ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse338942020.pdf
https://doi.org/10.30534/ijatcse/2020/338942020

Performance Evaluation of Routing Protocols in NS-2 and NS-3 Simulators

[1]Munsifa Firdaus Khan, [2]Indrani Das
[1,2] Department of Computer science, Assam University
Silchar, Cachar, Assam, India
[1]munsifa737@gmail.com
[2]indranidas2000@gmail.com

ABSTRACT

A real-world formation of a network is very hard in a network research area. It is very expensive and time consuming to deploy a single test bed. Deployment of a network becomes easy and cost effective with the help of a network simulator. In this paper, we discuss and compare the most popular routing protocols AODV and DSDV in open source network simulation tools, namely NS-2 and NS-3 using performance metrics throughput, delay and Packet Delivery Ratio (PDR). It is observed that the throughput is better using NS-3 and the PDR and delay is better using NS-2 for both the routing protocols. The performance of both the simulators varies highly due to the use of additional parameters used in the simulation environment for NS-3. We have also compared different properties of both the simulators which will help those researchers who feel difficult to select the appropriate network simulation tools for their research.

Key words: AODV, DSDV, Network Simulator**,** Performance Analysis.

1. INTRODUCTION

Simulation is a very significant technology for the present generation [6]. Simulation is actually an estimated replication of the function of a process or system. The modelling of real life and hypothetical objects on a computer is done with the help of computer simulation in order to understand how the system works. It can be applied into different fields [5], such as:

- In science which includes physics, chemistry and biology.
- In engineering, which includes civil, structural, mechanical, software, and computer engineering.
- In networking area like network simulation.

Computer simulation supports modeling and analysis of many natural systems. In Mobile Ad hoc Networks [MANETs], nodes are mobile which impacts the performance. Quality-of-Service is important for MANET [33], [34].The performance of a computer network can be predicted by a network simulator by computing the communication between the distinct network entities such as nodes, access points, routers, switches, links etc. [2], [10]. It is very expensive when all of these components are considered for setting up a network. The factors like efficiency, security, speed, etc. need to be considered besides cost [9]. Therefore, introduction of any new methods in a network is relatively risky. In the real world, the scenarios which are too expensive and time consuming are analyzed by the network developers with the help of a network simulator [10]. It is also very helpful to experiment with any new networking protocol. Researchers are developing several new protocols which are appealing to be more effective. Network simulators are also useful in analyzing existing protocols and modify those protocols to improve their performances in a controlled and consistent domain [6].

One of its advantages is that when designing real world system simulators are able to provide users a practical feedback which helps the user to determine the correctness and efficiency of a design before the system is actually constructed. Another advantage is network simulators allow system developers to study a problem at different levels of abstraction. Thirdly, for representing concepts to researchers or students, simulators that have computer graphics and animation are used effectively [13]. Finally, it is easy to maintain a simulator. At any stage of processing users have full control over it. It is easy and cheap to create and test a simulation scenario [8]. Despite of the advantages, simulators have disadvantages too. After initiation of a simulation there can be delays in producing its result. Delays may be due to the huge number of entities being simulated or due to the complex interactions that occur between the entities within the system being simulated. Therefore, due to hardware limitation of these simulators the computational demands of the simulator cannot be met. One of the methods to solve the aforesaid difficulty is to initiate untangle assumptions or heuristics into the simulator engine. Though the simulation time can be reduced using this technique, but the accuracy of the simulation results may not be corrected [13]. If the simulation environment scenarios are not set according to the actual requirement, then the result of the simulation may not be logically correct.

Network simulators can be commercial and open sourced.

- Commercial network simulator does not offer the source code of its simulation tool or the associated simulation tool packages for the common users free of cost. To use the simulation tool all the users have to buy the license or they have to pay to order specific packages for their own specific usage requirements [5]. Examples of these kinds of simulation tools are: are QualNet, NetSim and OPNET. The main advantage of commercial simulators is that it has ample and informed documentations. The maintenance of the software is done consistently by a specialized person of that organization [5]. However, the drawback of this network simulator is that it is very expensive and may not be affordable for all kinds of research institutions.

