IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 5, Ver. I (Sep.- Oct. 2017), PP 13-19
www.iosrjournals.org

Performance of Wireless Sensor Network Simulators using Ad-hoc On Demand Distance Vector (AODV) Protocol

*M.Khaleel Ullah Khan[1], Dr.K.S.Ramesh[2],
[1]Research Scholar, Department of ECE, K L University
[2]Professor, Department of ECE, K L University
Corresponding Author: M.Khaleel Ullah Khan

***Abstract**: Simulation in wireless networks is very important before the real time implementation of any project. For this purpose Design Engineers and Research community first simulate the networks design and then go for the implementation of any project. These will help to build new theories and hypothesis. Right from the invention of Wireless networks a number of simulators are available and up gradation, enhancement in characteristics is being done for the Network simulators. Few examples of these Simulators available are NS-2, NS-3, SWAN, JIST, OPNET, GloMoSim etc. The crucial decision for selecting the simulator depends upon its evaluating characteristics such as speed of evaluation, Memory usage for running the simulator, utilization of the CPU cycles and Scalability by simulating a routing protocol. Therefore it becomes important to choose the simulator accordingly depending upon the project.*

***Keywords**: Wireless Networks, Simulators, characteristics of simulators, AODV Routing.*

Date of Submission: 31-08-2017 Date of acceptance: 13-09-2017

I. Introduction

As Electronics and Communication industry is growing rapidly in leaps and bounds so as wireless technology is showing its impact on the newly emerging communication technology. Generally, when a design is proposed it cannot be implemented directly in the real world in large as it may give undesired results and later for rectification the whole project has to be modified. To avoid such situation new protocols or techniques/schemes are tested with the help of analytical modelling or simulators in computers. As simulation can be done on a prototype a modification can be done easily. After simulation if the result is yielded as desired then these protocols can be implemented for final design of the Project.

Analysis using Analytical method are not precise and also time consuming. It also consumes energy, CPU memory and power. Real world implementation is realistic. Therefore analysis of Physical design using the analytical method is not only a time consuming but also costly as it requires hardware and the manpower as well other resources. Hence, simulation is not only affordable which provides desired results and also cost effective. Therefore performance of any design or protocol is evaluated first with the help of simulators.

Earlier, Mathematical modelling as well as sample experimentation was common for the evaluation of communication networks. Now a day’s wireless networks have changed rapidly due to the increase in usage of the users and have become too complicated for analytical modelling. Computer system simulation is playing a key role in the research field and to help the researchers and network designers to understand the behaviour and performance of the networks and its protocols. Computer simulation is often used to evaluate performance of the planned capacity of wireless networks and satisfying the customer requirements. Simulation is also used to explore a wide range of potential protocol designs through rapid evaluation and iteration [1]. However, different simulators require variable time, memory and computation power for evaluating proposed protocols/techniques. This paper presents performance comparison of the network simulators like NS-2, NS-3, OMNET++, and GloMoSiM, as these are open source and vastly used network simulators in today ‘s simulation activities. This paper assists the design engineers in choosing the Network simulators for the simulation. It helps in research for effective utilization of memory and CPU speedy operation.

II. Implementation

The different routing protocols like AODV, AODV, DSR, DSDV, TORA, ZRP [2]are used to evaluate the performance of Network parameters[3] using simulators. In this paper AODV protocol selected for performance evaluation. NS-2 with OMNeT++ and QualNet by using radio propagation models are compared[4], [5].Architecture of NS-2 with TOSSIM are compared in [6],[7] &[8].

The performance of sensor network for designing in real time is compared using simulators JavaSim, SSFNET & NS-2 is compared in [9].The features and characteristics of Simulators like OPNET and NS-2 is done in [10].The MANET routing protocol, i.e. AODV, to evaluate the performance of the network simulator is used in this paper for evaluation of performance of different simulators.

III. Simulation Tools

The wireless sensor network simulators which are widely used in research are NS-3, NS-2, GloMoSim and OMNET++.

3.1: Network Simulator-2 (NS-2): NS-2 is an open-source simulation tool that runs on Linux. It is a discreet event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks[11,13]. It has many advantages that make it a useful tool, such as support for multiple protocols and the capability of graphically detailing network traffic. Additionally, NS2 supports several algorithms in routing and queuing. LAN routing and broadcasts are part of routing algorithms. Queuing algorithms include fair queuing, deficit round-robin and FIFO.

