

Enhancing the NS-2 Traffic Generator for the MANETs

¹Assist Prof. Dr. Saad Talib Hasson, ²Ahmed Jawad Kadhim, ³Zainab Saad Talib
^{1,2}Iraq- University of Babylon- College of Sciences- Computer Science Department
³University of Babylon- College of Sciences

Abstract: The network simulator (NS-2) structure is containing a file called "cbrgen". This file was used to generate the traffic rates by generating random connections between source nodes and destinations. The original "cbrgen" file generates only CBR and TCP traffics. It generates connections in random manner between 0 and 180s only. The purpose of this paper is to enhance and modify this file by rebuilding it to generate additional types of traffics such as exponential on/off and Pareto on/off, as well as adding the simulated application "Telnet" that can be implemented easily with NS-2. The generated connections was also developed by letting the time to be distributed randomly between 0 and the simulation time instead of the fixed time between 0 and 180s. All these modification were tested and implemented many times. Such modifications will make this simulator more practical and close to the real MANETs applications.

Keywords: MANET, traffic generators, simulations, performance metrics.

I. Introduction

MANET is a group of mobile nodes that work as a host and router without any infrastructures. The topology of MANET is changed frequently due to the movement of nodes. The network simulator version 2 (NS-2) represents a good tool to simulate this type of networks. The process of determining the traffic generator for a large number of hosts is too difficult and it may takes large time. Therefore, the CBR file was supported by NS-2 as a solution to perform this task. This file creates two types of traffic generators (TCP and CBR) and schedule them randomly based on the uniform distribution. This script file named "cbrgen" can be found under the following directory in the Linux environment "ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen", (ns-2.34 is the version of the NS-2 [1]).

In order to create new traffic generation between nodes, the user must first follow "ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen" directory and then write the following instruction in the Linux terminal [2].
`ns cbrgen.tcl -type <traffic_type> -nn <number> -seed <number> -mc <number> -rate <number> > file_name`

Traffic Generator

It models the behavior of the user depending on the predefined schedule. In NS-2 there are four types of traffic generators:

a. Exponential On/Off

During an on period, it sends fixed size payload for every randomized interval to an attached agent, but it stops sending during an off period. On and off periods are exponentially distributed. [3].

b. Pareto On/Off

It is similar to the exponential On/Off traffic generator, but it differs in that the durations of on and off periods which are Pareto distributions [3].

c. Constant Bit Rate (CBR)

This type sends payload of fixed size to the attached agent. By default, the interval between two successive payloads is fixed, but it can be optionally randomized by set the value of random parameter to 1 [4].

d. Trace Traffic

This type depends on the trace file from existing simulation where it selects starting place randomly in this trace file and begins generate traffic from it [4].

Simulated Applications

Simulated applications act as an actual application running and not depend on the predefined schedule for payload generation. FTP and Telnet are types of the simulated applications.

a. File Transfer Protocol (FTP)

It is a protocol which divides a given file by the source node into small pieces and transfers them to a destination node. Not as FTP in practice, in NS-2 the FTP module does not depend on an input file. It informs an attached sending transport layer agent of a file size in bytes. When receiving the demand of the used, the agent creates packets which can accommodate the file and forwards them to a connected receiving transport layer agent [3].

b. Telnet

It is an interactive application based on the client-server text. A Telnet client logs on to a server, and sends text messages to the server. The server will execute the received message and returns the result to that client. In other word, Telnet data traffic is created in response to user demand and not depends on a predefined schedule. However, NS-2 models a Telnet application in the same way as it does for traffic generators: sending a fixed size packet for every randomized interval [3].

