SSN College of Engineering, Kalavakkam
Department of Computer Science and Engineering
V Semester - CSE 'B'
UCS1511 NETWORKS LAB

Exercise 8: PERFORMANCE EVALUATION OF TCP AND UDP

Name **:** Mahesh Bharadwaj K
Roll No. **:** 185001089
Date **:** 20/10/2020

AIM: To write a tcl script to evaluate the performance of TCP and UDP sharing a bottleneck link.

ALGORITHM:

1. The Simulator class is used to create a new variable ns.
2. The color field here is used to discriminate the different data packets travelling across the nodes.
3. The namtrace is set for enabling animation to simulate the environment.
4. Then the various nodes n0,n1..etc are declared accordingly.
5. The duplex links between the nodes is set appropriately.
6. Following this the orientation of these nodes in the simulator is decided upon.
7. The queue limit is set to determine the capacity of the queue for any communication.
8. A UDP connection is set up between the node n0 and n5.
9. The CBR here facilitates this UDP connection.
10. Then TCP connection is setup between the nodes n0 and n4.
11. The FTP here is set as the application layer protocol that uses TCP.
12. The TCP connection requires a sink at the end and the UDP connection requires a NULL at the other end. This is also set up accordingly.
13. The run command is used to execute the simulation.

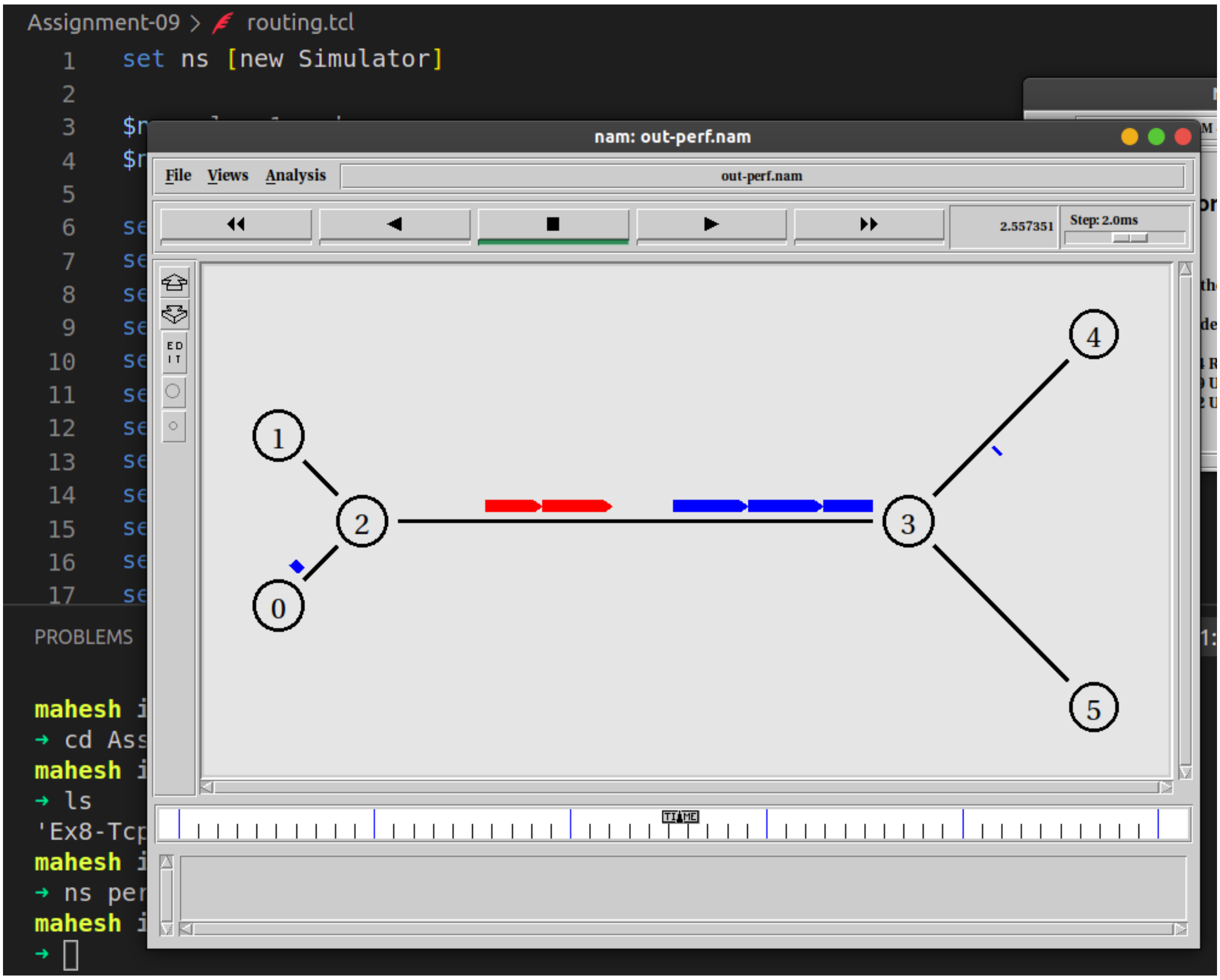
CODE:

```
#Create a simulator object
set ns [new Simulator]
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
$ns color 1 Blue
$ns color 2 Red
#Define a 'finish' procedure
proc finish {} {