NS2 started as a variant of the REAL network simulator. REAL is a network simulator originally intended for studying the dynamic behavior of flow and congestion control schemes in packet-switched data networks. Currently NS2 development by VINT group is supported through Defense Advanced Research Projects Agency (DARPA) with SAMAN and through NSF with CONSER, both in collaboration with other researchers including ACIRI (see Resources). NS2 is available on several platforms such as FreeBSD, Linux, SunOS and Solaris. NS2 also builds and runs under Windows. NAM interface contains control features that allow users to forward, pause, stop and play the simulation. The interface of ns-2 is shown in Figure 1.

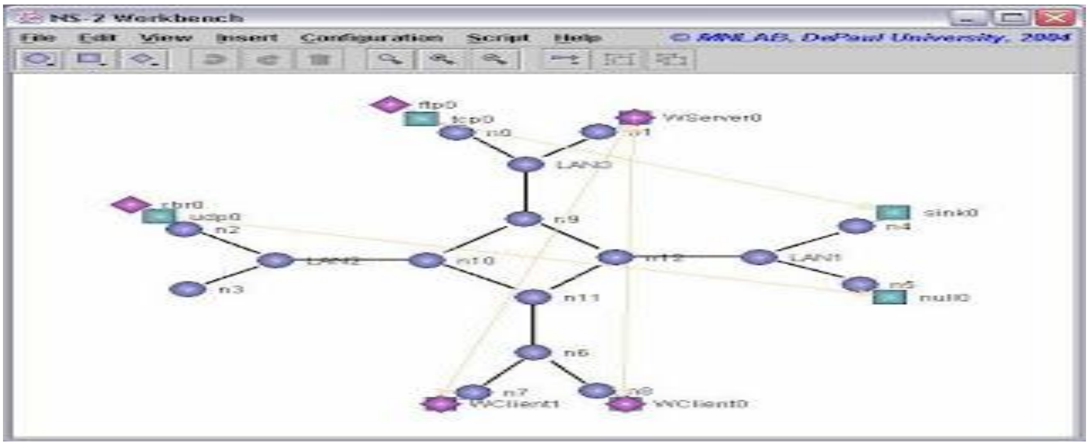

Figure 1: NS-2 Simulator Interface

3.2-Global Mobile Information System Simulator (GloMoSim): It is a scalable simulation environment for large wireless and wire line communication networks. [19] GloMoSim uses a parallel discrete-event simulation capability provided by Parsec. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols [20]. The following table lists the GloMoSim models currently available at each of the major layers: Layer Models Physical (Radio Propagation) Free space, Two-Ray Data Link (MAC) CSMA, MACA, TSMA, 802.11 Network (Routing) Bellman-Ford, FSR, OSPF, DSR, WRP, LAR, AODV Transport TCP, UDP Application Telnet, FT.The GloMoSim interface is shown in figure:2.

The node aggregation technique is introduced into GloMoSim to give benefits to the simulation performance. Initializing each node as a separate entity inherently limits the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system. Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation. In GloMoSim, each entity 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.

