

SSN College of Engineering, Kalavakkam
Department of Computer Science and Engineering
UCS1511 NETWORKS LAB

Exercise 8: PERFORMANCE EVALUATION OF TCP AND UDP

Name: Rahul Ram M

Roll No.: 185001121

Date: 05/11/2020

LEARNING OBJECTIVES:

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

ALGORITHM:

1. Create a new object ns with the class Simulator.
2. Open the nam trace file.
3. The namtrace is set for enabling animation to simulate the environment.
4. The color field here is used to discriminate the different data packets travelling across the nodes.
5. Declare the nodes namely n0, n1, ... to be used in the simulation.
6. The duplex links between the nodes is set appropriately.
7. Set the orientation of the nodes appropriately for proper representation.
8. The queue limit is set to determine the capacity of the queue for any communication.
9. A UDP connection is set up between the node n0 and n5.
10. The CBR (constant bit rate) here facilitates this UDP connection.
11. Then TCP connection is setup between the nodes n0 and n4.
12. The FTP here is set as the application layer protocol that uses TCP.
13. 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 Green  
$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 right-up
$ns duplex-link-op $n3 $n5 orient right-down
```

#Setting Queue Limit

```
$ns queue-limit $n2 $n3 5
```

#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
```

#Setup a UDP agent and attach it to node n0

```
set udp [new Agent/UDP]
$ns attach-agent $n1 $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_ 200
$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

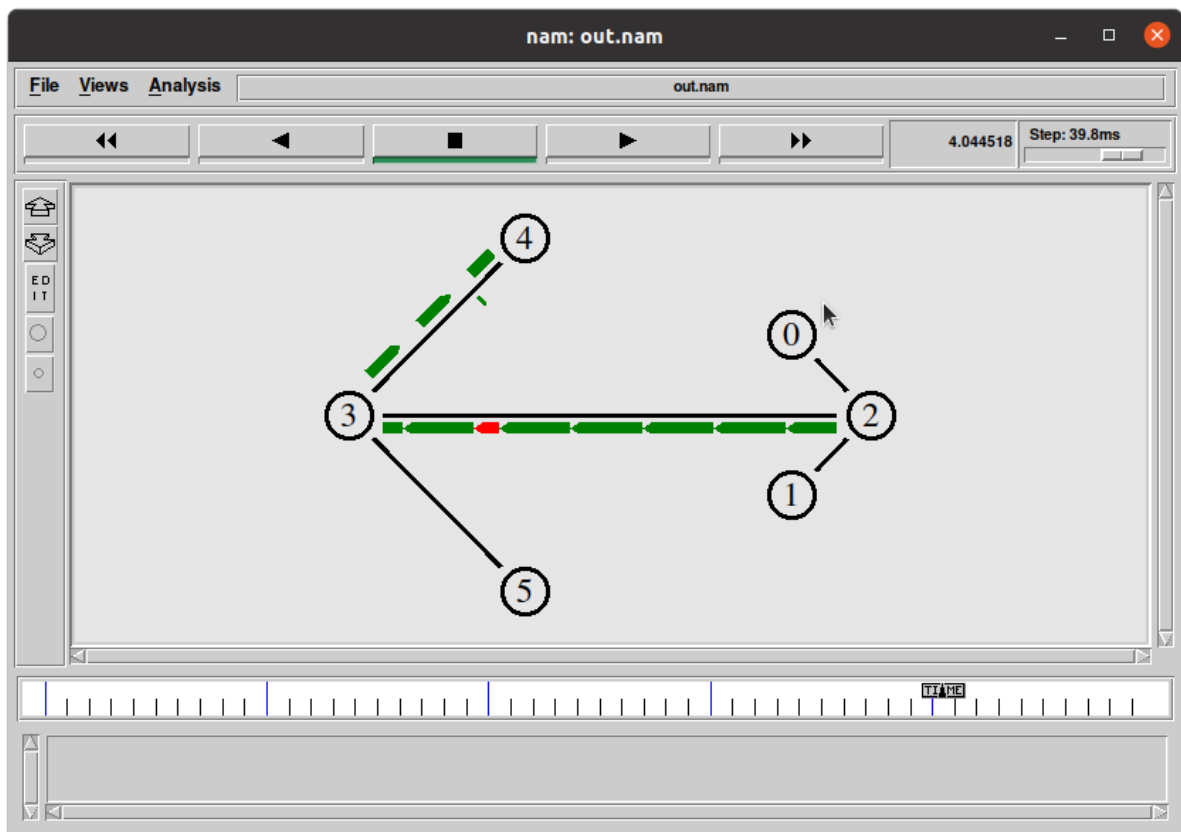
\$ns at 5.0 "finish"

#Run the simulation

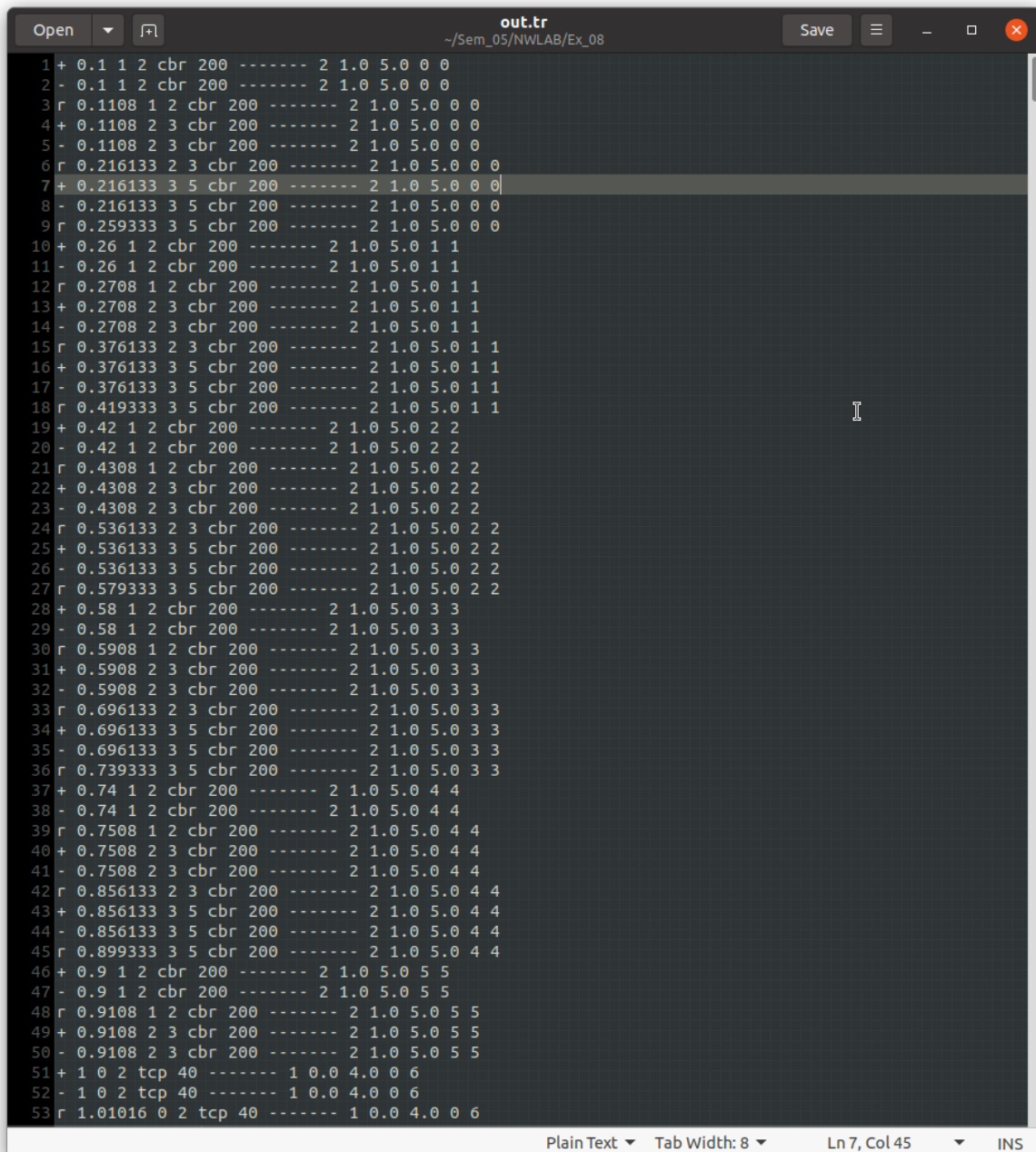
\$ns run

\$ns run

SCREENSHOT:



TRACE FILE:

A screenshot of a terminal window titled 'out.tr' with a subtitle '~/.Sem_05/NWLAB/Ex_08'. The window contains a list of 53 lines of network simulation output. Each line represents a packet event, starting with a sequence number, a plus or minus sign, a timestamp, and packet details. The details include source and destination IP addresses, port numbers, and protocol types (cbr for constant bit rate, tcp for transmission control protocol). The output shows a series of packets being sent and received, with timestamps increasing from 0.1 to 1.01016. The window has a dark background and standard window controls (Open, Save, etc.) at the top. At the bottom, it shows 'Plain Text', 'Tab Width: 8', 'Ln 7, Col 45', and 'INS'.

LEARNING OUTCOME:

This exercise helped me

- To understand the implementation of TCP and UDP connections using ns2.
- To understand how the packets are transferred using simulation and the problems associated such as packet loss, etc.
- To understand the performance of UDP and TCP connections.