Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Review on Wireless Sensor Network Simulator

Kartik N. Shah*, Himanshu Ghetia, Shantanu Santoki, Jaishankar N.
School of Computing Science,
VIT University, Vellore , India

Abstract— Wireless Sensor Network (WSN) is one of the most favorable areas for research work. WSNs are composed of large number of sensor nodes. So, it is very complex to model WSN for analysis purpose. To implement and evaluation of algorithms and protocols in WSN, simulators and emulators are used. There is always a concern that simulation result may not match with the original behavior of the system. So, we must be aware about the strengths and weakness of the simulators. There are lot many simulators and emulators available for WSNs. These simulators can be changed based on requirements. In this paper, we will discuss some of the famous simulators and emulators. Also we will compare those simulators based upon their merits and demerits. In this paper, we will consider simulators like NS2, OMNET++, TOSSIM and EmStar.

Keywords— Wireless Sensor Network, Simulation Tools, Review, Survey, WSN.

I. INTRODUCTION

Network simulation is an evolutionary area of computer networks. They are widely used for development of protocols and network architectures. An arbitrary model that specifies behavior of the nodes and communication channels are called as Network Simulators. As an example, if we want to investigate the characteristics of a new routing protocol then it is usually implemented in network simulator. Most of the network simulation toolkits are based on discrete event-based simulation (DES) paradigm [1]. To analyze the performance of wired and wireless networks, we can have methods called analytical methods, physical measurements and computer simulation. On the other hand, for wireless sensor networks cannot be analyzed using above methods easily as it can have energy problems, limited battery life etc. So, discrete- event simulation can be used for WSNs. It can simulate many jobs running on all the nodes [2]. Trace-driven simulation is commonly used in real system and the results are more accurate [2].

At the beginning stage of a protocol, simulators are used to develop and test. It can finish execution in very short time. It can simulate thousands of nodes in very less cost and less amount of time [3]. In the case of emulator, it uses firmware as well as hardware for simulation [3]. Emulator uses real nodes, so it can provide better performance compared to simulators. Many architectures can be combined and made a 'tiered architecture' which can have different capabilities, costs and battery backup [11][12].Wireless sensor nodes are very small in size and can be used in many fields as they are cheap in cost. It is having less complex functionality. So, if there are thousands of sensor nodes in the field then we cannot observe the energy consumption, network topology or new algorithm. So, we need to go for simulation environment. Fig. 1 shows the simple wireless sensor node.

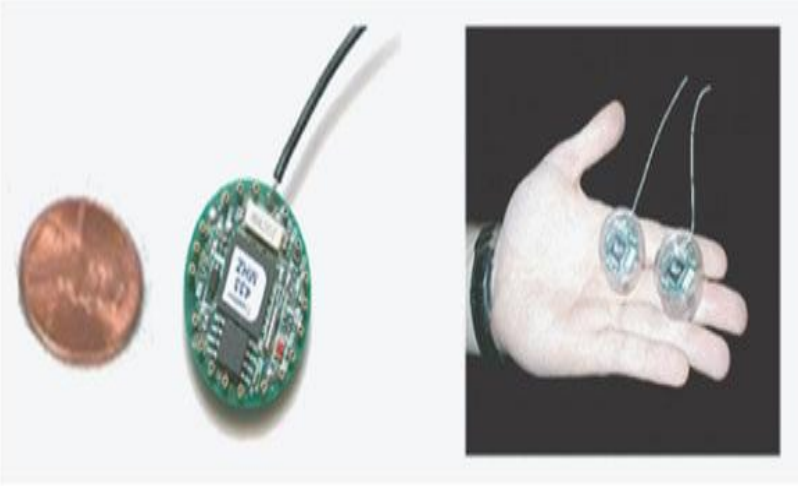

Fig 1: Simple Wireless sensor node

Figure 2 shows the Model for wireless sensor network. We can see the functionality that Agent will pass the information to nodes via environment and nodes will pass the information to sink nodes using radio channel.