- Open source network simulator comes under an open source license. The license fee is not needed for this simulation tool. All the users can easily study its source code, alter it, allocate it and find bugs in it [3]. In comparing to commercial simulators it is very flexible and development of new technologies is reflected faster here. The chief advantages of commercial network simulators are the main disadvantages of open source network simulators. Lack of systematic maintenance and complete updating of documentation by a specialized person leads to severe problems [5]. Examples include: NS-2, NS-3, OMNet++, SSFNet and J-Sim.

There are various network simulators with different characteristics. Some of them are: OPNET, NS-2, NS-3, NetSim, OMNeT++, REAL, J-Sim and QualNet. Different network simulators are compared on the basis of: connections, node's identification, range that lie between very simple to the complex and traffic between the nodes [3], [6], [10]. In this paper, we discuss about the chief characteristics, advantages and disadvantages of open source network simulators like NS2 and NS3. We have also compared different properties of both the network simulators. We have done simulation experiments of both the routing protocols, namely AODV and DSDV to check the performance with QoS metrics throughput, delay and PDR in NS-2 and NS-3. This paper will benefit students and researchers who are in the beginning stage of research. It will help them to select the appropriate network simulation tool for their research work. It will also help them to gain a brief knowledge about the popular open source network simulators.

The remaining paper is organized as: Section 2, gives a brief explanation about network simulators. In section 3, we discuss standard routing protocols. Section 4 presents the Simulation results. Section 5 discusses the experimental results and Section 6 concludes the paper.

2. NETWORK SIMULATORS

Different Network simulation tools have different features. In this paper, we present the basic concepts, architectures, benefits, limitations of the most popular open source discrete event simulators like NS-2 and NS-3. Table 1 shows the comparison. Both the simulators compared provides analysis and visualization tools.

2.1. Network Simulator version-2 (NS-2)

NS-2 is an open source, discrete event simulator widely used for research in the field of computer networking. In 1995, the Virtual Inter Network Testbed (VINT) developed NS-2. The project is jointly done by the people from the University of California at Berkeley, University of Southern California's Information Sciences Institute, Lawrence Berkeley National Laboratory and Xerox Palo Alto Research Center [3], [6]. It uses C++ and OTcl as the programming languages. The combination of these two languages has its characteristics. There is isolation in the implementation of control path and data path [2]. The core implementation is done with C++ whereas OTcl controls the simulation set-up and list the events [5]. In case of efficient execution C++ is quicker whereas for modification it is moderate. OTcl executes much slower than C++, but can be modified very quickly [3], [6]. The architecture of NS-2 is shown in Figure 1 [4]. TclCl combines C++ objects and OTcl objects as shown in the Figure 1. The input of NS-2 is the Tcl (.tcl) scripts and after simulation trace (.tr) files are produced as the outputs. The trace files can be interpreted interactively using tools like Network Animator (NAM), Xgraph and Gnuplot [2]. NS-2 provides simulation of various routing protocols like AODV, DSDV, DSR, TCP, UDP, HTTP, SRM and so on. The performance of these protocols is analyzed by using the trace files generated by NS-2 after simulation.

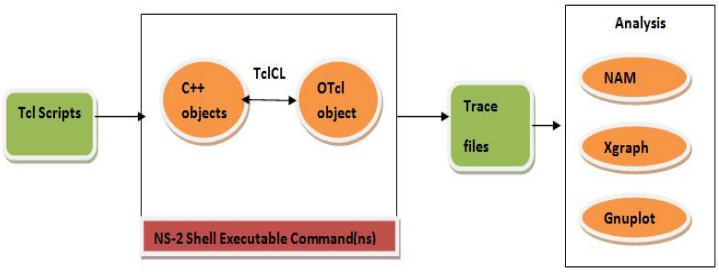

Figure 1: Architecture of NS-2 [1][2][3][4].

A. Network Animation Tool (NAM)

NAM is a Tcl/TK based animation tool for visualization of output in NS-2. It plays a very important role in NS-2 simulations [17]. Without NAM files output cannot be visualized. It is used to view the packet transmission at each link in a network. It supports layout of the topology, packet level animation and different data inspection tools.

