Routing Algorithm – RIP

Group Members Details: -

1). Name – Abhay Ray
Roll No - S20200010002

2). Name – Utkarsh Verma
Roll No – S20200010215

3). Name – S. Kodanda Rama Phaneendra
Roll No – S20200010184

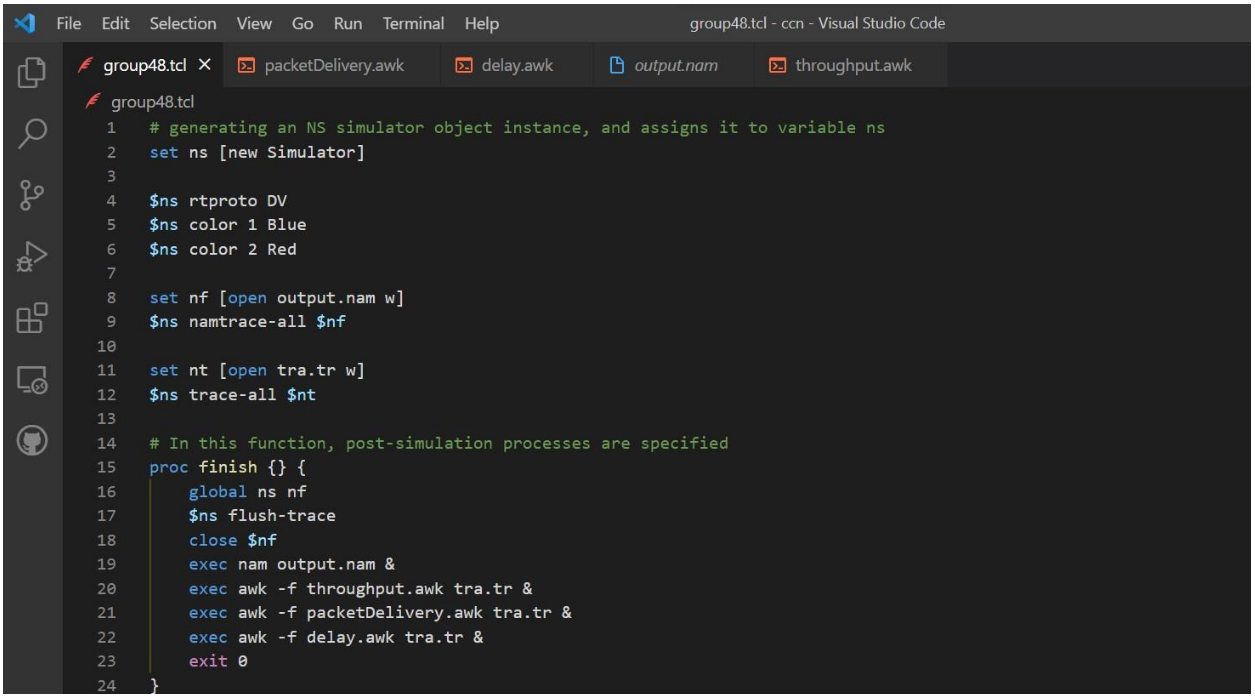
4). Name - P. Manohar Sashank
Roll No - S20200010163

In this project we tried to implement Routing Information Protocol, Sometimes known as RIP algorithm using ns2 and network animator(Xming in this project).

