Analysis of Simulation Tools for Underwater Wireless Sensor Networks

V.P. Dhviya
(Assistant Professor)
Department of Information Technology
K.S. Rangasamy College Of Technology
Namakkal, India
dhivyavp@gmail.com

R. Arthi
(PG Scholar)
Department of Information Technology
K.S. Rangasamy College Of Technology
Namakkal, India
santhirajarthi@gmail.com

Abstract— Our earth is enclosed by 75% of water that could be a river or sea or ocean. The underwater sensor network becomes more and more conventional for monitoring huge areas of the oceans where manual intervention is not possible. Underwater sensor Networks consist of an inconsistent number of sensors that are deployed to carry out monitoring tasks over a given area. The UWSNs offer continuous observation for various applications similar to ocean sampling networks, pollution monitoring, disaster prevention, submarine detection, etc. Simulation tools for underwater wireless sensor networks are gradually being used more to study sensor webs and to study new applications and protocols in this developing research field. This paper gives a simple analysis of simulation tools used for underwater sensor networks with a view to help researchers.

Keywords- Underwater wireless sensor networks, Simulator, Emulator, Aqua-Sim, QualNet.

I. INTRODUCTION

Wireless Sensor Networks (WSN) have become a dominant technology in our days, while their applications are creating a huge impact on the way that many processes being interconnected and share valuable information. Due to the recent developments in the WSN’s communication capabilities and in the improvements of the network infrastructure, this technology could be applied to numerous applications. Underwater sensor Networks consist of an inconsistent number of sensors that are organized to perform monitoring tasks over a given area. The acoustic communication, although more reliable and robust, is bandwidth restricted. In the underwater sensor network the acoustic signals are used for communication among nodes because the radio signal works with additional low frequency and it cannot travel far away in underwater [21]. Simulation-based testing can facilitate to signify whether or not the time and monetary investments are valuable. Simulation is the most common approach for the increasing and testing new protocol for Underwater sensor Networks. A Number of advantages are considered in this approach, including lower cost, ease of implementation, and realism of testing large-scale networks. With the intention of efficient development of any protocol based on simulations, it is essential to know the different tools available before designing. Simulation is not as perfect as real environment and that there are a number of popular underwater sensor network simulators available. Thus the designs of various simulators created are accurate and most useful for different situations/applications. Numerous network details of UWSNs are not finalized and consistent. Constructing a UWSNs tested is very costly. Executing real experiments on a testbed is costly and has complexity. In addition, repeatability is basically compromised and it will affect the experimental results. It is inflexible to isolate a particular aspect. Furthermore, running real experiments is constantly time consuming. Therefore, UWSNs simulation is significant for UWSNs developers. These simulators permit users to separate different factors of tuning configurable parameters. There are three kinds of simulation: Monte Carlo Simulation, Discrete-Event Simulations and Trace-Driven Simulation [22]. Discrete-event simulation is widely used in WSNs as it can easily simulate many jobs running on different sensor nodes. This simulation includes initial routines, input routines, output routines, and trace routines. In addition, this simulation gives dynamic memory management. Trace-Driven Simulation provides different services and is commonly used in real system. The simulation results have more reliability.

II. RELATED WORK

Simulators and Emulator

The simulator is commonly used to develop and test protocols of UWSNs, particularly in the starting stage of these designs. The cost of simulating a number of nodes is very low, and the simulation can be completed within very short execution time. Equally general and specialized simulators are offered for users to simulate UWSNs. The tool, which is using hardware as well as firmware to perform the simulation, is called an emulator. Emulation can unite both software and hardware implementation. Typically emulator has greater scalability, which can emulate several sensor nodes at the same time [22].

III. SIMULATION TOOLS

A) NS-2

Network Simulator version-2, NS-2 was designed by Defense Advanced Research Projects Agency (DARPA) and National Science Foundation. The discrete event Network Simulator NS-2 can be assembled using an object-oriented extension of Tool Command Language and the language C++ [1, 23]. Nowadays, many users run the NS-2 Simulator on Linux Operating System or Cygwin. NS-2 is well developed and accepted tool which can be used in both wired and wireless networks. NS-2 is an open source tool. Two languages can be used to increase the level of learning a curve. Tool Command Language (TcL) is most probably used for writing simulation code and also gives an absolute learning curve. NS-2 gives some further features of modeling sensor networks of sensor channel models, power models, scenario generation [24, 2-3].

