



New Developed Traffic Generator for NS-2

SAAD TALIB HASSON¹ and MOHAMMAD H. HUSAIN²

¹Head of Computer Science Department, College of Sciences, University of Babylon, Iraq.

²MSc. Student, University of Babylon, Iraq.

(Received: August 30, 2012; Accepted: June 04, 2011)

ABSTRACT

In this paper, a new version of the traffic generator script "cbrgen.tcl" to the Network Simulator (NS-2) was developed and applied in many simulation experiments. The main objective of this paper was to study the impacts of varying the transmitted (sent) data rate for the CBR and TCP traffic under varying number of nodes. The performance of MANETs traffic pattern was evaluated with respect to the quantitative metrics such as PDF, packet loss, throughput, end to end delay and NRL. As a result of this study with the comparative performance between the different data rates of CBR and TCP traffic over varying number of nodes shows a better performance in smaller rates.

Key words: MANET, Performance Metrics, Traffic Generator, TCP, CBR.

INTRODUCTION

A Mobile Ad Hoc Network is a collection of wireless mobile nodes that can be set up anywhere, anytime and connected via wireless link without using an existing or any fixed infrastructure. It represents a dynamic distributed system of arbitrarily moving wireless devices. Each of the nodes has a wireless interface and communicates with each other over a radio transmission range¹. In MANET, the nodes are free to move randomly while it communicates and can be connected to each other randomly as well as organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably². A MANETs might consist of several home-computing devices, including cellular phone, PDAs, laptops, MP3 player, notebooks, handheld PCs and so on. These nodes can be located in cars, ships, airplanes as well as other small personal electronic devices. Each node

will be able to communicate directly with other nodes that reside within its transmission range. For communicating with nodes that reside beyond this range, the node needs to use intermediate nodes to relay messages hop by hop³.

Traffic carried by future wireless networks is expected to be a mix of real-time traffic such as voice, multimedia conferences and games, and data traffic such as web browsing, messaging and files transfer. All of these applications will require widely varying and diverse Quality of Service (QoS) guarantees for the different types of offered traffic⁴.

The most expected traffic type in the networks is the traffic of the http applications. The web traffic is an incoming traffic for users, e.g. downloading requested html pages with different sizes from http servers in the Internet. The other general services in the Internet are FTP, real-time

communications (based on the CBR traffic) and email applications. The CBR server is used for generation of real-time streams such as video or speech traffic. The FTP server contains some data files of different sizes, these can be downloaded by users; every http server has a html page pool, users access html pages from these pools; the email server is used to receive the outgoing email traffic from users⁵.

Related Works

This section presents some of researchers that their works are related to this thesis.

Priti Garg and Asma Tuteja at 2011 has been evaluated a comparison between the performance of DSR and TORA routing protocols under CBR traffic through changing the time interval between the packets sent. Variation of time interval has done under 0.005, 0.010, 0.015, 0.020 and 0.025 sec. They used the performance metrics: throughput, average delay, PDF and routing load⁶.

Dong Sun and Hong Man at 2001 has been studied the performance of two routing protocols (i.e. DSR and AODV) based on TCP traffic flows (window size of TCP traffic). TCP window sizes of 5, 10 and 50 packets are used with 1, 5 and 10 TCP connections respectively. The performance comparisons based on several important performance metrics likes throughput, goodput and normalized routing overhead. The performance simulations show that AODV work better in a large window size and this need further improvements to provide better performance for TCP flow transmission⁷.

The effect of mobility i.e. speed (0 to 20 m/s) and pause time (0, 100, 200, 300, 400, 500 and 600) was evaluated by (Ronald Beaubrun and Badji Molo) at 2010. Traffic type (CBR, VBR) and traffic intensity (10, 20, 30 and 40 source) on the performance of MANET routing protocol i.e. DSR⁸.

Routing Protocols in MANETs

A routing in MANET was defined as a convention or standard that controls how nodes come to agree which way to route packets between computing devices in a MANET⁹. The routing protocol that used in this simulation study is AODV protocol.

