Assignment 02

Name : Kiran Patil

Course : Wireless Networks Lab

ID : 211070904

Sem : 06

Assignment 02

Aim : Basics of Network simulator

Objectives:- · Learn Basic concepts about NS-2, download NS-2, install and work with NS-2

Theory :- Network simulator has become an an Integral part of most reasearch works in the field of computer Networks. Whether it is for understanding the behaviour of existing protocols or determine the performance of a new protocol, one doesn't often get access to real network devices. This gap has been filled up by network simulation to a large extent.

Network simulator version 2 (NS-2) is a discrete event packet level simulator. the Network simulator covers a very large number of applications of diffrent kind of protocol of different network types. consisting of different network elements and traffic model. Ns-2 is a package of tools that simulate behaviour of networks such as creating networks topologies. log events that happens under any load analyze the events and understand the network.

NS-2 based on two languages C++ and OTcL (object oriented version of tool command language. while NS-2 core is written in C++. one use of OTcL is to write the simulation scripts.

- C++ helps in the following way:

1) Helps to increase efficiency of simulation
2) It is used to provide details of the protocols & their operations.
3) It is used to reduce packet and event processing time.

- OTcL helps in following way.

1) With the help of OTcL we can describe different network topologies
2) It helps us to specify the protocols and their applications.
3) It allows fast developement.
4) Tcl compatible with many platforms and it is flexible for integration.

5] TCL is very easy to use and it is available in free

There is a linkage between ett and OTcl, which allows us to run the simulation scripts.

- Basics of TcL programming for NS-2
Network simulator NS-2 would involve the following general steps.

1. Initialization and termination aspects of network simulator object

2. Defining the network topology: nodes, links, queues, mobility of nodes if any.

3. Defining the network traffic: creating agents and thier applications.

4. setting trace for the network Animator NAM

5. Tracing.

1. Write a TCL script to simulate following scenario.

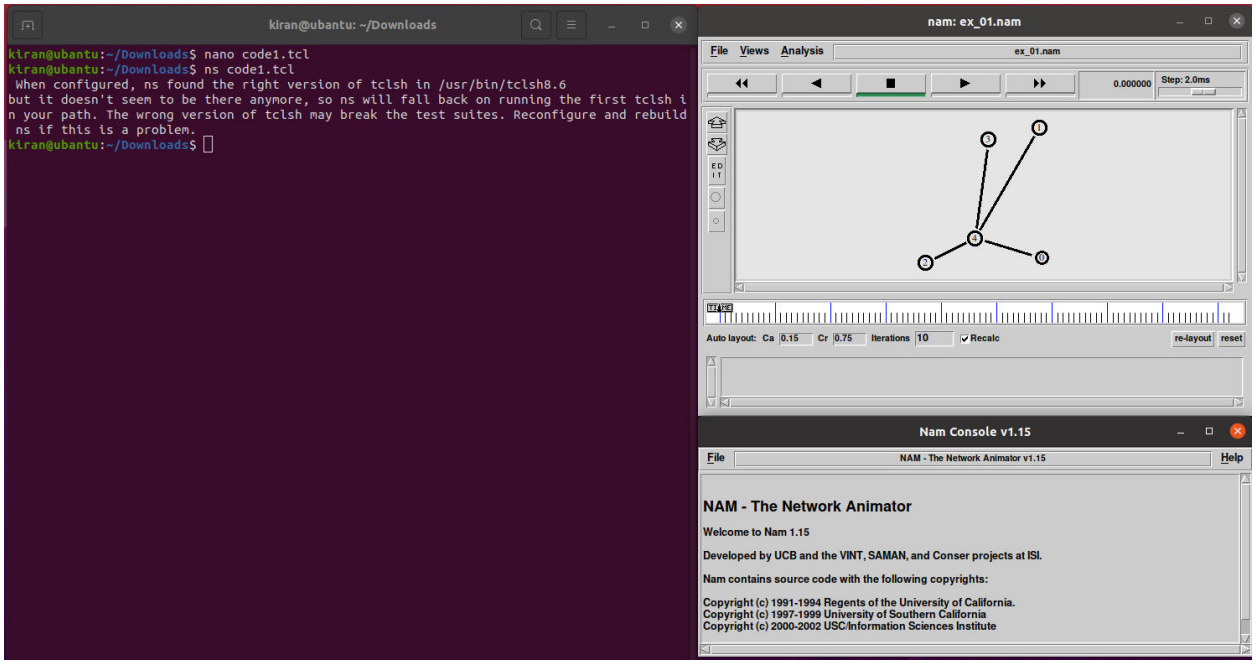
Consider a small network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 isat the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through thenode n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration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