Merits:

1. NS-2 supports a significant range of protocols in various layers.
2. Low cost
3. Users can easily edit the on-line documents and develop their own codes.

Demerits:

1. Tool command language is tedious to understand and write
2. NS-2 is more complicated
3. Time consuming for developing a protocol
4. NS-2 does not support GUI.

B) EmStar

EmStar is an Emulator particularly planned for WSN assembled in C and it was developed by university of California, Los Angels. EmStar is a trace-driven emulator which can be run in real-time, from pure simulation to actual deployment. Many users can execute this emulator on Linux Operating System. This emulator supports to grow WSN application for improved hardware sensors. EmStar is a location for WSNs assembles from Linux-class devices, so it is called as micro servers [25, 4-5]. EmStar contains both of the simulation and emulation tools, which use a modulator, but with severely layered architecture. EmStar gives a range of services that are used and joint to provide network functionality for wireless embedded systems [25, 4-5].

Merits:

1. Robustness
2. Easy to evaluate faults and error
3. More flexible
4. The master used to decrease bugs

Demerits:

1. Limited scalability
2. Decrease reality of simulation
3. The master can be accessed with only real time simulation

C) GloMoSim

Global mobile Information System Simulator (GLoMoSim) is a scalable simulation for huge wireless and wired Networks. In GLoMoSim, the node aggregation technique was introduced which gives major benefits of the simulation performance. GLoMoSim is also planned to be extensible using all protocols can be implemented as modules in the GLoMoSim library [26]. GLoMoSim can run using a numerous of synchronizing protocols, and was effectively implemented on equally shared memory and distributed memory computers. GLoMoSim provide fundamental functionality to stimulate wireless networks.

Merits:

1. Large scalability
2. It supports adhoc networking protocols
3. Good mobility models

Demerits:

1. GloMoSim supports only wireless network
2. Effectively limited to, IP networks
3. No specific routing protocols
4. Difficult to stimulate large sensor networks.

D) Shawn

Shawn is an open source and also a customizable sensor network simulator. Shawn will plan to maintain the large-scale network simulation. It is claimed to offer the maximum abstract level and maintain better network comparing to further simulators such as NS-2, SENSE, OMNet++ and GloMoSim [6]. The simulated nodes exist on a single world instance. Shawn (simulation/programming language) can be written in Java. It consists of features perseverance and decoupling of the simulation surrounding. They decouple state variables that allow for an easy implementation of persistence [7]. They join persistent and volatile data together to individual nodes and the world. Shawn has an ability to use various model interfaces that are used to control the simulation without knowing accurate implementation that implies an actual behavior of simulation [27, 8-9]. Shawn is mainly developed to maintain the primary gradual evolution of an idea of a centralized algorithm to a decentralized algorithm. In Shawn, two nodes can be easily communicated with exchanging messages that can be identified with communication models [10].

Merits:

1. Dense Protocol can be replaced by modifying
2. Easy to determine the effect of channel Parameters.

Demerits:

1. Simulation issues or lower layer issues are not considered.
2. Limited to generate a postscript file.

E) UWSim

UWSim is primarily designed for Underwater Sensor Network. Presently available simulators greater focus on ground-based sensors and adhoc networks, but they do not consider any factors that affect the underwater communication [11]. UWSim is mainly focusing on some specific handling scenarios like low-bandwidth, low frequency, high transmission and limited memory. It is based on component-based approaches slightly than a layer/protocol-based approach.

The software was built on Windows XP using Microsoft. Net framework 2.0 is developed to maintain all versions of windows include 64-bit machines. The real software development was carried out in an only object-oriented fashion using C# capabilities [12]. Nearly all Simulators assume the Uniqueness of ground-based wireless adhoc and sensor networks. Most simulators use radio frequency transmission, but UWSim needs some simulator, which used to simulate the acoustic network by UWSim [13]. Basically UWSim is based on a novel routing protocol which is proposed by Developers, distinct traditional simulators which are based on moreover proactive or reactive routing protocols (AODV and DSR). The different characteristics of Underwater Networks, for example, low bandwidth requires high frequency.

