

# Certificate

Name: MOHD. AFTAB ALAM

Class: CSE-6A

Roll No: 02013302717

Exam No:

Institution HMR INSTITUTE OF TECHNOLOGY AND  
MANAGEMENT

*This is certified to be the bonafide work of the student in the  
COMPUTER NETWORKS Laboratory during the academic  
year 20 / 20 .*

*No. of practicals certified \_\_\_\_\_ out of \_\_\_\_\_ in the  
subject of \_\_\_\_\_*

.....  
Teacher In-charge

.....  
Examiner's Signature

.....  
Principal

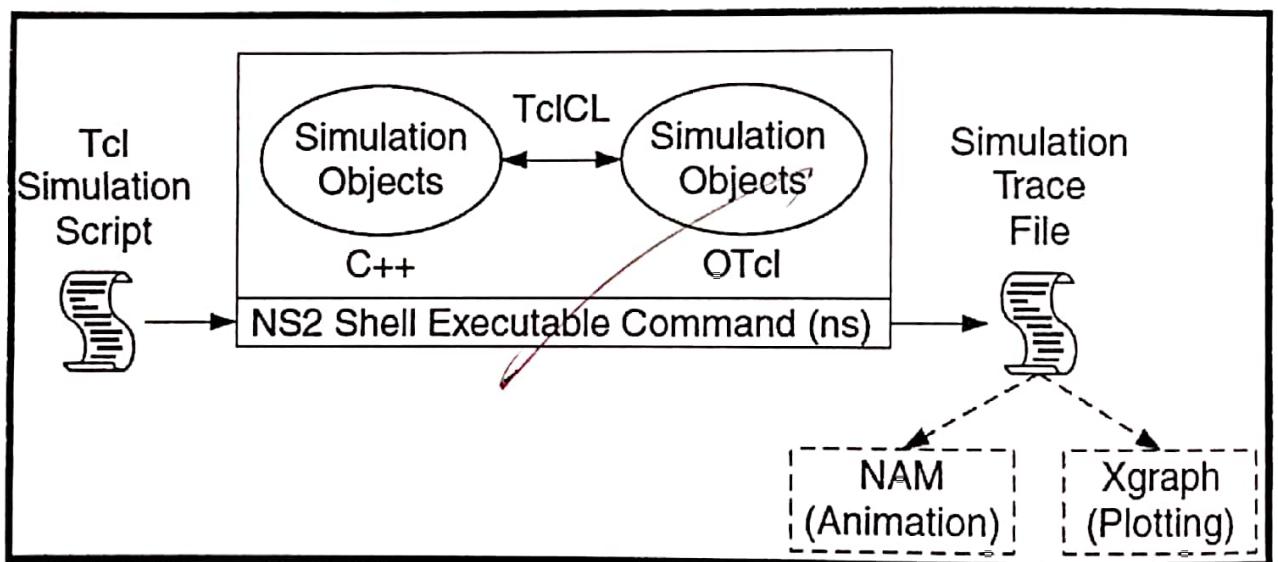
Date: .....

Institution Rubber Stamp

(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)

# Index

20/01/2020



## # ARCHITECTURE OF NS2

## Introduction.

### # Introduction to NS2.

- (\*) NS2 stands for Network Simulator Version 2. It is an open source event driven simulator designed specifically for research in computer communication networks. It is simply an event driven simulation tool. Useful in studying the dynamic nature of communication networks.

### # Features of NS2.

- (\*) It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, HTTP and DSR.
- (\*) It simulates both wired and wireless network.
- (\*) It is primarily UNIX based.
- (\*) Uses Tcl as its scripting language.
- (\*) OTcl - Object Oriented support.
- (\*) TclCL - C++ and OTcl linkage.
- (\*) Discrete event scheduler.

### # Basic Architecture of NS2.

- (\*) It consists of two key languages - C++ and OTcl. C++ defines the internal mechanism of the simulation object and the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. C++ & OTcl are linked using TclCL.

Teacher's Signature \_\_\_\_\_

20/01/2020

## # DIFFERENCES BETWEEN NS2 & NS3.

Sr. No.	Property	NS2	NS3
1.	Support	NS2 is not actively maintained and no longer support provided	NS3 is actively maintained with very good support. It is not backward compatible with NS2
2.	Architecture	NS2 is made with C++ and OTcl language. Tcl is used as scripting language for network simulation	NS3 is made with C++ and Python. Instead of scripting language, there is object oriented language and another option is Python
3.	Programming Language	C++, OTcl, Tcl script	C++, Python
4.	Recompilation	Recompilation is long process and many times fail	Recompilation is fast process and done in single command
5.	Post Analysis Supported File Formats	1. .nam for network animation 2. .tr for trace analysis 3. .xg for xgraph	1. .xml for network animation 2. .tr for trace analysis 3. .pcap for wireshark analysis
6.	Network Visualization	nam as network animator	1. PyViz as python visualizer 2. NetAnim as network animator
7.	Others	Only simulation	1. With simulation there is emulation also 2. DCE environment is provided which is used to simulate Linux kernel network stack

## Downloading and Installation of NS2 :

20/01/2020

Following are the steps to install NS2 on UNIX based system :

### **Downloading :**

Download ns-allinone-2.35 from here :

<https://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/>

It gets downloaded into your '/home/your\_user\_name/Downloads' directory.

Copy it to /opt folder by the following command :

```
cp /home/user_name/Downloads/ns-allinone-2.35.tar.gz /opt/
```

### **Installation :**

Open Terminal and Execute the following commands :

1. sudo apt-get update
2. sudo apt-get dist-upgrade
3. sudo apt-get update
4. sudo apt-get gcc
5. sudo apt-get install build-essential autoconf automake
6. sudo apt-get install tcl8.5-dev tk8.5-dev
7. sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev

Execute the following commands for Extraction :

1. tar -zvxf ns-allinone-2.35.tar.gz
2. cd ns-allinone-2.35
3. ./install

Open bashrc file to Set the Environment Variables. Execute following command on terminal :

```
sudo gedit ~/.bashrc
```

Copy the following lines at the end of the file :

```
# LD_LIBRARY_PATH  
OTCL_LIB=/opt/ns-allinone-2.35/otcl-1.14/  
NS2_LIB=/opt/ns-allinone-2.35/lib/  
USR_Local_LIB=/usr/local/lib/  
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$USR_Local_LIB  
# TCL_LIBRARY  
TCL_LIB=/opt/ns-allinone-2.35/tcl8.5.10/library/  
USR_LIB=/usr/lib/  
export TCL_LIBRARY=$TCL_LIBRARY:$TCL_LIB:$USR_LIB
```

20/01/2020

```
# PATH  
XGRAPH=/opt/ns-allinone-2.35/xgraph-12.2:/opt/ns-allinone-  
2.35/bin:/opt/ns-allinone-2.35/tcl8.5.10/unix:/opt/ns-allinone-  
2.35/tk8.5.10/unix/  
NS=/opt/ns-allinone-2.35/ns-2.35/  
NAM=/opt/ns-allinone-2.35/nam-1.15/  
export PATH=$PATH:$XGRAPH:$NS:$NAM  
#----
```

At last Execute the following commands :

1. source ~/.bashrc
2. ns

If you receive “%” sign, it means that NS is installed successfully.

## # Introduction to NS3.

- X) Similar to NS2, NS3 is also an open sourced discrete event network simulator which targets research and education use. NS3 is licensed under the GNU GPLv2 license and is available for research and development. NS3 is designed to replace the current popular NS2. However, NS3 is not an updated version of NS2 since NS3 is a new simulator and it is not backward compatible with NS2. The basic idea of NS3 comes from several different network simulators including NS2, YANs and GTNetS.