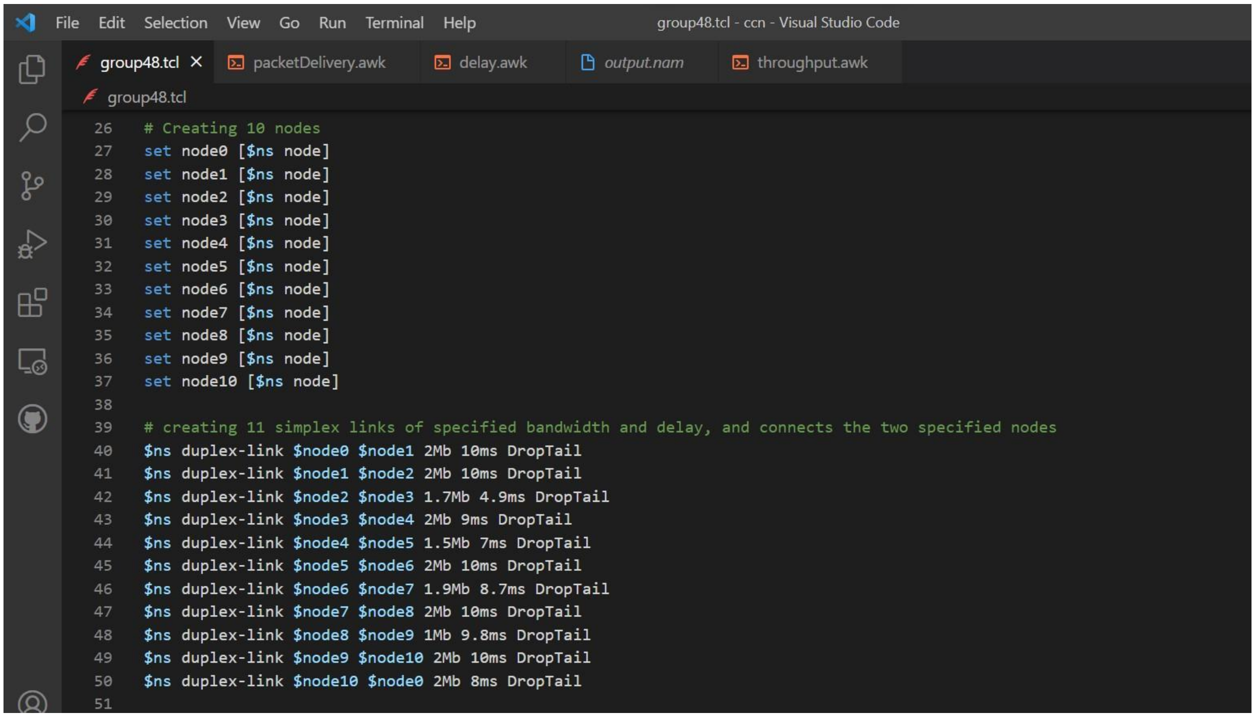
This project initially contains 4 files, one of which is a ".tcl" file and the rest three are ".awk" files.