II. Performance Metrics

There are many performance metrics used to evaluate the behavior of the simulated MANETs. In this paper only the following metrics were suggested to be used.

a. Throughput

Throughput represents the amount of data received by the destination nodes in certain period of time [5].

b. Average end-to-end delay (average E2E delay)

It represents the time that spent by the packet to reach to the destination.

$$E2E\ delay\ [packet_id] = received\ time\ [packet_id] - sent\ time\ [packet_id]$$

The average end-to-end delay can be calculated by summing the times taken by all received packets divided by their total numbers [6].

c. Packet delivery fraction (PDF)

PDF can be measured as the ratio of the received packets by the destination nodes to the packets sent by the source node [7].

$$PDF = (number\ of\ received\ packets / number\ of\ sent\ packets) * 100.$$

III. Modifying the Traffic Generator

The traffic generation file in NS-2 is the "Cbrgen". This file has an important role in each simulation scenario. the "Cbrgen" file plays an important role in reducing the required effort and time to build the traffic patterns. It also creates these traffics in random manner to assist the users in their simulation approaches. Due to our long and continuous dealing with NS-2, we found that it is useful to modify its behavior to exceed and achieve the new requirements of the MANETs applications. The proposed required modifications of this file can be done in two parts: (1) is to modify the scheduling time of this file and, (2) is to add a new ability for this generator to generate other near real traffic types in future simulation applications.

(1). Modify the Time

The old traffics were appeared at a uniformly distributed time between 0 and 180s. This is not practical in many simulation applications, because in many cases the time of the simulation is required to be smaller/ greater than 180s, so many traffics will become not useful when it was be scheduled with the simulation time. For this reason, we propose to modify this file and make it generates the traffics between 0 and the time of the set simulation time instead of 180s. The following instruction is used to generate the new traffic by the modified file:

```
ns cbrgen.tcl -type <traffic_type> -nn <number> -seed <number> -mc <number> -rate <number> -time <simulation time> > file_name
```

We add a new parameter (-time) to the old instruction. This parameter is used to match with the simulation time that used in the "tcl" file as well as it can be used to generate the suitable scenario files.

(2). Adding Traffic Generator and Application

In this section, we will present other modifications for the "cbrgen" file by adding two real types of the traffic generator which are the exponential On/Off and pareto On/Off. These generators are found and implemented in the network simulator NS-2 but the original "cbrgen" file was not assist them. As well as we added the one important simulation application (Telnet). The purpose of this modification is to help the researchers and NS-2 users in dealing with these random traffics and applications manner instead of writing or building these traffics manually. The manual process may not simulate the real or exact behavior of the wanted

distribution of the required traffic in the designed MANET. The following instruction is used to generate the required or designed traffic by the modified file:

```
ns cbrgen.tcl -type <agent_type> -traffic <traffic_type> -nn <number> -seed <number> -mc <number> -rate <number> -time <simulation time> > file_name.
```

In the above instruction a new parameter (-traffic) was added. The users must distinguish between the traffic generators used with UDP and the applications used with TCP. Where:

-type: means TCP or UDP and ,

-traffic: means (CBR, Exponential On/Off, and Pareto On/Off uses with UDP, (FTP or Telnet use with TCP).

Simulation Environment

In order to implement and apply these modifications, the following MANET simulation environment was suggested and evaluated. The used traffic rates in this simulation scenario will be randomly distributed as Exponential on/off , pareto on/off and Telnet.

Table (1) shows the suggested simulation environment of the MANET with the application of the new developed random traffics.

Table (1): suggested simulation environment parameters.

Parameter	Value
The simulator	NS-2.34
MAC	802.11
Propagation model	Two ray ground
Routing protocols	AODV
Simulation time	75s
Traffic generation	Exponential on/off, pareto on/off, Telnet
Antenna	Omni Antenna
Max connection	8
Transition rate	64 kb/second
Mobility model	Random way point model
Seed	1-50
Window size for Telnet application	32
Nodes number	20
Speed	10 m/s
Pause time	4s
Simulation area	1000m*1000m

IV. Simulation Results

For each traffic generator type, 100 traffic files were built (50 files based on the old "cbrgen" file, while the other 50 files are based on the modified "cbrgen" file) with different seeds. The maximum connections number was set to be "8" and the sending rate in this paper was set to be "64 kb/s".

Figure1 represents the throughput of exponential traffic before and after changing the scheduled time. Figure2 represents the average end-to-end delay of exponential traffic before and after changing the scheduled time. Figure3 represents the packet delivery fraction of exponential traffic before and after changing the scheduled time.

