Continuous and Discrete Time Analysis of Network in NS2

Preeti Wadhwa[1]
[1]*M.Tech. Student*, Department of ECE
DCRUST, Murthal
Sonipat, India
[1]*preetipooja.10@gmail.com*

Himanshi Saini[2]
[2]*Assistant Professor*, Department of ECE
DCRUST, Murthal
Sonipat, India
[2]*himanshi.4887@gmail.com*

Abstract- A Simulator is nothing but a system which can imitate the working of a real world process and a network simulator is also the system or the software which can establish and analyze a network similar to the network of the real world. There are basically two network simulation methods available, event-driven simulation and time-driven simulation. Both these methods have their own benefits like less simulation time, high efficiency, less operational overhead, user-friendly etc. depending on the application. Different methods of network simulation have different effect on the network parameters. This paper analyses the effect on the network parameters by converting the event-driven simulation of NS2 to time-driven simulation in Matlab.

Keywords: *Continuous, Discrete, NS2, MATLAB, Timestamps*

*****

I. INTRODUCTION

Simulation system simply contains a set of programs that models mathematical and logical relationships. In simulation, these models are evaluated using a set of commands by first collecting the data and then estimating & calculating the results of the targeted systems using this data. A Simulation model depicts state of evolution over time [1].

In Network Simulation, there are basically two methods available. These are event-driven simulation and time-driven simulation. In time-driven or time-stepping simulation, time is measured at small intervals giving the impression that system evolves continuously over time [1]. In an event-driven simulation which is also called as discrete-event simulation, time leaps through different points in time called events. In network simulation, there are many different events like packets transmitted, packets received, packets dropped etc. We can also combine both types of simulations depending on the application. But, here we are analyzing the effect on the network parameters by converting the discrete timestamps in NS2 to continuous timestamps in Matlab.

Network Simulation models have grown continuously in size. Higher degrees of detail and the desire for simulation of large networks have driven the requirement for high performance execution of such simulation models [2]. For analyzing the performance of network, effect of conversion of event-driven timestamps to time-driven on network parameters, is an important and significant step towards network survivability. Network survivability in itself is a crucial requirement for the rapidly increasing network traffic.

There are many different network simulators available for the simulation of small and large networks. Among them, Network Simulator version 2 abbreviated most commonly as

NS2 is a discrete event based simulator and also acts as a leader among the large number of competitors. It is an open source tool available for the field of networking. Since, it is an event-driven simulator, it leaps time when some kind of event occurs.

There are several different events that occur during simulation like packet enqueued, packet dequeued, packet received, packet dropped, packet acknowledged etc. Whenever any of these event occur, time leaps in NS2 and generates traces of these events in its trace file. These packets related events are required only to evaluate the network performance from user perspective. But there are many events like time at which packets left the queue etc, are of no use for calculating the network parameters like end-to-end delay, throughput, jitter etc. for increasing the network survivability. For evaluating the internal state of network entities or routing protocols, different events are required to be logged on. For this reason, it is required to convert these event based timestamps to time based vectors. Not only this, events are specific to this tool only and on the other hand, plotting different parameters with respect to time for analysis, is easy to understand and user-friendly.

There are many different tools available that can be used for this purpose. Matlab is one of them, which can be used effectively for parsing and plotting of NS2 traces. Parsing of NS2 trace file is done in Matlab to convert discrete event-based timestamps to continuous time-based vectors. This makes event-driven simulation as time-driven simulation that efficiently saves simulation time. After Parsing, plotting of different network parameters is done to analyze the effect of parsing on network parameters and thus network performance that will directly have significant impact on network survivability. The results can then be compared with results of NS2 generated using AWK files for further analysis.