The names of the four files are as follows: -

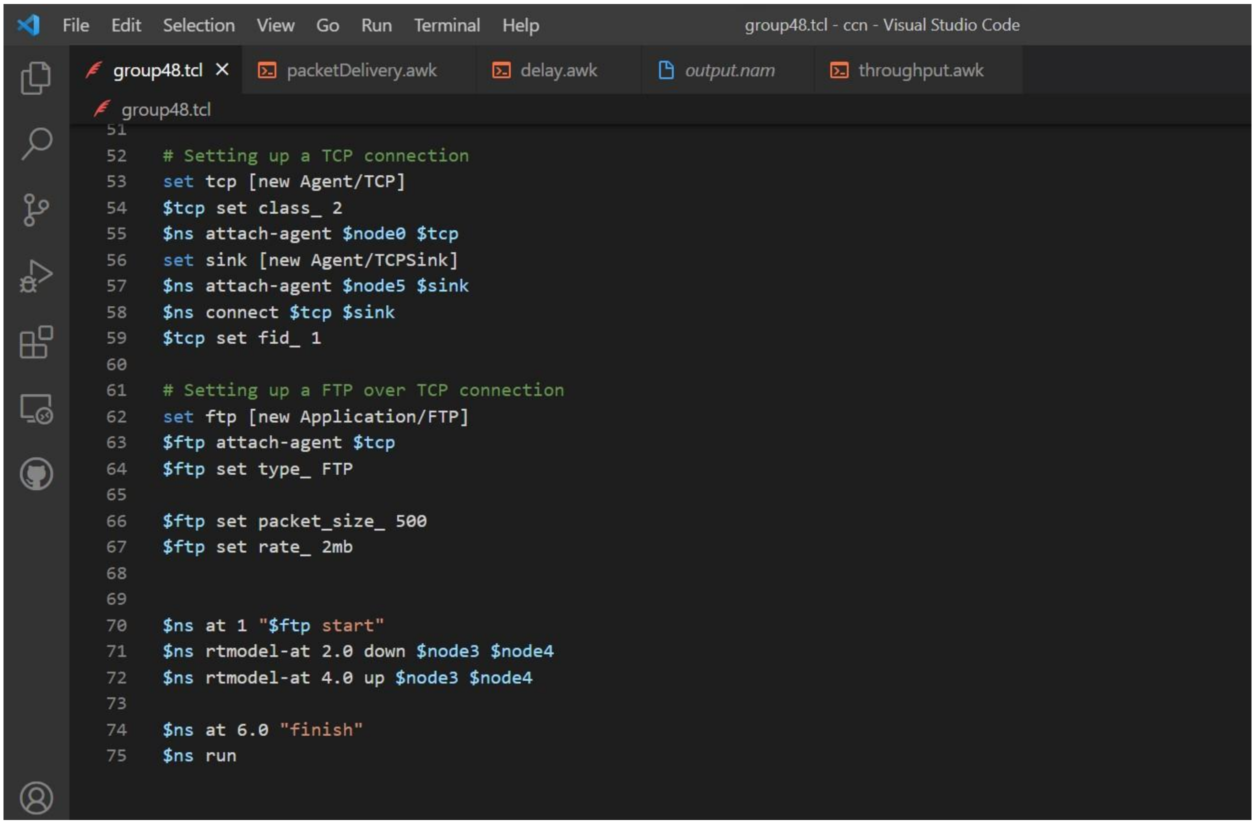
1. Group48.tcl (the code snippets are given below): -


File Edit Selection View Go Run Terminal Help
group48.tcl - ccn - Visual Studio Code
group48.tcl
packetDelivery.awk
delay.awk
output.nam
throughput.awk
group48.tcl

```
# generating an NS simulator object instance, and assigns it to variable ns
set ns [new Simulator]

$ns rtproto DV
$ns color 1 Blue
$ns color 2 Red

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

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

# In this function, post-simulation processes are specified
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam output.nam &
    exec awk -f throughput.awk tra.tr &
    exec awk -f packetDelivery.awk tra.tr &
    exec awk -f delay.awk tra.tr &
    exit 0
}
```

