All Sciences Proceedings

5th International Conference on Applied Engineering and Natural Sciences

July 10-12, 2023 : Konya, Turkey

All Sciences Proceedings
http://as-proceeding.com/

https://www.icaens.com/

Performance Analysis of Large-scale Networks in Ns-2

Ahmet Zengin*,1 and İdris Cesur 2

[1]Computer Engineering Department/Faculty of Computer and Information Sciences, Sakarya University, Türkiye
[2]Mechanical Engineering Department/Faculty of Technology, Sakarya Applied Science University, Türkiye

*(azengin@sakarya.edu.tr) Email of the corresponding author

Abstract **–** Wired communication refers to data transmission over a wire-based communication technology. Also, in computer engineering terms, a network is defined as a group of computers connected to share information or services (print services, multitasking, etc.). In this paper, the performance of a wired network that consists of more than 300 nodes which implement as an example of a big organization (i.e., Sakarya University), is investigated and analyzed by simulations using a network simulator tool (Ns-2). The analysis depends on many performance metrics such as throughput, delay, Jitter, etc. Results show that modeled network has enough performance in different scenarios.

Keywords – Wired, Networks, Ns-2 Simulator, Throughput, Delay, Jitter.

I. INTRODUCTION

A computer network is two or more computers connected to exchange data [1]. A small network can be as simple as connecting two computers with a single cable. Internet communication between two or more networks covers all aspects of computer communication. The main purpose of the campus LAN is to share resources and access information among users.

Network simulation provides the results of network design. The NS-2 network simulator is the most widely used open-source simulation simulator and is the most important, efficient and effective network simulation design tool for networking, language network research area and data objectives [2].

In this work, we investigate and analyze by simulations the performance of a wired large-scale network consisting of more than 300 nodes (exactly 311 nodes) implemented as an example of a big organization (i.e., Sakarya University) that analysis depends on many performance metrics such as throughput, delay, jitter and another metrics. Also, there is a proposed scenario.

The structure of this work is as follows. In Section 2, we make an overview of the methodology and performance metrics. In Section 3, we describe the simulation system design. In Section 4, we show the simulation results. Finally, the conclusion is given in Section 5.

II. MATERIALS AND METHOD

In this section, we have described the simulation environment, implementation details of network topologies, and performance metrics.

A. Simulation Environment (NS-2)

Vitality in Computers Models in computers to study fictional and real living objects. The network is simulated on a computer. A network emulator is a machine on a computer that uses a network. Our network state analysis is based on the Ns-2 test [2].

The Ns-2 provides extensive support for implementing TCP, routing and multicast protocols in wired and wireless (local and satellite) networks [3]. The performance of the network is also analyzed with the help of the simulation results of the demand [4].

B. Simulation Steps

To analyze the problem, a TCL (Vehicle Command Language) script must be written and simulated using Ns-2. The user program Ns-2 uses the OTcl (Object Tool Command Language)

scripting language. The OTcl letter will be as follows.

- Initiates an event scheduler.
- Set the network topology using the network object.
- Tells the traffic center when to start/stop sending packets according to the schedule.

Depending on the purpose of the user's OTcl simulation script, simulation results are stored as trace files that can be loaded by external requests for analysis [5]:

1. A NAM trace file (i.e., namFile.nam in our work) for use with the Network Animator Tool.
2. A Trace file (traceFile.tr) for use with XGraph, TraceGraph, or another's application.

The main simulation and network configuration in Ns-2 is done in the TCL script. After defining the network to NS2, run the TCL script and check the results of the output data. There are two exit points; a trace file containing all events (sending, receiving, packet loss, etc.) and a NAM file that can be used by network animators to view the simulation in graphical mode. In our experiment, we use authentication to validate tracking data and calculate network requirements such as access.

