Don Bosco Institute of Technology
Department of Information Technology
Wireless Technology

BE-IT **SEM 7**

Name: Christine Polly **Roll no: 17**

Experiment No 11

Title :
Network Simulator 2 - TCL script for TCP communication between four Clients and a End server.

Theory :
Introduction:

NS (network simulator) is a discrete event network simulator, specifically ns-1, ns-2 and ns-3. All of them are discrete-event network simulator, primarily used in research and teaching. ns-3 is free software, publicly available under the GNU GPLv2 license for research, development, and use. The goal of the ns-3 project is to create an open simulation environment for networking research that will be preferred inside the research community.
It should be aligned with the simulation needs of modern networking research.
It should encourage community contribution, peer review, and validation of the software.
Since the process of creation of a network simulator that contains a sufficient number of high-quality validated, tested and actively maintained models requires a lot of work, ns-3 project spreads this workload over a large community of users and developers.

Procedure/ Algorithm :
Description:

This network consists of 8 nodes (Client1, Client2, Client3, Client4, Router1, Router2 Router 3, Router 4, Router 5, Router 6 and Endserver1,). The duplex links between Client1, Client2, Client3, Client 4 and Router1 have 5 Mbps of bandwidth and 50 ms of delay. The duplex link between Router1 and Router2 has 5 Mbps of bandwidth and 50 ms of delay. The duplex link between Router 2 and Router 3 has 150 Kbps of bandwidth and 50 ms of delay and etc. Finally the duplex link between Router 6 and Endserver1 has 300 Kbps of bandwidth and 50 ms of delay. Each link uses a DropTail queue. A "TCP" agent is attached to Client1, Client2, Client3, Client 4 and a connection is established to a “TCP Sink" agent attached to Endserver1. As default, the maximum size of a packet that a "TCP" agent can generate is 1000 bytes. A "TCP Sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. The ftp is set to start at 0.50 sec and stop at 28.5 sec..