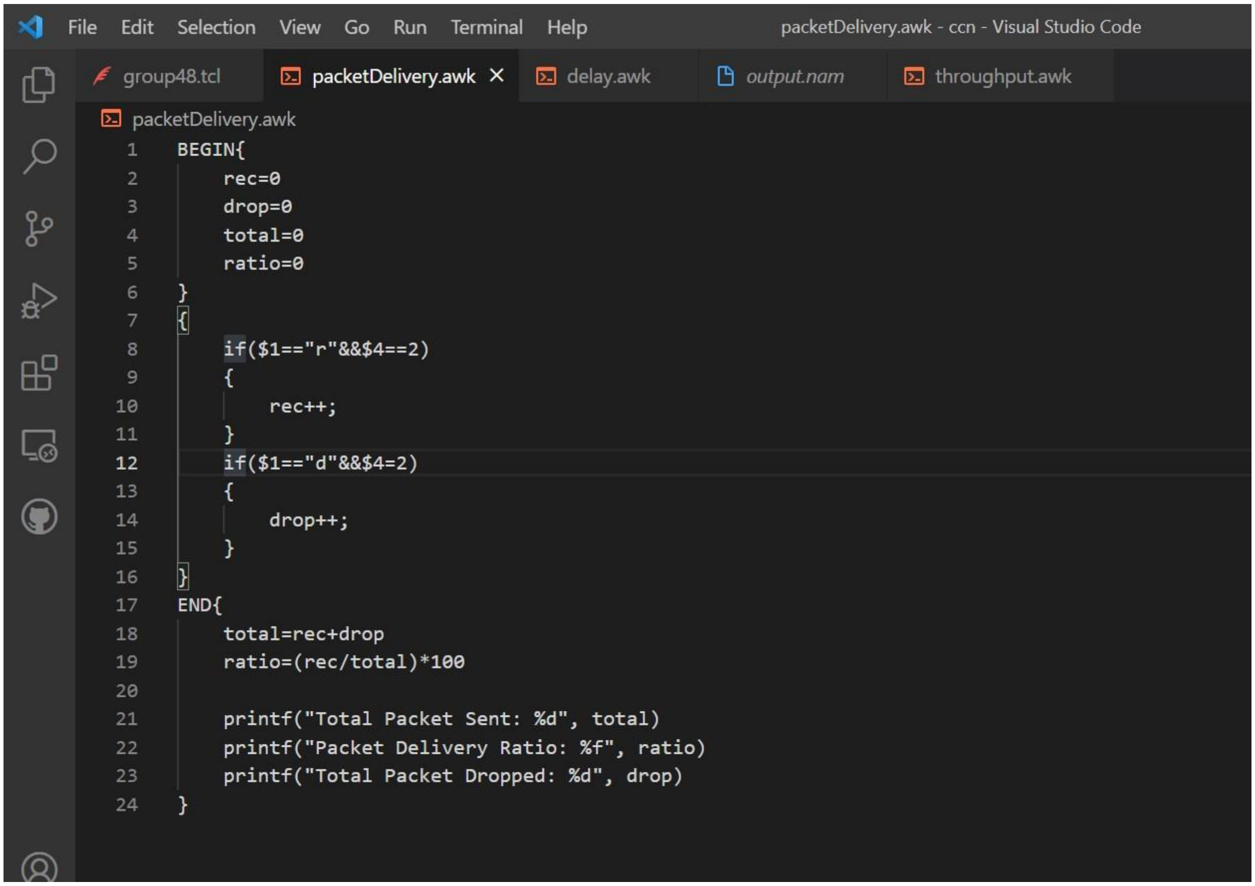

File Edit Selection View Go Run Terminal Help
group48.tcl - ccn - Visual Studio Code
group48.tcl
packetDelivery.awk
delay.awk
output.nam
throughput.awk
group48.tcl

```
# Creating 10 nodes
set node0 [$ns node]
set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]
set node5 [$ns node]
set node6 [$ns node]
set node7 [$ns node]
set node8 [$ns node]
set node9 [$ns node]
set node10 [$ns node]

# creating 11 simplex links of specified bandwidth and delay, and connects the two specified nodes
$ns duplex-link $node0 $node1 2Mb 10ms DropTail
$ns duplex-link $node1 $node2 2Mb 10ms DropTail
$ns duplex-link $node2 $node3 1.7Mb 4.9ms DropTail
$ns duplex-link $node3 $node4 2Mb 9ms DropTail
$ns duplex-link $node4 $node5 1.5Mb 7ms DropTail
$ns duplex-link $node5 $node6 2Mb 10ms DropTail
$ns duplex-link $node6 $node7 1.9Mb 8.7ms DropTail
$ns duplex-link $node7 $node8 2Mb 10ms DropTail
$ns duplex-link $node8 $node9 1Mb 9.8ms DropTail
$ns duplex-link $node9 $node10 2Mb 10ms DropTail
$ns duplex-link $node10 $node0 2Mb 8ms DropTail
```



