Veermata Jijabai Technological Institute, Mumbai 400019

Assignment No.: 03

Objective: Recreate experiment No.2 (Simulating a Local Area Network)**.**

Name : Kiran K Patil

Enrollment No.: 211070904

Branch : Computer Engineering

Batch: D

Theory :

Network simulation (NS) is one of the types of simulation, which is used to simulate the networks such as in MANETs, VANETs, etc. It provides simulation for routing and multicast protocols for both wired and wireless networks. NS is licensed for use under version 2 of the GNU (General Public License) and is popularly known

as NS2. It is an object-oriented, discrete event-driven simulator written in C++ and Otcl/Tcl.

NS-2 can be used to implement network protocols such as TCP and UDP, traffic source behavior such as FTP, Telnet, Web, CBR, and VBR, router queues management mechanism such as Drop Tail, RED, and CBQ, routing algorithms, and many more. In ns2, C++ is used for detailed protocol implementation and Otcl is used for the setup. The compiled C++ objects are made available to the Otcl interpreter and in this way, the ready-made C++ objects can be controlled from the OTcl level.

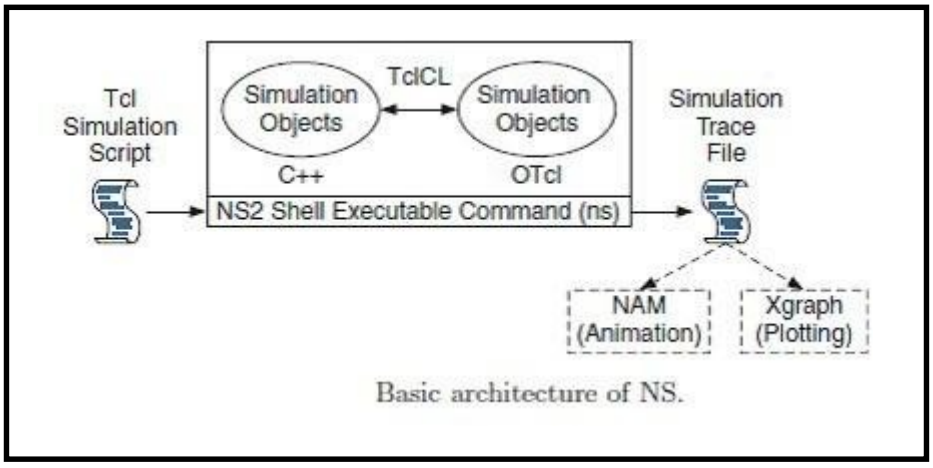
NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

Features of NS2

1. It is a discrete event simulator for networking research.
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.
3. It simulates wired and wireless network.
4. It is primarily Unix based.
5. Uses TCL as its scripting language.
6. Otcl: Object oriented support
7. Tclcl: C++ and otcl linkage
8. Discrete event scheduler