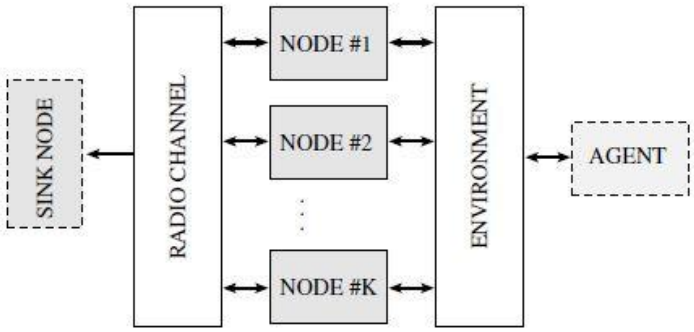

Fig 2: Wireless sensor network model [4]

In the following subsection we will take the overview of different simulation tools.

II. SIMULATION TOOLS

A. NS2 (Network Simulator)

Network Simulator (NS) is also a discrete event simulator which was developed at UC Berkeley. It was developed for simulation of IP networks at packet level [6]. It is generally used to investigate the protocols such as UDP, TCP, ECN, RED, CBQ etc [7]. Following figure shows the simplified view of network simulator.

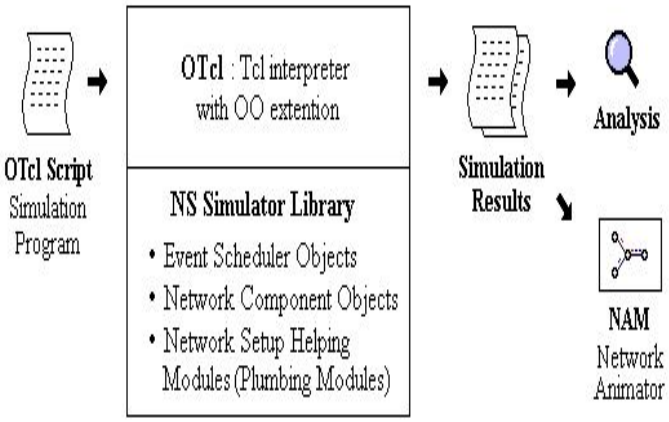

Fig 3: Simplified view of NS [6]

NS contains network simulation object libraries in Object- oriented Tcl (OTcl) script interpreter. NS2 architecture is having following five parts.

1. Event Scheduler
2. Network Components
3. Tclcl
4. OTcl library
5. Tcl 8.0 script language.

Figure 4 shows the architecture of NS2.

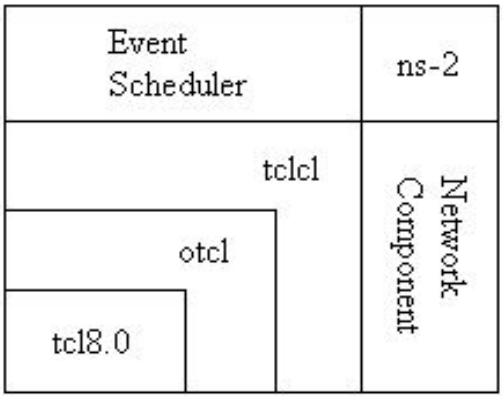

Fig 4: NS architecture [6]

Visualization tool for NS is Network AniMator(NAM). Following Figure 5 shows the screenshot of NAM.