## # Introduction to Discrete Event Simulation.

- ① Topology Definition : to ease the creation of basic facilities and define their interrelationships, NS3 has a system of containers and helpers that facilitates this process.
- ② Model Development : models are added to simulation ( UDP, IPv4) most of the time is done using helpers.
- ③ Node & Link configuration : models set their default values most of the time this is done using the attribute system.
- ④ Execution : simulation facilitates generate events, data requested by the user is logged.

Teacher's Signature \_\_\_\_\_

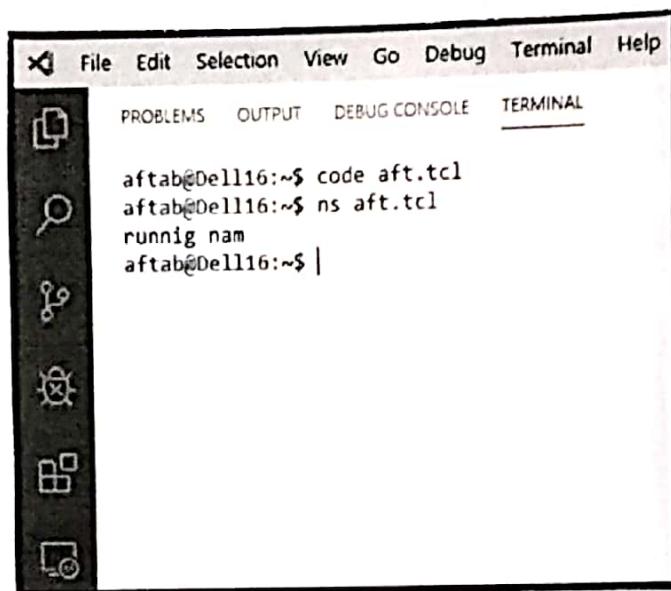
- (\*) Performance Analysis: after the simulation is finished and data is available as a time stamped event trace. this data can be statistically analysed with tools like R to draw conclusions.
- (\*) Graphical visualization: raw data collected in a simulation can be graphed using tools like gnuplot, matplotlib or XGRAPH.

## # Discrete Event Simulation Tools:

- (1) NS2 [Network Simulator 2]. }
- (2) NS3 [Network Simulator 3]. } explained earlier.
- (3) OMNet++.
- (\*) It is also a public source, component based network simulator with GUI support. Components are also called modules and are programmed in C++. The components are then assembled into larger components and modules by using a high level language. Its function is similar to that of OTcl in NS2 and Python in NS3. Modules are reusable and can be combined in various ways which is one of the main feature of OMNet++.

Teacher's Signature \_\_\_\_\_

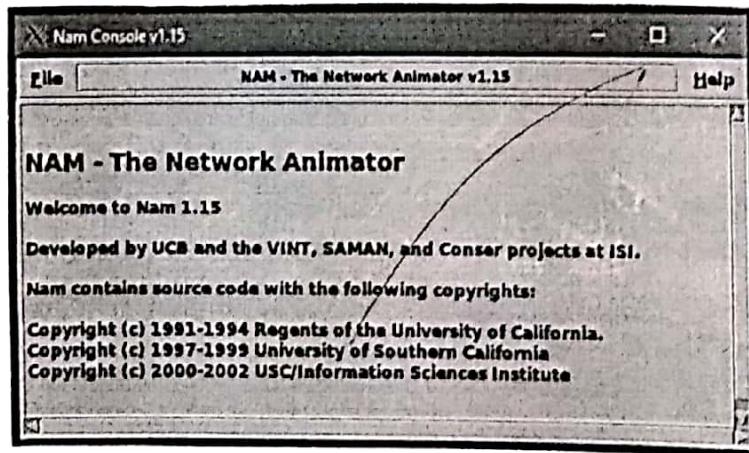
27/01/2020



A screenshot of a terminal window titled "Terminal". The window has a menu bar with File, Edit, Selection, View, Go, Debug, Terminal, and Help. Below the menu is a tab bar with PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, where TERMINAL is selected. The main area shows the following command-line session:

```
aftab@Dell16:~$ code aft.tcl
aftab@Dell16:~$ ns aft.tcl
runnig nam
aftab@Dell16:~$ |
```

# TERMINAL.



# NAM CONSOLE.

## Wired Network Simulation (4 Nodes)

# Aim - Write a Program for Wired Network simulation between 3 Source Nodes and 1 Destination Node.

# Code -

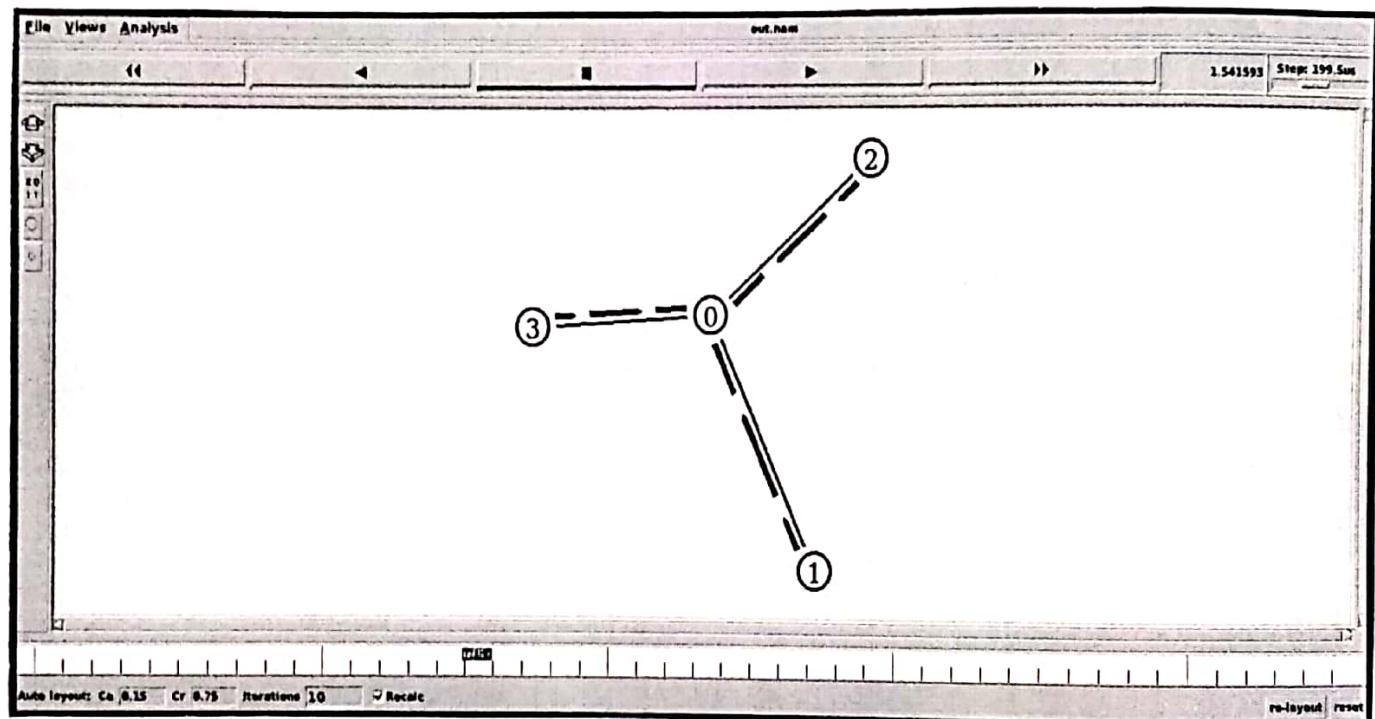
```

set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam.out &
    exit 0
}
set n0 n0[$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n1 $n0 1Mb 1ms DropTail
$ns duplex-link $n2 $n0 1Mb 1ms DropTail
$ns duplex-link $n3 $n0 1Mb 1ms DropTail
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set udp2 [new Agent/UDP]
$ns attach-agent $n2 $udp2
set udp3 [new Agent/UDP]