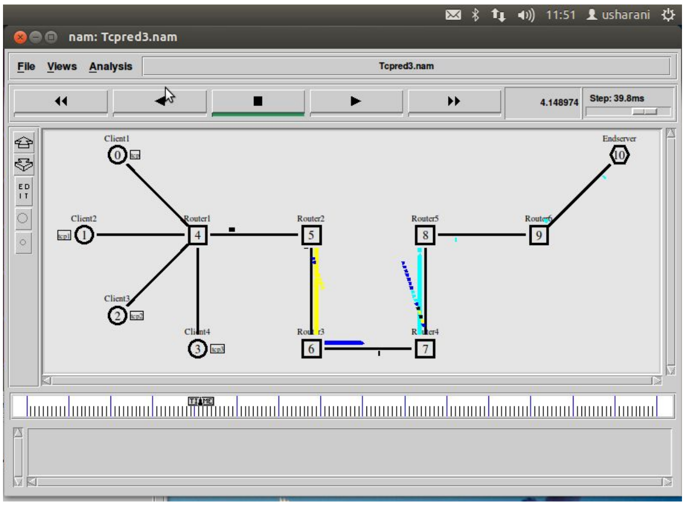
Code:
```
 #-------Event scheduler object creation--------#

set ns [ new Simulator]

 # ----------- CREATING NAM OBJECTS -----------------#

 set nf [open Tcpred3.nam w]
$ns namtrace-all $nf

 #Open the trace file
```

```
set nt [open Tcpred3.tr w]
$ns trace-all $nt

 set proto rlm

 $ns color 1 red
$ns color 2 blue
$ns color 3 yellow
$ns color 4 cyan
$ns color 5 maroon

 # --------- CREATING CLIENT - ROUTER -ENDSERVER NODES-----------#

 set Client1 [$ns node]
set Client2 [$ns node]
set Client3 [$ns node]
set Client4 [$ns node]
set Router1 [$ns node]
set Router2 [$ns node]
set Router3 [$ns node]
set Router4 [$ns node]
set Router5 [$ns node]
set Router6 [$ns node]
set Endserver1 [$ns node]

 # --------------CREATING DUPLEX LINK -----------------------#

$ns duplex-link $Client1 $Router1  5Mb 50ms DropTail
$ns duplex-link $Client2 $Router1  5Mb 50ms DropTail
$ns duplex-link $Client3 $Router1 5Mb 50ms DropTail
$ns duplex-link $Client4 $Router1 5Mb 50ms DropTail
$ns duplex-link $Router1 $Router2 5Mb 50ms DropTail
$ns duplex-link $Router2 $Router3 150Kb 50ms DropTail
$ns duplex-link $Router3 $Router4 300Kb 50ms DropTail
$ns duplex-link $Router4 $Router5 100Kb 50ms DropTail
$ns duplex-link $Router5 $Router6 300Kb 50ms DropTail
$ns duplex-link $Router6 $Endserver1 300Kb 50ms DropTail
#$ns duplex-link $Router6 $Endserver2 300Kb 50ms DropTail

  #-----------CREATING ORIENTATION -------------------------#

$ns duplex-link-op $Client1 $Router1 orient down-right
$ns duplex-link-op $Client2 $Router1 orient right
$ns duplex-link-op $Client3 $Router1 orient up-right
$ns duplex-link-op $Client4 $Router1 orient up
$ns duplex-link-op $Router1 $Router2 orient right
$ns duplex-link-op $Router2 $Router3 orient down
$ns duplex-link-op $Router3 $Router4 orient right
$ns duplex-link-op $Router4 $Router5 orient up
$ns duplex-link-op $Router5 $Router6 orient right
$ns duplex-link-op $Router6 $Endserver1 orient up-right
#$ns duplex-link-op $Router6 $Endserver2 orient right

  # --------------CREATING LABELLING -----------------------------#

 $ns at 0.0 "$Client1 label Client1"
$ns at 0.0 "$Client2 label Client2"
$ns at 0.0 "$Client3 label Client3"
$ns at 0.0 "$Client4 label Client4"
$ns at 0.0 "$Router1 label Router1"
$ns at 0.0 "$Router2 label Router2"
$ns at 0.0 "$Router3 label Router3"
$ns at 0.0 "$Router4 label Router4"
$ns at 0.0 "$Router5 label Router5"
$ns at 0.0 "$Router6 label Router6"
$ns at 0.0 "$Endserver1 label Endserver"
#$ns at 0.0 "$Endserver2 label Endserver2"
$ns attach-agent $Endserver1 $sink0

 $ns connect $tcp0 $sink0

 set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

 $ns add-agent-trace $tcp0 tcp
$tcp0 tracevar cwnd_
```

```
 # --------------- CONFIGURING NODES -----------------#

 $Endserver1 shape hexagon
$Router1 shape box
$Router2 shape square
$Router3 shape square
$Router4 shape square
$Router5 shape square
$Router6 shape square

 # ----------------ESTABLISHING QUEUES -------------#

 $ns duplex-link-op $Client1 $Router1 queuePos 0.1
$ns duplex-link-op $Client2 $Router1 queuePos 0.1
$ns duplex-link-op $Client3 $Router1 queuePos 0.5
$ns duplex-link-op $Client4 $Router1 queuePos 0.5
$ns duplex-link-op $Router1 $Router2 queuePos 0.1
$ns duplex-link-op $Router2 $Router3 queuePos 0.1
$ns duplex-link-op $Router3 $Router4 queuePos 0.1
$ns duplex-link-op $Router4 $Router5 queuePos 0.1
$ns duplex-link-op $Router5 $Router6 queuePos 0.5
$ns duplex-link-op $Router6 $Endserver1 queuePos 0.5

 # ----------------ESTABLISHING COMMUNICATION -------------#

#-------CLIENT1 TO ENDSERVER1 -------------#

 set tcp0 [new Agent/TCP]
$tcp0 set maxcwnd_ 16
$tcp0 set fid_ 4
$ns attach-agent $Client1 $tcp0

 set sink0 [new Agent/TCPSink]
$ftp2 attach-agent $tcp2
```

```
 $ns at 0.5 "$ftp0 start"
$ns at 28.5 "$ftp0 stop"

 # ---------------- CLIENT2 TO ENDSERVER1 -------------#

 set tcp1 [new Agent/TCP]
$tcp1 set fid_ 2
$tcp1 set maxcwnd_ 16
$ns attach-agent $Client2 $tcp1

 set sink1 [new Agent/TCPSink]
$ns attach-agent $Endserver1 $sink1

 $ns connect $tcp1 $sink1

 set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1

 $ns add-agent-trace $tcp1 tcp1
$tcp1 tracevar cwnd_

 $ns at 0.58 "$ftp1 start"
$ns at 28.5 "$ftp1 stop"

 # ----------------CLIENT3 TO ENDSERVER1------------#

  set tcp2 [new Agent/TCP]
$tcp2 set fid_ 0
$tcp2 set maxcwnd_ 16
$tcp2 set packetsize_ 100
$ns attach-agent $Client3 $tcp2
set sink2 [new Agent/TCPSink]
$ns attach-agent $Endserver1 $sink2
$ns connect $tcp2 $sink2

 set ftp2 [new Application/FTP]
$ns at 15.0 "finish"
$ns run
```

```
$ns add-agent-trace $tcp2 tcp2
$tcp2 tracevar cwnd_

 $ns at 0.65 "$ftp2 start"
$ns at 28.5 "$ftp2 stop"

  #--------------------CLIENT4 TO
ENDSERVER1----------------#

 set tcp3 [new Agent/TCP]
$tcp3 set fid_ 3
$tcp3 set maxcwnd_ 16
$tcp2 set packetsize_ 100
$ns attach-agent $Client4 $tcp3

 set sink3 [new Agent/TCPSink]
$ns attach-agent $Endserver1 $sink3

 $ns connect $tcp3 $sink3

 set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3

 $ns add-agent-trace $tcp3 tcp3
$tcp3 tracevar cwnd_

 $ns at 0.60 "$ftp3 start"
$ns at 28.5 "$ftp3 stop"

 # ---------------- FINISH PROCEDURE
-------------#

 proc finish {} {
          global ns nf nt

           $ns flush-trace
          close $nf
          puts "running nam..."
           exec nam Tcpred3.nam &
          exit 0
          }

#Calling finish procedure
```



Reference:
http://crazyguru9msrit.blogspot.com/2015/11/ns2-tcl-script-for-tcp-communication-bw.html