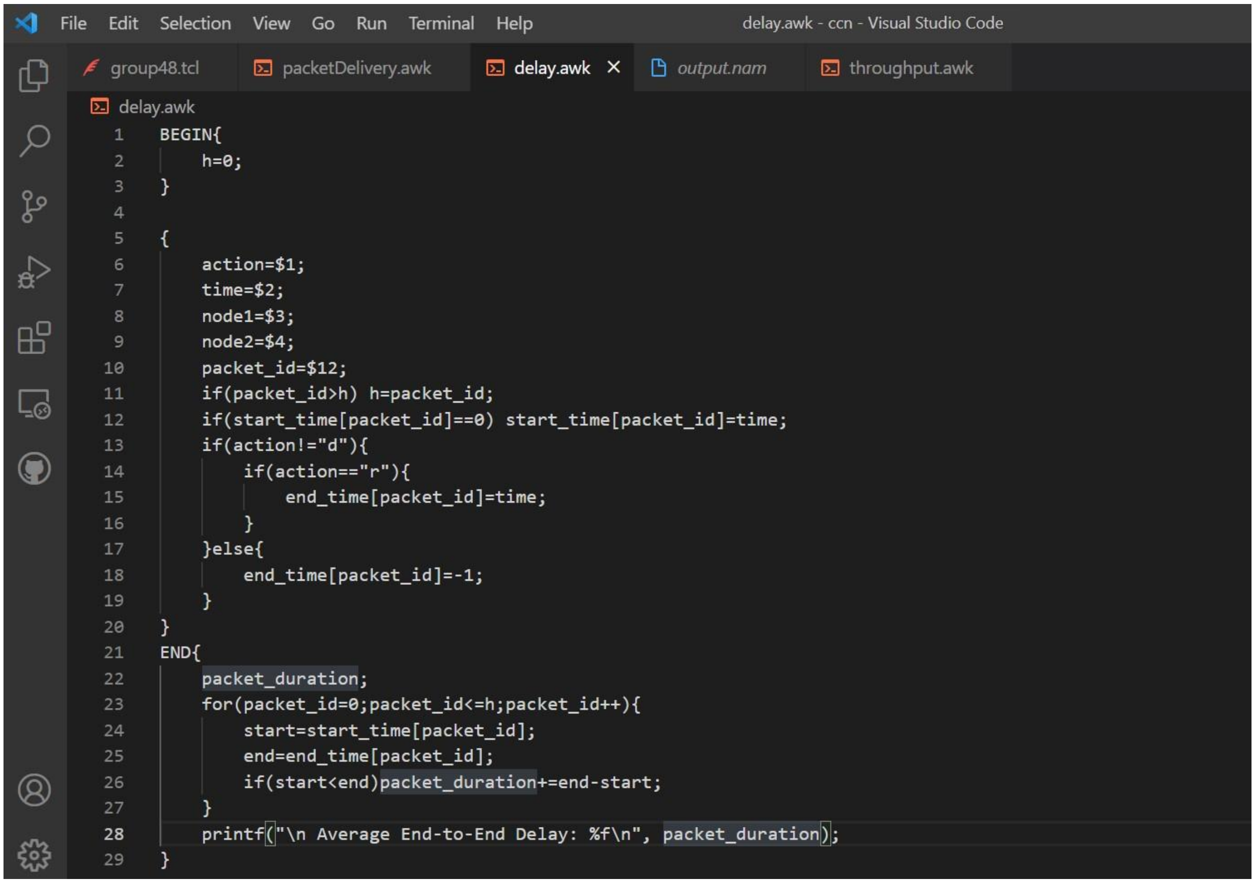
```
# Setting up a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $node0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $node5 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

# Setting up a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

$ftp set packet_size_ 500
$ftp set rate_ 2mb



$ns at 1 "$ftp start"
$ns rtmodel-at 2.0 down $node3 $node4
$ns rtmodel-at 4.0 up $node3 $node4

$ns at 6.0 "finish"
$ns run
```