Demerits:

1. Restrict the number of functionalities.
2. It cannot be used for any other sensor network of UWSN.

F) VisualSense

Designing of Wireless networks requires complicated representation and examine communication channel. Sensors, ad-hoc networking protocols consuming energy in sensor nodes. VisualSense is planned to maintain a component-based structure of such models. VisualSense offers an exact and extensible radio model. The radio model comes under a general energy propagation model that can be re-processing of physical phenomena. VisualSense offer a sound model based on this propagation model which is exact to use for localization. The development to Ptolemy consists of an only some new Java class and a few XML files [28]. The Classes are considered to be sub-Classed by model constructors for customization and non-trivial models can also be designed without writing Java code. It maintains actor-oriented description of network nodes, wireless communication channel. Customized Channels can be defined by sub classing the wireless Channel base class and by joining functionality define in Ptolemy II models. It is intended to permit the research community to split a model of disjoint aspects of the sensor nets problem and to construct models that include complicated elements from numerous aspects.

In VisualSense, heterogeneous systems are maintained. To stimulate something different than a pre-built model presented an idea of sound detection in sensor fields. VisualSense produce a model by mechanically creating a set of actors. VisualSense can be used to know the standards of WSNs [20].

Demerits:

1. It will provide a protocol only to the sound.

G) JSim

After designing of NS-2, J-Sim is a general purpose simulator model. J-Sim based on the concept of autonomous component architecture (ACA) [21-23]. Each node could be represented by an object by replacing the notion. J-Sim consists of three level components which are targeted nodes (which is used to produce stimuli), the sensor node (which reacts to the stimuli), sink node (the final destination for stimulus reporting). J-Sim gives GUI library, which service users to model or compile the Mathematical Modeling Language "text-based language" printed for J-Sim. Each component is broken down into various parts and modeled differently inside the simulator. The failure of each component makes it simple to use with various protocols in different simulation runs-Sim is more alike to NS, than can be written in two languages in J-Sim case-Java and Jacl, a Java version of Tcl. J-Sim has a number of improvements on NS-2 and other simulators. An Object oriented model is not a superior model for component based architecture used by NS-2. J-Sim also offers a script interface that allows integration with various script languages such as Perl, Tcl or Python.

Merits:

1. High quality of reusability and interchangeability
2. Independent platform
3. Lots of memory space.

Demerits:

1. Difficult to use
2. Execution time is high
3. Makes user hardly to use

H) OMNeT++

OMNeT++ [30, 17-19] is an Object-oriented distinct network simulation framework. OMNeT++ is not a simulator, but it slightly gives a framework and tools to write simulations. It is highly convenient so that it can be executed on the most common Operating System. OMNeT++ offers both noncommercial license and a commercial license. This simulator maintains a module programming model. OMNeT++ is an accepted non-specific network simulator, which can be used in equally wire and wireless environmental [20,31-32]. In OMNeT++, framework and simulation models are open source. OMNeT++ is free with full source code and is complimentary to use, alter and allocate in academic and educational institutions under its own license [36].

Merits:

1. Provide a powerful GUI.
2. Much easier simulator
3. Simulate power consumption problems in WSN
4. A Simulator can support MAC protocol as well as some localized protocol in WSN.

Demerits:

1. Available protocols are not large enough
2. Compatible problem will arise
3. High probability reports bugs

I) Aqua-Sim

Aqua-Sim can efficiently simulate the acoustic signal attenuation and packet collisions within underwater sensor networks. Aqua-Sim maintains a three-dimensional deployment environment additional in the two-dimensional area provided by other simulators. Aqua-Sim can easily be included with the accessible codes in NS-2. Aqua-Sim is equivalent with the CMU wireless simulation packages. Aqua-Sim which is not affected by any alters in the wireless package and it is independent of the wireless package [33]. Further, any modification to Aqua-Sim also limited to itself and does not have any collision on other packages in NS-2. Aqua-Sim can be designed with extensible and flexible options. It consists of three basic classes like Entities, Interfaces and Functions [35].