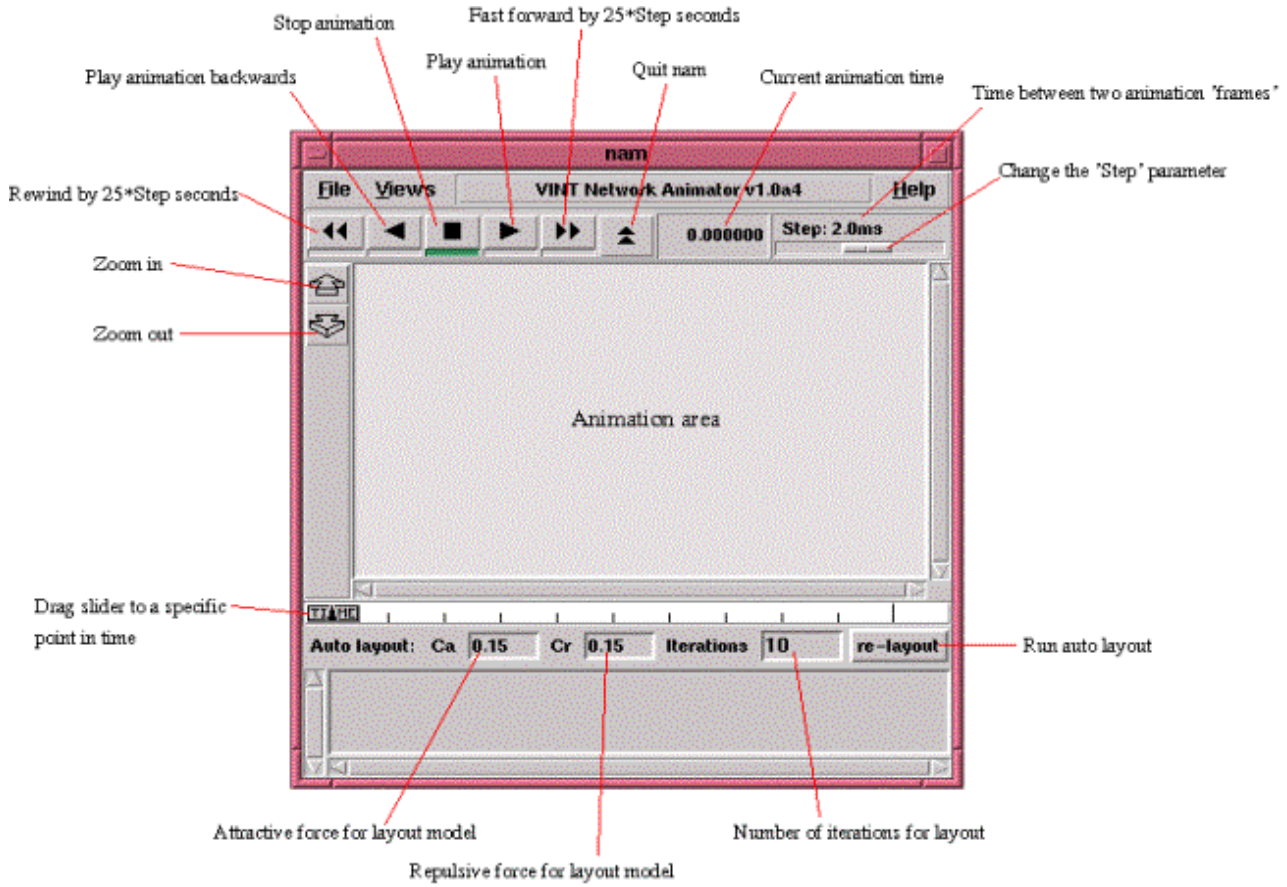

Fig 5: Network animator [8]

B. TOSSIM (For TinyOS Networks)

Different simulators have different focuses like NS simulators are simulation for packet level. TOSSIM was developed for Tiny OS sensor networks. It is discrete event simulator developed at UC Berkeley. It deals with the interaction and behaviour of the networks rather than the packet level. A small sensing and communication device called as Motes. They are less expensive and limited energy devices. Tiny OS is an OS used in sensors and the sensors that use this OS are called as Motes. Following Figure 6 shows the TOSSIM architecture.