Architecture of NS2

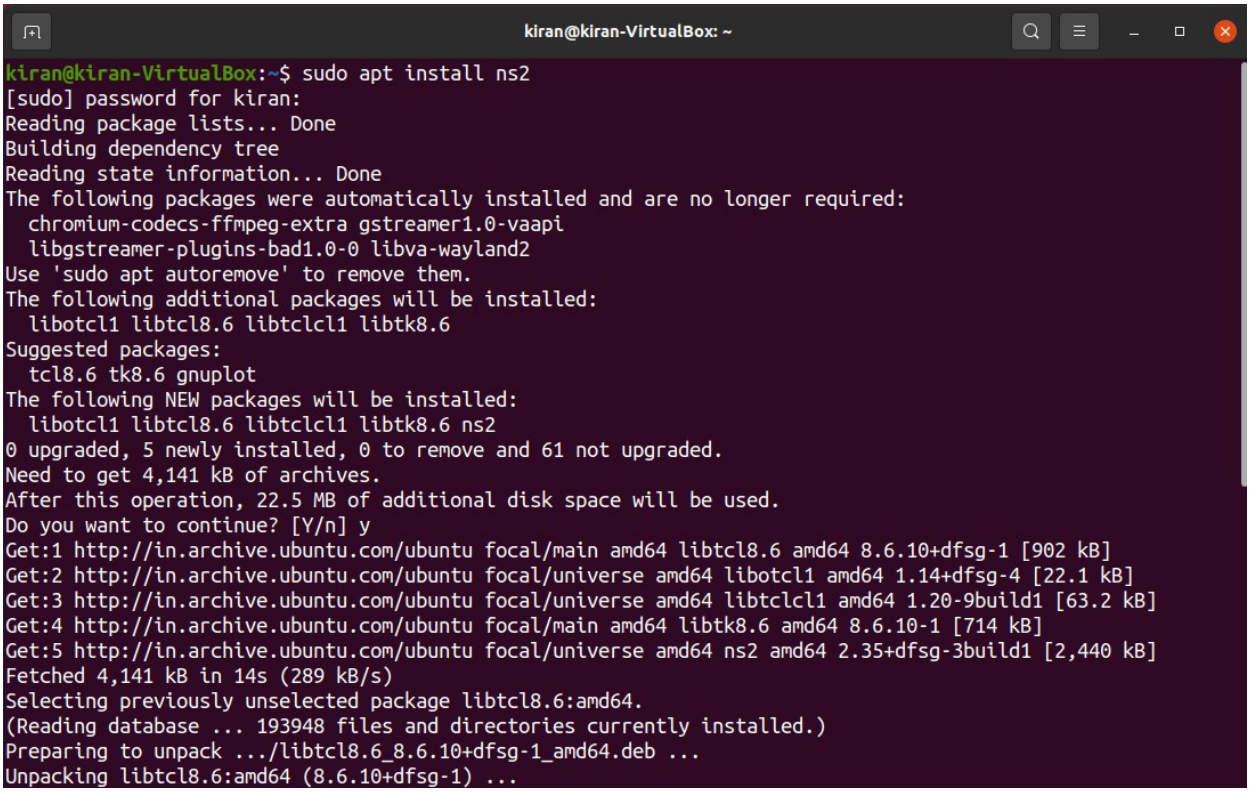
NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL

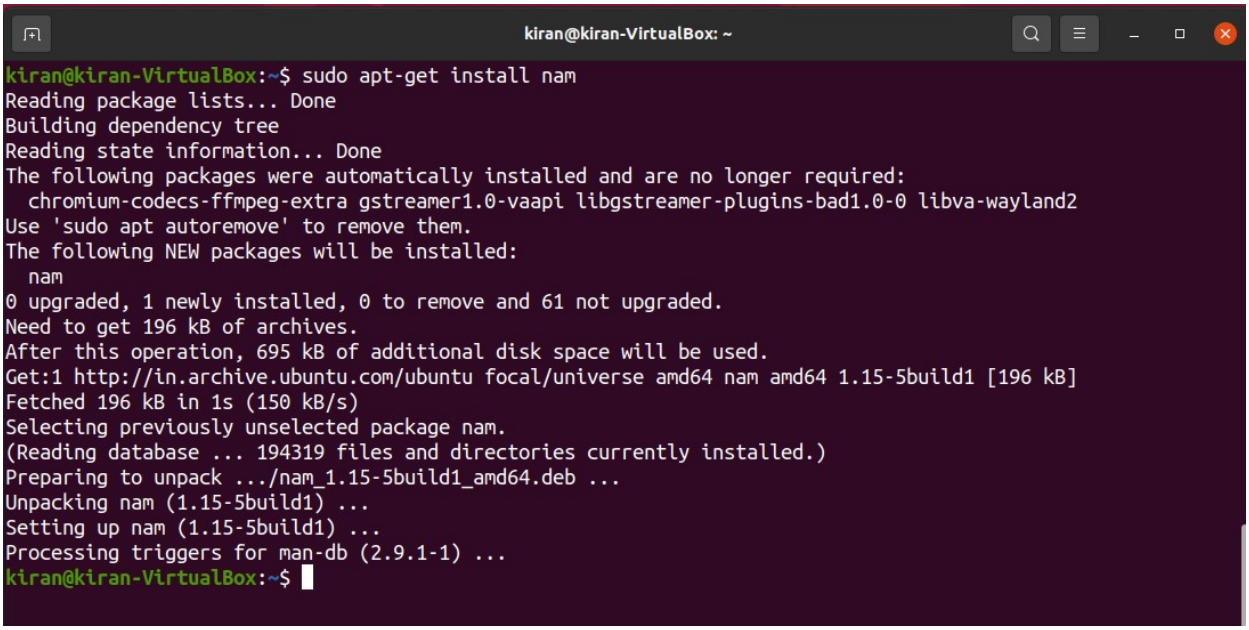


Basic architecture of NS.

Installation

- Install NS-2 using this command : sudo apt-get install ns2
- Nam is also needed to install. Nam (Network Animator) is an animation tool to graphically represent the network and packet traces.
- Use this command : sudo apt-get install nam


kiran@kiran-VirtualBox: ~
kiran@kiran-VirtualBox:~$ sudo apt install ns2
[sudo] password for kiran:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libotcl1 libtcl8.6 libtclcl1 libtk8.6
Suggested packages:
tcl8.6 tk8.6 gnuplot
The following NEW packages will be installed:
libotcl1 libtcl8.6 libtclcl1 libtk8.6 ns2
0 upgraded, 5 newly installed, 0 to remove and 61 not upgraded.
Need to get 4,141 kB of archives.
After this operation, 22.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libtcl8.6 amd64 8.6.10+dfsg-1 [902 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 libotcl1 amd64 1.14+dfsg-4 [22.1 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 libtclcl1 amd64 1.20-9build1 [63.2 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libtk8.6 amd64 8.6.10-1 [714 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 ns2 amd64 2.35+dfsg-3build1 [2,440 kB]
Fetched 4,141 kB in 14s (289 kB/s)
Selecting previously unselected package libtcl8.6:amd64.
(Reading database ... 193948 files and directories currently installed.)
Preparing to unpack .../libtcl8.6_8.6.10+dfsg-1_amd64.deb ...
Unpacking libtcl8.6:amd64 (8.6.10+dfsg-1) ...


kiran@kiran-VirtualBox: ~
kiran@kiran-VirtualBox:~$ sudo apt-get install nam
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
nam
0 upgraded, 1 newly installed, 0 to remove and 61 not upgraded.
Need to get 196 kB of archives.
After this operation, 695 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 nam amd64 1.15-5build1 [196 kB]
Fetched 196 kB in 1s (150 kB/s)
Selecting previously unselected package nam.
(Reading database ... 194319 files and directories currently installed.)
Preparing to unpack .../nam_1.15-5build1_amd64.deb ...
Unpacking nam (1.15-5build1) ...
Setting up nam (1.15-5build1) ...
Processing triggers for man-db (2.9.1-1) ...
kiran@kiran-VirtualBox:~$