Traffic Types in MANETs

The traffic carried over a MANET may have different characteristics. In this paper, two types of traffic were described in MANET network; Constant Bit Rate (CBR) traffic and File Transfer Protocol (FTP) traffic.

Constant Bit Rate (CBR)

Constant bit rate models generate packets at a fixed rate. The interval between two payloads (i.e., the sending rate) is fixed, but it can be optionally randomized. Applications such as digital non-compacted voice, audio and non-compacted video are typical examples of constant bit rate traffic. Implementation ease is the advantage of this traffic model, since inputs for the traffic generator are just one value for the packet size and one value for the interval between packets. The main drawback of this model is the fact that real applications normally vary its data injection rates¹⁰. The CBR traffic is used UDP as the transport layer because they do not require congestion management and reliability¹¹.

File Transfer Protocol (FTP)

FTP is a protocol which divides a given file into small pieces and transfers them to a destination host, that's means it used to simulate file transfer between two end points. The traffic usually uses TCP as the transport layer, which performed the appropriate congestion control and transmits the data reliably¹¹.

Mobility and Traffic Generation Model

It is easy to create and control mobility and traffic simulations for small number of nodes manually. Methods or algorithms are required to randomly implement and control mobility and traffic simulations for large number of nodes in a changed topology. This section can be divided into two subsections. The first subsection talks about CMU's node-movement generator "setdest" which can be used to create random waypoint node movements for mobile nodes. While the second subsection will deals with the traffic-pattern generating script. The main focus of this paper is on the traffic generator.

Traffic Generation Model

NS-2 provides traffic scenario generator script called "cbrgen.tcl" which was written in Tcl language. This script is used to create random traffic

connections of TCP and CBR. The “cbrgen.tcl” script unlike the mobility generates script and its original the format is as follows:

```
>>ns cbrgen.tcl [-type cbr/tcp] [-nn num_nodes] [-seed seed] [-mc max_conn] [-rate rate]>
[scenario_output_file]
```

Where type is used to define the type of traffic connection (CBR or TCP), nn is number of mobile nodes, seed is a random seed, if it not equal to 0, the traffic pattern will reappear if all the other parameters are the same, mc represents the maximum number of connections to be setup between the nodes, rate is the used packet sending rate for CBR traffic (The inverse value of the rate is used to represent the average interval time between two successive packets) and the scenario_output_file means the output name of traffic script¹¹.

In order to understand how the “cbrgen.tcl” script operates, one must first understand how CBR traffic and TCP traffic setting up connections between the mobile nodes. The following examples are developed to help explaining the use of this script:

Example 1

To create a CBR connection file between 30 nodes, having 25 as maximum connections, with a seed value of 1.0 and a rate of 2.0. Using the prompt type:

```
>>ns cbrgen.tcl -type cbr -nn 30 -seed 1.0 -mc 25 -rate 2.0 > cbr_traffic.
```

One segment of the resulted script of the cbr traffic generation output is shown below:

```
# nodes: 30, max connection: 25, Interarrival: 0.5, seed: 1.0
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.5
$cbr_(0) set random_ 1
```

```
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 “$cbr_(0) start”.
# 28 connecting to 29 at time 47.128346453946243
#
set udp_(24) [new Agent/UDP]
$ns_ attach-agent $node_(28) $udp_(24)
set null_(24) [new Agent/Null]
$ns_ attach-agent $node_(29) $null_(24)
set cbr_(24) [new Application/Traffic/CBR]
$cbr_(24) set packetSize_ 512
$cbr_(24) set interval_ 0.5
$cbr_(24) set random_ 1
$cbr_(24) set maxpkts_ 10000
$cbr_(24) attach-agent $udp_(24)
$ns_ connect $udp_(24) $null_(24)
$ns_ at 47.128346453946243 “$cbr_(24) start”
#
#Total sources/connections: 17/25
The last lines in cbr_traffic file shows how many actual nodes are used for the source and how many connections are created. From this run one can find that the number of source is not equal to number of connections and the simulation time for the CBR connections are randomly generated with a maximum value of 180 second.
```