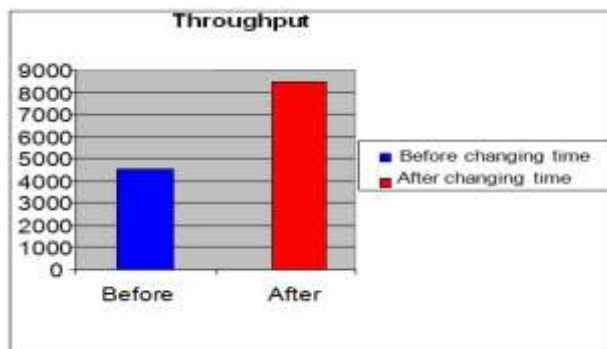


Fig1: The throughput of the exponential traffic.



Fig2: The average end-to-end delay of the exponential traffic.

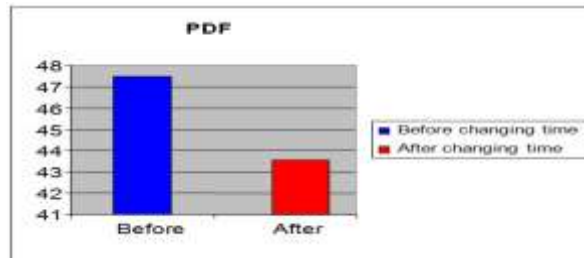


Fig3: The packet delivery fraction of the exponential traffic.

Figure4 represent the throughput of the Pareto traffic before and after changing the scheduled time.
 Figure5 represent the average end-to-end delay of the Pareto traffic before and after changing the scheduled time.
 Figure6 represent the packet delivery fraction of the Pareto traffic before and after changing the scheduled time.

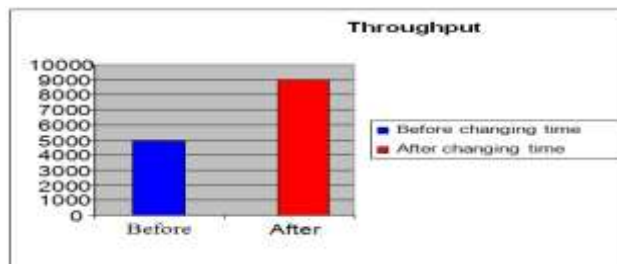


Fig4: The throughput of the Pareto traffic.

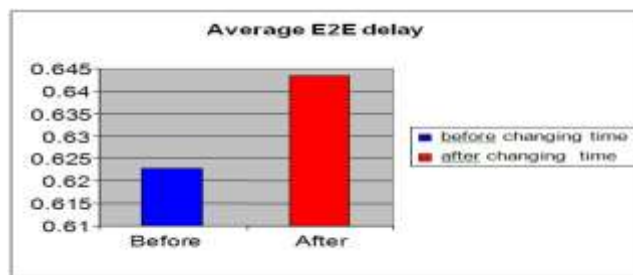


Fig5: The average end-to-end delay of the Pareto traffic.

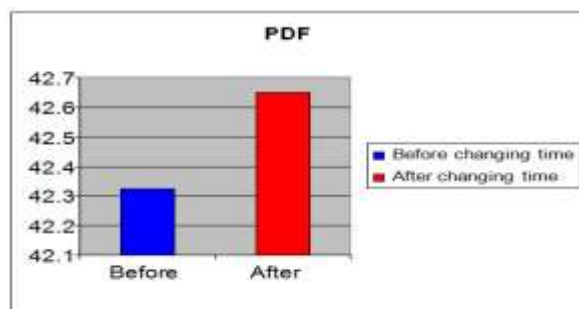


Fig 6: The packet delivery fraction of the Pareto traffic.

Figure 7 represents the throughput of the Telnet application before and after changing the scheduled time.

Figure 8 represent the average end-to-end delay of the Telnet application before and after changing the scheduled time.

Figure 9 represent the packet delivery fraction of the Telnet application before and after changing the scheduled time.