II. RELATED WORK

There are many tools available in literature that efficiently collect data from NS2 traces and then perform parsing and plotting of these traces to evaluate network performance. The reason for doing parsing is mainly that the use of traces directly for evaluating network performance requires huge amount of disk space which entails additional simulation time and high post-processing overhead. Since NS2 lacks efficient support for data collection and statistical analysis of simulation results, paper [3] presents a software framework to extend functionality of NS2 for efficient data collection and network performance evaluation for large simulation experiments. This framework consists mainly of two parts. First is a flexible C++ class named *Stat* to collect samples for different network measures and second is a pair of stand-alone tools *analyzer* and *loader* that are set up based on input configuration text files and whose output measures can directly be plotted using standard graph tools.

Parsing of NS2 traces is also done to analyze rapidly growing internet traffic. Pareto distribution is used in [4] for modeling modern internet traffic characteristics. The study attempted to determine whether Pareto generated data has the properties of exponential distribution. Inter-arrival times between packets generated by NS2 were calculated and subjected to Chi-Square goodness of Fit test and it was found that Pareto which allows traffic burst can efficiently be used in place of exponential distribution to model modern internet traffic. Further Colored Petri Net in [5] also makes use of NS2 traces for parsing and plotting in Matlab to increase efficiency of simulator and to study the performance of network parameters for network survivability. AODV routing protocol is modeled using Colored Petri Net and results of simulation are studied and compared with NS2. The study revealed that Colored Petri Nets are useful for validating the design and analyzing the properties & functionalities of the model.

The importance of analyzing the network performance metrics for network survivability has also been shown in [6] which focus on precision timestamping to correctly interpret measurement results. For recording network traffic, accurate timestamping is done by using network packet traces to investigate the impact on network performance metrics. Network performance between Flooding and Directed Diffusion Protocol has also been studied in [7] using NS2 and presented in the form of graph. It was found that by parsing trace file to extract necessary information performance metrics was effected and shortest delay occurred in diffusion directed protocol as compared to flooding protocol and even the end-to-end delay was not found to be increasing continuously. Moreover both these methods of simulation also been used for performance analysis of networked control systems also as in [10].

The most prominent and similar work to this paper has been done in [8] to study and implement VoIP network using NS2.

A VoIP network was created and tested for the different faults, so that improvements can be made for better network survivability. In this network several nodes were set up on either side of the two routers and exponential traffic source was used for the conversation over VoIP using different protocols UDP, RTP and TCP. Then different network parameters like packet loss, delay, jitter, throughput etc. were measured & plotted on graph and the results were compared with the theory based predictions. It was then concluded that TCP outperforms RTP & UDP and even no packet losses were observed except for maximum load. But for UDP & RTP percent loss was approximately 25%. The parameters like delay and jitter were low for RTP & UDP.

III. SIMULATION MODEL

To study the effect of event based simulation and time based simulation on the performance metrics of the network for improving the network survivability, NSFNET as simulation model has been designed. The Simulation Model for continuous and discrete time analysis of a network in NS2 is shown in Figure2. Event based simulation is studied using NS2 whereas MATLAB supports time based simulation, methodology approached for which is shown in Figure1.

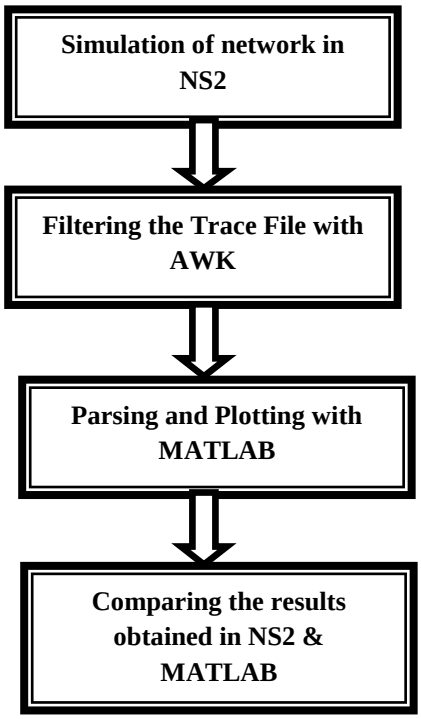

Figure1: Methodology Used

1. *Simulation with NS2*

NS2 is a discrete event based network simulator where timestamps are maintained by scheduler and is used to simulate different types of wired, wireless and optical networks. Here, NSFNET-14 has been designed in NS2 to analyze effect of continuous and discrete time simulation on the networks for improving network survivability. The network designed is a 14 nodes