C. Implementation

The performance of each system must be evaluated against certain standards; these standards lay the foundation for all operations. These metrics are called performance metrics. Consider the following performance metrics when evaluating network communications:

1. Throughput: The throughput is the amount of data sent for effective communication. It is the number of data rates sent to all terminals in the network. The passing criteria are:

Throughput = received data X 8 / data transmission period.

2 2. Delay: defined as the time required to send a packet to its destination. The delay should be as low as possible.

3. Jitter: It is defined as the delay time between packets in the same stream.

4. Packet Delivery Rate (PDR): The ratio of packets sent to the destination to the number sent by the destination. The formula used to calculate the package delivery fee is as follows:

PDR = received packets / generated packets * 100

III. SIMULATION SYSTEM DESIGN

In this section, we will describe the proposed scenario for the network under study and the code which implements this scenario.

A. Proposed scenario

In our project, we assumed there is a big organization (i.e., Sakarya University) that has a main router and ten faculty buildings. Each department has 30 hosts; furthermore, the star topology is selected to implement this network.

All devices in topology:

- connected by cables as duplex connection
- connection capacity 1.5 Mb, propagation delay 10 ms, using the DropTail queue scheduling algorithm.
- We need to send TCP data from host (13) (node3) to host (9) and from host (5) to host (12).
- We need to send UDP data from host (13) to host (6) at a rate of 1536 packets / 44 seconds, with a packet size of 1.5 KB.
- UDP traffic is also sent from Host(1) to Host(8) to generate 5632 packets of 5.5KB every 37 seconds.

Scheduling Operation:

- TCP Data: Master (13) - Node 3- Master (9) (Start: 0.2, Stop: 1.8), Master (5) to Host (12) (Start: 0.3, Stop) : 1.4)
- UDP data: host(13) to host(6) (start: 0.4, stop: 1.6), p1 to p8 (start: 0.7, stop: 1.7)
- routers Created by (1) Connection and Dept(2) drops at 0.7, then back at 1.0
- Drops connection between Router(1) and Dept(2) at 0.9, then 1.3 returns at
- stops at 2.0 simulation

B. Proposed scenario

Below is the code to generate the model with simple comments to clarify each step:

#----------#

```
# 1st, 2nd steps: Create a new instance from Simulator, and
trace, nam files #
#--------------------------------------------------------------------------#
set ns [new Simulator]

# creaet network animator file, nam
set nf [open namFile.nam w]
$ns namtrace-all $nf

# and create and open trace file for monitoring, traceFile
set tr [open traceFile.tr w]
$ns trace-all $tr
#--------------------------------------------------------------------------

#--------------------------------------------------------------------------#
# Create finish procedure to empty memo. and exec nam file
#
#--------------------------------------------------------------------------#
proc finish {} {
    global ns nf tr
    $ns flush-trace
    close $nf
    close $tr
    exec nam out.nam
    exit 0
}
#--------------------------------------------------------------------------

#--------------------------------------------------------------------------#
# 3ed step: Create topology (nodes and edges) with its style
#
#--------------------------------------------------------------------------#
#deifinig main variables for (Department, Hosts)
set deptCount 10
set hostCount 300
set hostPerDept [expr ($hostCount/$deptCount)]

# 3.1 create nodes and set its style
# 3.1.1 create "main router" node with its style
set Router(1) [$ns node]
$Router(1) shape hexagon
$Router(1) label Main_Router
$Router(1) label-color black
$Router(1) color red
$Router(1) add-mark Main_Router orange hexagon

# 3.1.2 create "Departments" nodes with its style
for {set i 1} {$i<=$deptCount} {incr i} {
 set Dept($i) [$ns node]
 $Dept($i) shape hexagon
 $Dept($i) color purple
 $Dept($i) label-color black
 $Dept($i) label Dept($i)
 #set x($i) [$ns node]
}

# 3.1.3 create "Hosts" nodes with its style
for {set i 1} {$i<=$hostCount} {incr i} {
 set Host($i) [$ns node]
 $Host($i) color blue
 $Host($i) label-color black
 $Host($i) label Host($i)
}

# 3.2 Creating Links (connections) for nodes
# 3.2.1 Creating Links (connections) between Main Router and
Dept.
# Bandwidth: 1 MB, Delay: 10 milli-seconds, Queue management
algorithm: DropTail
for {set i 1} {$i<=$deptCount} {incr i} {
$ns duplex-link $Router(1) $Dept($i) 1.5Mb 10ms DropTail
}

# 3.2.1 Create links (connections) between Dept. and Hosts
for {set i 1} {$i<=$deptCount} {incr i} {
 set index [expr (($i-1)*$hostPerDept+1)]
 set stopCond [expr ($i*$hostPerDept+1)]
 for {set j $index} {$j<$stopCond} {incr j} {
  $ns duplex-link $Dept($i) $Host($j) 1.5Mb 10ms DropTail
}
}
#--------------------------------------------------------------------------

#--------------------------------------------------------------------------#
# 4th step: create the routing (i.e. dynamic routing protocol)
#
#--------------------------------------------------------------------------#
$ns rtproto DV

#--------------------------------------------------------------------------#
# 5th step: create the transmission protocols (TCP, UDP)
(Transport)      #
#--------------------------------------------------------------------------#
#defining colors for data flow (packets color)
$ns color 0 black
$ns color 1 Red
$ns color 2 Blue
$ns color 3 Orange
$ns color 4 purple
$ns color 5 Green
$ns color 6 Yellow
$ns color 7 Magenta
$ns color 8 brown

# 5.1 creating tcp agents and attach it with Host(3, 9, 5, 12)
set tcp3 [new Agent/TCP]         ;# 1'st TCP agent (tcp3)
set tcpSink9 [new Agent/TCPSink]
$tcp3 set fid_ 8                          ;# packets color (from
tcp3)
$tcpSink9 set fid_ 5             ;# Color
$ns attach-agent $Host(3) $tcp3    ;# attach agent with specific
node
$ns attach-agent $Host(9) $tcpSink9 ;# attach tcpSink with
specific node
$ns connect $tcp3 $tcpSink9              ;# connect between tcp &
its sink

set tcp5 [new Agent/TCP]         ;# 2'nd TCP agent (tcp5)
set tcpSink12 [new Agent/TCPSink]
$tcp5 set fid_ 7
$ns attach-agent $Host(5) $tcp5
$ns attach-agent $Host(12) $tcpSink12
$ns connect $tcp5 $tcpSink12

# 5.2 creating udp agents and attaching it with Host(13,6,1,8)
set udp13 [new Agent/UDP]                        ;# 1'st UDP agent
(udp13)
set udpNull6 [new Agent/Null]                    ;# UDP dosent
contain Sink
$udp13 set fid_ 3
$ns attach-agent $Host(13) $udp13
$ns attach-agent $Host(6) $udpNull6
$ns connect $udp13 $udpNull6
```

```
set udp1 [new Agent/UDP]                    ;# 2'nd UDP
agent (udp1)
set udpNull8 [new Agent/Null]
$udp1 set fid_ 3
$ns attach-agent $Host(1) $udp1
$ns attach-agent $Host(8) $udpNull8
$ns connect $udp1 $udpNull8
#-------------------------------------------------------------------------

#-------------------------------------------------------------------------#
# 6th step: create the traffic model (FTP, CBR) (Application)
#
#-------------------------------------------------------------------------#
# 6.1 creating FTP application for tcp agents
set ftp3 [new Application/FTP]
set ftp5 [new Application/FTP]
$ftp3 set fid_ 6
$ftp3 attach-agent $tcp3        ;# ftp3 --> tcp3
$ftp5 attach-agent $tcp5        ;# ftp5 --> tcp5

# 6.2 creating CBR applications for udp
set cbr13 [new Application/Traffic/CBR]
# send 50 packets in 1 second i.e 1 packet every 1/50 second
$cbr13 set packetSize_ 1536
$cbr13 set interval_ 0.02
$cbr13 attach-agent $udp13

set cbr1 [new Application/Traffic/CBR]
# send 400 packets in 1 second i.e 1 packet every 1/400 second
$cbr1 set packetSize_ 5632
$cbr1 set interval_ 0.0025
$cbr1 attach-agent $udp1
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------#
# 7th step: Schedule events for the traffic, Simulation time (in
seconds)  #
#-------------------------------------------------------------------------#
# setting events
$ns rtmodel-at 0.7 down $Router(1) $Dept(2)
$ns rtmodel-at 1.0 up $Router(1) $Dept(2)

$ns rtmodel-at 0.9 down $Router(1) $Dept(5)
$ns rtmodel-at 1.3 up $Router(1) $Dept(5)

$ns at 0.2 "$ftp3 start"
$ns at 1.8 "$ftp3 stop"

$ns at 0.3 "$ftp5 start"
$ns at 1.4 "$ftp5 stop"

$ns at 0.4 "$cbr13 start"
$ns at 1.6 "$cbr13 stop"

$ns at 0.7 "$cbr1 start"
$ns at 1.7 "$cbr1 stop"

# Call the finish procedure after 2.0 seconds of simulation time
# finish proc closes temp trace files and opens the network
animator
$ns at 2.0 "finish"
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------#
# 8th step: Start the Simulation                    #
#-------------------------------------------------------------------------#
$ns run
```

