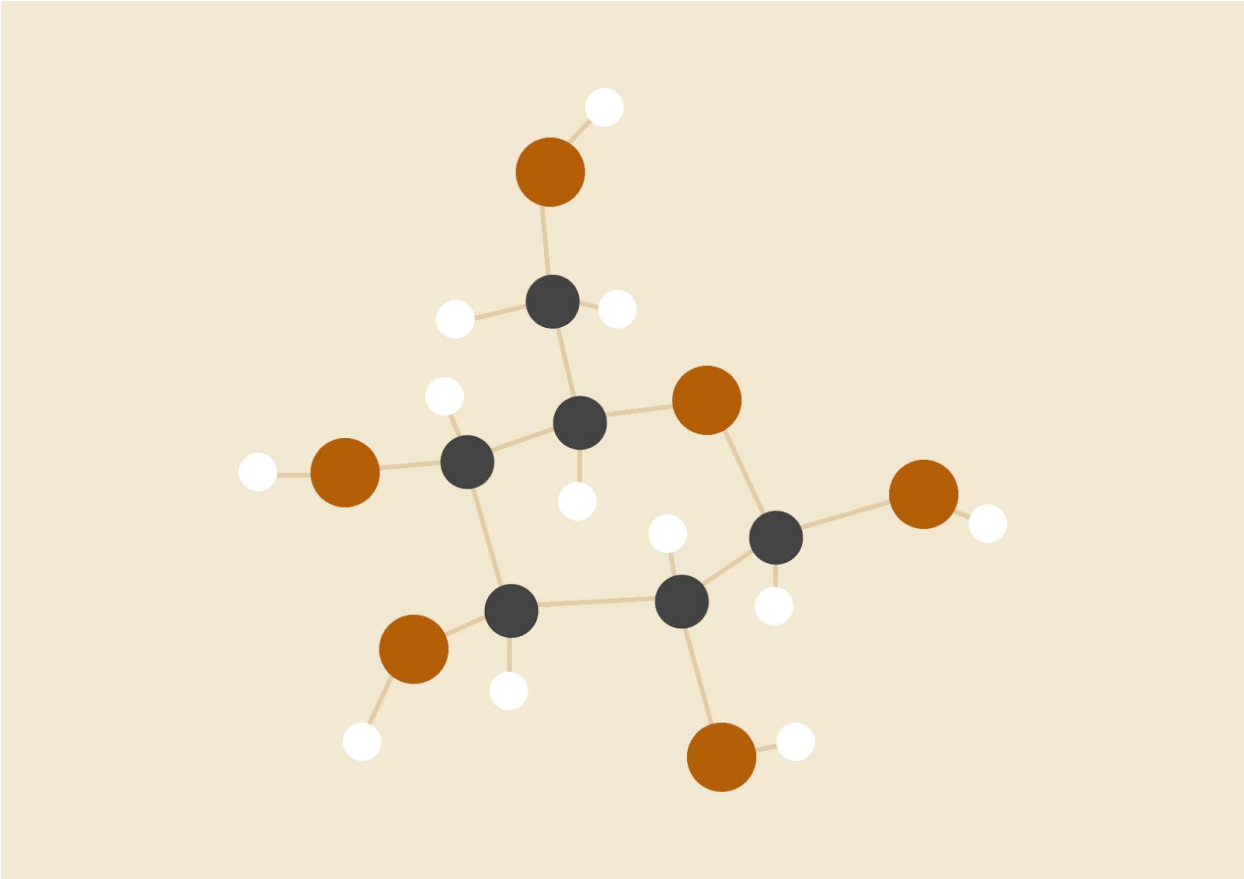
Lab Assignment 4



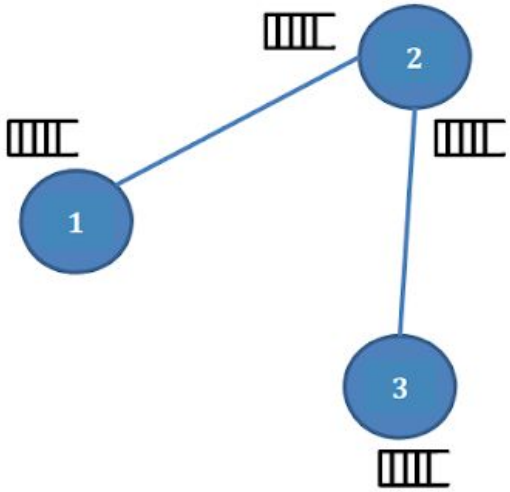
Shashank Kashyap

17114070

CSE III

Problem Statement 1

Write a Network Simulator (NS2) code to simulate a three node network with duplex links among them as shown in figure. Show the topology using NAM. Study the variation in number of packets dropped with the variation of the queue size in the nodes and with the variation of the bandwidth of the links.