network and has duplex links between the nodes with the bandwidth of 1.5 Mbps and 0.01s droptail. The traffic generator of the network is User Datagram Protocol most commonly known as UDP with static routing and queue limit of 4 packets and each packet of 1000 bytes in size. The snapshot of the network under analysis that has been designed in NS2 is shown in Figure2.

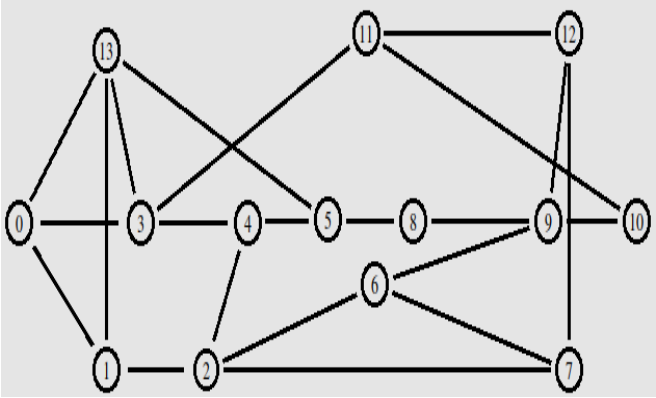

Figure2: Snapshot of network in NS2 (Simulation Model)

As discussed in [7, 3], simulation of network in NS2 generates two types of files trace file and NAM file. Traces and monitors are the only types of data collection methods supported by NS2. Traces record information of different events like packet enqueued, packet transmitted, packet size, flag etc. and the monitors give statistical information regarding the behavior of queues, amount of packets arrived and departed etc.

2. *Filtering the Trace File with AWK*

As already discussed, after simulation NS2 generates a trace file named *out.tr* which tracks all the information regarding the network in the form of events snapshot of which is shown in Figure3. This trace file consists of ASCII codes which are organized in 12 fields in the following order, Event Type, Timestamp, Source Node, Destination Node, Packet Type, Packet Size, Flags, Flow ID, Source Address, Destination Address, Sequence Number & Unique Packet ID.