1. Write a TCL script to simulate following scenario.

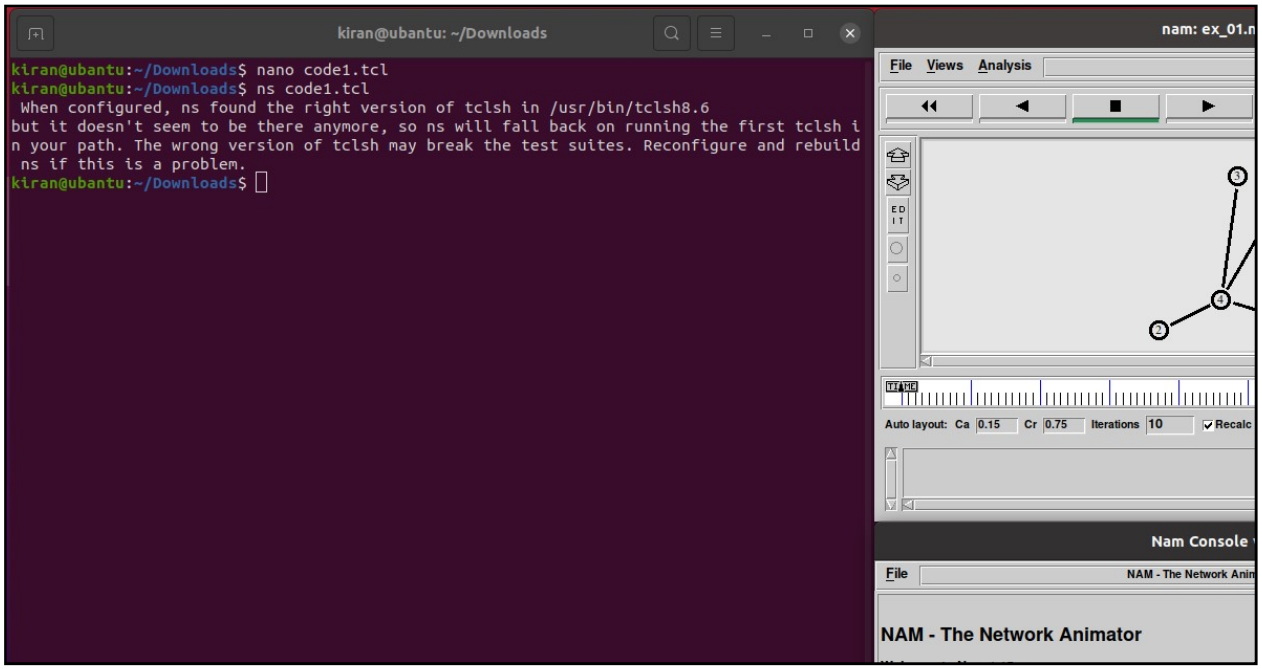
Consider a small network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 isat the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through thenode n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds.