Solution:

- We make a ".tcl" file (Q1.tcl) which runs the actual simulation in ns2. The queue_lengths and bandwidths are set in it. It saves the trace file and displays the simulation.

```
set netSim [new Simulator]

# the hyperparameters are set here
set ql1 100
set ql2 100
set bw1 10Mb
set bw2 10Mb

set d1 10ms
set d2 10ms
```

```
set d2 10ms

$netSim color 0 Red
$netSim color 1 Blue
$netSim color 2 Azure
$netSim color 3 Coral
$netSim color 4 Cyan

set f [open Q1.nam w]
$netSim namtrace-all $f

set t [open Q1.tr w]
$netSim trace-all $t

proc finish {} {
      global netSim f t
      $netSim flush-trace
      close $f
      close $t

      # Comment the below line to minimize the ns2 visualization
windows
      # exec nam Q1.nam &
      exec perl count_drops.p Q1.tr > packet_loss_q1.txt
      exit 0
}

set n0 [$netSim node]
set n1 [$netSim node]
set n2 [$netSim node]

$netSim duplex-link $n0 $n1 $bw1 $d1 DropTail
$netSim duplex-link $n2 $n1 $bw2 $d2 DropTail

$netSim queue-limit $n0 $n1 $ql1
$netSim queue-limit $n2 $n1 $ql2

$netSim duplex-link-op $n0 $n1 queuePos 0.5
$netSim duplex-link-op $n2 $n1 queuePos 0.5

set tcp_prot [new Agent/TCP]
```

```
$tcp_prot set class_ 0
$netSim attach-agent $n0 $tcp_prot

set sink [new Agent/TCPSink]
$netSim attach-agent $n1 $sink
$netSim connect $tcp_prot $sink
$tcp_prot set fid_ 1

set ftp_prot0 [new Application/FTP]
$ftp_prot0 set packetSize_ 500
$ftp_prot0 attach-agent $tcp_prot
$ftp_prot0 set type_ FTP

set tcp_prot [new Agent/TCP]
$tcp_prot set class_ 1
$netSim attach-agent $n2 $tcp_prot

set sink [new Agent/TCPSink]
$netSim attach-agent $n1 $sink
$netSim connect $tcp_prot $sink
$tcp_prot set fid_ 2

set ftp_prot2 [new Application/FTP]
$ftp_prot2 set packetSize_ 500
$ftp_prot2 attach-agent $tcp_prot
$ftp_prot2 set type_ FTP

$netSim at 0.1 "$ftp_prot0 start"
$netSim at 1.6 "$ftp_prot0 stop"

$netSim at 0.1 "$ftp_prot2 start"
$netSim at 1.6 "$ftp_prot2 stop"

$netSim at 2.0 "finish"

$netSim run
```

- We use a perl script (count_drops.p) which uses the trace file and counts the number of drops (occurrences of “d”) in it for each link. It stores these

values in a text file (packet_loss_q1.txt)

```
#!/usr/bin/perl

# script to find the amount of packet drops using
# the trace file saved as 1.tr

# works by finding the number of "d"
# which denotes a packet loss
# in the trace file

use :strict;
if($#ARGV<0){
printf("Usage: <trace-file>\n");
exit 1;
}

# to open the given trace file
open(Trace, $ARGV[0]) or die "Cannot open the trace file";
my $sc = 0; # sending counter
my $rc = 0; # receiving counter
my $rp = 0;
my $mc =0;
my $d_udp = [0,0];
my $d_tcp = [0,0];
my $t_tcp = [0,0];
my $t_udp = [0,0];

my %pkt_fc = (); #packet forwarding counter
while(<Trace>){ # read one line in from the file
my @line = split; #split the line with delimin as space


if($line[4] eq "cbr"){
      if($line[2] eq "0"){
      $t_udp[0]++;
      if($line[0] eq "d"){
            $d_udp[0]++;
      }
      }
      if($line[2] eq "2"){
```

```
	$t_udp[1]++;
	if($line[0] eq "d"){
		$d_udp[1]++;
	}
	}
}

if($line[4] eq "tcp"){
	if($line[2] eq "0"){
	$t_tcp[0]++;
	if($line[0] eq "d"){
		$d_tcp[0]++;
	}
	}
	if($line[2] eq "2"){
	$t_tcp[1]++;
	if($line[0] eq "d"){
		$d_tcp[1]++;
	}
	}
}
}
my $i = 0;

# whatever this is printing will get appended to packet_loss.txt
# rather than being displayed on terminal
while($i<2){
	if($i==0){
	printf("Node 0 to 1\n");
	}else{
	printf("Node 2 to 1\n");
	}
	printf("%f\n",$t_tcp[$i]);
	# Total tcp length
	# printf("Total udp length %f\n",$t_udp[$i]);

	printf("%f\n",$d_tcp[$i]);
	# Dropped tcp length
	# printf("Dropped udp length %f\n\n",$d_udp[$i]);
	$i++;
}
```