Example 2

TCP connection files can also be created using “type” as tcp. Using the following prompt type:

```
>>ns cbrgen.tcl -type tcp -nn 30 -seed 1.0 -mc 25 -rate > tcp_traffic
```

The output of the tcp_traffic file will be:

```
# nodes: 30, max connection: 25, Interarrival: 0.0, seed: 1.0
#
# 1 connecting to 2 at time 2.5568388786897245
#
set tcp_(0) [$ns_ create-connection TCP $node_(1) TCPSink $node_(2) 0]
$tcp_(0) set window_ 32
$tcp_(0) set packetSize_ 512
set ftp_(0) [$tcp_(0) attach-source FTP]
$ns_ at 2.5568388786897245 “$ftp_(0) start”.
# 28 connecting to 29 at time 47.128346453946243
#
set tcp_(24) [$ns_ create-connection TCP $node_(28) TCPSink $node_(29) 0]
$tcp_(24) set window_ 32
```

```
$tcp_(24) set packetSize_ 512
set ftp_(24) [$tcp_(24) attach-source FTP
$ns_ at 47.128346453946243 "$ftp_(24) start"
#
#Total sources/connections: 17/25
```

Enhancement of the Traffic Generator

Long repeated study was performed by this paper authors on MANET simulations using NS-2 to generate variable random traffic in different applications. It required deep interpretation of the algorithm steps used to generate the traffic by the build traffic generator script in NS-2. Such detailed study indicated the following limitations and weaknesses in the building program of the original traffic generator script (cbrgen.tcl).

- The time at each connection is uniformly distributed and fixed to be between 0 and 180.
- The results were repeated for each execution and the times per the connections are the same in all the executions.
- Always, the flow node or SDP is not equal the mc value.
- The original cbrgen.tcl script was made to generate specific number of connections between the source and the destination.

This paper will try to improve and develop a new version of the traffic generator script that can exceed these limitations and achieve certain reliable applicable improvements. The suggested enhancement was done by adding more required information's as shown in the following steps:

- Add w value to represent the window size of the TCP traffic.
- Add pktsize value to represent the packed size for TCP and CBR traffic.
- Changing the time value to be depended on the generated time of the random numbers (which was previously built to take the value between 0 and 180s). The enhancement is to let the time depends on the entered value by the user in the tcl script.
- Make the mc value express about number of max connections and also number of flow nodes that exchanged the name to SDP in the traffic generator script (cbrgen.tcl).

The developed format of traffic generator script will be:

```
>>ns cbrgen.tcl [-type cbr/tcp] [-nn num_nodes] [-seed seed] [-SDP source-destination-pairs and connection] [-rate rate_for_CBR] [-w window_size_TCP] [-pktsize packe_size] [-simtime simulation_time ] > [scenario_output_file]
```

Example 3

To create a CBR connection file between 30 nodes, having maximum connections of 25, with a seed value of 1.0, traffic rate of 2.0 kbps, the CBR packet size is 512 bytes and simulation time is set to be 300 seconds. The prompt type is:

```
>>ns cbrgen.tcl -type cbr -nn 30 -seed 1.0 -SDP 25 -rate 2.0 -w 32 -pktsize 512 -simtime 300 > cbr_traffic_new
```

The following code of the resulted traffic generator shows one segment of the new developed CBR traffic generation output called (cbr_traffic_new):

```
# nodes: 30, SDP and max connection: 25,
Interarrival for CBR: 0.5, seed: 1.0, packet size:512,
SimTime: 300.0 s
#
# 26 connecting to 27 at time 145.51421154547214
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(26) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(27) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.5
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$udp_(0) set fid_ 2
$ns_ at 145.51421154547214 "$cbr_(0) start"
# 3 connecting to 4 at time 114.85893470927093
#
set udp_(24) [new Agent/UDP]
$ns_ attach-agent $node_(3) $udp_(24)
set null_(24) [new Agent/Null]
$ns_ attach-agent $node_(4) $null_(24)
set cbr_(24) [new Application/Traffic/CBR]
$cbr_(24) set packetSize_ 512
$cbr_(24) set interval_ 0.5
$cbr_(24) set random_ 1
$cbr_(24) set maxpkts_ 10000
```

```

$cbr_(24) attach-agent $udp_(24)
$ns_ connect $udp_(24) $null_(24)
$udp_(24) set fid_ 2
$ns_ at 114.85893470927093 "$cbr_(24) start"
#
#Total sources/connections: 25/25
#

```

The last lines in `cbr_traffic_new` file shows how many actual nodes are used for the source and how many connections are created. From the above we found that the numbers of sources are equal to the number of connections and the simulation time for the CBR connections are randomly generated with a maximum value set at 300 second.