For large network traffic, the trace file generated is very large with size in hundreds of Mega Bytes. Thus to make parsing easy, the trace file is first filtered using AWK command in NS2. The command may vary according to the application.

```
+ 0 0 3 cbr 1000 ------- 1 0.0 10.0 0 0
- 0 0 3 cbr 1000 ------- 1 0.0 10.0 0 0
r 0.015333 0 3 cbr 1000 ------- 1 0.0 10.0 0 0
+ 0.015333 3 11 cbr 1000 ------- 1 0.0 10.0 0 0
- 0.015333 3 11 cbr 1000 ------- 1 0.0 10.0 0 0
+ 0.017857 0 3 cbr 1000 ------- 1 0.0 10.0 1 1
- 0.017857 0 3 cbr 1000 ------- 1 0.0 10.0 1 1
r 0.030667 3 11 cbr 1000 ------- 1 0.0 10.0 0 0
+ 0.030667 11 10 cbr 1000 ------- 1 0.0 10.0 0 0
- 0.030667 11 10 cbr 1000 ------- 1 0.0 10.0 0 0
r 0.03319 0 3 cbr 1000 ------- 1 0.0 10.0 1 1
+ 0.03319 3 11 cbr 1000 ------- 1 0.0 10.0 1 1
- 0.03319 3 11 cbr 1000 ------- 1 0.0 10.0 1 1
+ 0.035714 0 3 cbr 1000 ------- 1 0.0 10.0 2 2
- 0.035714 0 3 cbr 1000 ------- 1 0.0 10.0 2 2
r 0.046 11 10 cbr 1000 ------- 1 0.0 10.0 0 0
r 0.048524 3 11 cbr 1000 ------- 1 0.0 10.0 1 1
+ 0.048524 11 10 cbr 1000 ------- 1 0.0 10.0 1 1
- 0.048524 11 10 cbr 1000 ------- 1 0.0 10.0 1 1
r 0.051048 0 3 cbr 1000 ------- 1 0.0 10.0 2 2
+ 0.051048 3 11 cbr 1000 ------- 1 0.0 10.0 2 2
- 0.051048 3 11 cbr 1000 ------- 1 0.0 10.0 2 2
+ 0.053571 0 3 cbr 1000 ------- 1 0.0 10.0 3 3
- 0.053571 0 3 cbr 1000 ------- 1 0.0 10.0 3 3
r 0.063857 11 10 cbr 1000 ------- 1 0.0 10.0 1 1
r 0.066381 3 11 cbr 1000 ------- 1 0.0 10.0 2 2
+ 0.066381 11 10 cbr 1000 ------- 1 0.0 10.0 2 2
- 0.066381 11 10 cbr 1000 ------- 1 0.0 10.0 2 2
r 0.068905 0 3 cbr 1000 ------- 1 0.0 10.0 3 3
+ 0.068905 3 11 cbr 1000 ------- 1 0.0 10.0 3 3
- 0.068905 3 11 cbr 1000 ------- 1 0.0 10.0 3 3
+ 0.071429 0 3 cbr 1000 ------- 1 0.0 10.0 4 4
```

Figure3: Snapshot of trace file in NS2

3. *Parsing and Plotting with MATLAB*

The filtered file obtained in previous step is then parsed in MATLAB for calculating different parameters of the network required for network survivability. Parsing is done in MATLAB by saving the data present in each column into different vectors by using MATLAB commands. This parsed data is then used for determining throughput for different time intervals. The difference in calculating throughput in NS2 and MATLAB lies in the fact that here, throughput measurement is made based on different time intervals, whereas NS2 tracks events instantaneously. As a result, the data in the trace file is not listed at equal time intervals but in MATLAB, calculations are done based on vector indices of the parsed data for equal intervals of time.

Thus for calculating the throughput for the interval of 1-2 seconds, we have to scan the timestamp for obtaining the indices of the start and stop times, complete algorithm for which is shown in Figure4. Throughput is then calculated using the formula shown below in equation (1) and as also given in [9].

$$\text{Throughput (kbps)} = \frac{\text{Number of Bytes received}}{\text{Simulation Time}} * 0.008 \quad \textbf{(1)}$$

Algorithm for Parsing and Plotting in MATLAB

1. Initialize variables to get in values from trace file.
2. Open filtered trace file using *fopen* command in read mode.
3. Read the first line of the file and store the contents in a variable.
4. Now split the contents of a line using *regexp* command and store the values in different variables initialized in 1.
5. Repeat steps 3 & 4 for all the lines.
6. Now remove the first entries to get rid of the initialized values if required.
7. Close the file when parsing is done, using the command *fclose*.
8. For measurement, first find indices of the specified time interval.
9. Perform calculation and keep track of time.
10. At last, plot the different variables on graph using *plot* command.

Figure4: Algorithm for Parsing & Plotting in MATLAB

IV. RESULTS & ANALYSIS

To study the effect on network performance metrics for continuous and discrete time analysis, a network has been designed and implemented in NS2. Further the trace file has been filtered and parsing of filtered file has been done in MATLAB. Table1 below shows timestamps in seconds and throughput in kbps calculated for every 20th event in NS2.

