Roll No: 5117060

Group: 24

Name: Shivam Tiwari

Practical Examination

PROBLEM DEFINITION:

This network consists of 5 nodes (Client1, Client2, Router1, Router2 and Endserver1). The duplex links between Client1 and Router1 have 2 Mbps of bandwidth and 50 ms of delay. The duplex link between Router1 and Endserver1 has 100Kbps of bandwidth and 100 ms of delay. The duplex link between Client2 and Router2 has 100Kbps of bandwidth and 50 ms of delay. The duplex link between Endserver1 and Router2 has 100Mbps of bandwidth and 100 ms of delay. Each link uses a DropTail queue. A "TCP" agent is attached to Client1 and client2 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 1000bytes. A "TCPsink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. The packets are dropped down between Router1 to Endserver1 at 1.6 sec. The ftp is set to start at 0.5 sec and stop at 3.5 sec.

Code:

```
set ns [new Simulator]


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


set nf [open drop2.nam w]
$ns namtrace-all $nf


set nt [open drop2.tr w]
$ns trace-all $nt


set proto rlm


#------------COLOR DESCRIPTION---------------#


$ns color 1 red
$ns color 2 blue


# --------- CREATING SENDER - RECEIVER - ROUTER NODES-----------#


set Client1 [$ns node]
set Router1 [$ns node]
```

```
set Endserver1 [$ns node]

set Client2 [$ns node]

set Router2 [$ns node]


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


$ns duplex-link $Client1 $Router1 2Mb 50ms DropTail

$ns duplex-link $Router1 $Endserver1 100Kb 100ms DropTail

$ns duplex-link $Client2 $Router2 100Kb 50ms DropTail

$ns duplex-link $Router2 $Endserver1 100Kb 100ms DropTail


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


$ns duplex-link-op $Client1 $Router1 orient right

$ns duplex-link-op $Router1 $Endserver1 orient right

$ns duplex-link-op $Endserver1 $Router2 orient right

$ns duplex-link-op $Router2 $Client2 orient right


# --------------LABELLING ---------------------------#

$ns at 0.0 "$Client1 label Client1"

$ns at 0.0 "$Router1 label Router1"

$ns at 0.0 "$Endserver1 label Endserver1"

$ns at 0.0 "$Router2 label Router2"

$ns at 0.0 "$Client2 label Client2"


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

$Endserver1 shape hexagon

$Router1 shape square

$Router2 shape square


#-------------QUEUE SIZE DESCRIPTION---------------#

$ns duplex-link-op $Client1 $Router1 queuePos 0.5

$ns duplex-link-op $Router1 $Endserver1 queuePos 0.5

$ns duplex-link-op $Client2 $Router2 queuePos 0.5

$ns duplex-link-op $Router2 $Endserver1 queuePos 0.5
```

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


#-------------TCP CONNECTION BETWEEN NODES---------------#



set tcp1 [$ns create-connection TCP $Client1 TCPSink $Endserver1 0]
      $tcp1 set fid_ 1
      set ftp1 [$tcp1 attach-app FTP]
      $ftp1 set packetSize_ 1000
      $ftp1 set interval_ 0.5
      $ns at 0.5 "$Client1 color green"
      $ns at 1.5 "$Endserver1 color red"
      $ns at 0.5 "$ftp1 start"
      $ns at 3.0 "$ftp1 stop"


      $ns rtmodel-at 1.6 down $Router1 $Endserver1
      #$ns rtmodel-at 2.0 up $Router1 $Endserver1


#---------- client2 to endserver1------#

set tcp2 [$ns create-connection TCP $Client2 TCPSink $Endserver1 0]
      $tcp2 set fid_ 1
      set ftp2 [$tcp2 attach-app FTP]
      $ftp2 set packetSize_ 1000
      $ftp2 set interval_ 0.5
      $ns at 0.5 "$ftp2 start"
      $ns at 3.0 "$ftp2 stop"


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


proc finish {} {

global ns nf nt
$ns flush-trace
close $nf
```

```
puts "running nam..."

exec nam drop2.nam &

exit 0

}


$ns at 5.0 "finish"

$ns run
```

Output: