*International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)*
*Volume-2 Number-4 Issue-6 December-2012*

# Performance of TCP-Throughput on NS2 by Using Different Simulation Parameters

**Neeraj Bhargava[1], Ritu Bhargava[2], Anchal Kumawat[3], Bharat Kumar[4]**
[1]Associate Professor, Department of Computer Science , School of Engineering & Systems Science, MDS University, Ajmer ,Rajasthan, India.
[2]Lecture, Department of MCA, Govt. Women's Engineering College, Ajmer, Rajasthan, India.
[3]MCA Scholar, Deprtment of Computer Science, School of Engineering & Systems Science, MDS University, Ajmer, Rajasthan, India.
[4]Research Scholar, Department of Computer Science, MJRP University, Jaipur, Rajasthan, India.

## Abstract

***ns2 is a discrete event simulator for networking research, which works at the packet level. Here, we will be using ns2 to simulate traffic congestion of TCP and UDP packets inside a network. ns2 is popularly used in the simulation of routing and multicast protocols and is heavily used in ad-hoc networking research. ns2 supports network protocols (TCP, UDP, HTTP, Routing algorithms, MAC) etc. for offering simulation results for wired and wireless networks. When using TCP to transfer data the two most important factors are the TCP window size and the round trip latency. This paper deals the effect that the size of the flow control window has on the throughput of a TCP connection by using simulation parameters like-packet delay (sec), bandwidth, file-size (bytes) and to implement network fed with TCP traffic and background traffic. The objective of this paper is to observe the performance of TCP.***

## Keywords

***ns2, TCP, nam, gnuplot, Xgraph, TCP-throughput analysis***

## 1. Introduction

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite. TCP is one of the two original components of the suite, complementing the Internet Protocol (IP), and therefore the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered delivery of a stream of octets from a program to another program on different computers. TCP is the protocol used by major the internet applications such as the WWW (World Wide Web), email, remote administration, and file transfer. Other applications, which do not require reliable data stream service, may use the User Datagram Protocol (UDP), which provides a datagram service that emphasizes reduced latency over reliability [1].

TCP is a transport layer protocol used by applications that require guaranteed delivery. It is a sliding window protocol that provides handling for both timeouts and retransmissions. TCP is one of the main protocols in TCP/IP networks, whereas the IP protocol deals only with packets. TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees the delivery of data and the packets will be delivered in the same order as they were sent. The traditional methods for examining the performance of TCP have been simulation, implementations and measurements. However, the efforts have been made to characterize the throughput of TCP analytically as a function of parameters (like packet drop rate and round trip time) [2][8].

The TCP agent does not generate any application data on its own. Instead, the simulation user can connect any traffic generation module to the TCP agent to generate data. Two applications are commonly used for TCP: FTP and Telnet. FTP represents a bulk data transfer of large size, and telnet chooses its transfer sizes randomly [2].

There are two major types of TCP agents: one way agents and a two-way agent. One-way agents are further subdivided into a set of TCP senders (which obey different congestion and error control techniques) and receivers (sinks). The two-way agent is symmetric in the sense that it represents both a sender and receiver. TCP primarily designed for wired networks but it faces performance degradation when applied to the ad-hoc scenario.

*International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)*
*Volume-2 Number-4 Issue-6 December-2012*

## 2. Simulation

### a. Simulation model

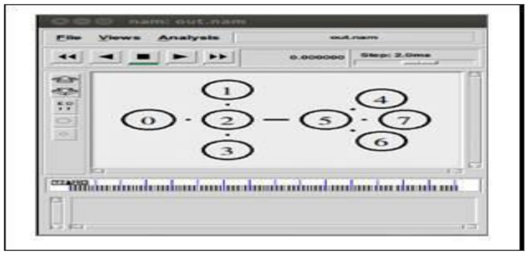


**Figure 1: Number Structure of Simulation Model.**

As shown in Figure 1. Which is a simulation network consisting of 8 nodes. The following are the steps to create simulation.

Node no 2 and node no 5 can be assume as routers. Node no 2 and node no 5 link is limited to 20 packets queue size and all other links have limited to 50 packets queue size.

Create FTP traffic on top of a TCP connection between node no. 1 and node no. 6 which act as TCP Reno. The TCP connection's maximum congestion window size is 40 packets, and the packet size is 280 bytes. Time out period of the TCP is to be set more than 0.2 seconds otherwise simulation does not create. The limit of FTP in this paper starts from 0.0 second and ends up to 15 seconds.