B. Trace Files

The output of NS-2 simulations is the trace files. The data in the trace files are in ASCII code. A trace file holds 12 fields in order to represent its data. The trace file format for NS-2 is shown in the Figure 2 below:

Figure 2: Trace file format [23]

Each trace line begins with an event (+, -, r, d) descriptor followed by the simulation time (in seconds) of that event. The field "from and to node" associates the link on which the event occurred [16]. After that flag (E, P, A, F, N) bit is in the row. Subsequently the type of a packet and its size is represented in bytes. Then flow id (fid) of an IP address which can be set by a user for each flow in the input script. The subsequent two fields are source and destination address in forms of "node port" [16]. The last field shows the network layer protocol's packet sequence number.

C. Benefits

- It provides parallel and distributed simulation. Parallelism refers to the simultaneous performance of dissimilar commands of the same program. It is used to speed up simulations. One the other hand, Distribution refers to the partitioning of program data or code (or both) on different computers. It is primarily used to bring scalability and/or to enable parallelism [24].

- It is much closer to the real system i.e. with its imitation ability, NS-2 can be associated with a real network [8].

- Availability of huge number of models like realistic mobility models, propagation models and energy models. [1], [8], [10].

- Complex scenarios like large wired and wireless network scenarios are easily tested [1], [2], [10].

- It is popular for its modularity [1][10]. NS-2 enables the co-existence of multiple modules within each layer of the protocol stack [25].

- It has strong and adaptable scripting and simulation set up [8]. It offers a complete documentation and a regularly updated manual for C++ and Otcl classes. There are also mailing lists and many web pages for its large user community which makes it easy to access information about the simulator.

D. Limitations

- NS-2 is a single threaded network simulator that preserves a priority queue of events. The event that arrived first is treated first at each time step and thus producing more events in the queue resulting in an inadequate scalability in terms of memory usage and computation time [26].

- It is difficult to use tracing system because it is just a block of ASCII data in a file [2].

- It is difficult in NS-2 to analyze and understand the code [2] because of its complex infrastructure [1], [10].

- With increasing number of nodes, the time taken by NS-2 for simulation also increases.

- Every time when the user changes the existing code, it is important to do a recompilation [1], [2], [10]. Sometimes, changes in the codes lead to re-installation of the simulation software.

2.2. Network Simulator version-3 (NS-3)

The NS-3 is an open source, discrete-event network simulator that is broadly used in academic research and educational purpose. In 2006, the NS-3 project was started and it is licensed under the GNU GPLv2 license [2], [3], [4], [6], [10]. The NS-3 simulation tool is devoted to constructing a hard simulation core that is finely documented, simple to apply and debug [12]. The core of NS-3 is written in C++ and with Python scripting interface. Now a day many researchers uses NS-3 to carry out their research work in the field of networking. Protocols which are newly designed as well as the existing ones can be written and tested in NS-3 using C++ language. NS-3 helps users to design protocol entities as close to real computers. It reduces the requirement of rewriting models for simulation and also supports the integration of more open-source networking software. It supports lightweight virtual machines [5]. NS-3 is trying to enable customization of the output without rebuilding the simulation core [5]. The basic architecture of NS-3 is given below in the Figure 3 [6], [2].

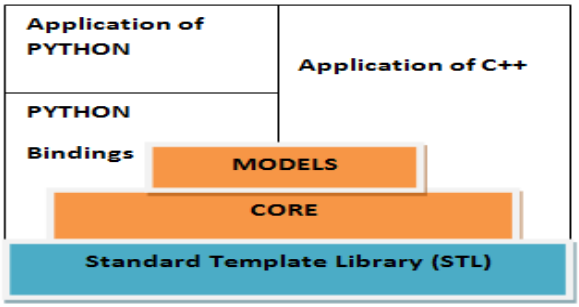

Figure 3: Architecture of NS-3 [1][2][3][6].

A. *Network Animator (NetAnim)*

NetAnim is an offline animation tool based on the Qt toolkit. It animates an XML trace file which is created during the simulation. George F Riley developed the first version [19]. It animates packets over wired and wireless networks provided inadequate support for LTE traces. It does not provide any support for Ipv6 [19]. It provides the flow of data transmission from a particular source to destination. It is basically the output of any simulation.