IV. SIMULATION RESULTS

A. Trace file and NAM screenshots

The trace file for this project which named is (traceFile.tr), as shown following:

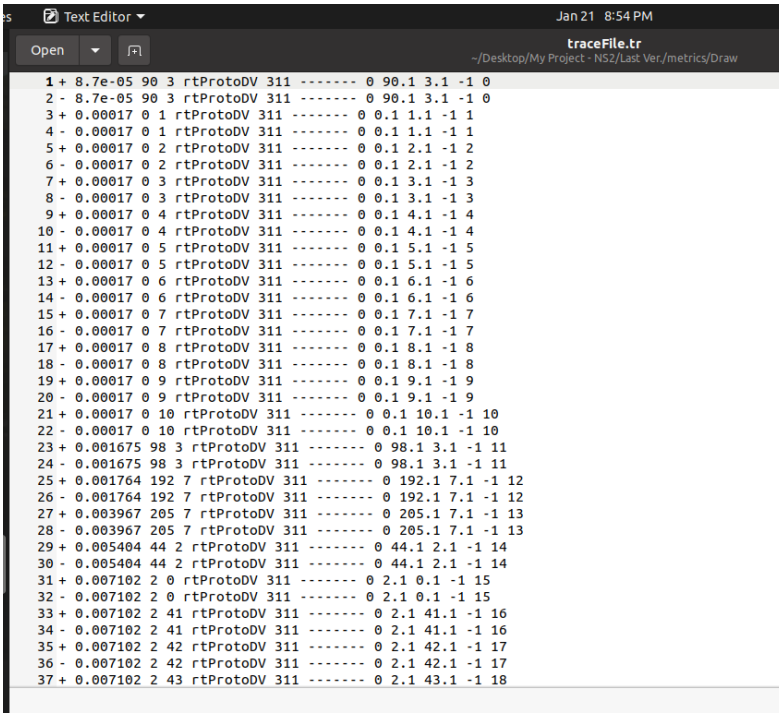

Fig. 1 A Screenshot of the Trace File for the project with (311 Nodes)

Figure 1 shows monitoring data; It was created after the animation in the NAM window finished. We will analyze our data with the help of tracking data.