Example 4

For the TCP traffic generation, the following form of the prompt can be used:

```

>>ns cbrgen.tcl -type tcp -nn 30 -seed 1.0 -SDP 25
-rate 2.0 -w 32 -pktsize 512 -simtime 300 >
tcp_traffic_new.

```

While the following code of the traffic generator result shows one segment of the tcp traffic generation output (`tcp_traffic_new`):

```

# nodes: 30, SDP and max connection: 25, window
size for TCP: 25 byte , seed: 1.0 , packet size:512 ,
SimTime: 300.0 s
#
# 2 connecting to 3 at time 49.118057381882359
#
set tcp_(0) [$ns_ create-connection TCP $node_(2)
TCPSink $node_(3) 0]
$tcp_(0) set window_ 25
$tcp_(0) set packetSize_ 512
set ftp_(0) [$tcp_(0) attach-source FTP]
$ns_ at 49.118057381882359 "$ftp_(0) start"
#
# 15 connecting to 16 at time 249.91520589679257
#
set tcp_(24) [$ns_ create-connection TCP
$node_(15) TCPSink $node_(16) 0]
$tcp_(24) set window_ 25
$tcp_(24) set packetSize_ 512
set ftp_(24) [$tcp_(24) attach-source FTP]
$ns_ at 249.91520589679257 "$ftp_(24) start"
#
#Total sources/connections: 25/25
#

```

This script can be used in a reliable manner to ensure generating connections between the source and the destination nodes that not exceeds the designed max connections value.

Performances Metrics

Performance metrics represents certain statistical measures of data routing performance¹². There are set of metrics that were used to evaluate the simulation performance of the MANETs network. The following metrics were suggested to be used in this study to measure the affects of the traffic pattern on the performance of the MANET network.

Packet Delivery Fraction (PDF)

PDF is the ratio between number of the received data packets successfully at the destinations and the number of data packets sent by the traffic sources as shown by equation (1).

$$PDF = \frac{\sum_{i=1}^n Packet_Receive(i)}{\sum_{i=1}^m Packet_Send(i)} \times 100 \quad \dots(1)$$

Where n is the number of received packets, and m is the number of sent packets^{13, 14}.

End to End Delay

The delay time is the spent time from the beginning of a packet transmission at a source node until received it by a destination node. The delay time in this study was measured in milliseconds (ms). It can be calculated by equation (2).

$$Delay = \left[\sum_{i=1}^n (Packet_Receive_Time) - (Packet_Sent_Time) \right] \quad \dots(2)$$

Where n is number of the received packets¹³.

Throughput

The throughput was defined as the total number of packets received by the destination per time unit that is delivered from one node through the channel. The throughput is measured in bps and the it is calculated by equation (3).

$$Throughput = \frac{\sum_{i=1}^n (number\ of\ delivered\ packet(i)) * 8}{total\ duration\ of\ simulation} \quad \dots(3)$$

Where n is the packets received¹⁵.

Packet Loss

The packet loss or drop is the failure of one or more transmitted packets to arrive at their destination. The packet loss is calculated by equation (4) ¹⁶:

$$\text{Packet Loss} = \text{No. of Packet Send} - \text{No. of Packet Received} \quad \dots (4)$$

Normalized Routing Load (NRL)

The normalized routing load is defined as the ratio of all routing control messages such as RREQ, RREP, RRER, HELLO and etc, by every node over the number of received data packets successfully at the destination nodes^{11,17}. The NRL is calculated by equation (5).

$$NRL = \frac{\sum_{i=1}^k \text{Routing_Packet}(i)}{\sum_{i=1}^n \text{Packet_Received}(i)} \quad \dots (5)$$

Where k is the number of routing packets and n is the number of received packets.