```

Teacher's Signature \_\_\_\_\_

27/01/2020



# OUTPUT WINDOW.

```

$ns attach-agent $n3 $udp3
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize - 500
$cbr1 set interval - 0.005
$cbr1 attach-agent $udp1
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize - 500
$cbr2 set interval - 0.005
$cbr2 attach-agent $udp2
set cbr3 [new Application/Traffic/CBR]
$cbr3 set packetSize - 500
$cbr3 set interval - 0.005
$cbr3 attach-agent $udp3
set null0 [new Agent/Null]
$ns attach-agent $n0 $null0
$ns connect $udp1 $null0
$ns connect $udp2 $null0
$ns connect $udp3 $null0
$ns at 0.5 "$cbr1 start"
$ns at 4.5 "$cbr1 stop"
$ns at 0.5 "$cbr2 start"
$ns at 4.5 "$cbr2 stop"
$ns at 0.5 "$cbr3 start"
$ns at 4.5 "$cbr3 stop"
$ns at 5.0 "finish"
puts "running nam"
$ns run

```

# result - Program is executed successfully.

Teacher's Signature \_\_\_\_\_

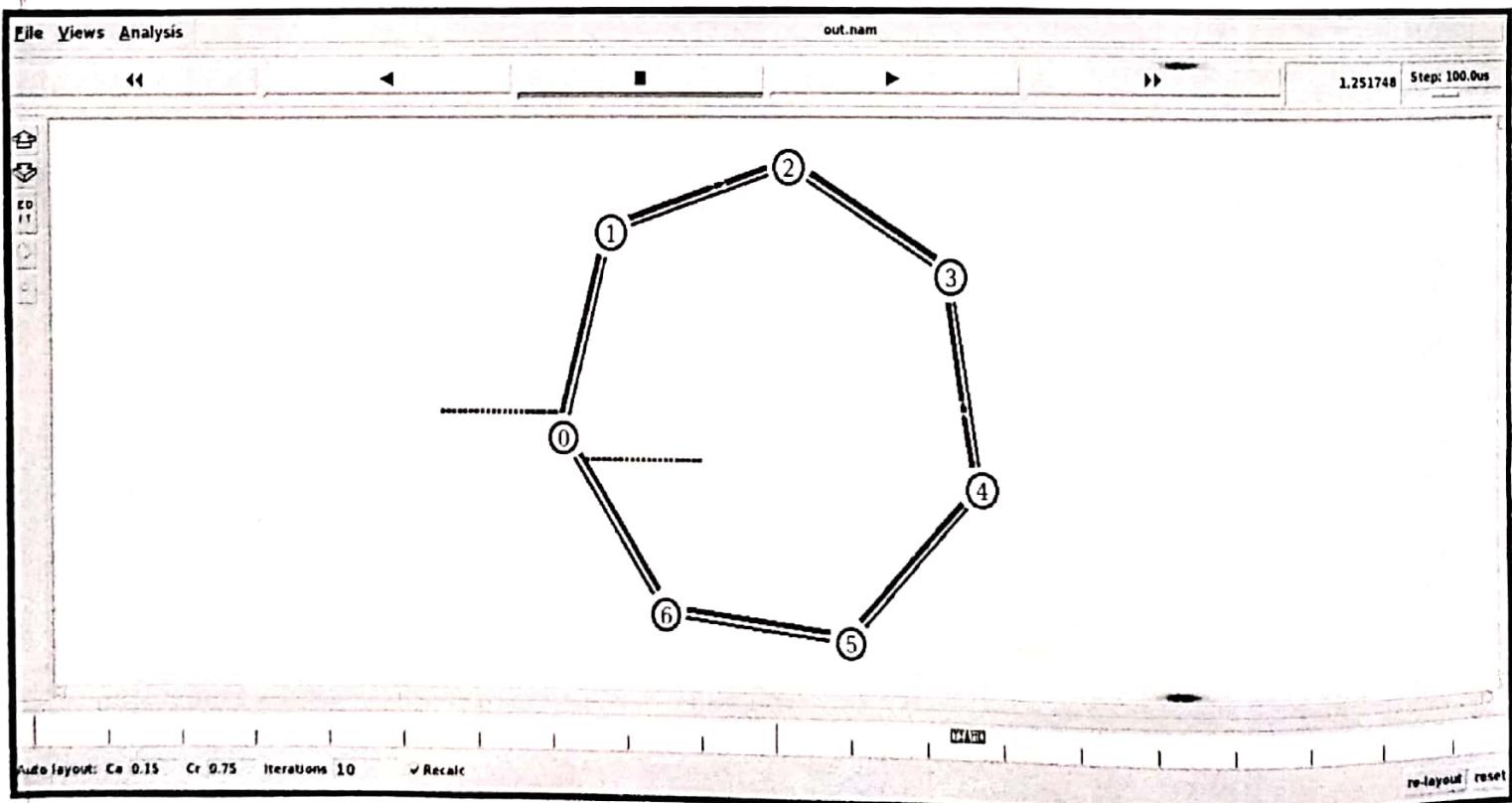
12/02/2020

The screenshot shows a terminal window with the following content:

```
aftab@Dell16:~$ code lab3.tcl
aftab@Dell16:~$ ns lab3.tcl
running nam
aftab@Dell16:~$ |
```

The terminal has a dark theme with icons on the left side. The tabs at the top are PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL being the active tab.

## # TERMINAL



## # OUTPUT WINDOW

Ring Topology:

# Aim- Write a Program for Ring Topology in a Network.

# Code-

```

set ns [new Simulator]
$ns rtproto DV
set nf [open out.nam w]
$ns monitor -all $nf
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam
    exit 0
}
for {set i 0} {$i < 7} {incr i} {
    set n($i) [ $ns node]
}
for {set i 0} {$i < 7} {incr i} {
    $ns duplex-link $n($i) $n([expr ($i+1)%7]) 512Kb 5ms
    dropTail
    $ns duplex-link -op $n(0) $n(1) queuePos 1
    $ns duplex-link -op $n(0) $n(6) queuePos 1
    set udp0 [new Agent/UDP]
    $ns attach-agent $n(0) $udp0
    set null0 [new Agent/Null]
    $ns attach-agent $n(3) $null0
}

```

Teacher's Signature \_\_\_\_\_

```

$ns connect $nfp0 $null0
set cbr0 [new Application/Traffic/TCBR]
$cbr set packetSize 1024
$cbr set interval 0.01
$cbr attach-agent $nfp0
$ns etmodel -at 0.4 down $n(2).$n(5)
$ns etmodel -at 1.0 up $n(2)$n(3)
$ns at 0.01 "$cbr0 start"
$ns at 1.5 "$cbr0 stop"
$ns at 2.0 "finish"
putl "running now"
$ns run