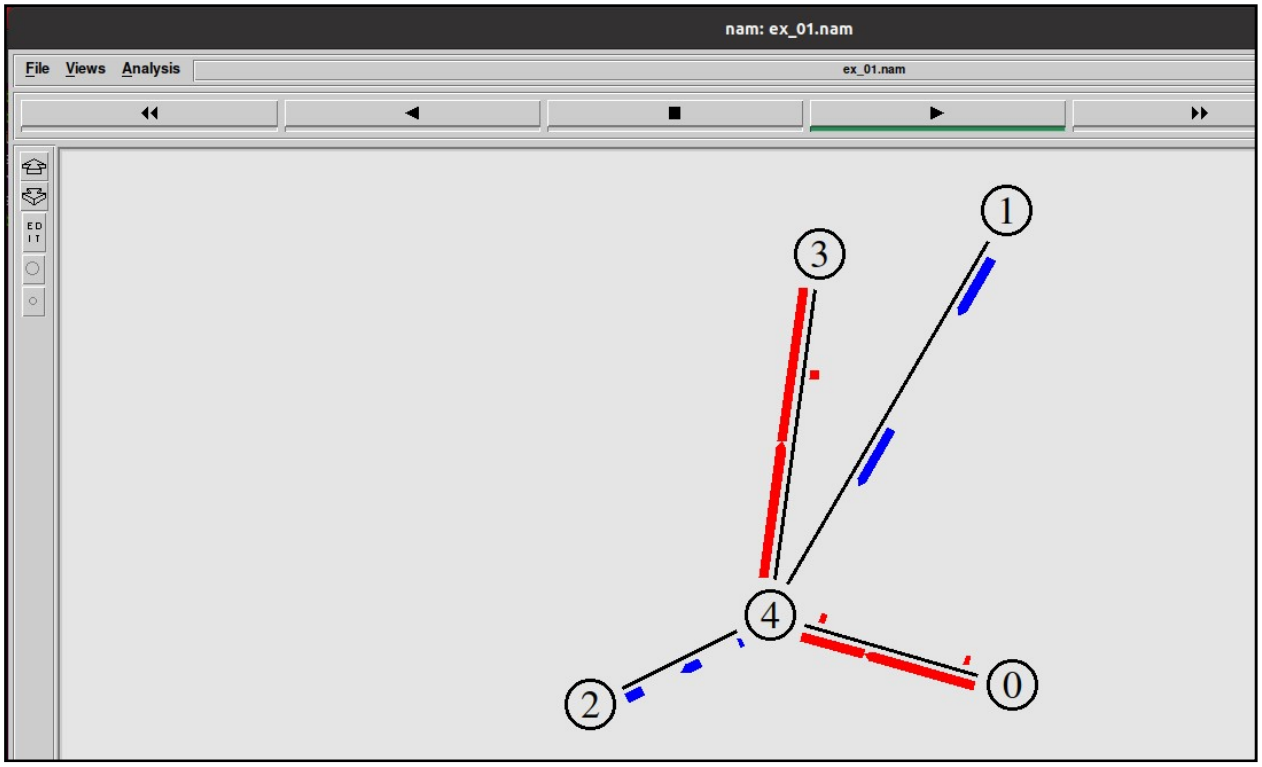
Script :

```
set ns [new Simulator]
set namfile [open ex_01.nam w]
$ns namtrace-all $namfile

set tracefile [open ex_01.tr w]
$ns trace-all $tracefile

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

$ns duplex-link $n0 $n4 1Mb 10ms DropTail
$ns duplex-link $n1 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n3 1Mb 10ms DropTail
$ns duplex-link $n4 $n2 1Mb 10ms DropTail

set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink

$ns connect $tcp $sink

set ftp [new Application/FTP]
$ftp attach-agent $tcp

set udp [new Agent/UDP]
```

```
$ns attach-agent $n1 $udp

set null [new Agent/Null]
$ns attach-agent $n2 $null

$ns connect $udp $null

$udp set class_ 1
$ns color 1 Blue

$tcp set class_ 2
$ns color 2 Red

set cbr [new Application/Traffic/CBR]
$cbr set packetsize_ 500
$cbr set interval_ 0.005
$cbr attach-agent $udp

$ns at 0.0 "$cbr start"
$ns at 0.0 "$ftp start"
$ns at 9.0 "$cbr stop"
$ns at 9.0 "$ftp stop"

proc finish {} {
global ns namfile tracefile
$ns flush-trace
close $namfile
close $tracefile
exec nam ex_01.nam &
exit 0
}

$ns at 10.0 "finish"
$ns run
```

Output :


kiran@ubantu: ~/Downloads
kiran@ubantu:~/Downloads$ nano code1.tcl
kiran@ubantu:~/Downloads$ ns code1.tcl
When configured, ns found the right version of tclsh in /usr/bin/tclsh8.6
but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh i
n your path. The wrong version of tclsh may break the test suites. Reconfigure and rebuild
ns if this is a problem.
kiran@ubantu:~/Downloads$
nam: ex_01.n
File Views Analysis
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
Nam Console
File
NAM - The Network Animator