 global ns nf
 $ns flush-trace
 #Close the trace file
 close $nf
 #Execute nam on the trace file
 exec nam out.nam &
 exit 0
}
# Creating Nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
#Setting Links
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n2 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mb 40ms DropTail
#Setting Topology
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient left-up
```

```
$ns duplex-link-op $n3 $n2 orient left
$ns duplex-link-op $n3 $n4 orient up
$ns duplex-link-op $n3 $n5 orient right-up
#Setting Queue Limit
$ns queue-limit $n2 $n3 10
#Setup a TCP connection over 0 and 4 and its flow id, window size, packet size
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Create a UDP agent and attach it to node n0
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
# Create a CBR traffic source and attach it to udp0
set cbr [new Application/Traffic/CBR]
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
$cbr set random_ false
$cbr attach-agent $udp
#Create a Null agent (a traffic sink) and attach it to node n1
set null [new Agent/Null]
$ns attach-agent $n5 $null
#Connect the traffic source with the traffic sink
$ns connect $udp $null
#Set Flow ID, Packet Size and Window Size
$udp set fid_ 2
$udp set window_ 8000
$udp set packetSize_ 552
#Start and stop the cbr and ftp
```

```
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.5 "$ftp stop"
$ns at 5.0 "$cbr stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

SCREENSHOT:



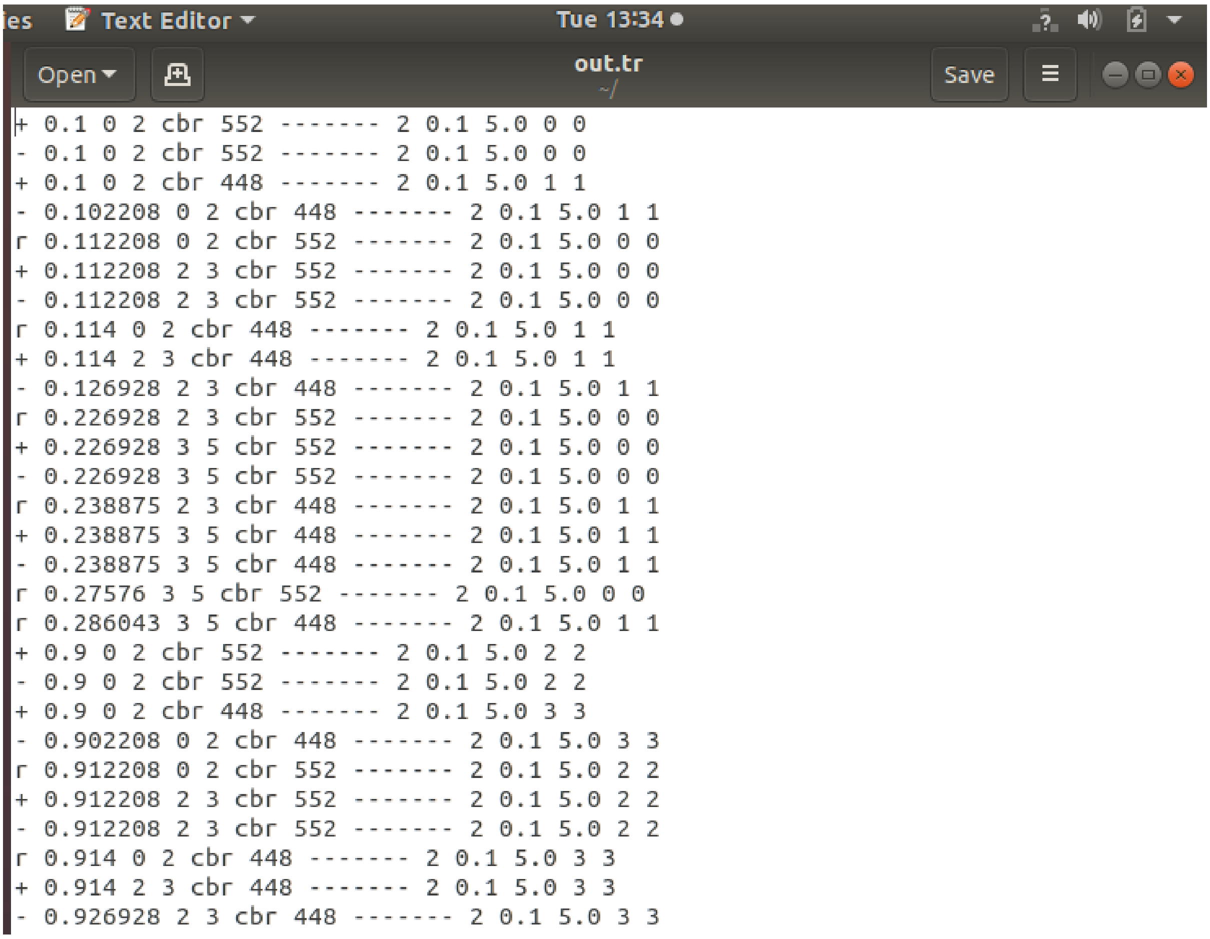
Assignment-09 > routing.tcl