Simulation Environments

The Network simulator NS-2 version 2.34 was used to evaluate the performance of the traffic patterns in MANET routing protocol. NS-2 was used to simulate the events like: sending, receiving, forwarding and dropping packets. The simulation

results were done in LINUX environment (debian 5). NS-2 can simulate a layered network from the physical radio transmission channel to high-level applications [6]. The channel characteristics and the parameters that related to the network environment must be defined firstly. The other characteristics of the channel are shown in table 1:

The node mobility movements in this simulation experiments were adjusted according to "Random Way-Point Model"; it includes the speed and the pause time. The speed of the node refers to the average node movement speed in the simulation area. The CBR traffic source was suggested to provide varying values of transmission rates of 2 to 25 kbps, also TCP traffic source was suggested to provide varying values of window sizes of 2 to 25 bytes. The changing of the data rate for CBR traffic and window size for TCP traffic was studied to show their impacts on the performance of the traffic pattern in the MANET. The CBR and TCP traffic source were suggested to provide a constant value data packets of 512 bytes. The suggested network parameters are shown in Table 2.

Simulation Results and Analysis

A set of experiments were done and repeated in order to evaluate the performance of MANET traffic. Certain values were kept constant in every simulation run. In this paper, 26 different simulation scenarios were performed. These

**Table 1: Simulation Parameters
(Channel Parameters)**

Parameter	Value
Transmission Range of each Node	550 m
Propagation Model	Two-Ray Model
Channel Type	Wireless Channel
Network Interface Type	Phy/ WirelessPhy
Interface Queue type	Queue/Drop tail
Link Layer	LL
Antenna	Antenna / Omni
Maximum Packet In IFQ	50
MAC Protocol	IEEE 802.11
Routing Protocols	AODV

Table 2: Network Parameters

Parameter	Value
Simulation Area	600 × 600 Meter
Packets Size Channel Nodes	512 Byte
Placement	Random
Mobility Model Random	Waypoint Model
No. of Nodes	10,20,30,40,50,60,70
Pause Time	10 Seconds
Send Rate for CBR Traffic	(2/5/8/16/20/25) kbps
Window Size for TCP Traffic	(2/5/8/16/20/25) b
Nodes Speed	20 Meter/Second
SDP	25 Source
Simulation Time	300 sec.

scenarios were used to test the effects of varying data rate for CBR traffic and TCP traffic on the performance of the MANETs. The Performance comparison of TCP and CBR traffic patterns at varying data sending rate under varying number of nodes. The used performance metrics are: PDF, drop packets, throughput, average delay and NRL. The following results were recorded.

Packet Delivery Fraction

Fig. 1 shows the PDF of CBR traffic for five different rates under varying number of nodes.

The behavior of PDF with different CBR traffic rates under variable number of nodes is seemed to be decreased with increasing number of nodes.

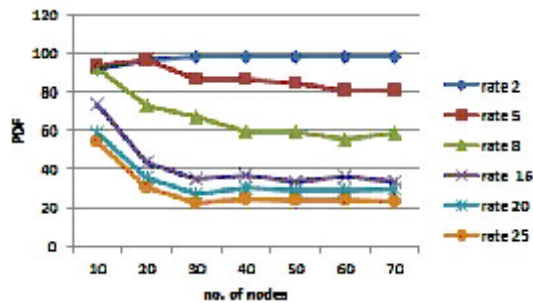


Fig. 1: PDF results with different CBR traffics and variable no. of nodes

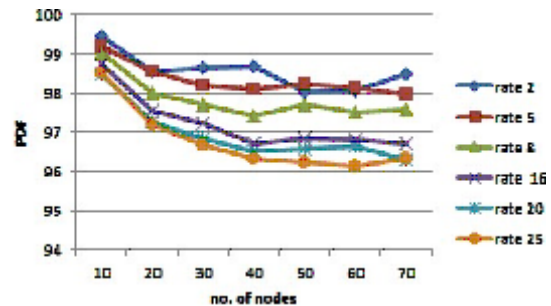


Fig. 2: PDF results with different TCP traffics and variable No. of nodes.