Add CBR traffic on top of a UDP connection between node no. 0 and node no. 6. The CBR service generates and sends packets at the rate of 100 packets per second. The CBR source starts at 2.0 second and lasts for 7 seconds.

Further add a VBR video traffic with rate of 600 KBPS between node no. 3 and node no. 6, use toggle for exponential traffic, set On period as 150ms, and Off period as 100 ms. The size of each CBR and VBR packet is 280 bytes. The VBR starts at 7.0 seconds and ends at 11 seconds.

The congestion window size and bandwidth of CBR and VBR traffic should be monitored and plotted during the entire simulation time. The average throughput of TCP, CBR, and VBR will be obtained from iteration [9][15].

### b. Simulation Parameters

This network model will be generated with the help of network animator tool, after running TCL script considering the following simulation parameters as shown in table 1.

**Table 1: Simulation Parameters**

| Channel | Channel/Wireless |
|---|---|
| Network Interface | Wireless |
| NS Version | ns-2 |
| CBR Packet Size | 1280 bytes |
| Interface Queue | Drop Tail |
| Queue Length | 50 |
| No. of Nodes | 8 |
| Simulation Area Size | 800*600 |
| Simulation Duration | 60 second |
| VBR Packet Size | 1280 bytes |
| Packet Rate | 1600k |
| Burst Time | 750ms |
| Idle Time | 1100 |

## 3. Methodologies

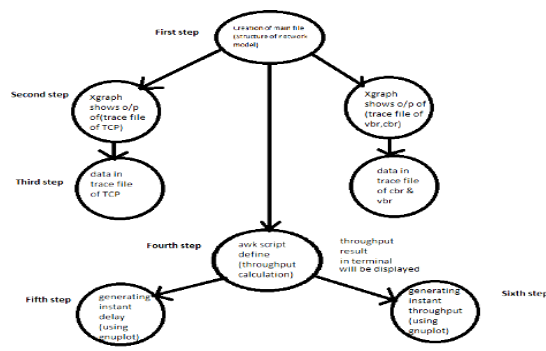


**Figure 2: DFD (Data Flow Diagram) of Methodologies**

### A. First step

Create a main file that runs the simulation and prints the results to "out.nam" file. The structure of dynamic network-model, draw as Figure 3(a). The Creation of main file includes the following steps :

- Begin with creating the simulator object.
- Define the different colors for data flows.
- Open the output files & network animator trace file.
- To display the results initiate Xgraph.
- Execute network animator on the trace file
- Create eight nodes & links between the nodes & also set the orientation of all 8 nodes.
- Monitor the queue for the link between different nodes.

- Create a UDP agent and attach it to node no. 0 & create a CBR traffic source and attach it to UDP on node no. 0.
- Create a TCP agent and attach it to node node no. 1 to node no. 6 after that setup TCP connections.
- Create a VBR traffic source and attach it to another node i.e. node no. 3.

A procedure will be created which periodically records the bandwidth received by the two traffic sinks sink 0/1 and writes it to the three files f0/1.

- To call again the procedure set the time as 0.05 seconds & then find how many bytes have been received by the traffic sinks further calculate the bandwidth (in MBit/s) and write it to the desired file.
- Reset the above calculated bytes values on the traffic sinks .
- Re-schedule the procedure for which time has been set as 0.05 seconds.
- Create two Loss monitors such as a traffic sink & traffic source later on attach them to node no. 6 .
- Start logging with the received bandwidth. Schedule events for the CBR agents & Call the finish procedure after 5 seconds of simulation time.

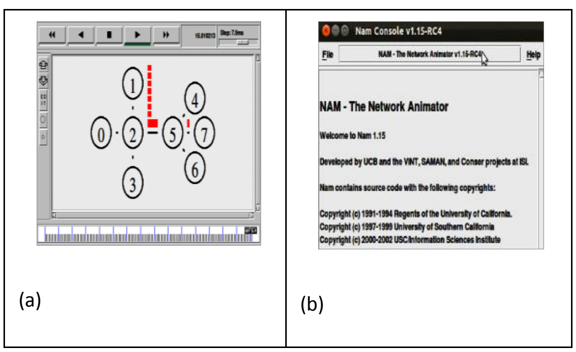


**Figure 3 (a): Dynamic Network Model, (b) 'Nam' Window.**

### B. Second step

After completing the simulation we will generate two Xgraph windows as shown in Figure 4(a) & Figure 4(b) and a simulation window is also known as nam window as Figure 3(b).