A. *Trace Files*

NS-3 generates trace (.pcap and .tr) files after compilation of .cc scripts. The performance analysis of existing and new protocols can be done by using the trace files generated by NS3. The motto of development of NS3 Trace Analyzer is to make researchers comfortable in analyzing the performance of a protocol using trace files. The format of trace files in NS-3 and NS-3 is completely different from each other. The tracing of a file in NS3 can be done in two ways:

Ascii tracing: Like NS-2, Ascii Trace Helper generates .tr files but the format of trace files in NS-2 and NS-3 are different from each other [18]. Trace files generated are very huge so .awk scripts and perl scripts are required to extract data from these files which are later used for performance analysis of any given protocol using metrics like throughput, delay, PDR etc. The format of trace file is shown in Figure 4 below:

Figure 4: Trace file format [18]

The starting field of a trace file is an event (+, -, r, d) descriptor succeeded by the simulation time (in seconds) of that event. The next two fields indicate starting and ending nodes at which, an event has occurred. After that packet type and its size (in bytes) is represented followed by flag bit. Then the next field indicates class of the packet, which can help to recognize a TCP connection. Subsequent two fields represent the address of source and destination. Finally, the line ends with the sequence number and identifier of the packet.

Pcap tracing: Wireshark, GUI application and tcpdump, shell command are used to analyze .pcap files [18].

B. *Benefits*

- In NS-3, models may be enabled/disabled, users may maintain their own models separately and several external animators, data analysis and visualization tools can be used with NS-3 which makes it highly modular in compared to NS-2 [1], [2], [10], [14].

- NS-3 is flexible in comparison to other simulators [1], [2]. Changing the existing codes is much easier in comparison to NS-2.

- NS-3 is planned for integration into test-bed and virtual machine environment [14]. Integration with real networks is done with the help of a special feature called emulation mode [2]. Simulators can be amalgamated with and connected to real networks to send and receive traffic of these networks [27].

- In NS-3, mailing list like the user mailing list, NS-developer mailing list is maintained in a responsive manner [2].

- Validation of the NS-3 model becomes much easier and reliable with the support of tools provided by the NS-3 testing environment. The process of getting the NS-3 model behavior to agree with the desired target system behavior as defined by the model qualification process is called model validation [14].

- NS-3 is widely used in the extension, modification and optimization of existing models.

C. *Limitations*

- NS-3 has limited scalability concerning the memory usage and the required computation time [2]. Every node, channel and the other components need memory, so their number is restricted by the accessible memory. Events are handled in a certain time and the number of events that can be handled is restricted to assume that the simulation designer defines a maximum time for the computation of the simulation [27].

- Lack of credibility and reliability in NS-3[2]. Simulation reliability is a challenge and all characteristics of reality is not possible to implement in a simulation hence negotiations have to be made. [27].

- In NS-3, python does not support cent percent of the API like for scripting and visualization because some of the APIs involve pointers that require information about memory passing semantics and such information is not involved in the function signatures or is either documented or sometimes it is not even documented [2], [14].

- Specialized maintainers are needed to highlight the features of NS-3 like frequent code reviews, correcting

errors, and answering to questions on mailing lists, pertaining to their respective modules [14].

2.3. Comparison of NS-2 and NS-3

We have compared different features of both the network simulators namely NS-2 and NS-3 as shown in Table1 given below.

Table 1: Different features of NS-2 and NS-3 [3], [4], [31], [32]

Features	NS-2	NS-3
Based on	NS-1 and REAL simulator	NS-2, GTNets, YANS
Language Supported	C++, OTCL	C++, Python
Platform	Windows, Linux, Unix, Cygwin	Windows, Linux, Unix, Mac OS
GUI support	Poor	Good
Cost and License	Free and Open Source	Free, GNU General Public License
API	Event Base	low-level, users can mix and match between the simpler API
Network Support	Wired Network, Wireless Ad-Hoc mode, Wireless Managed mode, Wired cum Wireless, cannot simulate problems of the bandwidth or the power consumption in Wireless Sensor Network	Wired Network, Wireless Network, Wireless Sensor Network
Simulation Output	NAM	NS-3-viz, pyviz
Simulation	Virtual	Real
Computation time	High	Low
CPU utilization	Maximum 2.0% yearly (low)	Maximum 41.7M% yearly (high)
Execution	Moderate	Best
Complexity	More complex	Less Complex
Compatibility	More Compatible	Less Compatible
Memory Consumption	High	Low

3. ROUTING PROTOCOLS

We have considered here two routing protocols. One is proactive namely DSDV and another is reactive namely AODV routing protocol.