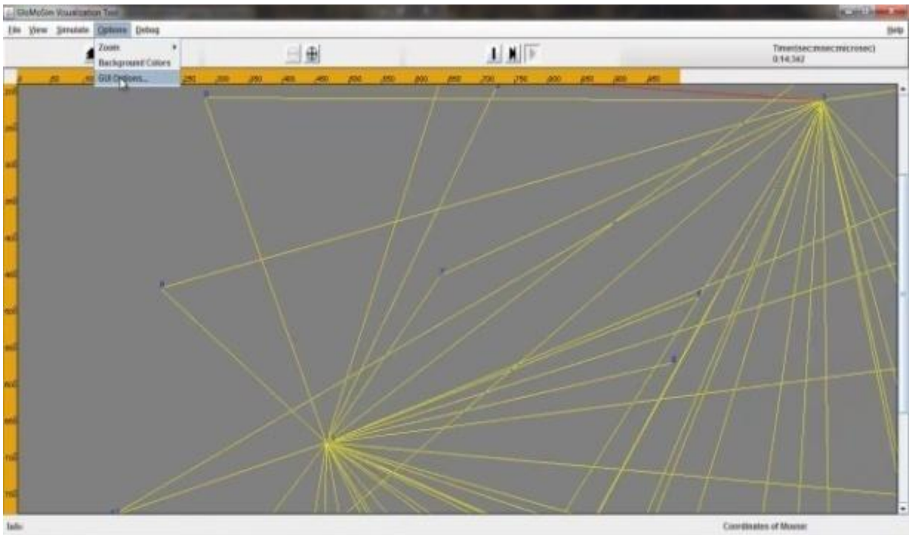

Figure -2 : GloMoSim Interface

3.3-The OMNeT++ Integrated Development Environment: It is based on the Eclipse platform, and extends it with new editors, views, wizards, and additional functionality [17]. OMNeT++ adds functionality for creating and configuring models (NED and ini files), performing batch executions, and analyzing simulation results, while Eclipse provides C++ editing, SVN/GIT integration, and other optional features (UML modeling, bugtracker integration, database access, etc.) via various open-source and commercial plug-ins[18].
The NED Editor shown in figure (3) which can edit NED files both graphically or in text mode, and the user can switch between the two modes at any time, using the tabs at the bottom of the editor window.

Figure-3:NED editor

In graphical mode, one can create compound modules, channels, and other component types. Submodules can be created using the palette of available module types [25]. Visual and non-visual properties can be modified in the Properties View, or by dialogs invoked from the context menu. The editor offers many features such as unlimited undo/redo, object cloning, constrained move and resize, alignment of objects, and zooming.

Submodules can be pinned (having a fixed position), or unpinned (auto-layout). Graphical features that can be edited are background image, background grid, default icons (via display string inheritance), icon sizing and colouring, transmission range, and many others.

3.4-Network Simulator-3(NS-3): The NS-3 simulator is a discrete-event network simulator targeted primarily for research and educational use. NS-3 is open-source, and the project strives to maintain an open environment for researchers to contribute and share their software. NS -3 is not a backwards-compatible extension of NS-2. The two simulators NS-2 and NS-3 are both written in C++ but NS-3 is a new simulator that does not support the NS-2 APIs. Some models from NS-2 have already been ported from NS-2 to NS-3. NS-3 has been developed to provide an open, extensible network simulation platform, for networking research and education.[15,16] In brief, NS-3 provides models of how packet data networks work and performs, and provides a simulation engine for users to conduct simulation experiments. Some of the reasons to use NS-3 include performing studies that are more difficult or not possible to perform with real systems, to study system behaviour in a highly controlled, reproducible environment, and to learn about how networks work. Users will note that the available model set in NS-3 focuses on modelling how Internet protocols and networks work, but NS-3 is not limited to Internet systems; several users are using NS-3 to model non-Internet-based systems. Many simulation tools exist for network simulation studies. Below are a few distinguishing features of NS-3 in contrast to other tools [23].NS-3 is designed as a set of libraries that can be combined together and also with other external software libraries. While some simulation platforms provide users with a single integrated graphical user interface environment in which all tasks are carried out, NS -3ismoremodularinthisregard. Several external animators and data analysis and visualization tools can be used with NS-3. However, users should expect to work at the command line and with C++ and/or Python software development tools [24].NS-3 is primarily used on Linux systems, although support exists for FreeBSD, Cygwin (for Windows), and native Windows Visual Studio support is in the process of being developed. The NS-3 interface is shown in Figure:4