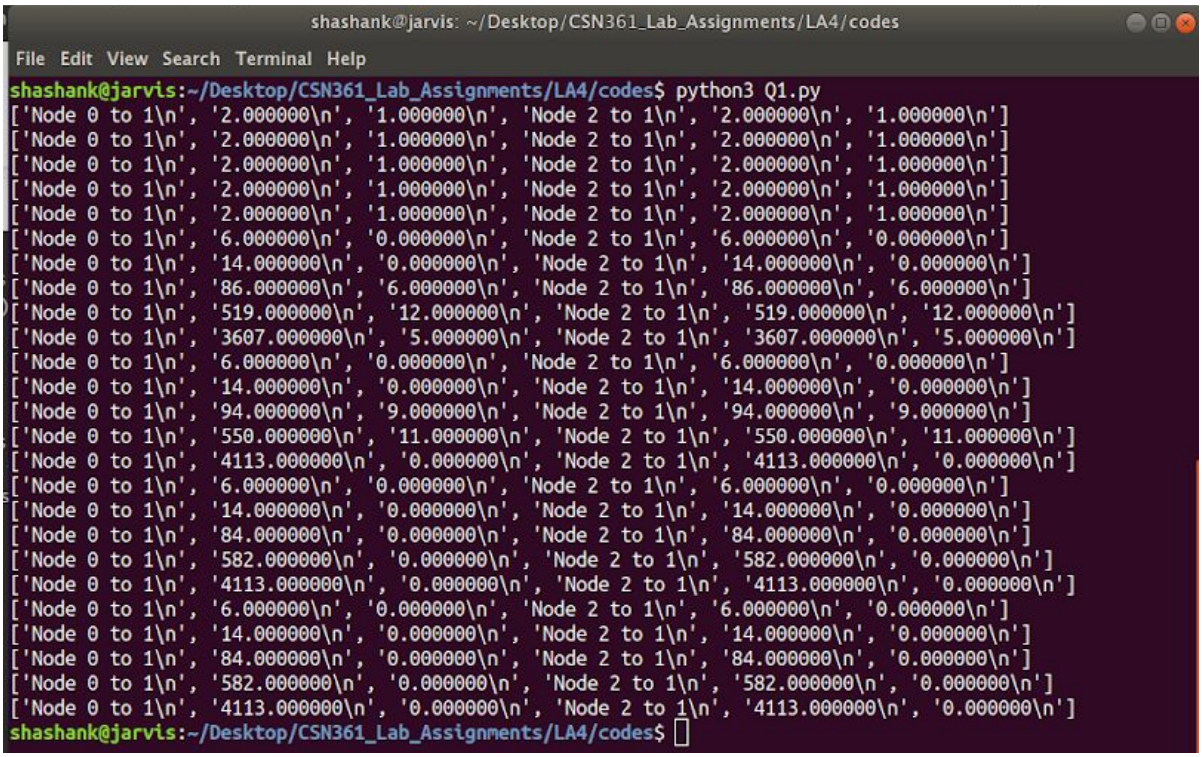
- In order to run Q1.tcl file with a bunch of queue length and bandwidth settings and keep track of packet_loss in all those cases to finally make a graph, we use a python script (Q1.py). It stores the data (packet loss information) in a csv file for better visualization.

```
# Script to automate the running of different
# combinations of queue_lengths and bandwidth
# and writing them into csv file

import os
import csv

# declare the possible bandwidths and queue_lengths
queue_lengths = ['1','5','10','50','100']
bandwidths = ['1Kb', '10Kb', '100Kb', '1Mb', '10Mb']

# open csv file for editing
csv_file = open('Q1_analysis.csv','w')
w = csv.writer(csv_file)

for queue_length in queue_lengths:
      for bandwidth in bandwidths:
      # for each combination of qlen and bandwidth
      with open('Q1.tcl','r') as file:
            data = file.readlines()
      l = []

      # Set the appropriate values in Q1.tcl file
      l.append(queue_length)
      data[2] = "set ql1 " + queue_length + "\n"

      l.append(queue_length)
      data[3] = "set ql2 " + queue_length + "\n"

      l.append(bandwidth)
      data[5] = "set bw1 " + bandwidth + "\n"

      l.append(bandwidth)
      data[6] = "set bw2 " + bandwidth + "\n"

      with open('Q1.tcl','w') as file:
```

```
            file.writelines(data)

      # run the simulation
      os.system("ns Q1.tcl")

      # see result in packet_loss.txt and add to csv file
      with open('packet_loss_q1.txt','r') as file:
            data = file.readlines()
      print((data))
      l.append(data[1])
      l.append(data[2])
      l.append(data[4])
      l.append(data[5])

      w.writerow(l)

csv_file.close()
```