```

# Result - Program is executed successfully.

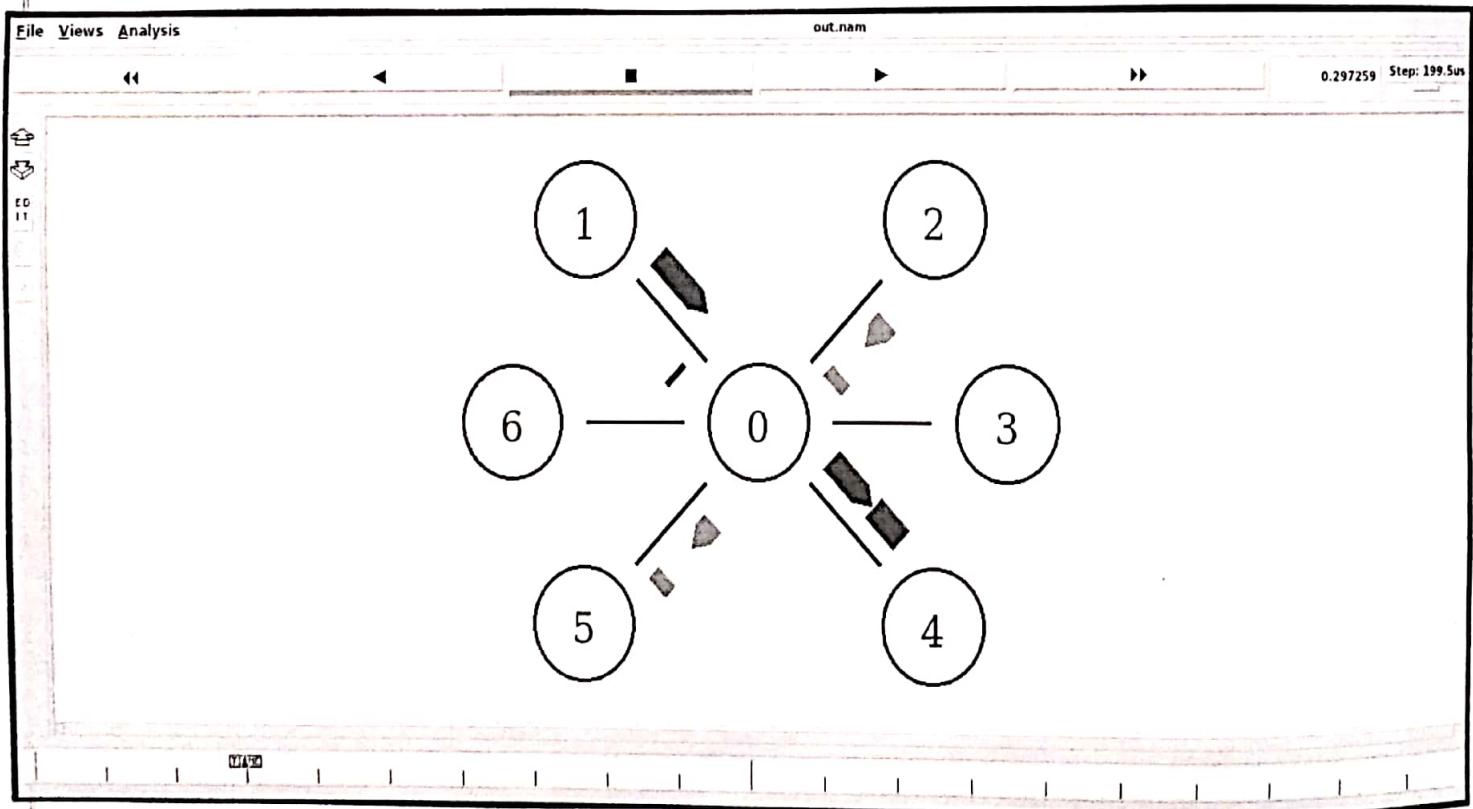
Teacher's Signature \_\_\_\_\_

12/02/2020

A screenshot of a terminal window titled "TERMINAL". The window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal content shows the following command-line interaction:

```
aftab@Dell16:~$ code lab4.tcl
aftab@Dell16:~$ ns lab4.tcl
running nam
aftab@Dell16:~$ |
```

### # TERMINAL



### # OUTPUT WINDOW

Star Topology.

# Aim- Write a program for Star Topology in a Network.

# Code-

```

set ns [new Simulator]
$ns color1 blue
$ns color2 red
$ns setproto DV
set nf [open out.nam w]
$ns montrace-all $nf
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam.out nam
    exit 0
}
for {set i 0} {$i < 7} {incr i} {
    set n($i) [$ns node]
}
for {set i 1} {$i < 7} {incr i} {
    $ns duplex-link $n(0) $n($i) 512kb 10ms SFQ
}
$ns duplex-link -op $n(0) $n(1) orient left-up
$ns duplex-link -op $n(0) $n(2) orient right-up
$ns duplex-link -op $n(0) $n(3) orient right
$ns duplex-link -op $n(0) $n(4) orient right-down

```

Teacher's Signature \_\_\_\_\_

```

$ns duplex-link $n(0) $n(5) orient left-down
$ns duplex-link $n(0) $n(6) orient left
set tcp0 [new Agent/TCP]
$tcp0 set class -1
$ns attach-agent $n(1) $tcp0
set sink0 [new Agent/TCP Sink]
$ns attach-agent $n(4) $sink0
$ns connect $tcp0 $sink0
set udp0 [new Agent/UDP]
$udp0 set class -2
$ns attach-agent $n(2) $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set rate 256Kb
$cbr0 attach-agent $udp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns rtmodel-at 0.5 down $n(0) $n(5)
$ns rtmodel-at 0.9 up $n(0) $n(5)
$ns rtmodel-at 0.7 down $n(0) $n(4)
$ns rtmodel-at 1.2 up $n(0) $n(4)
$ns at 0.1 "$ftp0 start"
$ns at 1.5 "$ftp0 stop"
$ns at 0.2 "$cbr0 start"
$ns at 1.3 "$cbr0 stop"
$ns at 2.0 "finish"
puts "running now"
$ns flush