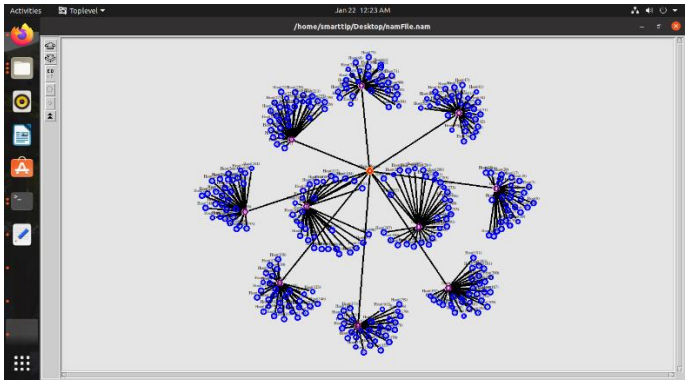

Fig. 2 A Screenshot of the NAM window for the project with (311 Nodes)

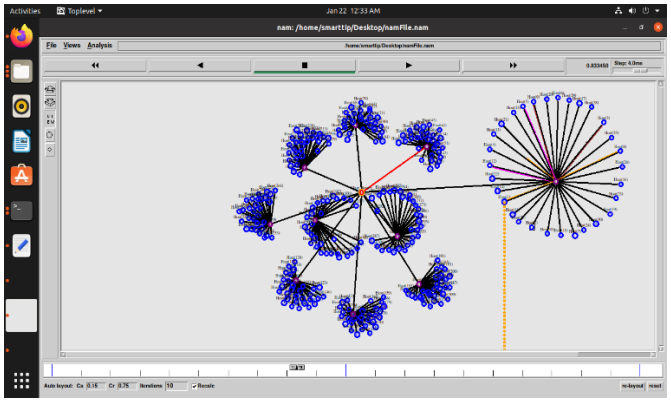

Fig. 3 A Screenshot of the NAM window for the project with (311 Nodes)

Figures 2 and 3 show the data flow and drop packets in a node (11), which is named Dept (1); we maximized the network for Dept (1) for clarification purposes.

B. Simulation Result

As a result of our network simulation, we got information from (traceFile.tr), so we can extract many metrics from it. These metrics give us indications about the performance of the network. Finally, we represented some of them as values or plots, as shown below:

B.1. Throughput

Figure 4 shows the network's throughput, and after applying some "awk" script, we got the received packets are 13056 and the average throughput [kbps]: 19688.8.

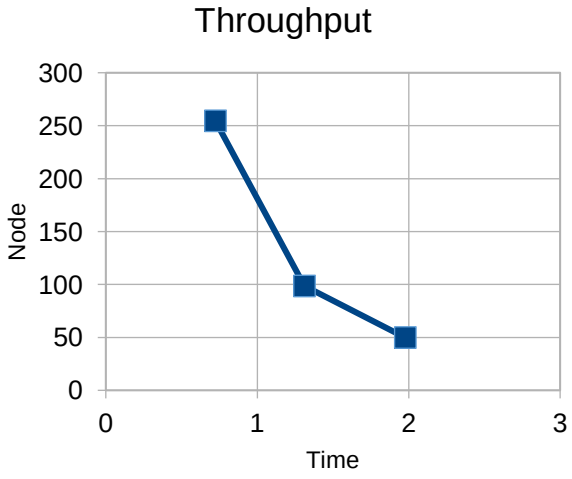

Fig. 4 Throughput

B.2. Delay

Here we used the awk script to extract the delay and GNUPLOT to plot the result, as shown in Figure 5.

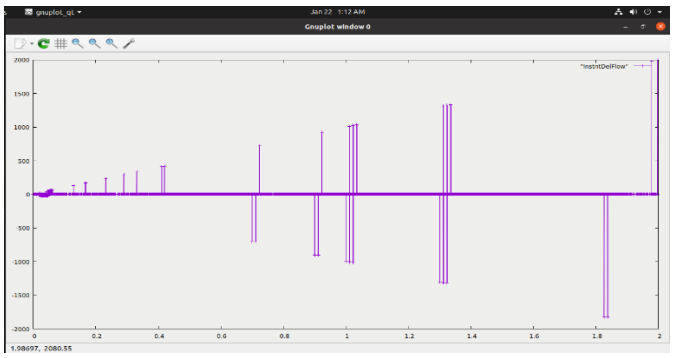

Fig. 5 Instant delay (By gnuplot tool)