Table1: Discrete-time analysis in NS2

EVENT NO.	INSTANTANEOUS TIME (seconds)	THROUGHPUT (kbps)
1	0	0
21	0.051048	940.30
41	0.089286	1344.01
61	0.135286	1344.01
81	0.173524	1673.73
101	0.211762	1344.01
121	0.247476	1568.01
141	0.285714	1255.30
161	0.334238	1217.40
181	0.369952	1568.01
201	0.40819	1464.51
221	0.446429	1343.97
241	0.492429	1344.01
261	0.530667	1464.51
281	0.568905	1344.01
301	0.604619	1568.01
321	0.642857	1255.30
341	0.691381	1217.40
361	0.727095	1568.01
381	0.765333	1464.51
401	0.803571	1343.97
421	0.849571	1343.97
441	0.88781	1673.68
461	0.926048	1344.01
481	0.961762	1568.01
501	1.01281	1154.07

In the above table, throughput has been calculated for discrete timestamps and it has been found that calculating network parameters at different instants of time using discrete timestamps requires large simulation time because of large size of trace file and also results in larger operational overhead as compared to the calculations done for the continuous intervals of time as shown in Table 2, 3 & 4.

Table2: Throughput for 0.05 seconds interval

CONTINUOUS TIME INTERVAL (seconds)	THROUGHPUT (kbps)
0-0.05	960
0.05-0.10	1280
0.10-0.15	1440
0.15-0.20	1600
0.20-0.25	1280
0.25-0.30	1440
0.30-0.35	1280
0.35-0.40	1440
0.40-0.45	1600
0.45-0.50	1280
0.50-0.55	1440
0.55-0.60	1280
0.60-0.65	1440
0.65-0.70	1600
0.70-0.75	1280
0.75-0.80	1440
0.80-0.85	1280
0.85-0.90	1440
0.90-0.95	1600
0.95-1.00	1600

Table3: Throughput for 0.075 seconds interval

CONTINUOUS TIME INTERVAL (seconds)	THROUGHPUT (kbps)
0-0.075	1173.3
0.075-0.150	1386.7
0.150-0.225	1386.7
0.225-0.3	1386.7
0.3-0.375	1280.0
0.375-0.45	1493.3
0.45-0.525	1386.7
0.525-0.6	1386.7
0.6-0.675	1386.7
0.675-0.75	1280.0
0.75-0.825	1493.3
0.825-0.9	1386.7
0.9-0.975	1386.7

Table4: Throughput for 0.1 seconds interval

CONTINUOUS TIME INTERVAL (seconds)	THROUGHPUT (kbps)
0-0.1	1120
0.1-0.2	1440
0.2-0.3	1360
0.3-0.4	1360
0.4-0.5	1360
0.5-0.6	1360
0.6-0.7	1440
0.7-0.8	1360
0.8-0.9	1360
0.9-1.0	1520

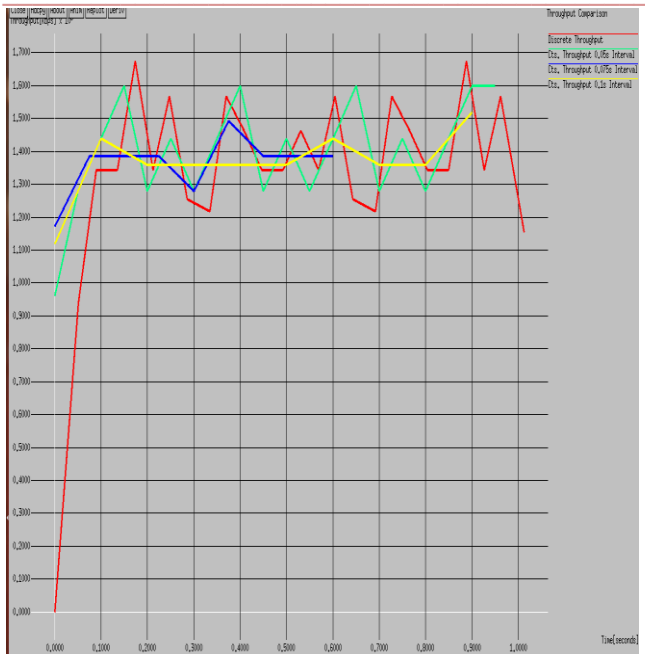

Figure4: Comparison of Throughput for Discrete and Continuous times

From the above analysis, it has been found that the value of average throughput was more in case of continuous time analysis in MATLAB as compared to discrete-time event based analysis in NS2 as also shown in Figure4. It was because of large simulation time of NS2. Moreover, as we decrease the time interval, value of Throughput goes on increasing.