Network entity classes depict concrete network entities. Pure Interface classes are simply effective and represent common interfaces and provide as the base classes for others. Common Function classes serve some common functions of other classes and can be integrated into any classes in Aqua-Sim.

Merits:

1. Discrete-event driven network simulator
2. Support 3D networks and mobile networks
3. Simulate underwater acoustic channels with high fidelity
4. Easily import new protocols

Demerits:

1. In underwater, acoustic signals are very slow.

J) QualNet

QualNet is a planning, testing and training tool that “mimics” the activities of an actual communication network. Simulation is a cost-valuable method of developing, deploying and managing network-centric systems during their complete life cycle. Users can estimate the basic activities of a network. QualNet gives a complete environment for design Protocol, creating and animating network scenarios [34]. QualNet consists of graphical tools that show more numbers of metrics gathered during simulation of network scenario. QualNet can maintain real-time speed to allow software-in-the-loop. QualNet can execute on a cluster, multi-core and multi- processor systems. Users can increase a protocol model in QualNet [34]. QualNet can join together hardware and software applications, like OTB, real networks and third party visualization software.

Merits:

1. High speed
2. It can model, thousands of nodes
3. Scalability
4. Extensibility.

IV. COMPARISON TABLE

In this table 1, we have discussed about some of the simulators with its program language/Platform and analyze that the simulator is an open source or commercial and also its limitations. The NS-2 simulator is mostly used for networking. NS-2, EmStar, GloMoSim, Shawn, J-Sim, Aqua-Sim is the open source simulators. From this survey Aqua-Sim and QualNet simulators are most probably used in Underwater Sensor Networks.

TABLE 1

S. No.	Simulator	Programming Language/Platform	Open Source and Online Documents	Limitations
1	NS-2	C++	Yes	Complicated, Time consuming GUI didn't support.
2	EmStar	Linux	Yes	Limited scalability, Decrease reality of simulation, Accessed by only a real time simulation
3	GloMoSim	Parsec	Yes	Currently supports only wireless network, Effectively limited to IP networks, No specific routing protocols
4	Shawn	Java	Yes	Low layer issues are not considered, Limited to generate postscript files.
5	UWSim	C++		Restricted number of functionalities, Not usable for any sensor networks except UWSN.
6	VisualSense	Ptolemy II		Provides a protocol only to the sound signal.
7	J-Sim	Java	Yes	Difficult to use, Execution time is high.
8	OMNeT++	C++	Noncommercial License, commercial License	Compatible problem, High probability of bug report.
9	Aqua-Sim	C++	Yes	An acoustic signal is very slow
10	QualNet	C++	Commercial license	

V. CONCLUSION

The goal of this paper is to offer a simple analysis of generic tools that are used in underwater sensor networks. Due to the major difference between underwater sensor networks and terrestrial radio networks, many simulators are introduced. Among them Aqua-Sim and QualNet are the tools that are frequently used in underwater sensor networks. QualNet is a user friendly tool and also gives more efficiency, less time consuming when compared to NS-2 tool. Aqua-Sim is a great simulation tool, with high reliability and flexibility, for underwater networking research. According to diverse targets to decide different simulation tools of WSNs will be more capable and effective.

REFERENCES

[1] Downard I., DC, N., “Simulating sensor networks in ns-2,”2004.

[2] Park, S., Savvides, A., Srivastava, M., “SensorSim: a simulation framework for sensor networks,” in Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems. ACM New York, NY, USA, pp. 104–111, 2000.

[3] “Sensor Sim: A Simulation Framework for Networks Sensors”, Electrical Engineering Department, University of California, Los Angeles, Retrieved October, vol. 16, 2006.

[4] Elson, J., Bien, S., “Emstar: An environment for developing wireless embedded systems software,” Center for Embedded Networked Sensing (CENS) Technical Report, vol. 9, 2003.

[5] Girod, L., Ramanathan, N.," A Software Environment for Developing and Deploying Heterogeneous Sensor Actuator Networks,” Center for Embedded Network Sensing, p. 101, 2007.