```

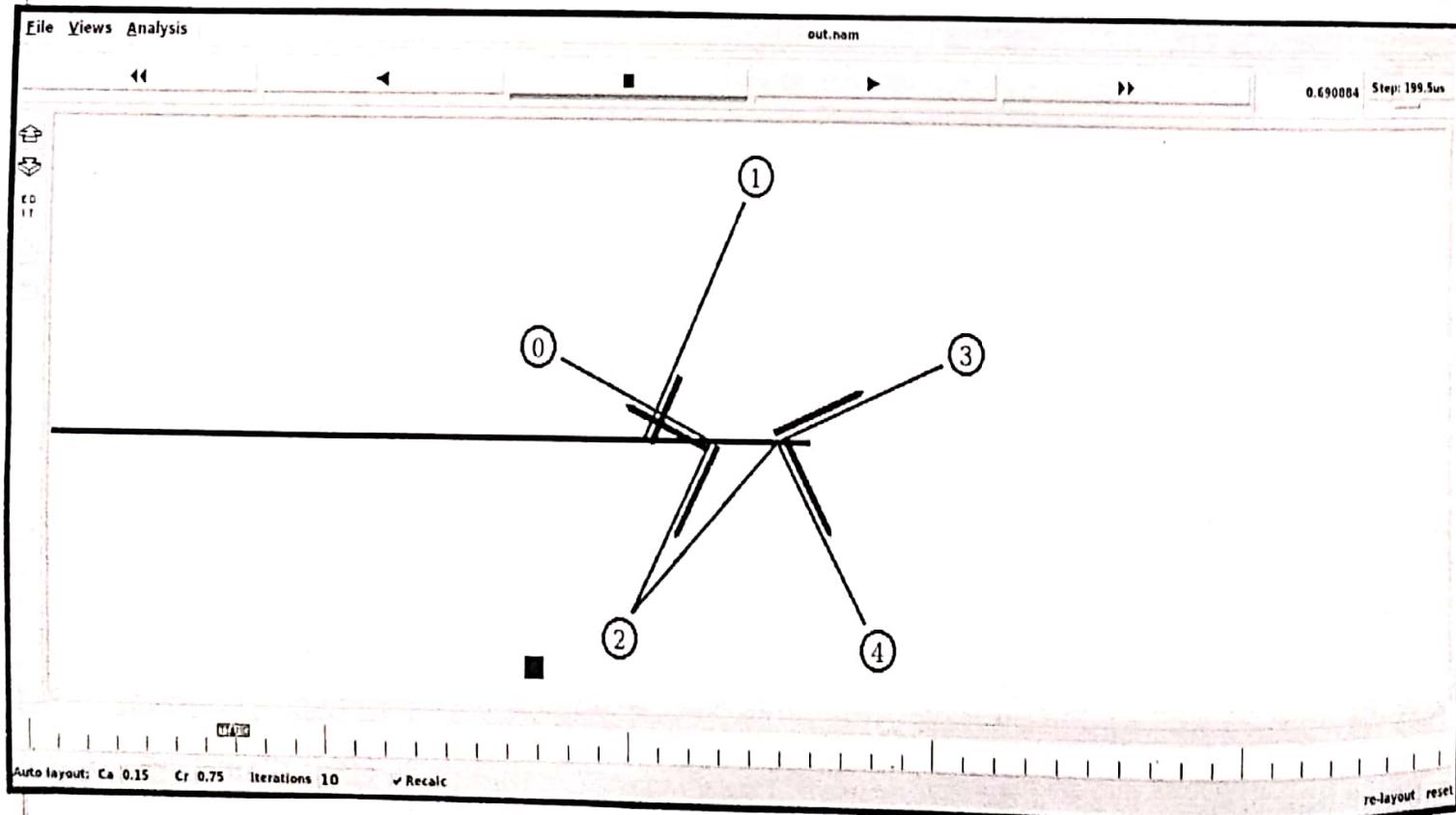
Teacher's Signature \_\_\_\_\_

# Result- program is executed successfully.

12/02/2020

```
aftab@Dell16:~$ code lab5.tcl
aftab@Dell16:~$ ns lab5.tcl
warning: no class variable LanRouter::debug_
see tcl-object.tcl in tclcl for info
running nam
aftab@Dell16:~$ |
```

## # TERMINAL



## # OUTPUT WINDOW

Bus Topology:

# Aim- Write a program for Bus Topology in a Network.

# Code-

```

set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish { } {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam&
    exit 0
}
set n0[$ns node]
set n1[$ns node]
set n2[$ns node]
set n3[$ns node]
set n4[$ns node]
set lno0[$ns newlorn "fno $n1 $n2 $n3 $n4" "0.5Mb 40ms LL
    Queue/DropTail MAC/Csmma/Cd channel]
set tpo[new Agent/TCP]
$tpo set class_ 1
$ns attach-agent $n1 $tpo
set sink0[new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tpo $sink0

```

Teacher's Signature \_\_\_\_\_

```
set cbr0 [new Application/Traffic(CBR)]
$cbr0 set packetSize -500
$cbr0 set interval -0.01
$cbr0 attach-agent $step
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
puts "running now"
$ns run
```

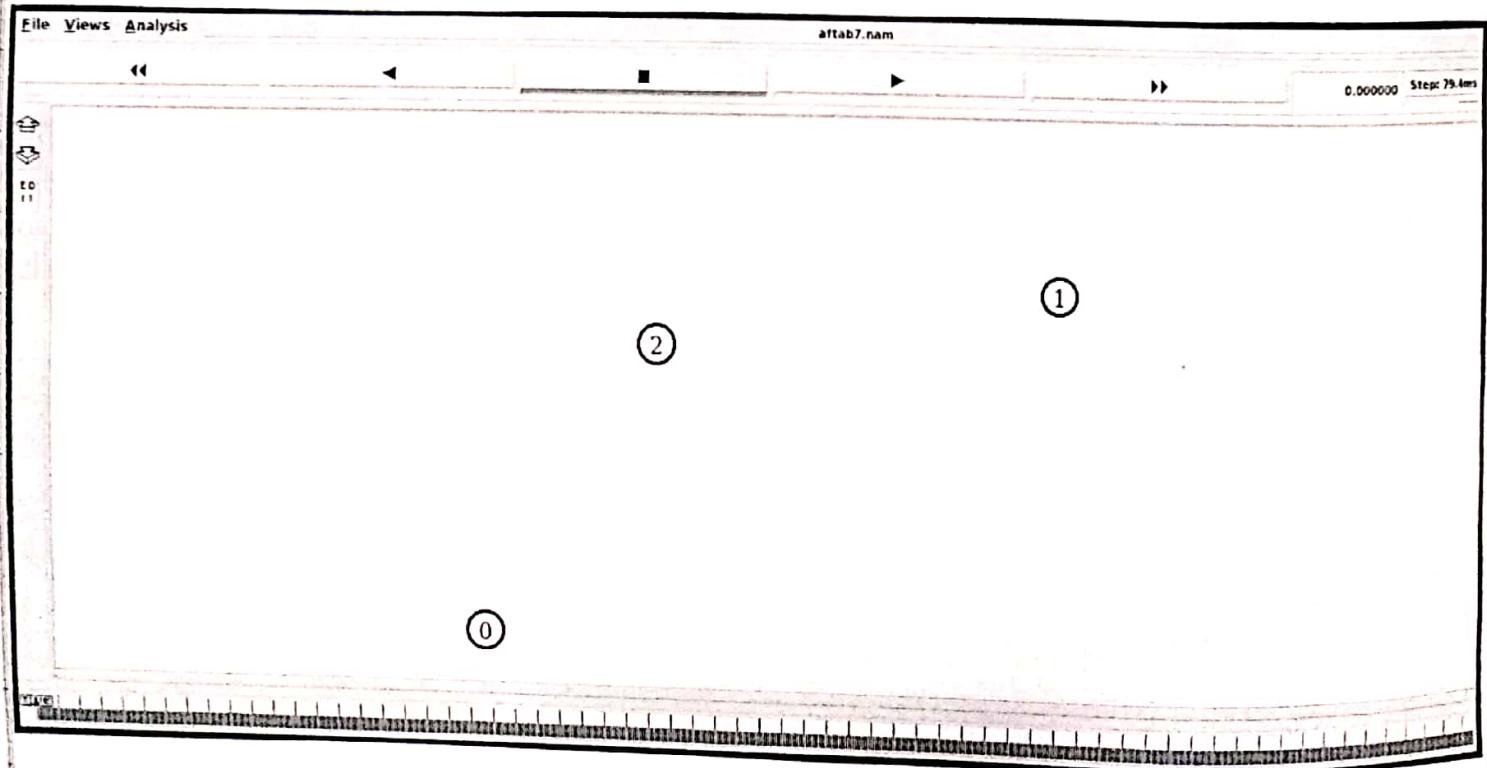
# Result - Program is executed successfully.