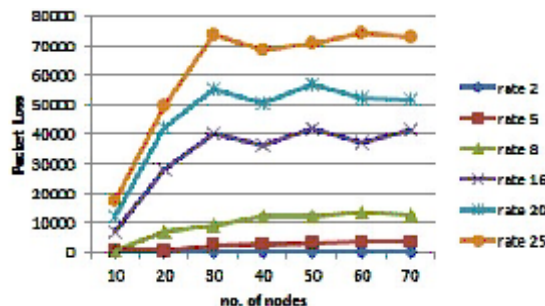


Fig. 3: packet loss with different CBR traffic rates and variable No. of nodes

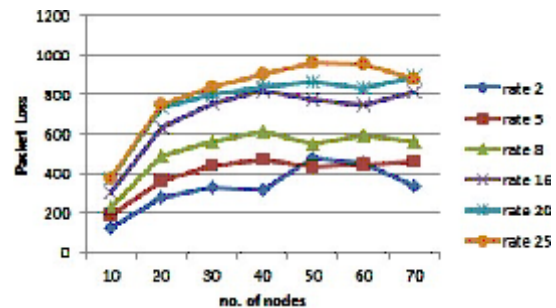


Fig. 4: packet loss with different TCP traffic rates and variable no. of nodes.

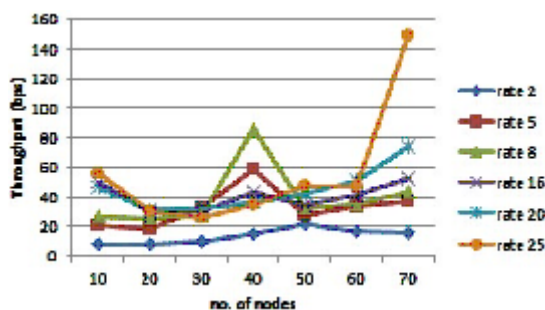


Fig. 5: CBR result for throughput with no. of nodes.

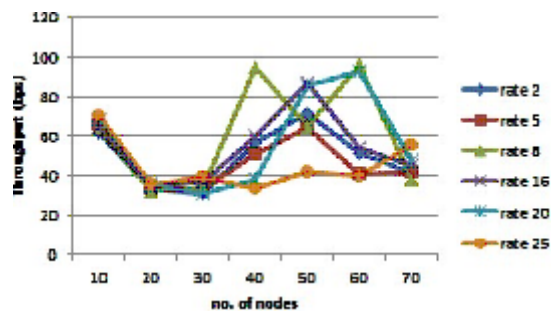


Fig. 6: TCP result for throughput with no. of nodes

Fig. 2 shows the PDF of TCP traffic for five rates under varying number of nodes.

The behavior of PDF with different TCP traffic rates and variable number of nodes is seemed to be decreased gradually with increasing number of nodes.

Packet Loss

Fig. 3 shows the packet loss of five CBR traffic rates with varying number of node.

The performance of the packet loss with different CBR traffic rates and varying number of nodes are seemed to be better at lower rates values. When the network size increased the performance of CBR traffic is increasing gradually. The highest drop packet value of 73817 packets for the performance of CBR traffic at rate 25. when the number of nodes are 30, while the lowest drop packet value is 131 packets at rate 2 when the number of nodes are 20.

Fig. 4 shows the packet loss of five TCP traffic rates with varying number of nodes.

The performance of the packet loss with variable TCP traffic rates and variable number of nodes are seemed to be better at lower rates values. The lowest drop packet value of 124 packets at rate 2 when number of nodes are 10 while highest drop packet value of 964 packets at rate 25 and number of nodes are 50.

Throughput

Fig. 5 shows the throughput of CBR traffic for different rates with varying number of nodes.