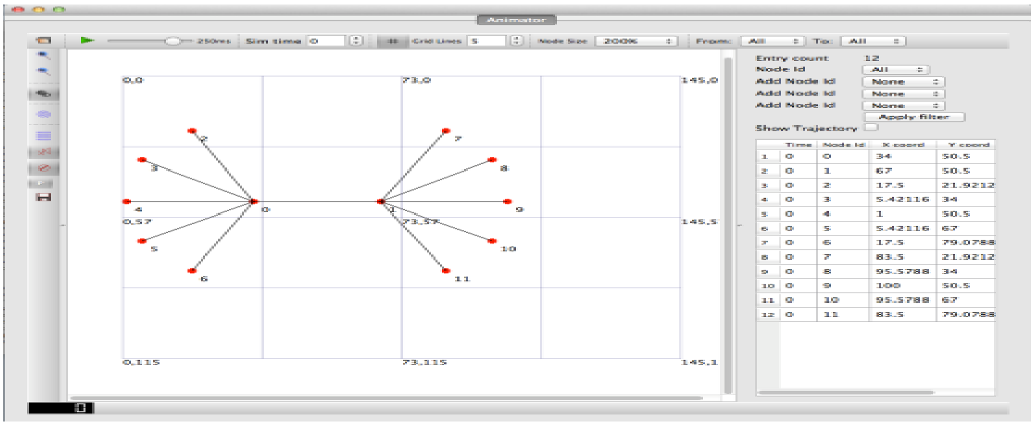

Figure-4: NS-3 Simulator

IV. 4-Ad-hoc On Demand Vector Protocol (AODV):

In AODV, the network remains silent unless a connection is needed by any node in the network. When a connection is required, the source node broadcasts a connection request. Referring to Figure 4, node A wants to communicate with node H, therefore, node A broadcasts a route request (RREQ) message in the network. [12] Every node in the network forwards the RREQ message to its neighbors and records the previous node from where the request was received. When the destination node H receives the RREQ message, it sends back a unicast route reply (RREP) message to the source node through the node that delivered the RREQ message. The intermediate nodes that receives the RREP message forwards it to the next node with the smallest distance towards the source node as shown in Figure 5. The entries that are not used in the routing tables are recycled after a time. If a link fails, a routing error message is sent back to the transmitting node and the route discovery process is repeated.

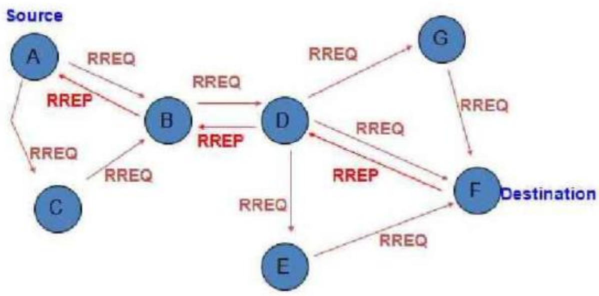

Figure-5: AODV Protocol

The advantage of AODV is that it generates no extra traffic for communication along existing links. In addition, distance vector routing is simple and does not require much memory or calculation. However, AODV requires more time to establish a connection, and the initial communication to establish a route is higher than some other protocols.[14,21 & 22] The main advantage of this protocol is that the routes are established on demand, and destination sequence numbers are used to find the latest route towards the destination. The disadvantage of this protocol is that intermediate nodes can lead to inconsistent routes if the source sequence number is very old. If the intermediate nodes do not have the latest destination sequence number, stale entries may exist. In addition, multiple RouteReply packets may be created in response to a single RouteRequest packet that can lead to high control packet overhead. Moreover, AODV periodic beaconing leads to unnecessary bandwidth consumption

V. Simulation and Results

To evaluate the performance of state of the art simulators, we simulated AODV routing protocol on selected simulators and evaluated the performance.

5.1. Simulation Setup: Before the start of a simulation, we configure connection establishments between pre-determined nodes, e.g. node A sends data to node B. As the communication starts, the source node starts transmitting at a regular interval of 0.2 seconds. During the simulation, the number of nodes was varied from 400 to 2000. In addition, each simulation was executed for 500 seconds in a simulation area of 1000 * 1000 (X * Y). The parameters and different versions of Simulators are summarized in Table 1& 2.

Table 1: Simulation setup

Simulation Time	500s
X, Y Dimensions	1000 x 1000
Mobility Model	None
Packet size	512 kb
Number of nodes	400-2000
Routing protocol	AODV

Table-1: Simulation / Scenario

NS-2 version	2.34
NS-3 version	3.10
GloMoSim version	2.03
OMNET++ version	4.2

Table-2: Simulator Versions

The simulations tools were executed on the Linux platform.

5.2- Results: The performance comparison was done based on the following parameters: memory usage (MB), CPU utilization (percent), scalability, and computation time (seconds).

5.2.1- Memory Usage

We simulated the AODV protocol for 500 seconds while varying the number of nodes from 400 to 2000. As shown in the Figure 6, NS-2 uses the highest amount of memory while NS-3 uses the lowest amount of memory compared to OMNET++ and GloMoSim. As the number of nodes increases, there is a linear growth in memory consumption for all simulators with minor difference. NS-3 was found to be the most efficient in memory usage among selected simulators.