Teacher's Signature \_\_\_\_\_

19/02/2020

```
aftab@Dell16:~$ code wireless.tcl
aftab@Dell16:~$ ns wireless.tcl
num_nodes is set 3
warning: Please use -channel as shown in tcl/ex/wireless-
INITIALIZE THE LIST xListHead
running nam
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5, distCST_ = 550.0
SORTING LISTS ...DONE!
end simulation
aftab@Dell16:~$ |
```

## # TERMINAL



## # SCREENSHOT ①

### Wireless Network Simulation (3 Nodes).

# Aim- Write a program for wireless network simulation between 3 nodes.

# Code-

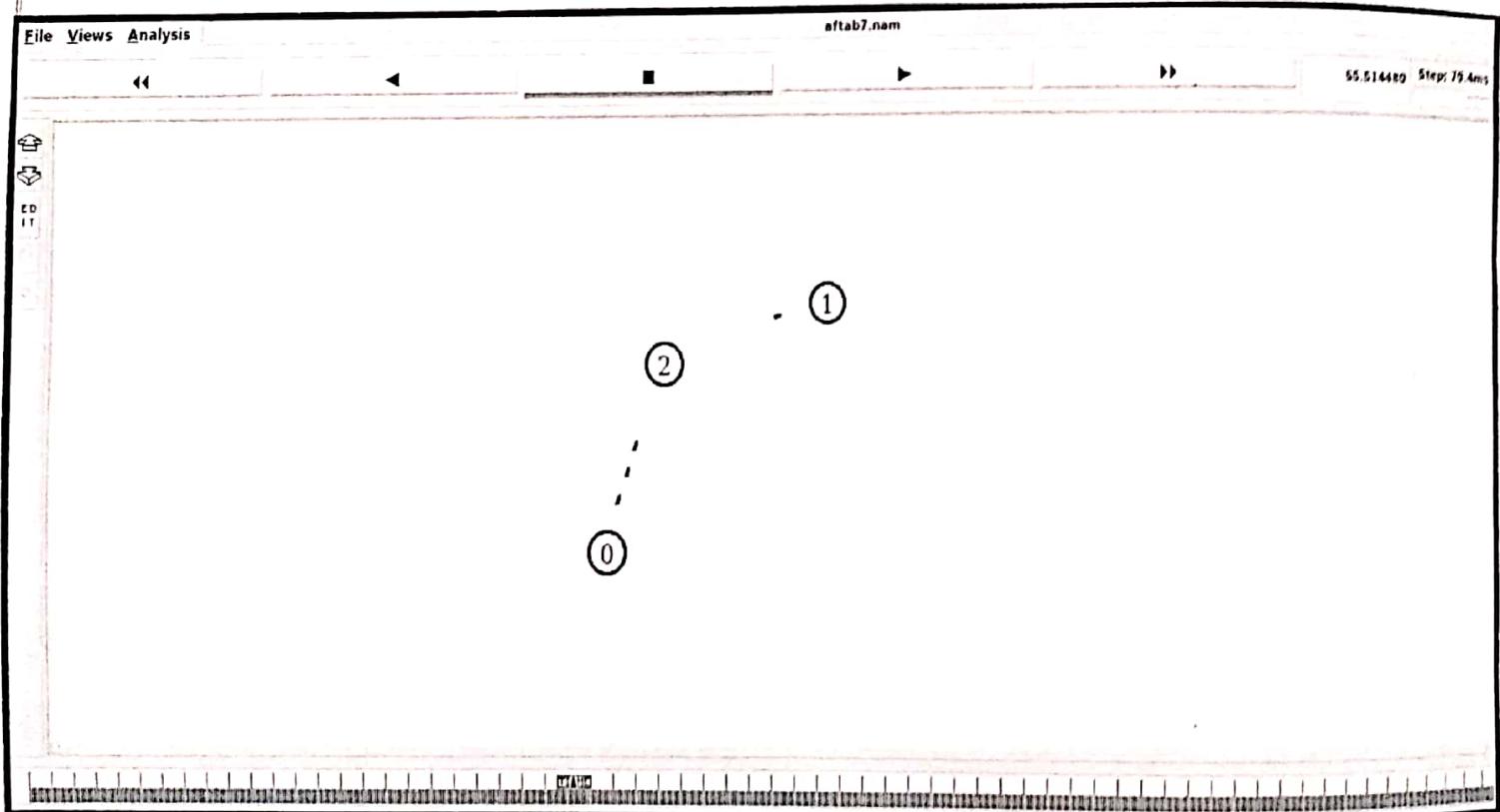
```

set val(chan) Channel/WirelessChannel;
set val(prop) Propagation/TwoRayGround;
set val(netif) Phy/WirelessPhy;
set val(mac) Mac/802-11;
set val(ifq) Queue/DropTail/PriQueue;
set val(ll) LL;
set val(ant) Antenna/OmniAntenna;
set val(ifqlen) 50;
set val(nn) 3;
set val(rp) DSDV;
set val(x) 500;
set val(y) 400;
set val(stop) 150;
set ns [new Simulator]
set tracefd [open xwrls-simple.tr w]
set windowxstimer [open xwrls-simple-win.tr w]
set nwidbocd [open xfbabt.nam w]
$ns trace-all $tracefd
$ns nwidbocd-wireless $nwidbocd $val(x) $val(y)
set topo [new Topology]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

```

Teacher's Signature \_\_\_\_\_

19/02/2020



#SCREENSHOT · (2)

@ time = 55.51 sec

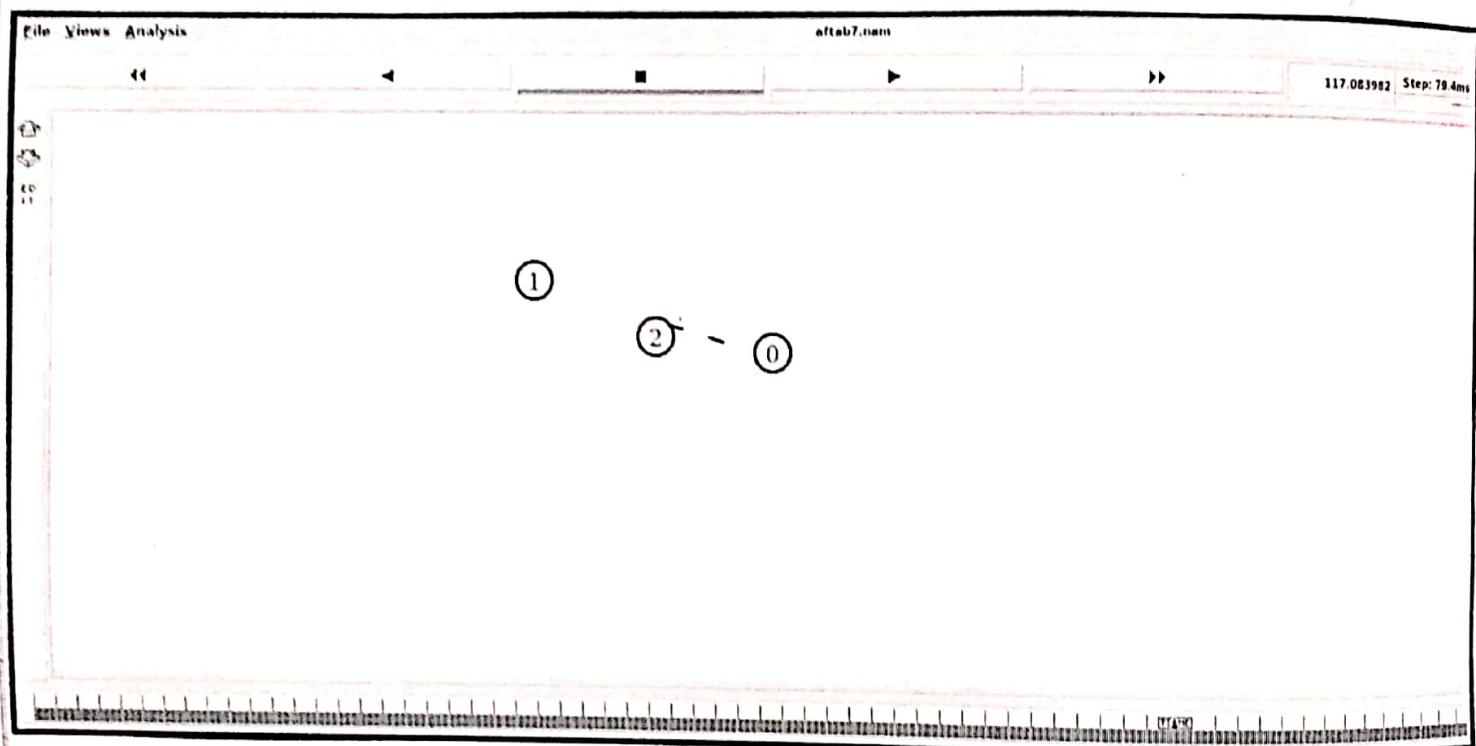
$\$ns$  node - config - adhocRouting \$val(rif)  
 - ll-type \$val(ll)  
 - mac-type \$val(mac)  
 - ifq-type \$val(ifq)  
 - ifqlen \$val(ifqlen)  
 - out-type \$val(out)  
 - prop-type \$val(prop)  
 - phy-type \$val(phy) (notif)  
 - channel-type \$val(chan)  
 - topo-instance \$topo  
 - agentTrace ON  
 - routerTrace ON  
 - mocTrace OFF  
 - movementTrace ON