3.1. Destination Sequenced Distance Vector (DSDV)

Destination Sequence Distance Vector (DSDV) is a proactive i.e., a routing table is used where each node maintains up to date route information about every node in a network [29]. This protocol solves the routing loop problem [30]. As this protocol has routing information in advance so delay is minimized here. This protocol is not suitable for huge or dynamic network [22].

3.2. Ad Hoc On Demand Distance Vector (AODV)

Ad Hoc On Demand Distance Vector (AODV) is a reactive, i.e. a route is initiated on need [22], [28]. The use of sequence number in this protocol creates loop free routes [28]. AODV offers route maintenance, i.e. due to node mobility if a route breaks then its neighbor nodes respond to that failed route and changes in network topology in a timely manner. One of its advantages is the flexibility for extremely dynamic environment [20].

4. PERFORMACE ANALYSIS

4.1. Simulation Environment:

We have performed the simulation using NS-2 and NS-3. The Simulation parameters have been given in the Table 2.

Table 2: Simulation Environment for NS-2 and NS-3

Parameters	Values
Simulation area	1000x1000
Number of nodes	10, 20, 30, 40, 50, 60, 70, 80, 90 and 100
Simulation Start time	10 seconds
Simulation end time	150 seconds
Speed	Min:0 m/s and Max:20 m/s
Mobility Model	Random Way Point
Channel	Wireless
Routing Protocol	AODV and DSDV
MAC	802.11
Pause Time	0 seconds
Number of Destinations	2
Propagation Model	Free Space

For performance analysis, in NS-2, we have considered the trace files of AODV and DSDV whereas in NS-3 we have considered the flowmon files of AODV and DSDV. The QoS parameters that we have used for performance analysis are: throughput, delay and Packet Delivery Ratio (PDR).

4.2. Simulation Results

A. Throughput

It is defined as the amount of data packets transmitted through a path [22]. It is used as one of the performance

measures to check the QoS in MANET using simulation software, namely NS-2 and NS-3 on standard routing protocols like AODV and DSDV. We have observed in Table 3 that the highest and the lowest throughput obtained for AODV is using NS-3 which are 48.6125 kbps and 0.1080 kbps for 80 and 30 nodes respectively. It is also noticed that the throughput for AODV is equal for some set of nodes using NS-2. For nodes like 20, 40, 70 and 90 the throughput is 0.0202 kbps and nodes like 10 and 100 the throughput of AODV is 0.0201 kbps. It is seen that the throughput for AODV is higher for a large number of nodes using NS-3. Moreover, it is also analyzed that the throughput is better using NS-3 for all set of nodes. On the other hand, we have observed in Table 4 that the highest throughput obtained for DSDV is 1.1055 kbps for 20 nodes using NS-3 whereas the smallest throughput achieved is 0.0178 kbps for 100 nodes using NS-2. It is noticed that the throughput obtained for DSDV is better using NS-3 in comparison to NS-2. It is examined that the throughput obtained by AODV is higher and better than that of DSDV. Furthermore, for both the routing protocols, it is noticed that the throughput is better using NS-3.

Table 3: Throughput of AODV

Number of nodes	NS-2	NS-3
10	0.0201	0.1443
20	0.0202	8.2744
30	0.0193	0.1080
40	0.0202	18.0105
50	0.0203	22.8606
60	0.0200	15.4291
70	0.0202	22.9033
80	0.0158	48.6125
90	0.0202	22.2002
100	0.0201	23.8125

Table 4: Throughput of DSDV

Number of nodes	NS-2	NS-3
10	0.0194	0.7655
20	0.0197	1.1055
30	0.0190	0.2035
40	0.0195	0.1987
50	0.0190	0.8835
60	0.0188	1.0322
70	0.0190	0.4409
80	0.0186	0.1183
90	0.0188	0.2362
100	0.0178	0.7452