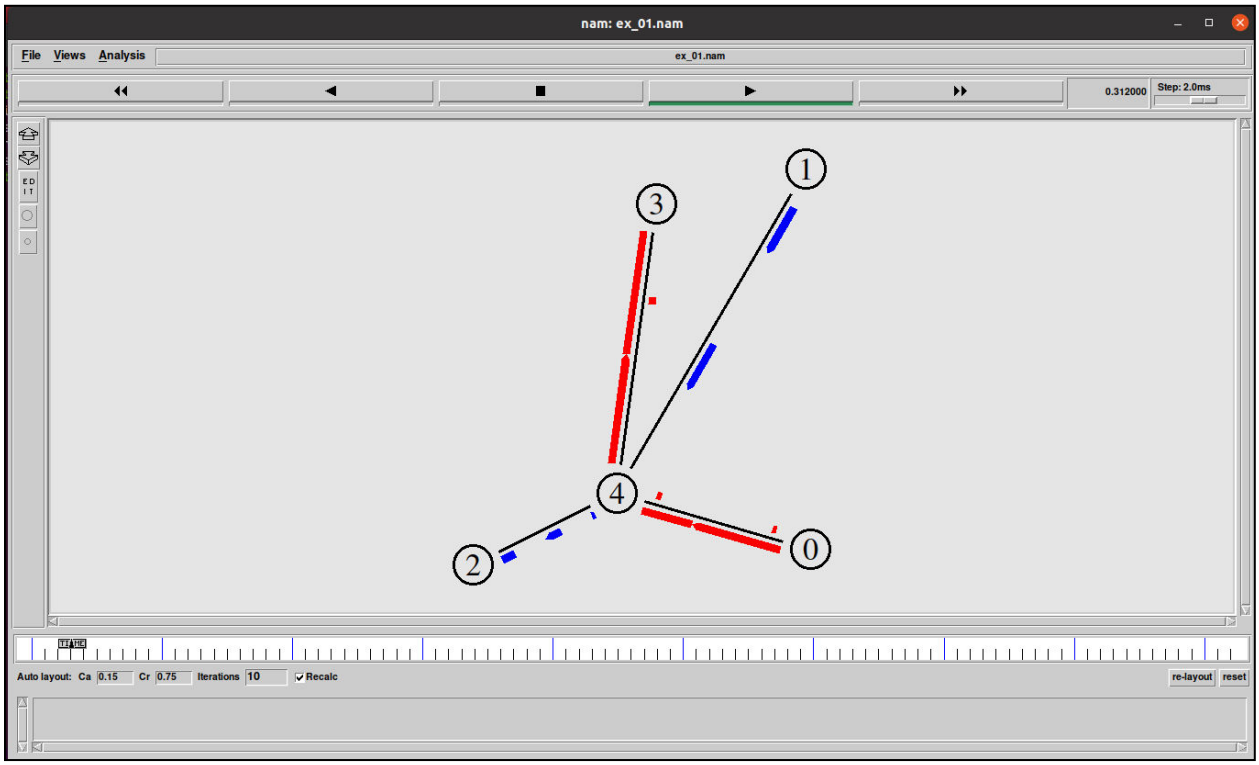
Script :

```
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.nam
File Views Analysis
ex_01.nam
0.000000
Step: 2.0ms
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout reset
Nam Console v1.15
File
NAM - The Network Animator v1.15
Help
NAM - The Network Animator
Welcome to Nam 1.15
Developed by UCB and the VINT, SAMAN, and Conser projects at ISI.
Nam contains source code with the following copyrights:
Copyright (c) 1991-1994 Regents of the University of California.
Copyright (c) 1997-1999 University of Southern California
Copyright (c) 2000-2002 USC/Information Sciences Institute


nam: ex_01.nam
File Views Analysis
ex_01.nam
0.312000
Step: 2.0ms
1
3
4
2
0
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout reset

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

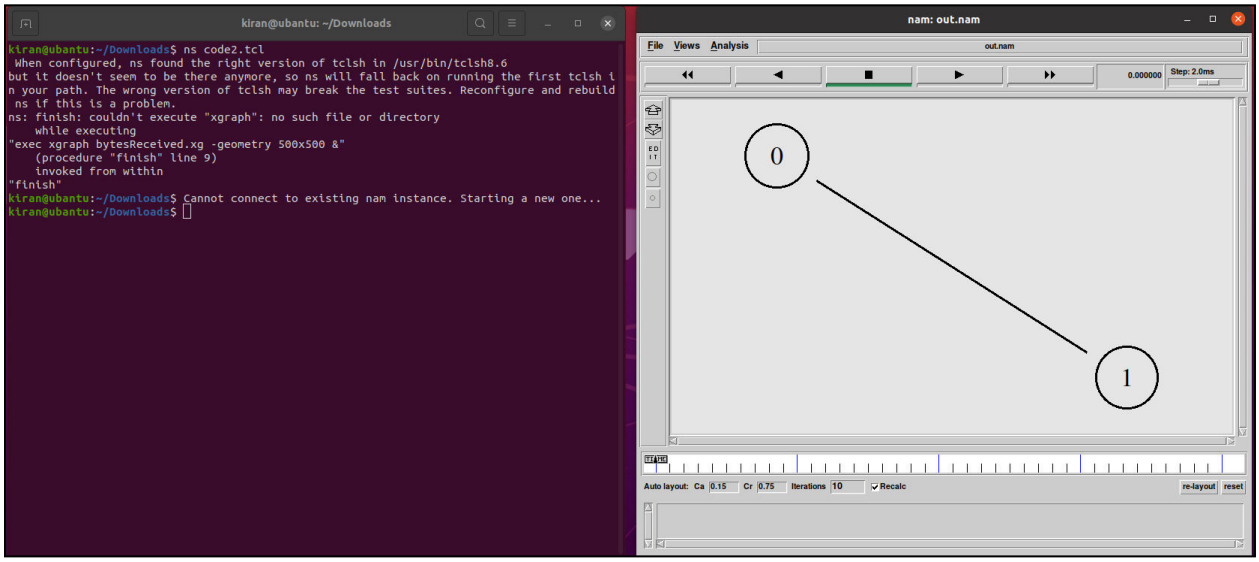