for {set i 0; \$i < \$val(nn); \$i++} {  
 set node-(\$i) [\$ns node]  
}

$\$node-(0)$  set X - 5.0  
 $\$node-(0)$  set Y - 5.0  
 $\$node-(0)$  set Z - 0.0  
 $\$node-(1)$  set X - 490.0  
 $\$node-(1)$  set Y - 285.0  
 $\$node-(1)$  set Z - 0.0  
 $\$node-(2)$  set X - 150.0  
 $\$node-(2)$  set Y - 240.0  
 $\$node-(2)$  set Z - 0.0  
 $\$ns at 10.0 " \$node(0) setdest 250.0 250.0 3.0 "$   
 $\$ns at 15.0 " \$node(1) setdest 45.0 285.0 5.0 "$   
 $\$ns at 110.0 " \$node(2) setdest 480.0 300.0 5.0 "$

Teacher's Signature \_\_\_\_\_

19/02/2020



# SCREENSHOT (3)

@ time = 117.08 sec.

```

set tcp [new Agent/TCP/Newreno]
$tcp set class-2
set sink [new Agent/TCPSink]
$ns attach-agent $node-(0) $tcp
$ns attach-agent $node-(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
proc plotWindow { tpsource file } {
    global ns
    set time 0.01
    set now [$ns now]
    set wond [$tpsource set wond_]
    puts $file "$now $wond"
    $ns at [expr $now + $time] "plotWindow $tpsource $file"
    $ns at 10.1 "plotWindow $tcp $window<Time2"
    for {set i 0} {$i < $val(n)} {incr i} {
        $ns initial-node-pos $node-($i) 30
    }
    for {set i 0} {$i < $val(n)} {incr i} {
        $ns at $val(stop) "$node-($i) reset";
    }
    $ns at $val(stop) "$ns now-end-wireless $val(stop)"
    $ns at $val(stop) "stop"
    $ns at 150.01 "puts \\"end simulation\\"; $ns halt"
    proc stop {} {
}

```

Teacher's Signature \_\_\_\_\_

global ns tracefd nomtrace

\$ ns flush-trace

close \$tracefd

close \$nomtrace

exec nom aflat7.nom &

}

- puts "running nom"

\$ ns run

# result - Program is executed successfully.

Teacher's Signature \_\_\_\_\_

13/03/2020

(1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12)

event	time	from node	to node	pkt type	pkt size	fogs	fid	src addr	dst addr	seq num	pkt id

#FORMAT OF TRACE STRING [12 FIELDS]:

## Trace Format of NS2:

# Aim - Show & Analyze the Trace Format of NS2.

# Trace Files & Description -

(\*) the file written by an application to store coverage information or overall network information in NS2, it is known as trace file. its extension is .tr. for all the programs written in NS2, a unique trace file is generated by user.

# Format of Trace File -

(1) Event or Type Identifier-

+ : packet enqueue event

- : packet deque event

r : packet reception event

d : packet drop event

c : packet collision at the MAC level

(2) Time - at which the packet tracing string is created.

(3)(4) source and Destination Node - source and destination ID's of tracing objects.

(5) Packet Name - name of the packet type.

(6) Packet Size - size of packet in bytes.

(7) Flags - 7 digit flag string.

Teacher's Signature \_\_\_\_\_

11/03/2020

```
File Edit Selection View Go Run Terminal Help
outstr - Visual Studio Code
1 + 0.1 e 1 tcp 40 ----- 0.0.0 1.0 2 0
2 - 0.1 e 1 tcp 40 ----- 0.0.0 1.0 0 0
3 ^ 0.10416 0 1 tcp 40 ----- 0.0.0 1.0 0 0
4 + 0.10416 1 0 ack 40 ----- 0 1.0 0.0 0 1
5 - 0.10416 1 0 ack 40 ----- 0 1.0 0.0 0 1
6 ^ 0.10832 1 0 ack 40 ----- 0 1.0 0.0 0 1
7 + 0.10832 0 1 tcp 1040 ----- 0.0.0 1.0 1 2
8 - 0.10832 0 1 tcp 1040 ----- 0.0.0 1.0 1 2
9 ^ 0.10832 0 1 tcp 1040 ----- 0.0.0 1.0 2 3
10 - 0.11248 0 1 tcp 1040 ----- 0.0.0 1.0 2 3
11 ^ 0.11648 0 1 tcp 1040 ----- 0.0.0 1.0 1 2
12 + 0.11648 1 0 ack 40 ----- 0 1.0 0.0 1 4
13 - 0.11648 1 0 ack 40 ----- 0 1.0 0.0 1 4
14 ^ 0.12064 0 1 tcp 1040 ----- 0.0.0 1.0 2 3
15 + 0.12064 1 0 ack 40 ----- 0 1.0 0.0 2 5
16 - 0.12064 1 0 ack 40 ----- 0 1.0 0.0 2 5
17 ^ 0.12064 1 0 ack 40 ----- 0 1.0 0.0 1 4
18 + 0.12064 0 1 tcp 1040 ----- 0.0.0 1.0 3 6
19 - 0.12064 0 1 tcp 1040 ----- 0.0.0 1.0 3 6
20 + 0.12064 0 1 tcp 1040 ----- 0.0.0 1.0 4 7
21 ^ 0.1246 1 0 ack 40 ----- 0 1.0 0.0 2 5
22 + 0.1246 0 1 tcp 1040 ----- 0.0.0 1.0 5 8
23 + 0.1246 0 1 tcp 1040 ----- 0.0.0 1.0 6 9
24 - 0.1248 0 1 tcp 1040 ----- 0.0.0 1.0 4 7
25 ^ 0.1288 0 1 tcp 1040 ----- 0.0.0 1.0 3 6
26 + 0.1288 1 0 ack 40 ----- 0 1.0 0.0 3 10
27 - 0.1288 1 0 ack 40 ----- 0 1.0 0.0 3 10
28 ^ 0.12896 0 1 tcp 1040 ----- 0.0.0 1.0 5 8
29 - 0.13296 0 1 tcp 1040 ----- 0.0.0 1.0 4 7
30 + 0.13296 1 0 ack 40 ----- 0 1.0 0.0 4 11
31 - 0.13296 1 0 ack 40 ----- 0 1.0 0.0 4 11
32 ^ 0.13296 1 0 ack 40 ----- 0 1.0 0.0 3 10
33 + 0.13296 0 1 tcp 1040 ----- 0.0.0 1.0 7 12
```

## # WIRED NETWORK TRACE FILE.

"-" : disable

1st = "E" : ECN (Explicit Congestion Notification) echo is enabled.