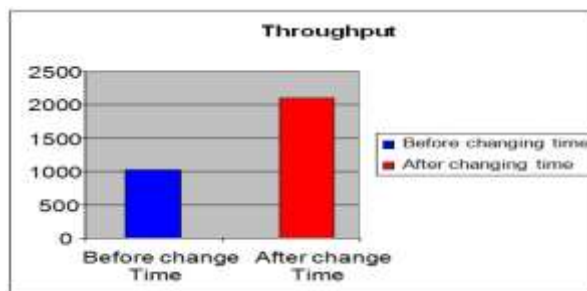


Fig 7: The throughput of the Telnet application.

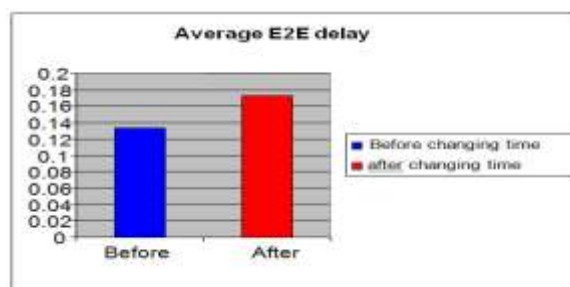


Fig 8: The average end-to-end delay of the Telnet application.

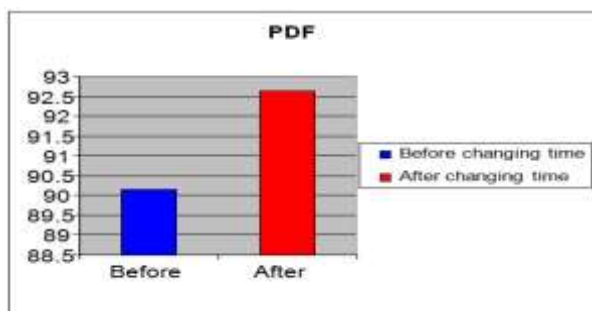


Fig9: The packet delivery fraction of the Telnet application.

V. Conclusion

From the modified "cbrgen" file, we conclude that the users will be able to schedule many traffic generators easily in an optimal manner because all the traffics will begins and finished through the assigned simulation period. This process will prevent the traffic generators to operate out of the simulation period.

The new users and MANETs designers will be able to design wide ranges of traffics depending on many distributions which helps in improving the behavior of the real and applicable MANETs.

References

- [1] Lee Kok Thong, "Performance Analysis of Mobile Ad hoc Networking Routing Protocols", MSc. Thesis, Naval Postgraduate School, Monterey, California, 2004.
- [2] Kevin Fall and Kannan Varadhan, "The ns Manual (formerly ns Notes and Documentation)", 2010.
- [3] Teerawat Issariyakul and Ekram Hossain, "Introduction to Network Simulator NS2", Springer, ISBN: 978-0-387-71759-3, 2009.
- [4] Björn Wiberg, "Porting AODV-UU Implementation to ns-2 and Enabling Trace-based Simulation", MSc. thesis, Information Technology, Department of Computer Systems, Uppsala University, Sweden, 2002.
- [5] Ducksoo Shin, Jonghyup Lee, Jaesung Kim, and Jooseok, "A²OMD : AN Adaptive Ad Hoc On-Demand Multipath Distance Vector Routing Protocol Using Dynamic Route Switching", Journal of Engineering Science and Technology, Vol. 4, No. 2, 171 – 183, 2009.
- [6] Aliff Umair Salleh, Zulkifli Ishak, Norashidah Md. Din, and Md Zaini Jamaludin, "Trace Analyzer for NS-2", IEEE, pp. 29-32, Student Conference on Research and Development (SCORED), Malaysia, 2006.
- [7] Er. Deepinder Singh Wadhwa and Er. Tripatjot Singh Panag, "Performance Comparison of Single and Multipath Routing Protocols in Adhoc Networks ", Int. J. Comp. Tech. Appl., ISSN: 2229-6093, Vol 2 (5), 1486-1496, 2011.