Consider a client and a server. The server is running a FTP application (over TCP). The client sends a request to download a file of size 10 MB from the server. Write a script to simulate this scenario. Let node #0 be the server and node #1 be the client. TCP packet size is 1500 B. Assume typical values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

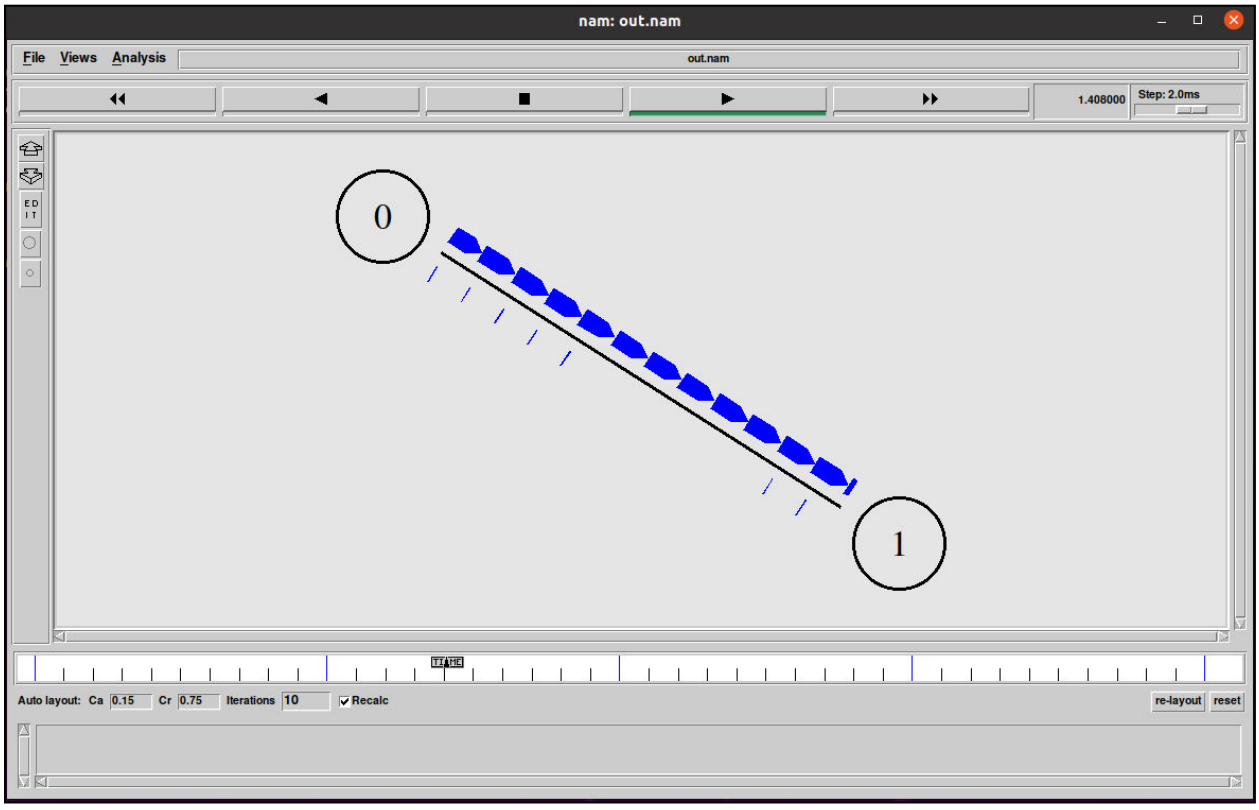
Script :

```
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 c
an be plotted # using xgraph or gnuplot. this example uses xgrap
h 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
out.nam
0.000000
Step: 2.0ms
0
1
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout reset


nam: out.nam
File Views Analysis
out.nam
1.408000
Step: 2.0ms
0
1
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout reset

Conclusion :

- 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.