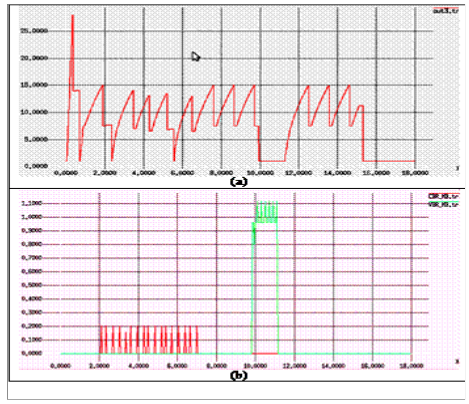


**Figure 4.a : Xgraph window show trace file of TCP,**
**Figure 4. b : Xgraph window shows trace file of CBR & VBR**

### C. Third step

The data of both above Xgraph will be shown in out.nam file that can be express as follows -

+ -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 7.0 0 ------- null}

- -t 0 -s 1 -d 2 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 7.0 0 ------- null} + -t 0.001064 -s 2 -d 5 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 7.0 0 ------- null}

In this section there are only TCP and TCP acknowledgment packets, but real out.nam file is like 12 MB. Some characters are shown as –

r : received, + : queued, - : dequeued

h : routed, d : dropped.

The first line will be interpreted as -

+ -t 0.001064 -s 2 -d 5 -p tcp -e 40 -c 2 -i 0 -a 2 -x {1.0 7.0 0 ------- null}

It elaborate as packet queued, at time 0.001064, from source node no. 2, to destination node no. 5, protocol is TCP, packet size is 40 bytes, flow id (-c) is 2 (cbr and vbr do not have flow id), packet number is (-i) 0. Assign the no. of each element before writing an awk scripts like $1, $2, $3…,$15, where first element in the row $1 (r, t, +,-, d) represented as status, second element $2 is not used for specific purpose, third element in the row $3 represented as time & $15 is represented as flow id.

### D. Fourth step

Awk script defines instantaneous delay and throughput in main file where calculation has done with different parameters such as packet size, acknowledge of data, cbr rate, cbr rate of data & simulation time ,vbr rate, vbr rate of data & simulation time , tcp rate, tcp data & simulation time & packet size to find throughput & Instantaneous delay. After defining all parameters the awk file will be displayed & output of awk file will store in NAM file. The result will be shown as in Figure 5. TCP sent 1238 acks and 1201 of them were delivered.CBR sent 34 data packets and 33 were delivered by the throughput: 34.5546 Kbps , VBR sent 414 packets and 161 were delivered by the throughput: 990.032 Kbps System throughput is in Kbps.

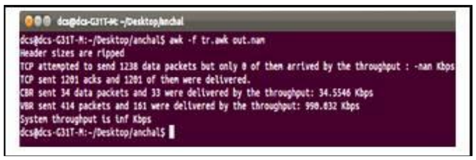


**Figure 5: Output window display the throughput results**

### E. Fifth step

For creating instant delay file with the help of following code as shown in Figure 6. To plot the instant delay over the time using gnuplot. gnuplot is a visualization tool which is shown as follows:-

```
BEGIN
{ send[""]=50;
recv[""]=0;
received =0
settime=0
gr=0.1
arr=1 }
if (($1 == "+")&&($9 == "tcp")&&($5 == 1)){
send[$15]=$3 }
if (($1 == "r")&&($9 == "tcp")&&($7 == 6)){
received= received+1
recv[$15] = $3
average+= recv[$15]-send[$15]
if ($3 <= settime) {
arr=arr+1
subtotal=subtotal+recv[$15]-send[$15]
#printf ("eskiarr:%g", arr) }
else { subtotal= subtotal/arr
printsettime, subtotal > "insde.txt"
set=set+1
arr=1
settime=settime+gr
#printf ("yeniarr:%g", arr)
subtotal=recv[$15]-send[$15] } } }
END { #average = average / received
#printf("Average delay is %g seconds with total %g received packets \n",average, received)
subtotal= subtotal/arr
printsettime, subtotal > "insde.txt" }
```

**Figure 6: gnuplot for instant delay code**

The Graph will be generated after defining data set & experiments will be done through gnuplot. After plotting the data the result will be shown as in Figure 7.