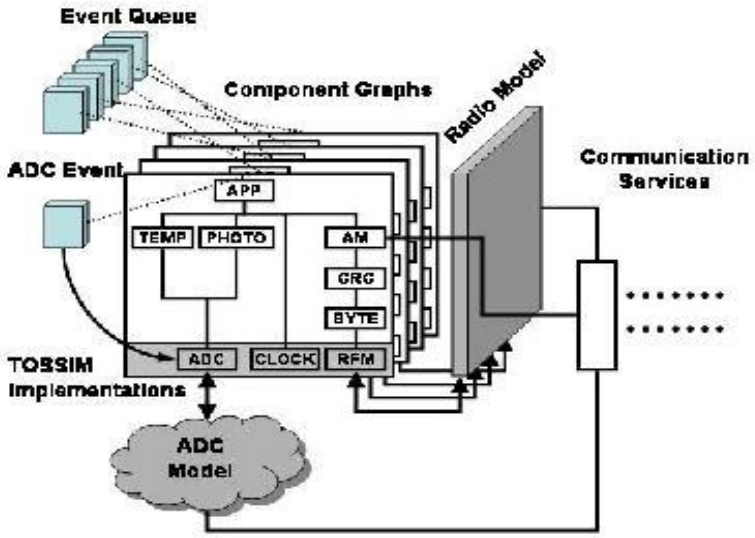

Fig 6: TOSSIM Architecture [5]

TOSSIM architecture is made up of following parts [5]

1. Frames (TinyOS component graphs)
2. Events (Execution model)

3. Models (Radio and ADC)
4. Hardware abstraction components
5. Communication Services

It comes with Java visualization and actuation environment called TinyViz. [5].The following figure 7 shows the TinyViz for TOSSIM.

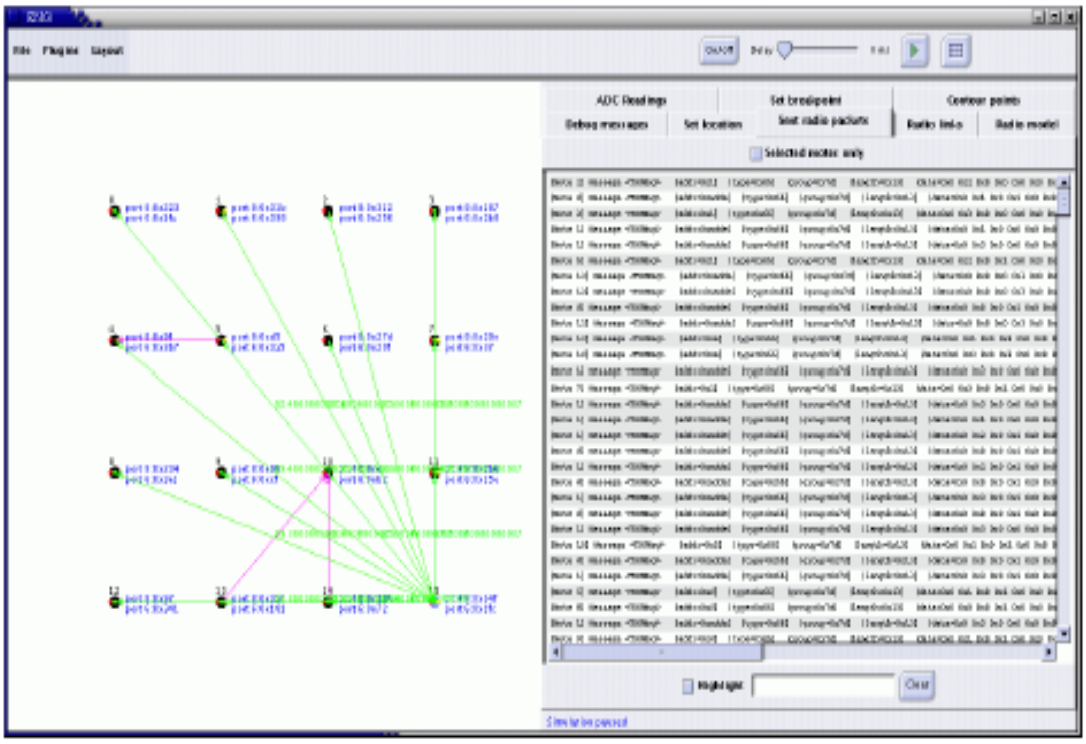

Fig 7: TinyViz connected to TOSSIM[5]

C. OMNET++

OMNET++ is a C++ based discrete event simulator for simulation of communication networks, distributed and parallel networks. It is widely used in simulation for complex IT systems and hardware architecture as well. It is free for academics (non-commercial) use. The modules of OMNET++ modules can communicate using message passing model. Active modules are also called as simple modules. These simple modules can be combined to a single module and called as composite module [9]. Figure 8 shows the logical architecture of OMNET++.