B. Delay

It is defined as a time taken by a network for transmission of data packets from a particular source node to a destination node [22]. It is also used as the one of the major performance measures to check the impact of QoS in MANET. It is observed in Table 5 that the least delay for AODV is 120.5814 seconds for 90 nodes using NS-2 whereas the largest delay for AODV is 398.9098 seconds for 100 nodes using NS-3. The simulation result clearly shows that the delay is higher using NS-3 for AODV. On the other hand, Table 6 shows that the least delay for DSDV is 0.8819 seconds for 70 nodes, whereas the largest delay is 8419.0 seconds for 80 nodes using NS-3. It is also observed that using NS-3 the delay value obtained for DSDV is highly varied for some set of nodes like 20, 30, 50 and 70 gives the lower delay values, whereas for 10, 40, 60, 80, 90 and 100 the delay value obtained is much higher. However, it is seen that the delay value obtained for DSDV using NS-2 is uniform. It is observed that the delay is higher using NS-3 in contrast to NS-2 which implies that the delay is better using NS-2. Moreover, it is noticed that for 80, 90 and 100 nodes the delay value is higher for DSDV using both the simulation tools in compared to other set of nodes. So, it is concluded that for a large number of nodes the performance of DSDV is not better with respect to delay. Furthermore, it is examined that using both the simulation tools, the delay obtained by AODV is lower in comparison to the delay obtained by DSDV. Therefore, it is concluded that the performance of AODV is better in comparison to DSDV with respect to delay.

Table 5: Delay of AODV

Number of nodes	NS-2	NS-3
10	233.9846	320.1923
20	239.5950	393.5263
30	214.1067	188.0
40	214.5289	214.0235
50	237.7458	206.6679
60	235.0969	166.0170
70	157.4674	331.4749
80	167.4926	262.7533
90	120.5814	304.6885
100	234.7496	398.9098

Table 6: Delay of DSDV

Number of nodes	NS-2	NS-3
10	248.7765	5096.0
20	245.3507	18.0
30	247.4802	22.0
40	244.5499	636.5
50	246.9507	5.0
60	245.8011	6963.5
70	247.4450	0.8819
80	257.6843	8419.0
90	255.7319	5319.5
100	271.5086	6694.0

C. PDR

It is defined as the ratio between the number of received data packets by a particular destination node to the number of data packets generated by the particular source node [22]. It is also another performance measure for checking QoS in MANET. It is observed in Table 7 that the highest PDR for AODV obtained is 0.9989 for 90 nodes using NS-2 whereas the least is 0.2162 for 30 nodes using NS-3. It is noticed in Table 7 that the PDR for AODV is same for some set of nodes using NS-2. Nodes like 20, 50, 60 and 100 gives the equal PDR value of 0.9983 and nodes like 30 and 70 gives PDR value of 0.9981. Moreover, it is analyzed that using NS-3 PDR for AODV is better for larger number of nodes. However, it is also noticed that the PDR for AODV is better using NS-2 in comparison to NS-3. On the other hand, Table 8 shows that the lowest PDR for DSDV is 0.0172 for 80 nodes using NS-3 whereas the largest PDR is 0.9982 for 40 nodes using NS-2. It is observed that the PDR value for DSDV is uniform using NS-2. It is observed that the PDR for DSDV is better using NS-2 in contrast to NS-3. Moreover, it is examined that the PDR value of AODV is better in comparison to DSDV using both the simulation tools. In other words, the performance of AODV is well with respect to PDR in contrast to DSDV.

Table 7: PDR of AODV

Number of nodes	NS-2	NS-3
10	0.9966	0.5929
20	0.9983	0.6795
30	0.9981	0.2162
40	0.9982	0.8016
50	0.9983	0.9214
60	0.9983	0.9002
70	0.9981	0.8838
80	0.9974	0.9335
90	0.9989	0.8662
100	0.9983	0.8561

Table 8: PDR of DSDV

Number of nodes	NS-2	NS-3
10	0.9973	0.1552
20	0.9980	0.0689
30	0.9969	0.1034
40	0.9982	0.1379
50	0.9957	0.0345
60	0.9947	0.2759
70	0.9959	0.1034
80	0.9972	0.0172
90	0.9978	0.1552
100	0.9969	0.4310

