Stathopoulos, D. Estrin, "EmStar: An Environment for Developing Wireless Embedded Systems Software", 2003.

[13] Lewis Girod, Jeremy Elson, Alberto Cerpa, Thanos Stathopoulos, Nithya Ramanathan, Deborah Estrin, " EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks", USENIX Technical Conference, 2004.

TABLE 1: COMPARISON OF MAIN-STREAM SIMULATION TOOLS

DES = Discrete-Event Simulation TDS = Trace-Driven Simulation
GS = General Simulator SS = Specific Simulator

Tools	Type	DES/ TDS	GUI	Open Sources & Online Documents	GS/SS	Summary
NS-2	Simulator	DES	No	Yes	GS	• cannot simulate more than 100 nodes • cannot simulate problems of the bandwidth or the power consumption in WSNs
TOSSIM	Emulator	DES	Yes	Yes	SS for WSNs	• can support thousands of nodes simulation • can emulate radio models and code executions • only emulate homogeneous applications • have to use Power TOSSIM to simulate power consumption
EmStar	Emulator	TDS	Yes	Yes	SS for WSNs	• cannot support large number of sensors simulation • only run in real time simulation • only apply to iPAQ-class sensor nodes & MICA2 motes
OMNeT++	Simulator	DES	Yes	No	GS	• can support MAC protocols and some localized protocols in WSN • simulate power consumptions and channel controls • limited available protocols
J-Sim	Simulator	DES	Yes	Yes	GS	• can simulate large number of sensor nodes, around 500 • can simulate radio channels and power consumptions • its execution time is much longer
ATEMU	Emulator	DES	Yes	Yes	SS for WSNs	• can emulate different sensor nodes in homogeneous networks or heterogeneous networks • can emulate power consumptions or radio channels • the simulation time is much longer
Avrora	Simulator	DES	No	Yes	SS for WSNs	• can support thousands of nodes simulation • can save much more execution time
SENS	Simulator	DES	No	Yes	SS for WSNs	• Balanced consideration of modelling methodology and simulation efficiency • Memory-efficient, fast, extensible, and reusable Not accurate evaluation of WSN research • Lacks a comprehensive set of models Absence of a visualization tool
COOJA Java (Simulations In C)	Simulator	DES	No	Yes	GS	• Concerning both simulated hardware and software. • Larger-scale behaviour protocols and algorithms can be observed
Castalia	Simulator	DES	No	Yes	GS	• Physical process modelling, sensing device bias and noise, node clock drift, and several MAC and routing protocols implemented. • Highly tunable MAC protocol and a flexible parametric physical process model. • Not a sensor specific platform. • Not useful if one would like to test code compiled for a specific sensor node platform.