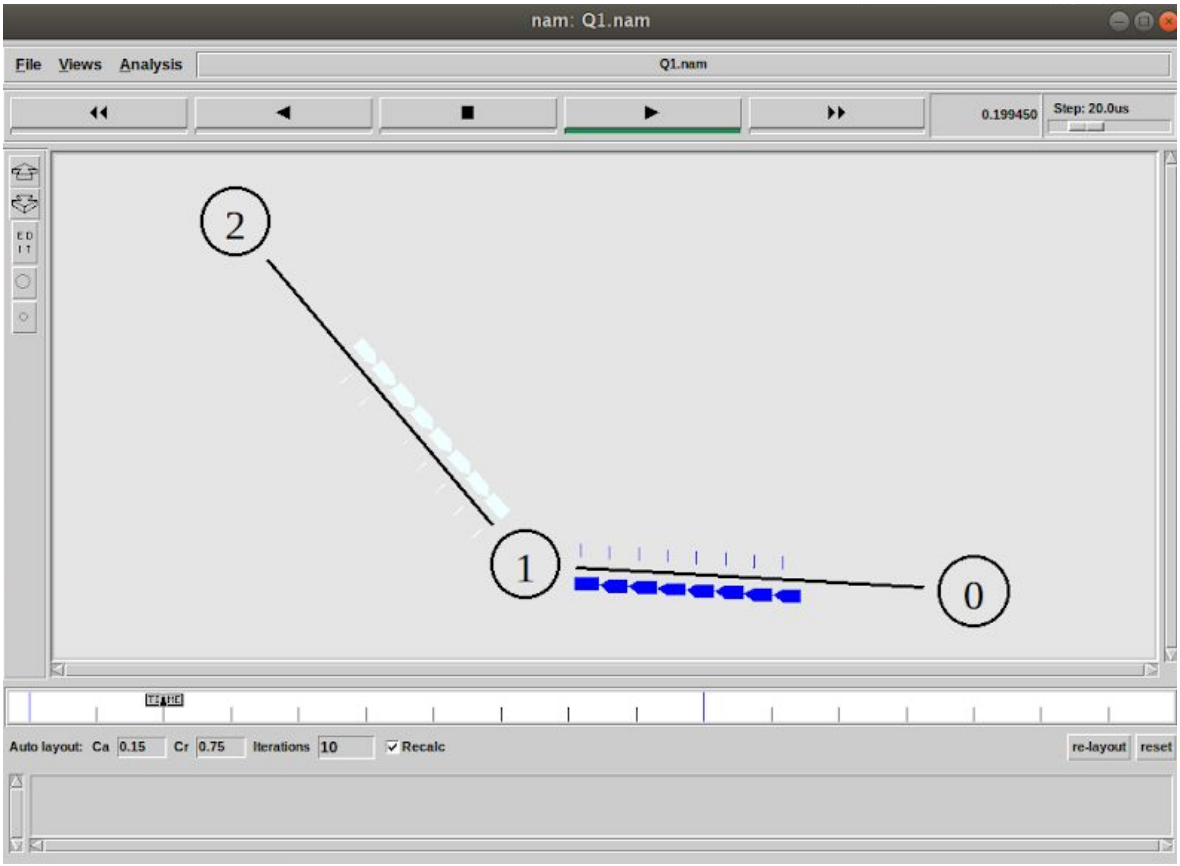
Execution:

- Terminal displays the packet loss information for all the cases.