5. RESULTS AND DISCUSSION

We have done a performance analysis of network simulation tools, namely NS-2 and NS-3 using routing protocols, namely AODV and DSDV with QoS metrics like throughput, PDR and delay. It is observed in Table 3 and Table 4 that the throughput for AODV and DSDV is higher using NS-3 in compared to NS-2. The higher the throughput the better is the QoS. Moreover, it is noticed from Table 5 and Table 6 that the delay value is higher for both the routing protocol using NS-3 which means that the delay is better using NS-2 for both AODV and DSDV because the lower the delay value the higher is the QoS. Additionally, it is analyzed that the PDR is higher using NS-2 for both the routing protocols as shown in Table 7 and Table 8. The higher the PDR value the better is the QoS. We have observed all the three QoS metrics like throughput, delay and PDR for both AODV and DSDV using both the simulation tools. It is noticed that though both the simulators conclude that AODV outperforms well in comparison to DSDV with respect to throughput, delay and PDR but the values of the performance metrics are highly varying with both the simulators. In other words, QoS is better using NS-2 with respect to delay and PDR whereas using NS-3 QoS is better for throughput. These differences in the result are due to the use of additional parameters like random rectangular position allocator along with a random way point mobility model in NS-3. Moreover, the use of On/Off helper for traffic control in NS-3 is another reason. Another reason is the use of Constant Speed propagation delay model in NS-3. Both the simulation tools are not identical which in turn leads to difficulty in creating the same simulation environment. While doing simulation using both the simulators it is observed that NS-3 takes much time during execution in compared to NS-2.

6. CONCLUSION AND FUTURE WORK

Network simulators help the network designers to test new networking protocols or to modify the existing protocols in a restricted and reproducible way. We have done performance analysis of standard open source network simulation tools, namely NS-2 and NS-3 using standard routing protocols like AODV and DSDV to check the impact of QoS using QoS metrics namely throughput, PDR and delay. It is observed that the throughput for both AODV and DSDV is higher using NS-3. However, the PDR and delay for both AODV and DSDV is better using NS-2. It is analyzed that higher the throughput, better is the efficiency and larger the PDR better is the reliability and lower the delay lesser is the congestion. This paper will help researchers to choose a particular network simulation tools for their research work. This paper can be further extended by analyzing other routing protocols like DSR, OLSR, AOMDV etc.