The performance of CBR traffic at rate 8 having the higher throughput value of 86.1 bps when the number of node is 40. While the throughput for the rate 2 performed a less value in the network.

Fig. 6 shows the throughput of TCP traffic for different rates with varying number of nodes.

The Throughput performance of different TCP traffic rates initially shows high value initially and decreases with increasing the number of nodes to certain level and then started increasing again. The best result of the throughput at the rate 8 when

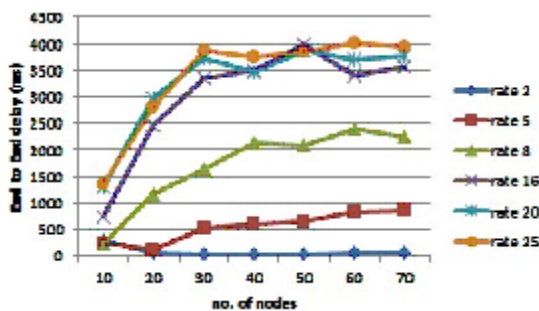


Fig. 7: End to end delay for the CBR with no. of nodes

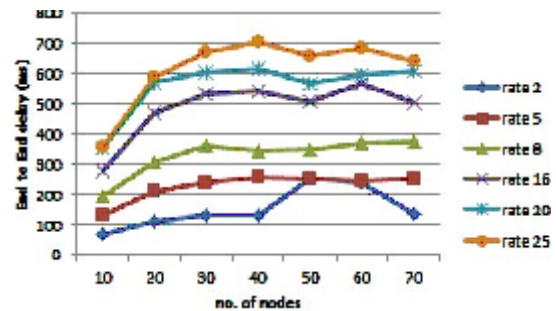


Fig. 8: End to end delay for the TCP with no. of nodes

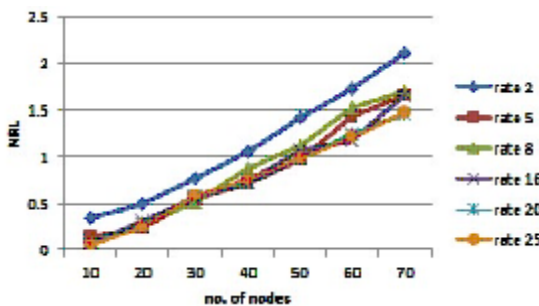


Fig. 9: NRL for the CBR with no. of nodes

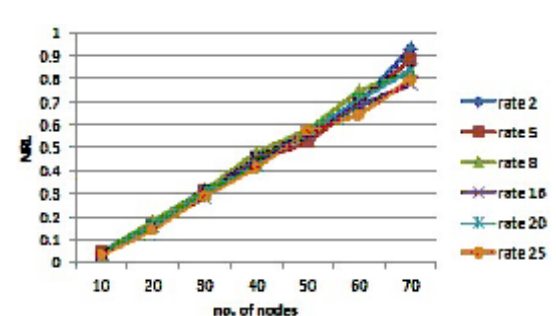


Fig. 10: NRL for the TCP with no. of nodes

numbers of nodes are 40 and 70. There are good performance at rates 2, 5, 16 and 20.

End to End Delay

Fig. 7 shows the end to end delay of CBR traffic for different rates under varying number of nodes.

The performances of the delay with the CBR traffic at lower rates and varying number of nodes have less value than other rates. When the number of nodes increased, the performance of CBR traffic at rate 2 has the best delay value.

Fig. 8 shows the end to end delay of the TCP traffic for different rates under varying number of nodes.

The performance of TCP traffic at lower rates with varying number of nodes having delay less than other rates. It has the less delay at rate 2 compared with other rates. When the number of nodes increased, the delay will increased gradually for all rates.

Normalized Routing Load

Fig. 9 shows the NRL of CBR traffic for different rates under varying number of nodes.

NRL value is increased with increasing number of nodes for different rates. The large NRL value performed in this network at rate 2.

Fig. 10 shows the NRL of TCP traffic for different rates under varying number of nodes.

The NRL values shows approximately similar increasing at all rates with increasing number of nodes.