[6] L. Girod, et. al., “EmStar: An environment for developing wireless embedded systems Software,” Technical report, Center for Embedded Networked Sensing, University of California, Los Angeles, CENS Technical Report 009, 2003.

[7] L. Girod, et. al., “Emstar: A software environment for developing and deploying heterogeneous sensor-actuator networks,” ACM Transactions on Sensor Networks, vol. 3, no. 3, article 13, Aug. 2007.

[8] Kroeller, A., Pfisterer, D., “Shawn: A new approach to simulating wireless sensor networks,” Arxiv preprint cs/0502003, 2005.

[9] Fekete, S., Kroller, A., Fischer, S., Pfisterer, D., “Shawn: The fast, highly customizable sensor network simulator,” in Networked Sensing Systems, 2007. INSS’07. Fourth International Conference on, pp.299–299, 2007.

[10] A. Kr¨oller, D. Pfisterer, C. Buschmann, S. P. Fekete, and S. Fischer, “Shawn: A new approach to simulating wireless sensor networks,” in Proceedings of DASD'05, Design, Analysis, and Simulation of Distributed Systems, (San Diego, USA), 5 April 2005.

[11] S. Dhurandher, S. Misra, M. Obaidat, and S. Khairwal, “UWSim: A simulator for underwater sensor networks,” Simulation, vol. 84, no. 7, pp. 327-338, 2008.

[12] J. Cui., J. Kong, M. Gerla and S. Zhou, “Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications,” UCONN CSE Technical Report: UbiNET-TR05-02, University of Connecticut, USA, 2005.

[13] R. Jurdak, C. V. Lopes and P. Baldi, “Battery lifetime estimation and optimization for underwater sensor networks,” Sensor Network Operations, Wiley IEEE, pp. 397–420., May 2006.

[14] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, and Y. Zhao, “VisualSense: Visual Modeling for Wireless and Sensor Network Systems,” in Technical Memorandum UCB/ERL M05/25, University of California, Berkeley, USA, 15 July 2005.

[15] Sobeih, A., Chen, W., Hou, J., Kung, L., “J-sim: A simulation environment for wireless sensor networks,” in Proceedings of the 38th annual Symposium on Simulation. IEEE Computer Society Washington, DC, USA, pp. 175–187, 2005.

[16] Hou, J., Kung, L., “J-Sim: A Simulation and emulation environment for wireless sensor networks,” IEEE Wireless Communications Magazine, vol. 13, no. 4, pp. 104–119, 2006.

[17] Varga, A., et al., “The OMNeT++ discrete event simulation system,” in Proceedings of the European Simulation Multiconference (ESM’2001), pp. 319–324, 2001.

[18] Varga, A., “OMNeT++ Discrete event simulation system. User Manual,” Technical University of Budapest, Dept. of Telecommunications, 2006.

[19] Varga, A., Hornig, R., “An overview of the OMNeT++ simulation environment,” in Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops table of contents. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) ICST, Brussels, Belgium, Belgium, 2008.

[20] [Egea05]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, pp.2-9, 2005. URL: http://ait.upct.es/~eegea/pub/spects05.pdf

[21] http://www.ijareeie.com/upload/2014/february/13Q_Survey.pdf

[22] http://www.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html

[23] http://www.isi.edu/nsnam/ns/, 2012/.

[24] http://nesl.ee.ucla.edu/projects/sensorsim/, 2012.

[25] http://www.lecs.cs.ucla.edu/emstar/, 2012.

[26] http://pcl.cs.ucla.edu/projects/glomosim/

[27] http: //shawn.sourceforge.net/, 2012.

[28] http://www.mathworks.com/products/matlab/

[29] http://sites.google.com/site/jsimofficial, 2012.

[30] http://www.omnetpp.org/

[31] http://en.wikipedia.org/wiki/Omnet%2B%2B

[32] http://www.omnetpp.org/home/what-is-omnet

[33] http://obinet.engr.uconn.edu/wiki/index.php/Aqua-Sim

[34] http://web.scalable-networks.com/content/qualnet

[35] http://obinet.engr.uconn.edu/wiki/images/Aqua-Sim_An_NS-2_Based_Simulator_for.pdf

[36] http://en.wikipedia.org/wiki/OMNeT++