nam: ex_01.nam
File Views Analysis
ex_01.nam
1
3
4
2
0

2. Write a TCL script to simulate a file transfer with ns2

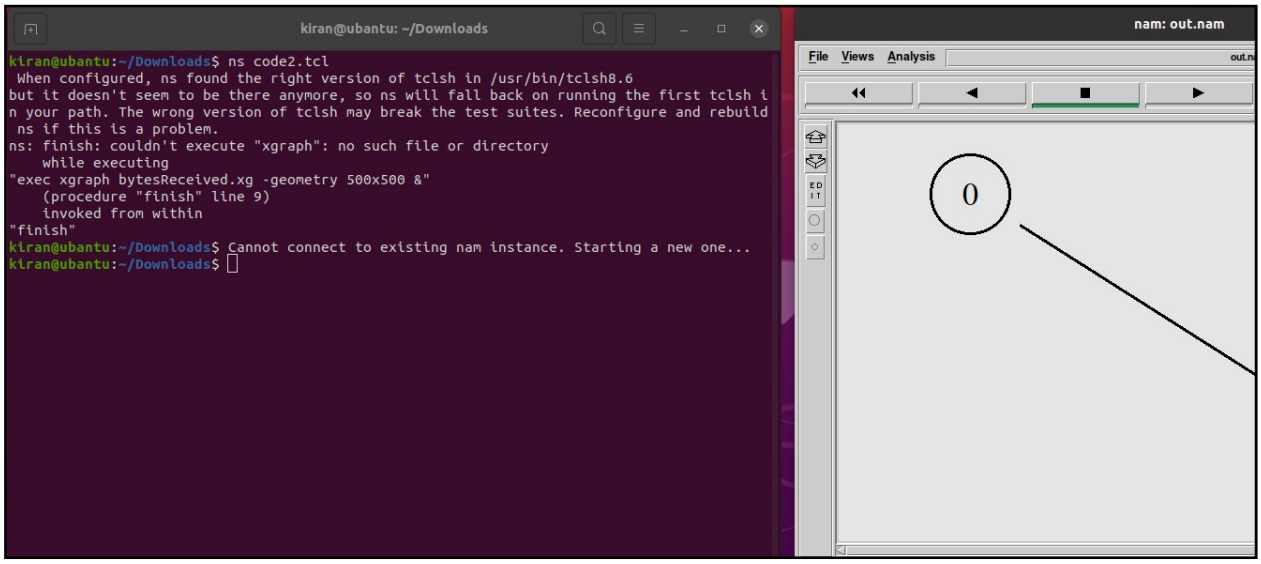
Consider a client and a server. The server is running a FTP application (over TCP). The client sends a request to download a file of size 10 MB from the server. Write a script to simulate this scenario. Let node #0 be the server and node #1 be the client. TCP packet size is 1500 B. Assume typical values for other parameters.