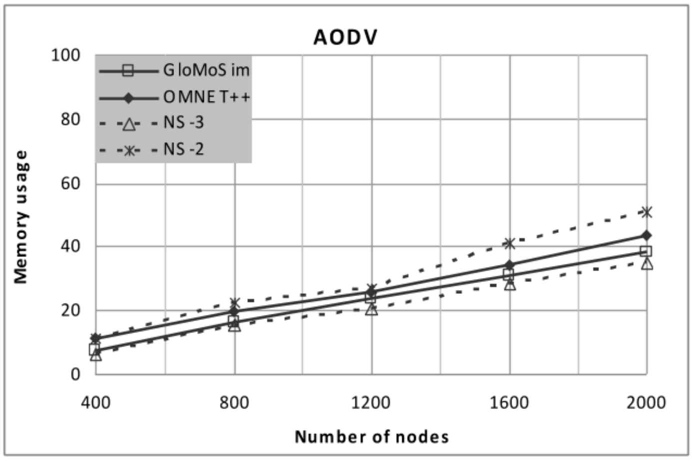

Figure-6: Number of Nodes vs Memory Usage

5.2.2- CPU Utilization

CPU utilization was measured while varying the number of nodes. For all simulators, there is little effect on CPU utilization as the number of nodes increases. Figure 7 shows that the CPU utilization of NS-2 and NS-3 is almost similar (5% variation) and is much higher compared to GloMoSiM and OMNET++. GloMoSiM and OMNET++ shows a CPU utilization of only up to 35% with very little difference between them. The behavior of NS-2 and NS-3 was analyzed based on CPU utilization in detail by executing different applications in parallel with simulation tool. The simulations usually take a long time to execute while researchers use other applications, waiting for the results. We found that when other applications are executed in parallel, the CPU utilization of NS-2 and NS-3 drops to about 50%, hence allowing other applications to execute in parallel.

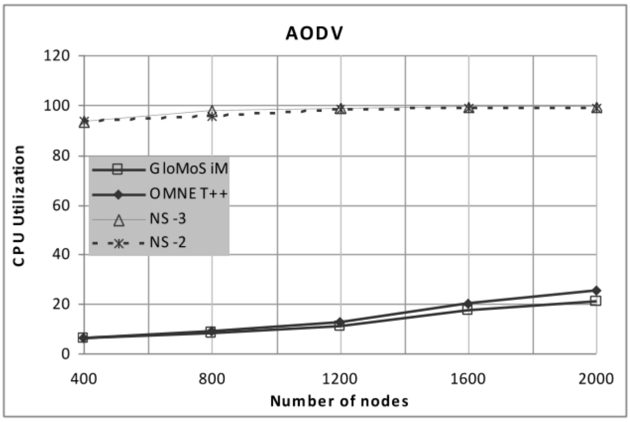

Figure-7: Number of Nodes vs CPU utilization

5.2.3- Computational Time

The computational time was calculated by simulating AODV protocol for 500 seconds while increasing the number of nodes. As illustrated in Figure 8, NS-2 has the highest computation time. In addition, NS-2's computation time increases rapidly with increasing number of nodes, which means NS-2 is not scalable. For large number of nodes, it may take a very long time compared to the other simulators. The computation time of the other simulators is quite low compared to NS-2. In terms of computation time and scalability, NS-3 appears to be the most efficient.

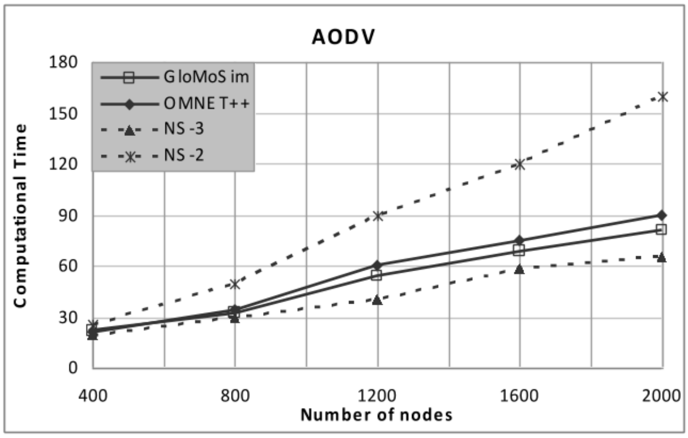

Figure-8: Number of Nodes vs Computational Time

VI. Conclusion

In this paper, we evaluate the performance of four network simulators with respect to different parameters. Based on the simulation results, we conclude that NS-3, OMNET++, and GloMoSiM are capable of carrying out large scale network simulations. NS-3 has proven to be the fastest simulator among the selected simulators in terms of computation time. In addition, NS-2 and NS-3 fully utilize the CPU, but is able to reduce CPU utilization when other applications are executed in parallel. Despite being quite new and is still under development, NS-3 demonstrates the best performance among all.