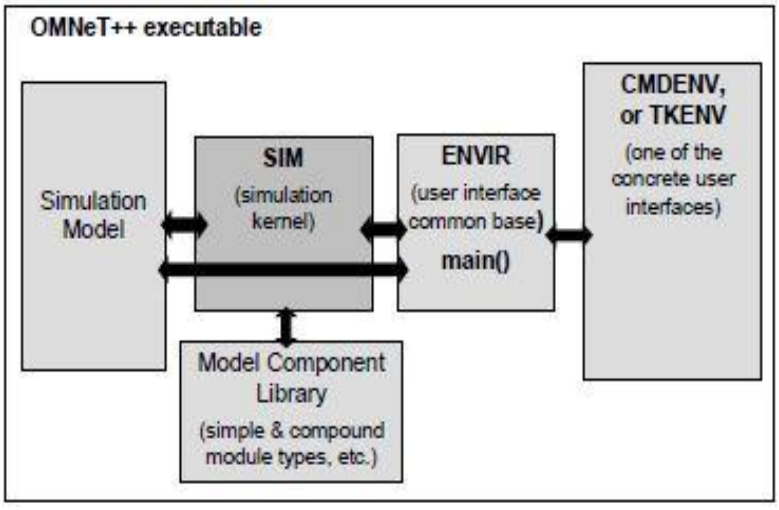

Fig 8: OMNET++ logical architecture [9]

The components of OMNET++ are as follows: [10]

1. Simulation kernel library
2. NED topology description lang
3. IDE (based upon Eclipse)
4. Tkenv
5. Cmdenv
6. Utilities
7. Documentation

Following figure shows the screenshot of OMNET++.