2. packetDilevrey.awk



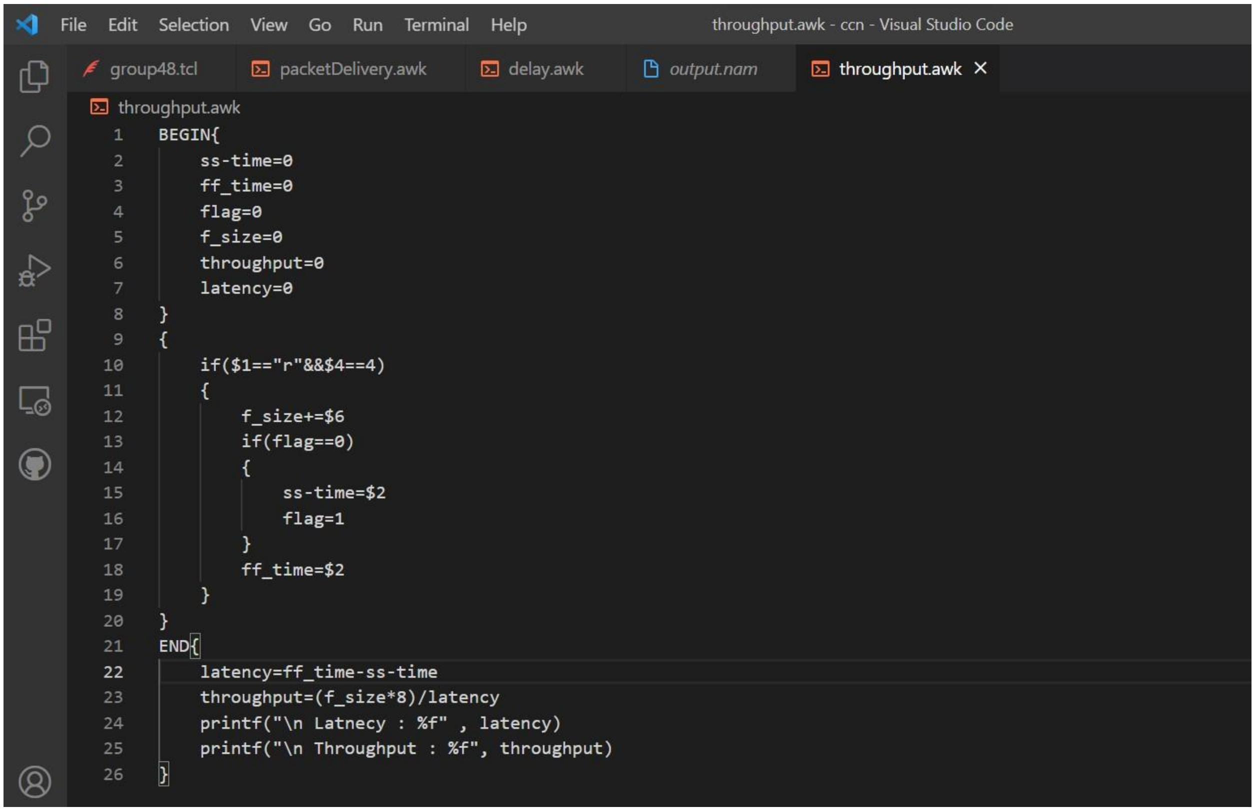
```
BEGIN{
    rec=0
    drop=0
    total=0
    ratio=0
}
{
    if($1=="r"&&$4==2)
    {
        rec++;
    }
    if($1=="d"&&$4=2)
    {
        drop++;
    }
}
END{
    total=rec+drop
    ratio=(rec/total)*100

    printf("Total Packet Sent: %d", total)
    printf("Packet Delivery Ratio: %f", ratio)
    printf("Total Packet Dropped: %d", drop)
}
```

3. delay.awk



```
BEGIN{
    h=0;
}

{
    action=$1;
    time=$2;
    node1=$3;
    node2=$4;
    packet_id=$12;
    if(packet_id>h) h=packet_id;
    if(start_time[packet_id]==0) start_time[packet_id]=time;
    if(action!="d"){
        if(action=="r"){
            end_time[packet_id]=time;
        }
    }else{
        end_time[packet_id]=-1;
    }
}
END{
    packet_duration;
    for(packet_id=0;packet_id<=h;packet_id++){
        start=start_time[packet_id];
        end=end_time[packet_id];
        if(start<end)packet_duration+=end-start;
    }
    printf("\n Average End-to-End Delay: %f\n", packet_duration);
}
```