- Example packet_loss.txt file

- The simulation



- Final analysis table

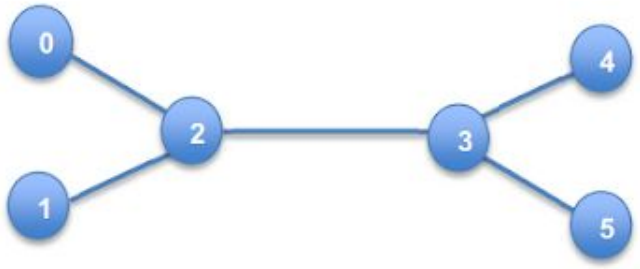
Queue Length	Bandwidth	Node 0 to 1		Node 2 to 1	
		Total sent	Packet dropped	Total sent	Packet dropped
1	1Kb	2	1	2	1
1	100Kb	2	1	2	1
1	10Mb	2	1	2	1
10	1Kb	6	0	6	0
10	100Kb	94	9	94	9
10	10Mb	4113	0	4113	0
100	1Kb	6	0	6	0
100	100Kb	84	0	84	0
100	10Mb	4113	0	4113	0

Problem Statement 2

Write a Network Simulator (NS2) code to simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion. Study the variation in number of packets dropped with the variation of the queue size in the nodes and with the variation of the bandwidth of the links.

Nodes are connected as follows: 0-2, 1-2, 2-3, 3-4 and 3-5

Packet transmissions: 0-4 and 5-1



Solution:

- We have a Q2.tcl file which is the file used to run the ns2 simulation. We define the queue length and bandwidth for the links in this file and the trace file is generated.

```
set netSim [ new Simulator ]

set nf [ open Q2.nam w ]
$netSim namtrace-all $nf

set tf [ open Q2.tr w ]
$netSim trace-all $tf

set node0 [$netSim node]
set node1 [$netSim node]
set node2 [$netSim node]
```

```
set node3 [$netSim node]
set node4 [$netSim node]
set node5 [$netSim node]

set queue_length_02 10
set queue_length_23 10
set bandwidth_02 10Kb
set bandwidth_23 10Kb

$netSim duplex-link $node0 $node2 $bandwidth_02 1ms DropTail
$netSim duplex-link $node1 $node2 $bandwidth_02 1ms DropTail
$netSim duplex-link $node2 $node3 $bandwidth_23 1ms DropTail
$netSim duplex-link $node3 $node4 $bandwidth_02 1ms DropTail
$netSim duplex-link $node3 $node5 $bandwidth_02 1ms DropTail

set pingagent0 [new Agent/Ping]
$netSim attach-agent $node0 $pingagent0
$pingagent0 set packetSize_ 50000
$pingagent0 set interval_ 0.0001

set pingagent1 [new Agent/Ping]
$netSim attach-agent $node1 $pingagent1

set pingagent2 [new Agent/Ping]
$netSim attach-agent $node2 $pingagent2

set pingagent3 [new Agent/Ping]
$netSim attach-agent $node3 $pingagent3

set pingagent4 [new Agent/Ping]
$netSim attach-agent $node4 $pingagent4

set pingagent5 [new Agent/Ping]
$netSim attach-agent $node5 $pingagent5
$pingagent5 set packetSize_ 30000
$pingagent5 set interval_ 0.00001

$netSim queue-limit $node0 $node2 $queue_length_02
$netSim queue-limit $node2 $node3 $queue_length_23
$netSim queue-limit $node3 $node5 $queue_length_02
$netSim queue-limit $node5 $node3 $queue_length_02
```

