

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.

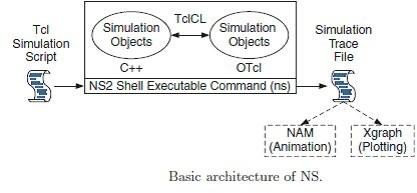
NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

**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

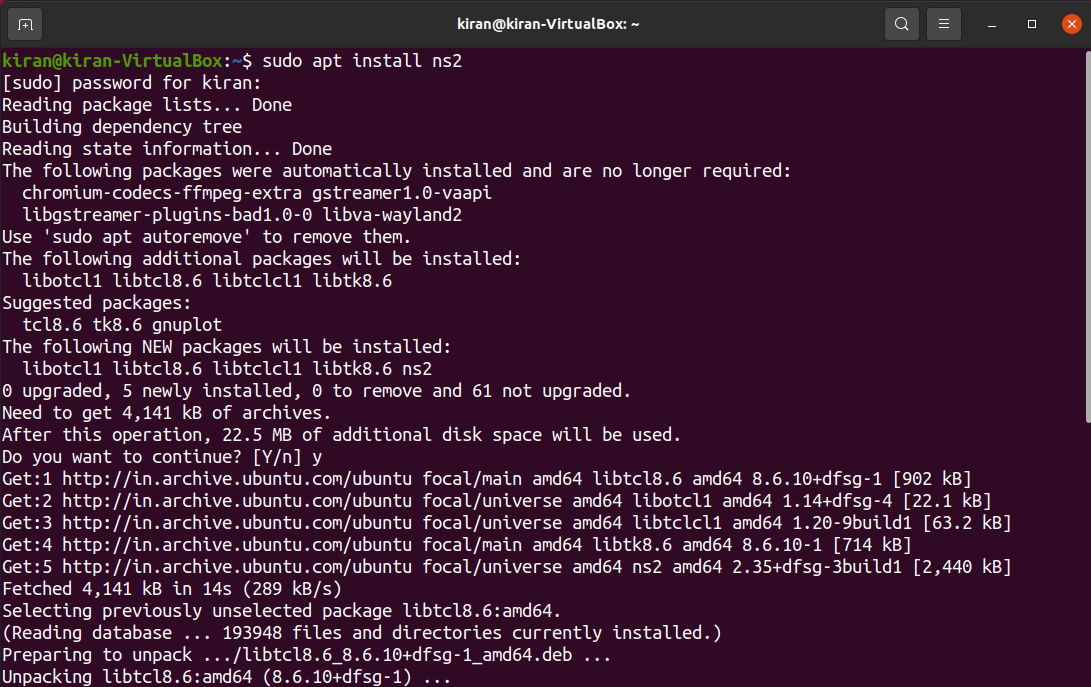
# Architecture of NS2

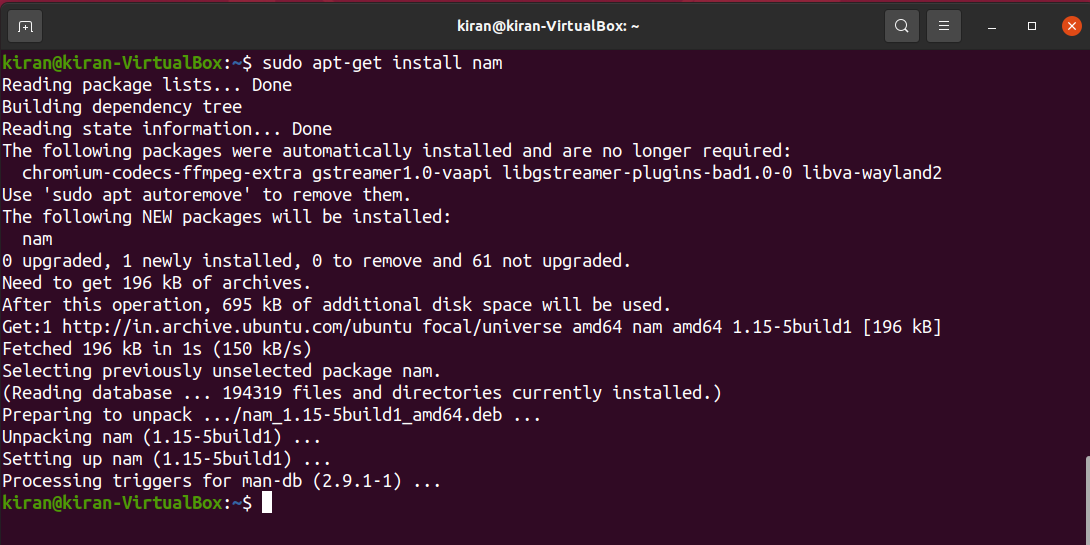
NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the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