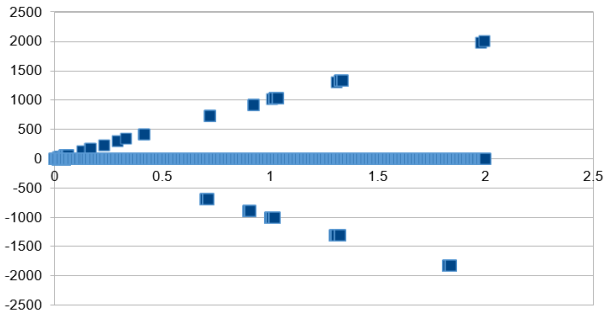

Fig. 6 Average of delay (By Excel software)

Instant delay:

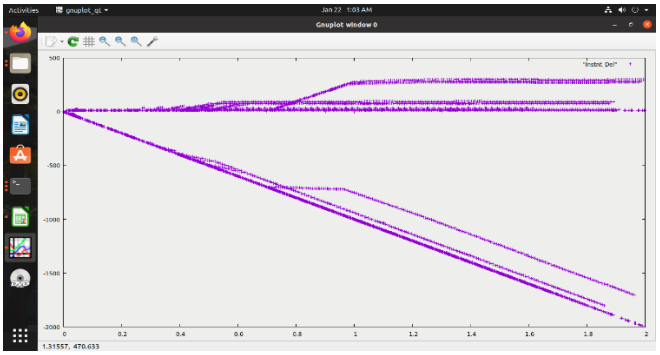

Fig. 7 Instantions of delay

B.3. Jitter

As shown below, the jitter metric shows the high jitter before finishing the simulation, as shown in Figure 7.

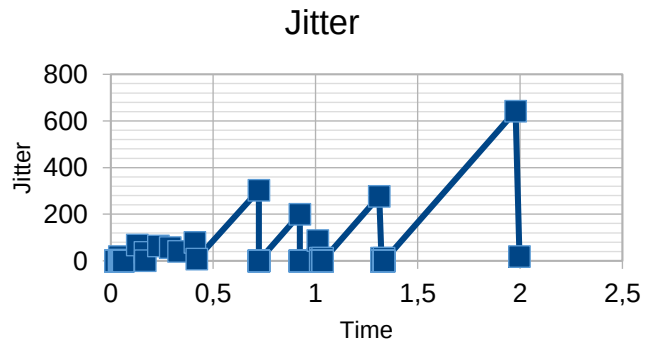

Fig. 7 Plot of Jitter

V. CONCLUSIONS

This article presents a performance analysis of a 311-node network under different conditions and different packages; The simulation was done by Ns-2. Metrics such as throughput, latency, and packet delivery are used to analyze network performance. These requirements are different for FTP and CBR traffic. After the scenario was presented, we extracted some parameters such as efficiency, delay and vibration from the generated graphics after performing the simulation.

REFERENCES

[1] Steenstrup, Martha, Routing in Communications Networks, Prentice Hall, 1995

[2] The Network Simulator - ns-2, https://www.isi.edu/nsnam/ns/ (last accessed on 21.05.2023)

[3] Ghodichor, F. S., & Madan, B. Network Analyzer and Report Generation Tool for NS-2 using TCL. 2017

[4] Payal, J. S. K. "TCP traffic based performance investigations of DSDV, DSR and AODV routing protocols for manet using ns2." Int. J. Innov. Technol. Explor. Eng 3.2 (2013): 2278-3075.

[5] Mukeshkumar, Ganesh Kumar “To Analyze and Compare Ring and Mesh Topologies with Varying Traffic Patterns” International Journal For Technological Research In Engineering, Volume 2, Issue 11, July-2015