```
set ns [new Simulator]

$n
$n

se
se
se
se
se
se
se
se
se
se
se
se
```

PROBLEMS

```
mahesh i
→ cd Ass
mahesh i
→ ls
'Ex8-Tcp
mahesh i
→ ns per
mahesh i
→ 
```

TRACE FILE:



```
+ 0.1 0 2 cbr 552 ------- 2 0.1 5.0 0 0
- 0.1 0 2 cbr 552 ------- 2 0.1 5.0 0 0
+ 0.1 0 2 cbr 448 ------- 2 0.1 5.0 1 1
- 0.102208 0 2 cbr 448 ------- 2 0.1 5.0 1 1
r 0.112208 0 2 cbr 552 ------- 2 0.1 5.0 0 0
+ 0.112208 2 3 cbr 552 ------- 2 0.1 5.0 0 0
- 0.112208 2 3 cbr 552 ------- 2 0.1 5.0 0 0
r 0.114 0 2 cbr 448 ------- 2 0.1 5.0 1 1
+ 0.114 2 3 cbr 448 ------- 2 0.1 5.0 1 1
- 0.126928 2 3 cbr 448 ------- 2 0.1 5.0 1 1
r 0.226928 2 3 cbr 552 ------- 2 0.1 5.0 0 0
+ 0.226928 3 5 cbr 552 ------- 2 0.1 5.0 0 0
- 0.226928 3 5 cbr 552 ------- 2 0.1 5.0 0 0
r 0.238875 2 3 cbr 448 ------- 2 0.1 5.0 1 1
+ 0.238875 3 5 cbr 448 ------- 2 0.1 5.0 1 1
- 0.238875 3 5 cbr 448 ------- 2 0.1 5.0 1 1
r 0.27576 3 5 cbr 552 ------- 2 0.1 5.0 0 0
r 0.286043 3 5 cbr 448 ------- 2 0.1 5.0 1 1
+ 0.9 0 2 cbr 552 ------- 2 0.1 5.0 2 2
- 0.9 0 2 cbr 552 ------- 2 0.1 5.0 2 2
+ 0.9 0 2 cbr 448 ------- 2 0.1 5.0 3 3
- 0.902208 0 2 cbr 448 ------- 2 0.1 5.0 3 3
r 0.912208 0 2 cbr 552 ------- 2 0.1 5.0 2 2
+ 0.912208 2 3 cbr 552 ------- 2 0.1 5.0 2 2
- 0.912208 2 3 cbr 552 ------- 2 0.1 5.0 2 2
r 0.914 0 2 cbr 448 ------- 2 0.1 5.0 3 3
+ 0.914 2 3 cbr 448 ------- 2 0.1 5.0 3 3
- 0.926928 2 3 cbr 448 ------- 2 0.1 5.0 3 3
```

LEARNING OUTCOME:

- I learnt to implement TCP and UDP connections using ns2
- I learned to analyse the performance
- I Learned to run simulations using nam