# 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





**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 &

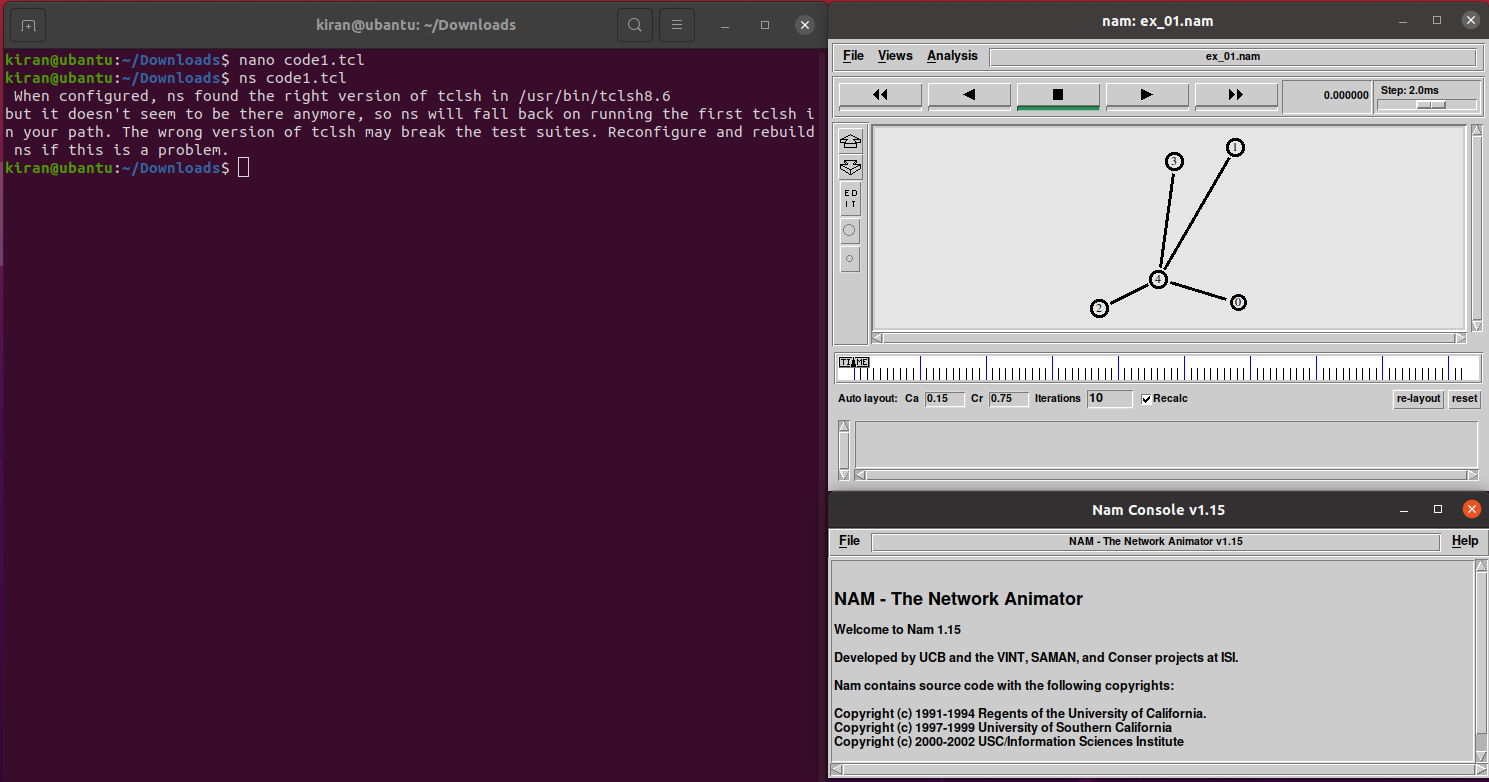
exit 0

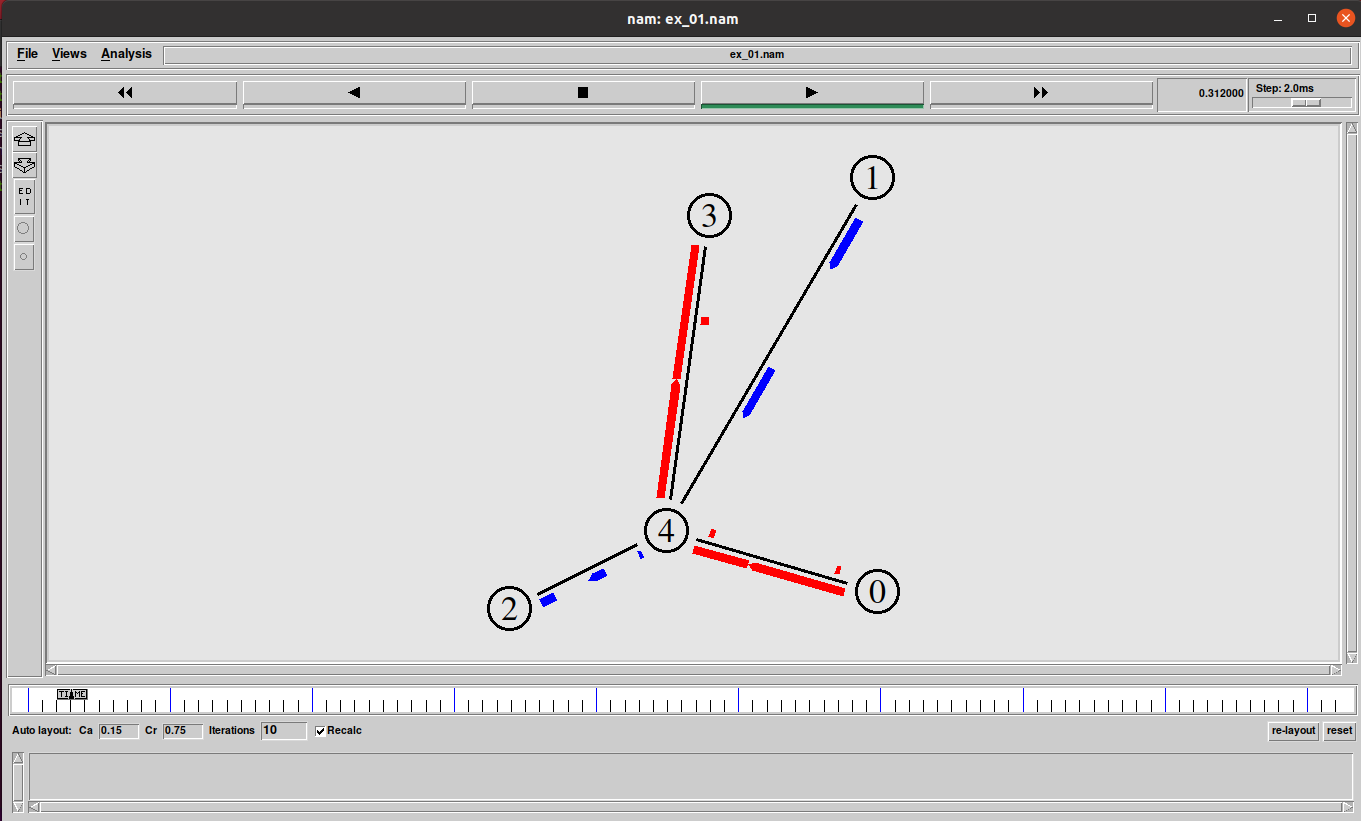
}

$ns at 10.0 "finish"

$ns run

**Output :**





**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"

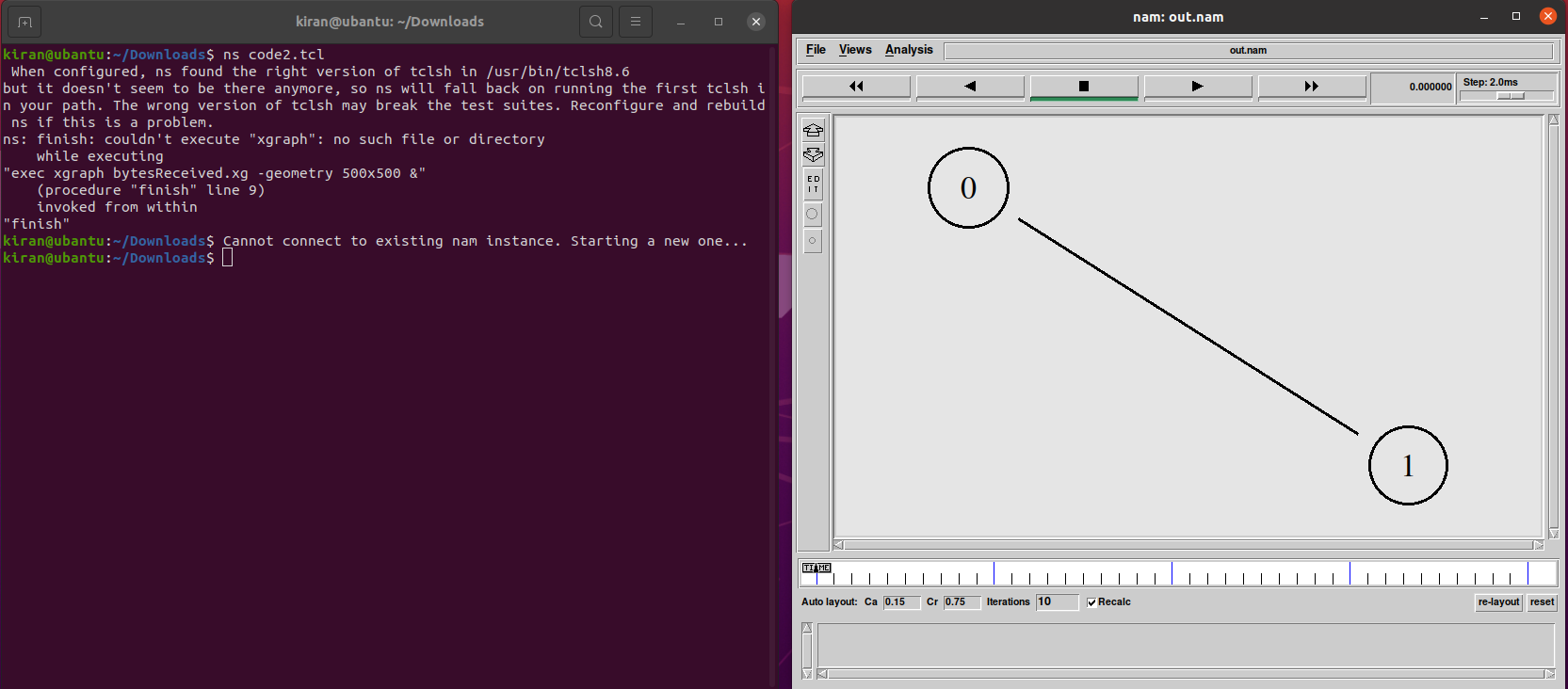
#Call the finish procedure after 5 seconds of simulation time