References

[1]. J. Heidemann, K. Mills, S. Kumar, "Expanding Confidence in Network Simulation," IEEE Network Magazine, Vol. 15, No. 5, Apr. 2000, pp. 5863.
[2]. S. R. Das, C.E. Perkins, E.M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," INFOCOM, Tel-Aviv, Israel, Aug. 2000.
[3]. A. R. Khan, S. A. Madani, K. Hayat, S. U. Khan, "Clustering-based power-controlled routing for mobile wireless sensor networks," International Journal of Communication Systems, Vol. 25, No. 4, pp. 529-542, 2012.
[4]. J. Broch, D. A. Maltz, D. B. Johnson, Y.C. Hu, J. Jetcheva, "A Performance Comparison of MultiHop Wireless Ad Hoc Network Routing Protocols," ACM/IEEE international conference on Mobile computing and networking, 1998.
[5]. K. M Reineck, K. Jonas, K. Uhde, "Evaluation and Comparison of Network Simulation Tools" Master Thesis, Aug. 2008.

[6]. Karl M., "A Comparison of the architecture of network simulators NS-2 and TOSSIM" In Proc. of Performance Simulation of Algorithm and Protocols Seminar Institute of parallel and distributed systems, January 2005
[7]. E. Weingarten, H. V. Lehn, K. Wehrle, "A performance comparison of recent network simulators," IEEE International Conference on Communications, 2009.
[8]. D. M. Nicol, "Comparison of Network Simulators Revisited," Dartmouth College, http://www.ssfnet.org/Exchange/gallery/dumbbell/ dumbbellperformance-May02.pdf, May 2002, accessed Jan 25th, 2012.
[9]. H. Sundani, H. Li, V. K. Devabhaktuni, M. Alam, "Wireless Sensor Network Simulators A Survey and Comparisons," International Journal of Computer Networks, Vol. 2, Apr 2011, pp. 249265.
[10]. B. Schilling, J. Hahner, "Qualitative Comparison of Network Simulation Tools, Modeling and Simulation of Computer Systems" Technical report, Institute of Parallel and Distributed Systems, University of Stuttgart, January 2005.
[11]. A. Hassan, "VANET Simulation," Masters Thesis in Electrical Engineering, School of Information Science, Computer and Electrical Engineering, Halmstad University,May 2009.
[12]. P. Yinfei, "Design Routing Protocol Performance Comparision in NS2: AODV Comparing to DSR as Example", Department of Computer Science, SUNY Binghamton, Vestal, 2010.
[13]. C. H. Vrije, M. Hofmann, "Performance Comparison of Reliable Multicast Protocols using the Network Simulator ns-2" Annual conference on local computer networks, 1998.
[14]. G. Carneiro, P. Fortuna, M. Ricardo, "FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3)," Fourth International ICST Conference on Performance Evaluation Methodologies and Tools, 2009.
[15]. Wireshark. http://www.wireshark.org, accessed Jan 10th 2012.
[16]. T. Saeed, H. Gill, Q. Fei, Z. Zhang, B.T. Loo, "An Open-source and Declarative Approach towards Teaching Large-scale Networked Systems Programming," ACM SIGCOMM Education Workshop, Toronto, Canada, 2011.
[17]. A. Varga, "THE OMNET++ DISCRETE EVENT SIMULATION SYSTEM" European Simulation Multiconference , 2003.
[18]. A. Varga, A. Y. Sekercioglu, "Parallel Simulation Made Easy With OMNeT++" 15th European Simulation Symposium, 2003.
[19]. R. Bagrodia, R. Meyerr, "PARSEC: A Parallel Simulation Environment for Complex System" UCLA technical report, 1997.
[20]. GloMoSiM, http://pcl.cs.ucla.edu/projects/glomosim/, accessed Sept 20th 2011.
[21]. X. Zeng, R. Bagrodia, M. Gerla, "GloMoSiM: A Library for parallel simulation of Large scale wireless networks" Workshop on Principles of Advanced and Distributed Simulation, pp. 154161, 1998.
[22]. L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, M. Gerla, "GloMoSiM: A Scalable Network Simulation Environment," IEEE Transactions on Reliability - TR, 1999
[23]. The network simulator, http://www.isi.edu.nsnam/ns, accessed Sept 20th 2011.
[24]. T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 project goals," workshop on ns-2: the IP network simulator, ACM, New York, USA, 2006.
[25]. A. Varga, R. Hornig, "An overview of the OMNeT++ simulation environment," International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, Mar. 2008.

IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) is UGC approved Journal with Sl. No. 5016, Journal no. 49082.

M.Khaleel Ullah Khan. "Performance of Wireless Sensor Network Simulators using Ad-Hoc On Demand Distance Vector (AODV) Protocol." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), vol. 12, no. 5, 2017, pp. 13–19.