4. Throughput.awk


File Edit Selection View Go Run Terminal Help
throughput.awk - ccn - Visual Studio Code
group48.tcl
packetDelivery.awk
delay.awk
output.nam
throughput.awk
throughput.awk

```
BEGIN{
    ss-time=0
    ff_time=0
    flag=0
    f_size=0
    throughput=0
    latency=0
}
{
    if($1=="r"&&$4==4)
    {
        f_size+=$6
        if(flag==0)
        {
            ss-time=$2
            flag=1
        }
        ff_time=$2
    }
}
END{
    latency=ff_time-ss-time
    throughput=(f_size*8)/latency
    printf("\n Latnecy : %f" , latency)
    printf("\n Throughput : %f", throughput)
}
```

After running the group48.tcl file in the terminal it will create two more file

1) output.nam
2) tra.tr

How to see the output of the project: -

To run the group48.tcl file the steps are as follows

Step 1 – Extract the .zip file in any folder.

Step 2 – Open the folder in the terminal.

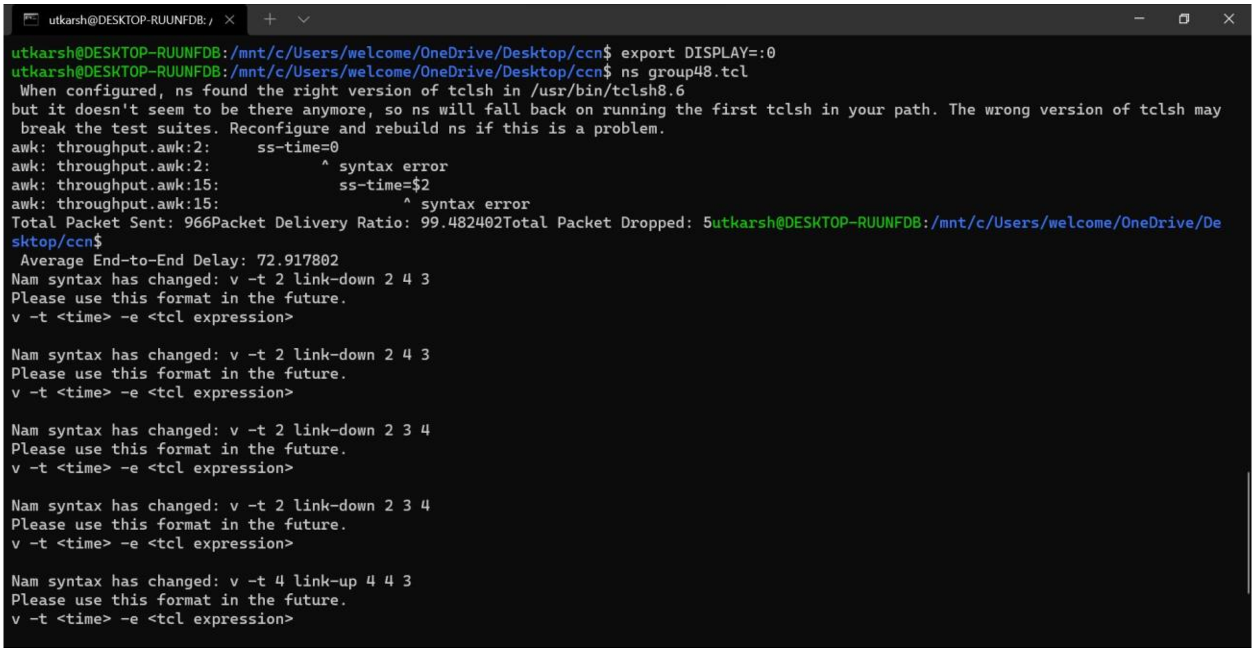
Step 3 – Start your Xming server by launching Xlaunch.