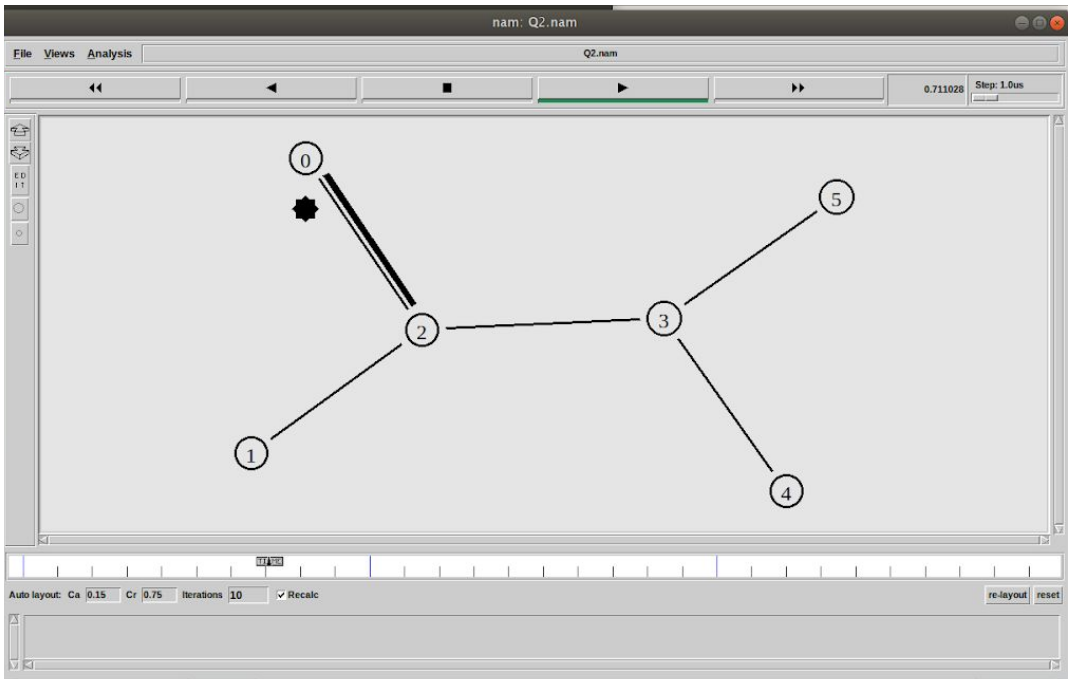
```
$netSim queue-limit $node3 $node2 $queue_length_02
$netSim queue-limit $node2 $node1 $queue_length_02
$netSim queue-limit $node3 $node4 $queue_length_02

Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_ id]received answer from $from with round trip time
$rtt msec"
}

$netSim connect $pingagent0 $pingagent4
$netSim connect $pingagent5 $pingagent0

proc finish { } {
global netSim nf tf
$netSim flush-trace
close $nf
close $tf
exec awk -f drop_count.awk Q2.tr &
exec nam Q2 &
exit 0
exit 0
}

for {set i_int 1} {$i_int < 30} {incr i_int} {
set i [expr {$i_int * 0.1}]
$netSim at $i "$pingagent0 send"
}

for {set i_int 1} {$i_int < 30} {incr i_int} {
set i [expr {$i_int * 0.1}]
$netSim at $i "$pingagent0 send"
}

$netSim at 3.0 "finish"
$netSim run
```

- We then use drop_count.awk file to count the total number of drops in the nodes.

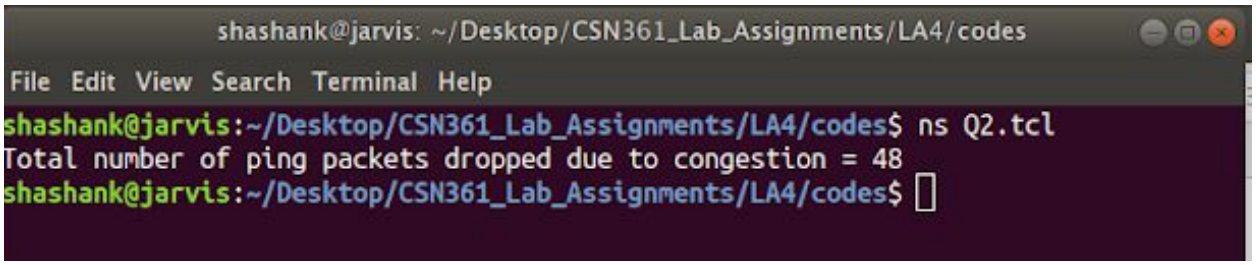
```
# count the number of "d" in the Q2.tr file
# to find the number of packet loss

BEGIN {
  drop=0;
}

{
  if($1=="d") {
      drop++;
  }
}

END{
  printf("Total number of %s packets dropped due to congestion =
%d\n",$5,drop);
}
```

Execution:

- The NS2 simulation



- Total number of packets dropped



- The analysis can be made by trying (setting in Q2.tcl file) out different values of queue_length and bandwidth and observing the packet_loss.