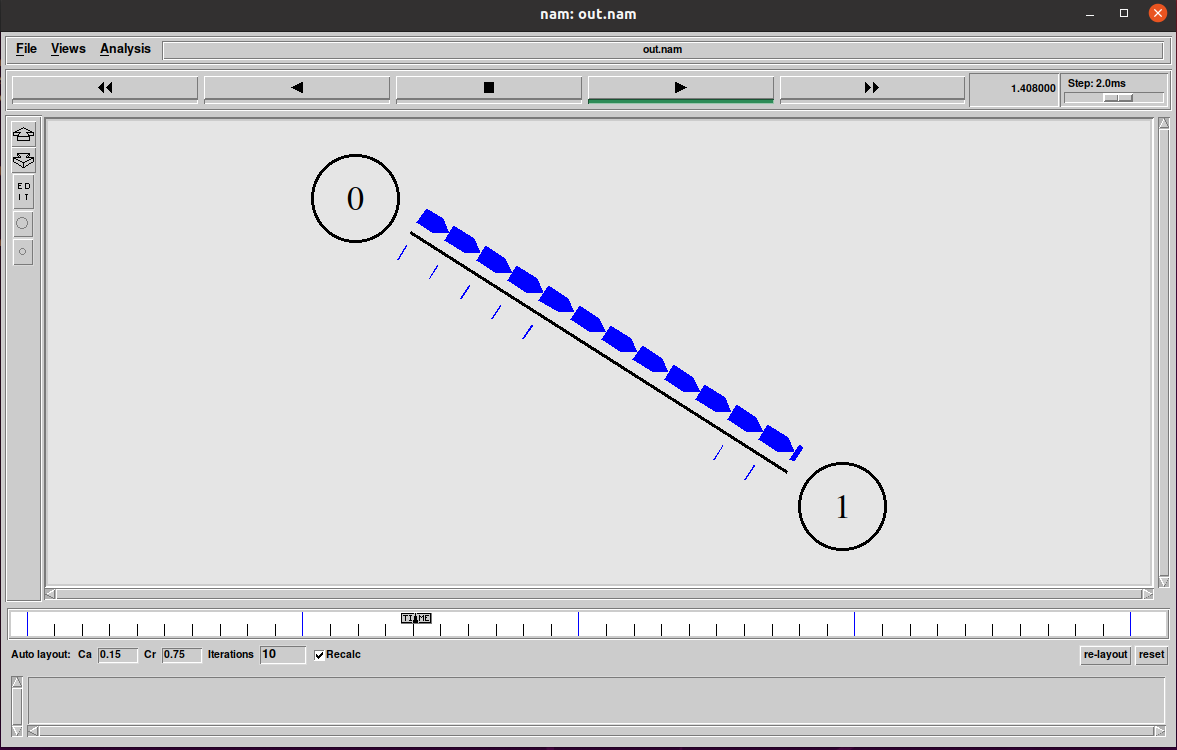
$ns at 6.0 "finish"

#Run the simulation

$ns run

**Output :**

****

****

**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.