Step 4 – Now type “export DISPLAY=:0” as it is in the terminal.

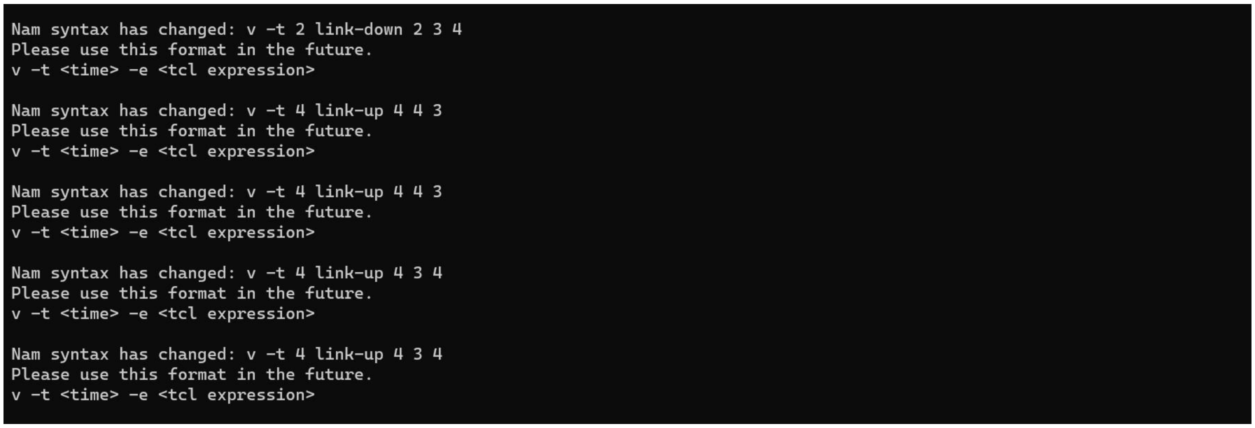
Step 5 – After entering the above command enter “ ns group48.tcl” this will open the animator used in this project

Step 6 – Click on start button and when the time reaches 1s you will be able to see the flowing of the packets. After 2s the you will see that the packets from duplex line 3-4 are being dropped and hence the algorithm finds another path to reach the destination.

Output in terminal: -



```
utkarsh@DESKTOP-RUUNFDB:/mnt/c/Users/welcome/OneDrive/Desktop/ccn$ export DISPLAY=:0
utkarsh@DESKTOP-RUUNFDB:/mnt/c/Users/welcome/OneDrive/Desktop/ccn$ ns group48.tcl
 When configured, ns found the right version of tclsh in /usr/bin/tclsh8.6
but it doesn't seem to be there anymore, so ns will fall back on running the first tclsh in your path. The wrong version of tclsh may
 break the test suites. Reconfigure and rebuild ns if this is a problem.
awk: throughput.awk:2:     ss-time=0
awk: throughput.awk:2:            ^ syntax error
awk: throughput.awk:15:            ss-time=$2
awk: throughput.awk:15:                   ^ syntax error
Total Packet Sent: 966Packet Delivery Ratio: 99.482402Total Packet Dropped: 5utkarsh@DESKTOP-RUUNFDB:/mnt/c/Users/welcome/OneDrive/De
sktop/ccn$
 Average End-to-End Delay: 72.917802
Nam syntax has changed: v -t 2 link-down 2 4 3
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2 link-down 2 4 3
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2 link-down 2 3 4
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2 link-down 2 3 4
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 4 link-up 4 4 3
Please use this format in the future.
v -t <time> -e <tcl expression>
```



```
Nam syntax has changed: v -t 2 link-down 2 3 4
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 4 link-up 4 4 3
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 4 link-up 4 4 3
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 4 link-up 4 3 4
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 4 link-up 4 3 4
Please use this format in the future.
v -t <time> -e <tcl expression>
```