CONCLUSIONS

In this simulation work, the transport layer behavior regarding TCP and CBR protocols was evaluated. Several simulation scenarios in NS-2, over different network topologies and data flow patterns were carried out. The results of both TCP and CBR traffic on various data sending rate have been analyzed. The performance metrics to evaluate the performance of TCP and CBR traffic pattern includes PDF, packet loss, throughput, delay, NRL for MANET. In this paper, an enhancement of the traffic generator script in "NS-2" and carried out in an efficient manner. Depending on the new developed traffic generator for many experiments shows reliable well performing results compared with the old traffic generator in different metrics.

REFERENCES

1. Ning Yang, "Congestion-Aware Cross-Layer Design for Wireless Ad Hoc Networks", Master thesis of Science in Electrical Engineering Department of Electrical Engineering College of Engineering University of South Florida, (2004).
2. Sudip Misra, Isaac Woungang and Subhas Chandra Misra, "Guide to Wireless Ad Hoc Networks", Springer-Verlag London, (2009).
3. Stefano Basagni, Marco Conti, Silvia Giordano and Ivan Stojmenovic, "MOBILE AD HOC NETWORKING", Wiley- IEEE Press, published simultaneously in Canada, (2004).
4. Sanjay Shakkottai, Theodore S. Rappaport, and Peter C. Karlsson, "Cross-layer design for wireless networks "IEEE Communications Mag., 41: 74-80 (2003).
5. D.Staehle, K.Leibnitz and P.Tran-Gia, "Source Traffic Modeling of Wireless Applications", *International Journal of Electronics and Communications*, 55, (2001).
6. Priti Garg and Asma Tuteja, "Comparative Performance Analysis of Two Ad-hoc Routing Protocols", International Conference on Network and Electronics Engineering, IACSIT Press, Singapore, (2011).
7. Dong Sun and Hong Man, "TCP Flow-based Performance Analysis of Two On-demand Routing Protocols for Mobile Ad Hoc Networks ", (2001).
8. Ronald Beaubrun and Badji Molo, "Using DSR for Routing Multimedia Traffic in MANETs", *International Journal on Computer Science and Engineering (IJCSE)*, (2010).

9. Wojciech Wojcikiewicz, "Ad-hoc routing protocols A comparison between OLSR and ARA", November (2007).
10. Lydia Parziale, David T. Britt, Chuck Davis, Jason Forrester, Wei Liu, Carolyn Matthews and Nicolas Rosselot "TCP/IP Tutorial and Technical Overview", Eighth Edition, IBM redbooks, (2006).
11. Ekram Hossain and Teerawat Issariyakul, "Introduction to Network Simulator NS-2", Second Edition, Springer New York Dordrecht Heidelberg London, (2012).
12. S. Corson and J. Macker, "Routing Protocol Performance Issues and Evaluation considerations", RFC2501, IETF Network Working Group, (1999).
13. Sabina Barakovic, Suad Kasapovic, and Jasmina Barakovic, "Comparison of MANET Routing Protocols in Different Traffic and Mobility Models", *Telfor Journal*, 2(1), (2010).
14. Nilesh P. Bobade and Nitiket N. Mhala, "Performance Evaluation of Ad Hoc On Demand Distance Vector in MANETs with varying Network Size using NS-2 Simulation", *International Journal on Computer Science and Engineering (IJCSE)* 2(8), (2010).
15. Vikas Singla and Parveen Kakkar, "Traffic Pattern based performance comparison of Reactive and Proactive protocols of Mobile Ad-hoc Networks", *International Journal of Computer Applications* (0975 – 8887), (2010).
16. Nisarg Gandhewar and Rahila Patel "Performance Evaluation of AODV protocol in MANET using NS2 Simulator", Proceedings published in *International Journal of Computer Applications (IJCA)*, 2nd National Conference on Information and Communication Technology (NCICT), (2011).
17. Yasser Kamal Hassan, Mohamed Hashim Abd El-Aziz and Ahmed Safwat Abd El-Radi, "Performance Evaluation of Mobility Speed over MANET Routing Protocols" , *International Journal of Network Security*, 11(3), 128-138, (2010).