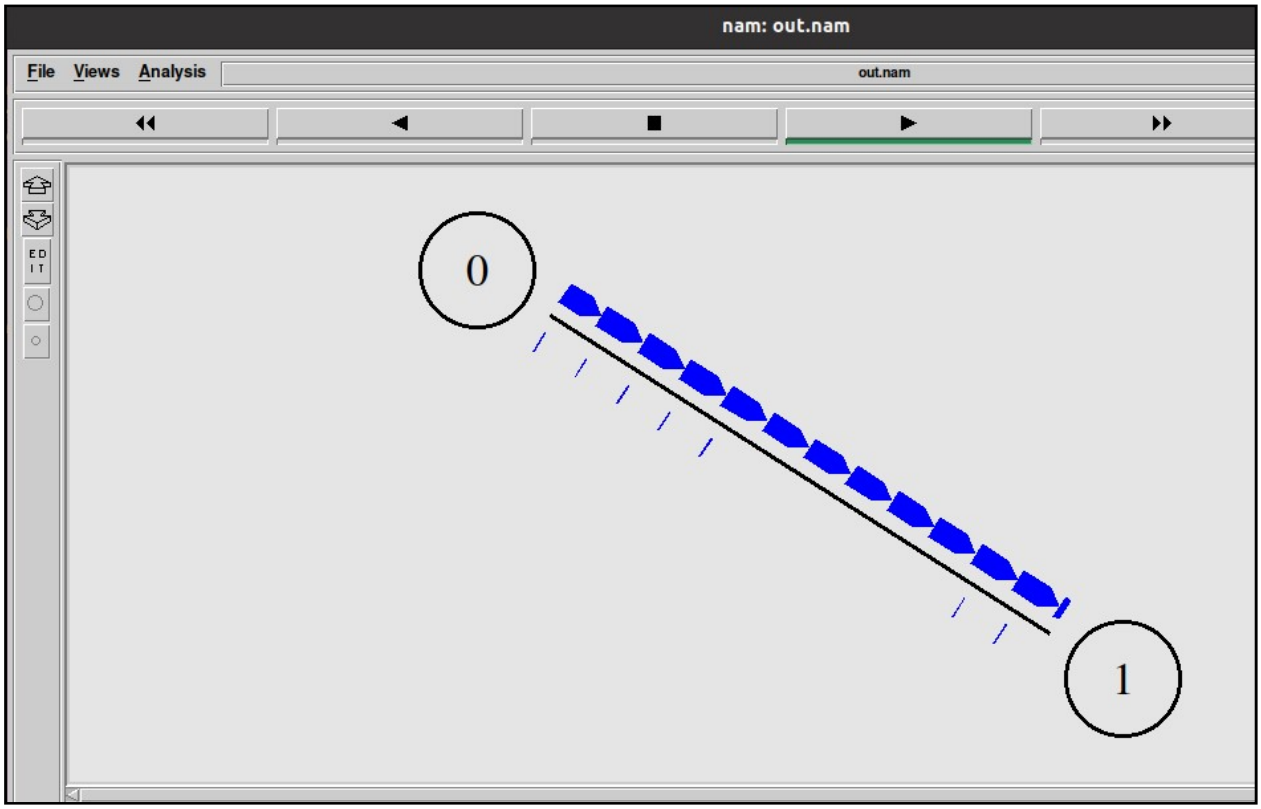
Script :

```
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
set outfile [open  "bytesReceived.xg"  w]

# procedure to plot the bytesReceived window
proc plotWindow {tcpSource outfile} {
   global ns
   set now [$ns now]
   set cwnd [$tcpSource set cwnd_]

# the data is recorded in a file called bytesReceived.xg (this can
 be plotted # using xgraph or gnuplot. this example uses xgraph to
 plot the cwnd_
   puts  $outfile  "$now $cwnd"
   $ns at [expr $now+0.1] "plotWindow $tcpSource  $outfile"
}

proc finish {} {
        global ns nf
        $ns flush-trace
        #Close the NAM trace file
        close $nf
```

```
        #Execute NAM on the trace file
        exec nam out.nam &
        exec xgraph bytesReceived.xg -geometry 500x500 &
        exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n1 10Mb 10ms DropTail

#Setup a TCP connection $n0->$n1
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP


#schedule
$ns  at  0.0  "plotWindow $tcp  $outfile"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 6.0 "finish"

#Run the simulation
$ns run
```

Output :


kiran@ubantu: ~/Downloads
kiran@ubantu:~/Downloads$ ns code2.tcl
When configured, ns found the right version of tclsh in /usr/bin/tclsh8.6
but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh i
n your path. The wrong version of tclsh may break the test suites. Reconfigure and rebuild
ns if this is a problem.
ns: finish: couldn't execute "xgraph": no such file or directory
while executing
"exec xgraph bytesReceived.xg -geometry 500x500 &"
(procedure "finish" line 9)
invoked from within
"finish"
kiran@ubantu:~/Downloads$ Cannot connect to existing nam instance. Starting a new one...
kiran@ubantu:~/Downloads$
nam: out.nam
File Views Analysis
0


nam: out.nam
File Views Analysis
out.nam
0
1

Conclusion:

- We have learned to install the ns2 network simulator and understood its features.
- From this experiment we have learned to install the NS2 then executed the the script observed the animated output and various fields of the network simulator.
- while running the script two more files created 1. Filename.tr and 2.filename .nam The .nam file gives the simulation.
- The NS2 has various options like files, views, analysis also we can start, stop fast forward/backword the simulation using NS2.
- We have observed the trace file.