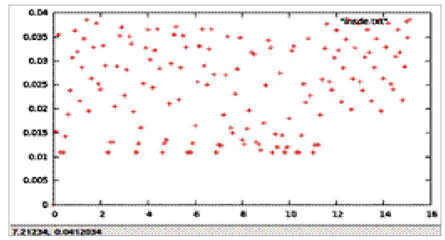


**Figure 7: gnuplot for instant delay**

### F. Sixth step

For finding instant throughput the following code will be generated as shown in Figure 8. After writing the code the result will be shown as in figure 9.

```
BEGIN
{settime=0
gr=0.1
subtotal=0
total=0
set=0 }
 if (($1 == "r") && ($9 == "tcp")&& ($7 == 6))
{ if($3 <= settime) {
subtotal=subtotal+$11
total=total+(subtotal/gr)*8/1000
set=set+1}
else {
subtotal= (subtotal/gr)*8/1000
printsettime, subtotal > "insth.txt"
set=set+1
total=total+subtotal
settime=settime+gr
subtotal=$11 }
} }
END {
subtotal= (subtotal/gr)*8/1000
#printf ("total %g", total)
printsettime, subtotal > "insth.txt"
printsettime> "time.txt" }
```

**Figure 8: gnuplot for instant throughput code**

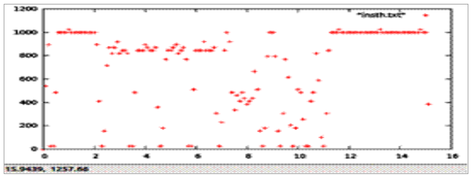

**Figure 9 : gnuplot for instant throughput**

## 4. Conclusion

This paper summarizes to measure the performance of TCP and its Simulations generated with the help of ns2 software. Several simulations have been run with Ns2 in order to acquire a better understanding of these parameters .It shows that ns2 is a perfect tool for achieving such goal.

## References


[1] Transmission Control Protocol http://en.wikipedia.org/wiki/Transmission_Control_Protocol#cite_note-12.

[2] Introduction to Network Simulator Ns2 ,second edition . TeerawatIssariyakul EkramHossain Springer.

[3] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. Comparative Study of Wireless Network Simulators. The Seventh International Conference on Networking, 517-523, 2008. IEEE.

[4] G. F. Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M. J. Reed. OPNET Modeler and ns-2 - Comparing the Accuracy of Network Simulators for Packet-level Analysis Using a Network Testbed. WSEAS Transactions on Computers, 2(3):700-707,July 2003.

[5] M. Fleury, G. F. Lucio and M. J. Reed. Clarification of the "OPNET NS-2 Comparison" Paper with regards to OPNET Modeler. July 2003.

[6] Organizing server assets: Methodology for organizing server assets, 2003. From the OPNET Methodology and Case Studies Website, part of the Technical Resources papers.

[7] SNS: Staged Simulation in ns-2. http://www.cs.cornell.edu/people/egs/sns/ .

[8] J.Padhye, V.Firoiu, D.Towsley, J.Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", In Proc. ACM SIGCOMM '98 1998.

[9] TCP Extensions for High Performance, Internet Engineering Task Force RFC 1323. 9-11.

[10] The Network Simulator ns-2: Tips and Statistical Data for Running Large Simulations in NS; http://www.isi.edu/nsnam/ns/ns-largesim.html.

[11] W. G. Bragg. Which network design tool is right for you? IEEE IT Pro Magazine, 2(5): 23-31,2000.

[12] J. Heidemann and K. Mills. Expanding confidence in network simulations. IEEE Network Magazine, 2001.

[13] K. Pawlikowski, H-D. J. Jeong, and J-S. R. Lee. On the credibility of simulation studies of telecommunication networks. IEEE Communications Magazine, 2002.

[14] C. Zhu, O. W. W. Yang, J. Aweya, M. Oullette, and D. Y. Montuno. A comparison of active queue management algorithms using the OPNET Modeler. IEEE Communications Magazine, 40(6): 158-167, 2002.

[15] S. Fortin-Parisi and B. Sericola. "A Markov model of TCP throughput, goodput and slow start." Performance Evaluation (58):89-108, Nov.2004.



**Dr. Neeraj Bhargava** : Presently working as Associate Professor & Head, Department of Computer Science, School of Engineering & System Sciences, MDS University , Ajmer. He has more than 23 yr of experience for teaching in the University. His areas of interest are Spatial Database, Image Processing and Ad hoc Network.

**Dr. Ritu Bhargava**: Presently working as lecturer, Department of MCA, Govt. Women's Engineering College, Ajmer. She has more than 9 yr of experience for teaching MCA. Her areas of research are Web GIS, Wireless Network.

**Anchal Kumawat** : Pursuing MCA from Department of Computer Science, School of Engineering & System Sciences, MDS University , Ajmer. Presd ently she is involve in Ad hoc network and Wireless Network.

**Bharat Kumar: Pursuing** Ph.D Computer Science from MJRP University Jaipur. Presently he is involved in Ad hoc network research.