2nd = "P" : priority in the IP header is enabled.

3rd = Not in use

4th = "A" : congestion action

5th = "E" : congestion has occurred

6th = "F" : ~~FTP~~ the TCP fast start is used

7th = "N" : Explicit Congestion Notification (ECN) is on.

(8) Flow ID - it shows the ID of packet.

(9) (10) Source and Destination Address - the format of these two fields is "a:b", where "a" is the address and "b" is the port.

(11) Sequence Number - it shows the sequence number.

(12) Packet Unique ID - the last field shows the unique ID of packet.

### # Trace File Description -

s - send packet

r - received packet

d - packet dropped

f - packet forwarded

c - collision of packet at MAC level

t - time at which packet tracing started

Hs - ID of hop

Hd - ID of next hop towards destination

Ni - Node ID

Teacher's Signature \_\_\_\_\_

11/03/2020

xwrls-simple.tr - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

1 s 0.032621055 \_1\_RTR --- 0 message 32 [0 0 0 0] ----- [1:255 -1:255 32 0]
2 s 0.178591260 \_2\_RTR --- 1 message 32 [0 0 0 0] ----- [2:255 -1:255 32 0]
3 s 1.113421886 \_2\_RTR --- 2 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
4 " 10.00000 0 (5.00, 5.00, 0.00), (250.00, 250.00), 3.00
5 s 10.000000000 \_0\_AGT --- 3 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
6 r 10.000000000 \_0\_RTR --- 3 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
7 s 12.530838300 \_0\_RTR --- 4 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
8 s 13.000000000 \_0\_AGT --- 5 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
9 r 13.000000000 \_0\_RTR --- 5 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
10 s 13.630359915 \_2\_RTR --- 6 message 32 [0 0 0 0] ----- [2:255 -1:255 32 0]
11 s 14.288428760 \_1\_RTR --- 7 message 32 [0 0 0 0] ----- [1:255 -1:255 32 0]
12 " 15.00000 1 (490.00, 285.00, 0.00), (45.00, 285.00), 5.00
13 s 19.000000000 \_0\_AGT --- 8 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
14 r 19.000000000 \_0\_RTR --- 8 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
15 s 25.369352837 \_0\_RTR --- 9 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
16 r 25.370532808 \_2\_RTR --- 9 message 32 [0 ffffffff 0 800] ----- [0:255 -1:255 32 0]
17 s 26.306541963 \_2\_RTR --- 10 message 32 [0 0 0 0] ----- [2:255 -1:255 32 0]
18 r 26.337502727 \_0\_RTR --- 10 message 32 [0 ffffffff 2 800] ----- [2:255 -1:255 32 0]
19 s 27.274269218 \_1\_RTR --- 11 message 32 [0 0 0 0] ----- [1:255 -1:255 32 0]
20 s 31.000000000 \_0\_AGT --- 12 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
21 r 31.000000000 \_0\_RTR --- 12 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
22 s 37.3799295458 \_2\_RTR --- 13 message 32 [0 0 0 0] ----- [2:255 -1:255 32 0]
23 r 37.381016113 \_0\_RTR --- 13 message 32 [0 ffffffff 2 800] ----- [2:255 -1:255 32 0]
24 s 37.381016233 \_1\_RTR --- 13 message 32 [0 ffffffff 2 800] ----- [2:255 -1:255 32 0]
25 s 37.425085764 \_0\_RTR --- 14 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
26 r 37.427046440 \_2\_RTR --- 14 message 32 [0 ffffffff 0 800] ----- [0:255 -1:255 32 0]
27 s 38.426532792 \_0\_RTR --- 15 message 32 [0 0 0 0] ----- [0:255 -1:255 32 0]
28 r 38.427533459 \_2\_RTR --- 15 message 32 [0 ffffffff 0 800] ----- [0:255 -1:255 32 0]
29 s 40.043941990 \_1\_RTR --- 16 message 32 [0 0 0 0] ----- [1:255 -1:255 32 0]
30 r 40.046982728 \_2\_RTR --- 16 message 32 [0 ffffffff 1 800] ----- [1:255 -1:255 32 0]
31 s 40.493686470 \_2\_RTR --- 17 message 32 [0 0 0 0] ----- [2:255 -1:255 32 0]
32 r 40.494587098 \_0\_RTR --- 17 message 32 [0 ffffffff 2 800] ----- [2:255 -1:255 32 0]
33 r 40.495597163 \_1\_RTR --- 17 message 32 [0 ffffffff 2 800] ----- [2:255 -1:255 32 0]

# WIRELESS NETWORK TRACE FILE.

$N_x, N_y, N_z$  - Coordinates of the nodes situated

$Ne$  - Node energy level

$NI$  - Trace level

$Nw$  - Reason of the event

AGT - Agent

RTR - Routing

END - DROP End of simulation

COL - DROP MAC collision

DUP - DROP MAC Duplicate

DERR - DROP MAC Packet Error

RET - DROP MAC Retry Count Exceed

STA - DROP MAC Invalid state

BSY - DROP MAC Busy

NRTE - DROP RTR - NO ROUTE

LOOP - DROP RTR ROUTE Loop

TTL - DROP RTR TTL has reached zero

TOUT - DROP RTR QTIME OUT Expired

l<sub>s</sub> - Source address of source port

l<sub>d</sub> - Destination address of destination port

l<sub>l</sub> - Packet Size

l<sub>f</sub> - Flow ID

l<sub>i</sub> - Unique ID

l<sub>v</sub> - TTL value next hop info



### MAC LAYER INFORMATION -

M<sub>a</sub> - MAC Layer Duration

M<sub>d</sub> - Destination Ethernet Address

M<sub>s</sub> - Source Ethernet Address

M<sub>t</sub> - Ethernet Type

Teacher's Signature \_\_\_\_\_

(\*)PACKET INFORMATION -

-Parp - address resolution protocol

-Po - ARP Request / Reply

Pm - Source MAC Address

Ps - Source Address

Pa - Destination MAC Address

Pd - Destination Address

Pn - Nodes Transversed

Pq - F log

Pi - Route Request Sequence Number / Sequence Number

Pp - F log

Pq - Reply Length

Pc - Src of source routing

Pw - Errors Report F log

Pc - Report to whom

Pb - Link error from link a to link b

-P\_cbr - CBR data

Pf - how many level packet leave

Po - Optimal number of Forward

-PTCP - TCP flow

-Ps - sequence number

Pu - acknowledgement

Pf - Packet Failure

# result - Trace format of NS2 is Analyzed Successfully.

Teacher's Signature