V. Conclusion And Future Scope

From the continuous and discrete time analysis of network in NS2 in this paper, it has been found that there is increased efficiency and improvement in network performance when performance metrics are calculated for continuous intervals of time in MATLAB instead of discrete timestamps in NS2. Continuous time analysis not only saves time and efforts but also aids network survivability. There is less operational overhead and many additional features like statistical analysis of internal state of the network etc. supported by continuous-time analysis of the network. Moreover, these events are specific to this tool only and require thorough understanding, whereas plotting and calculating parameters with respect to time is easy to understand even for novice researchers.

In future, to save time and further improve network survivability, co-simulation of NS2 and MATLAB can be performed. There is a lot of work that can be done by integrating Matlab with NS2 for co-simulation and perform continuous time analysis directly for network survivability without the need of parsing later on.

References

[1]. Dr.N.Sairam & Dr.R.Seethalakshmi, "Parallel Discrete Event Simulation", NPTEL – Computer Science & Engineering – Parallel Algorithms, School of Computing, SASTRA Univeristy, Thanjavur-613401.

[2]. Mirko Stoffers, Sascha Schmerling, Georg Kunz, James Grossz, Klaus Wehrle, "Large-Scale Network Simulation: Leveraging the Strengths of Modern SMP-based Computer Clusters", _Communication and Distributed Systems, RWTH Aachen University, zSchool of Electrical Engineering, KTH Royal Institute of Technology.

[3]. Claudio Cicconetti, Enzo Mingozzi, Giovanni Stea, " An Integrated Framework for Enabling Effective Data Collection and Statistical Analysis with ns-2", Dipartimento di Ingegneria dell Informazione, University of Pisa, Via Diotisalvi, 2-56100 Pisa, Italy.

[4]. Alakiri Harrison. O, Oladeji Florence. A, Benjamin Benjamin. C, Okolie Cletus. C, Okikiola Folasade. M, " The Desirability of Pareto Distribution for Modeling Modern Internet Traffic Characteristics", Department Computer Technology, University of Lagos, Nigeria.

[5]. Garimidi Hareesh, "Dynamic Modeling of Routing Protocol using Colored Petri Nets", Master's Thesis, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, 769008, India.

[6]. Jörg Micheel[1,2], Stephen Donnelly[1], Ian Graham[1], "Precision Timestamping of Network Packets", [1]WAND, Computer Science, The University of Waikat Hillcrest Road, Gate 8,Hamilton, New Zealand, [2]NLANR MOAT, SDSC, USC 10100, John Hopkins Dr La Jolla, CA 92093-0505.

[7]. Deepesh Dewangan, Ms.Manisha Rajpoot, "Performance Analysis Between Flooding and Directed Diffusion Protocol", Deptt. Of Computer Science & Engineering, RCET, Bhila, Chhattisgarh, Bhilai – India, ISSN: 2278-7844.

[8]. Behdad Jamshidi, Victor Mateescu, "VoIP in NS2 Simulation", Ensc 427 Communication Networks Spring 2012.

[9]. Chane, Mekides Abebe, Tekelmariam, Hailay Mezgebo, "Evaluation of Smartphones Network Performance", Master Thesis, Electrical Engineering, April, 2012.

[10]. Guosong Tian, Colin Fidge and Yu-Chu Tian, "Hybrid System Simulation of Computer Control Applications over Communication Networks", School of Information Technology, Queensland University of Technology,Brisbane QLD 4001, Australia.