REFERENCES

1. M. H. Kabir, S. Islam, M. J. Hossain and S. Hossain. **Detail Comparison of Network Simulators,** *International Journal of Scientific & Engineering Research*, Vol. 5(10), pp. 203-218, October 2014.
2. R. L. Patel, M. J. Pathak, and A. J. Nayak. **Survey on Network Simulators**. *International Journal of Computer Applications*, vol. 975, pp. 8887, 2018.
3. S. G. Gupta, M. M. Ghonge, P. D. P. M. Thakare and P Jawandhiya. **Open-source network simulation tools: An overview.** *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol.2(4), pp.1629-1635, 2013.
4. T. Arvind. **A comparative study of various network simulation tools**. *Int. J. of Comput. Sci. and Eng*, vol. 7(8), pp. 374-378, August 2016.
5. J. Pan, and R. Jain. **A survey of network simulation tools: Current status and future developments.** *IJCSET, vol.* 7(8), pp. 374-378, 2016.
6. S. Saba, A. Gupta, and R. Badgujar. **Network simulation tools survey.** *International Journal of Advanced Research in Computer and Communication Engineering, vol.* 1(4), pp. 199-206, 2012.
7. S. Kamal and R. Prakash. **Improved survey on network simulation tools.** *International Journal of Engineering Research & Technology (IJERT),* vol. 3(4),2014.
8. M. Köksal, **A survey of network simulators supporting wireless networks**. *línea: http://www. ceng. metu. edu. tr/~ e1595354/A% 20 Survey*, October 2008.
9. K. Aseri. **A Comprehensive Overview of Simulation Tools for Virtual Network Implementation**. International Journal of Advanced Research in Computer Science and Software Engineering. Vol. 5(3), pp. 567-571, 2015.
10. L. Raja. **Study of Various Network Simulators** *International Research Journal of Engineering and Technology (IRJET)* vol.5(12), pp. 1192-1201, 2018.
11. C. Gayathri and R. Vadivel. **An Overview: Basic Concept of Network Simulation Tools**. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE*, vol. 6(1), pp. 19-22, 2017.
12. https://en.wikipedia.org/wiki/Network_simulation. (accessed on dtd 10/04/2020)
13. http://web.cs.mun.ca/~donald/msc/node6.html. (accessed on dtd 10/04/2020)
14. http://www.nsnam.org/ns-3-13/download/. (accessed on dtd 10/04/2020)
15. https://www.isi.edu/nsnam/ns/doc/node6.html. (accessed on dtd 10/04/2020)
16. A. U. Salleh, Z. Ishak, N. M. Din, M. Z. Jamaludin, **Trace Analyzer for NS-2**, 4th Student Conference on Research and Development (SCOReD 2006), Shah Alam, Selangor, MALAYSIA, June 2006, pp. 29-32.
17. https://ns2blogger.blogspot.com/p/network-animator-nam.html. (accessed on dtd 08/04/2020)
18. U. R. Pujeri, and V Palaniswamy**, Trace Analyzer for NS3**, *Advances in Information Sciences and Service Sciences(AISS*), vol.7(5), pp. 61-67, October 2015.
19. https://www.nsnam.org/wiki/NetAnim. (accessed on dtd 08/04/2020)
20. P. Landge, A. Nigavekar **Modified aodv protocol for energy efficient routing in manet**. *Int J Eng Sci Res Technol*, vol.5(3), pp.523–529, 2016.
21. M. F. Khan and I. Das. **A study on quality-of-service routing protocols in mobile ad hoc networks.** *In: 2017 international conference on computing and communication technologies for smart nation (IC3TSN). IEEE*, pp 95–98, 2017.
22. M. F. Khan and I. Das. **An Investigation on Existing Protocols in MANET**. © *Springer Nature Singapore Pte Ltd. H. S. Saini et al. (eds.), Innovations in Computer Science and Engineering, Lecture Notes in Networks and Systems 74*, pp 215-224, 2019.
23. https://ns2blogger.blogspot.com/p/the-file-written-by-application-or-by.html. (accessed on dtd 08/04/2020)
24. L. Hogie, P. Bouvry, and F. Guinand. **An overview of manets simulation**. *Electronic notes in theoretical computer science* vol.150(1), pp. 81-101,2006. https://doi.org/10.1016/j.entcs.2005.12.025
25. http://telecom.dei.unipd.it/pages/read/58. (accessed on dtd 09/04/2020)
26. S. Bak. **Large-scale network simulation scalability and an FPGA-based network simulator**, 2012.
27. S. Rampfl. **Network simulation and its limitations**. *Proceeding zum Seminar Future Internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und Autonomous Communication Networks (ACN)*, vol. 57, pp.57-63, 2013.
28. M. E. Elizabeth and C. E. Perkins. **An implementation study of the AODV routing protocol.** *2000 IEEE Wireless Communications and Networking Conference. Conference Record (Cat. No. 00TH8540)*, IEEE, vol. 3. 2000.
29. G. F. Ahmed, R. Barskar, and N. Barskar. **An improved DSDV routing protocol for wireless ad hoc networks**. *Procedia Technology* 6, 2012, pp. 822-831.

30. A. Kaur and A. K. Gupta. **Performance Evaluation of AODV and DSDV using NS-3.** *Int. J. Innov. Eng. Technol* 6, 2016, pp. 560-563.
31. Vidhi, A. Malik, H. Saini. **Network Simulators: A Comparative Survey**. *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* pp.52-56, 2015.
32. S. Suchita and B. G Prasanthi. **A Study of Network Simulation Tools**. *Research in Digital Revolution and New India,* pp. 270-272.
33. A. Rajesh, R. Shankari. **A Novel Opinion Computational model of multi path POR Routing Protocol based on subjective logic in Mobile Ad hoc Networks.** *International Journal of Advanced Trends in Computer Science and Engineering,* vol.8(5), pp. 2167-2177, 2019. https://doi.org/10.30534/ijatcse/2019/48852019
34. J. Gopal, J Vellingiri, J. Gitanjali, K. Arivuselvan and S. Sudhakar. **An Improved Trusted On-Demand Multicast Routing with QoS for Wireless Networks**. *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 9(1), pp.261-265, 2020. https://doi.org/10.30534/ijatcse/2020/39912020