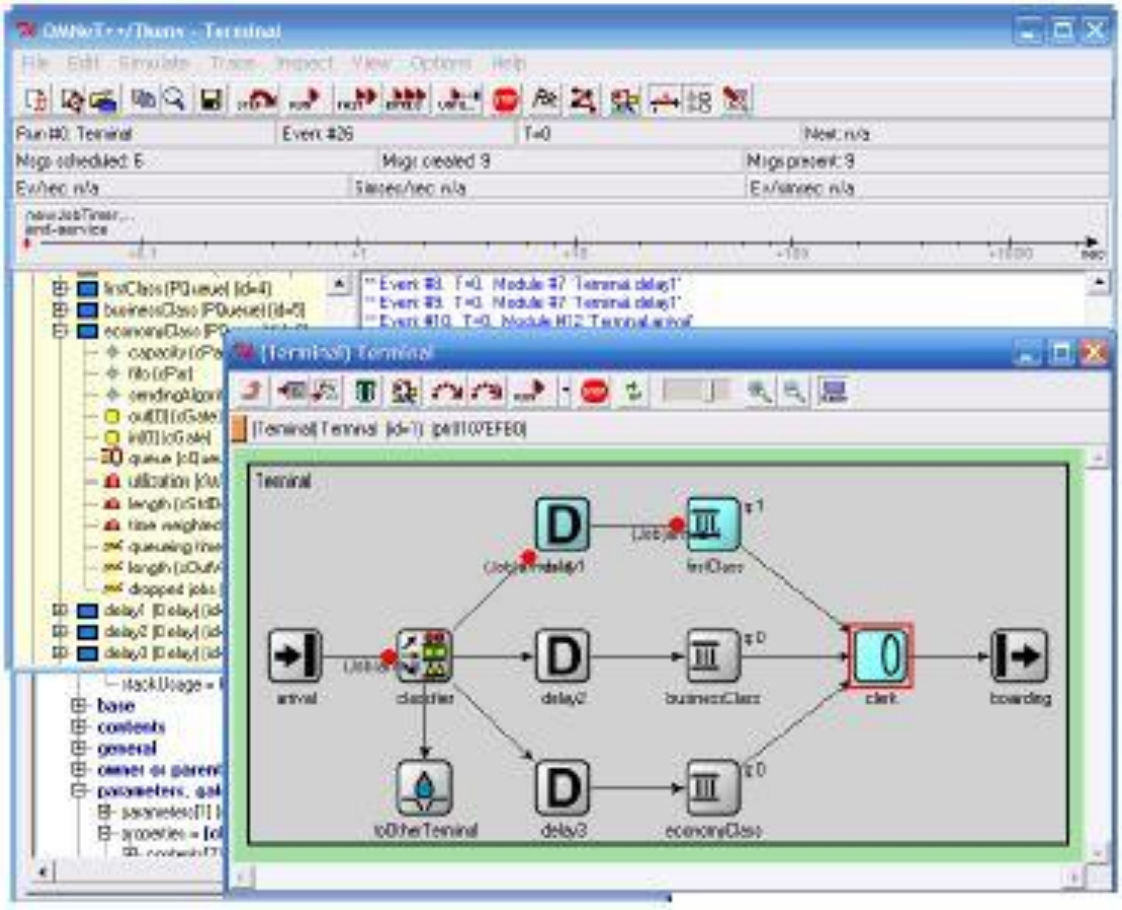

Fig 9: Screenshot of Tkenv of OMNET++ [9]

D. EmStar

EmStar is an emulator. It was designed for WSN built in C [13-14]. It was developed by University of California, Los Angeles. It is trace driven emulator for real time [14]. As it uses C language, we can do modular programming in it. So, we can have more flexibility of making different modules for different functionalities. We can also reuse the modules whenever needed. EmStar is GUI based emulator. So, it is easy to use. But the drawback is limited scalability. We can have limited number of nodes in this emulator. The layered framework of EmStar can be shown in figure 10.

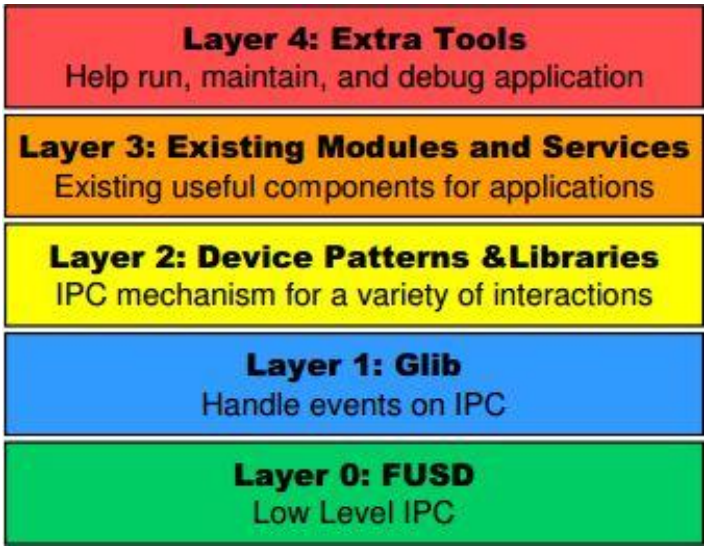

Fig10 Layered Framework for EmStar[15]

Emstar environment is consists of libraries, tools, services and microkernel extension. The most widely used tools are:

EmSim : Simulator for microserver

EmCee: Interface for real low-power radios

EmTOS : Extension of Emstar for TinyOS applications. The visualization of EmStar (EmView) can be shown as in following figure 11.

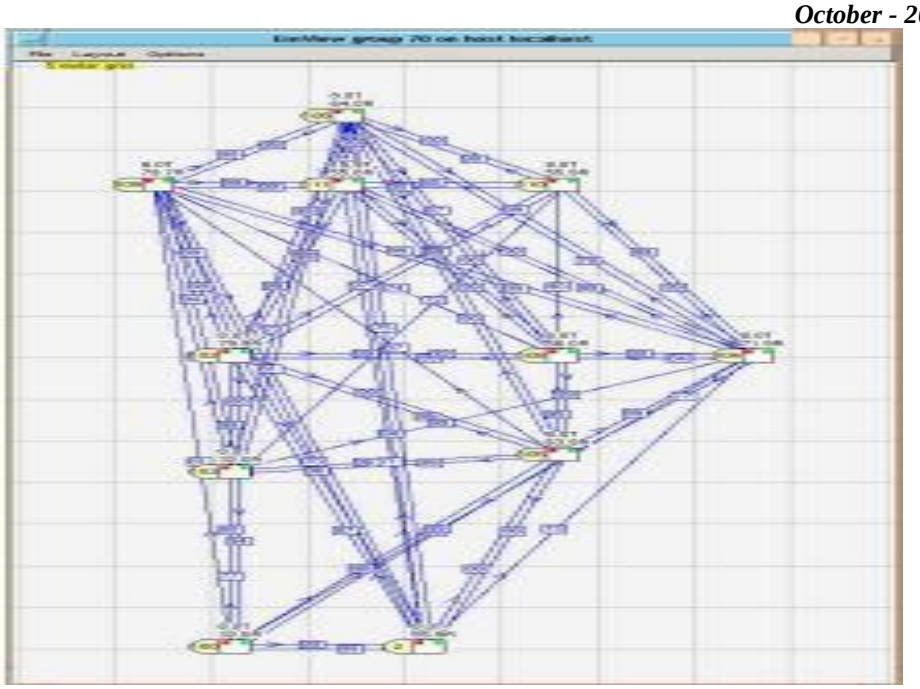

Fig 11: Emstar Visualizer (EmView)[15]

III. CONCLUSION

We can conclude from the above discussion that every simulator is made with a special purpose. Each simulator is having its merits as well as its demerits. So, based upon requirements we can select a proper simulator. Network Simulators (NS) are made for packet level analysis of data while TOSSIM was made for the study of behavior and interaction of networks. We can summarize as follows.

	NS2	TOSSIM	OMNET++	EmStar
Simulator/Emulator	Simulator	Emulator	Simulator	Emulator
Discrete-Event (DE)/ Trace driven(TD)	DE	DE	TD	DE
Open Source	No	Yes	Yes	Yes
GUI	No	Yes	Yes	Yes
General Purpose	Packet Level	TinyOS sensor	Communication networks	Real-time

ACKNOWLEDGMENT

The authors would like to thank the School of Computer Science and Engineering, VIT University, for giving them the opportunity to carry out this project and also for providing them with the requisite resources and infrastructure for carrying out the survey.

REFERENCES

[1] G. S. Fishman. Principles of Discrete Event Simulation. John Wiley & Sons, Inc., New York, NY, USA, 1978.
[2] 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.
[3] 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.
[4] E. Egea-López, J.Vales-Alonso, A.S. Martínez- Sala, P. Pavón-Mariño, J. García-Har ,"Simulation Tools for Wireless Sensor Networks", Summer Simulation Multiconference - SPECTS 2005, pp. 2-9.
[5] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003).
[6] J. Chung, M. Claypool. NS by Example. Worcester Polytechnic Institute (WPI) http://nile.wpu.edu/NS/purpose.
[7] L. Breslau, D. Estrin, K. Fall. Advances in Network Simulation. 2000.
[8] Marc Greis's tutorial (now maintained by VINT group)http://www.isi.edu/nsnam/ns/tutorial/index.html.
[9] András Varga and Rudolf Hornig , "AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT", SIMUTools, March 03 – 07, 2008, Marseille, France.
[10] http://www.omnetpp.org/home/what-is-omnet.
[11] A. Cerpa et. al. Habitat Monitoring: Application Driver for Wireless Communications Technology. In 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, April 2001.

[12] H. Wang, D. Estrin, and L. Girod. Preprocessing in a Tiered Sensor Network for Habitat Monitoring. In in EURASIP JASP Special Issue on Sensor Networks, number 4, pages 392-401, 2003.

[13] 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.

[14] 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.

[15] Lewis Girod , Jeremy Elson , Nithya Ramanathan, "Emstar: a Software Environment for Developing and Deploying Heterogeneous Sensor-Actuator Networks" , ACM Journal, Vol. V, No. N, August 2006, Pages 1-34.

