

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

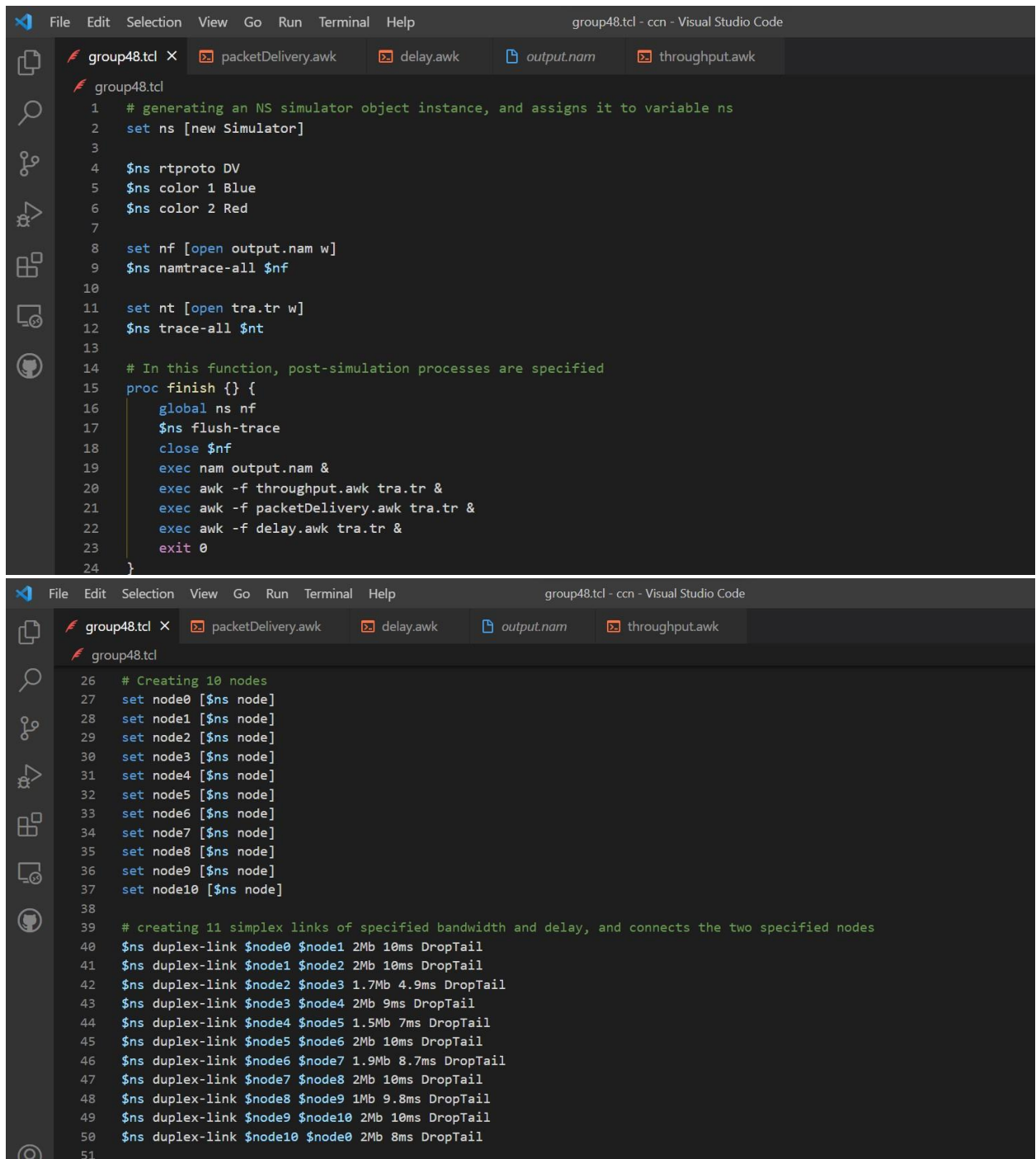
- 4). Name - P. Manohar Sashank
Roll No - S20200010163

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.

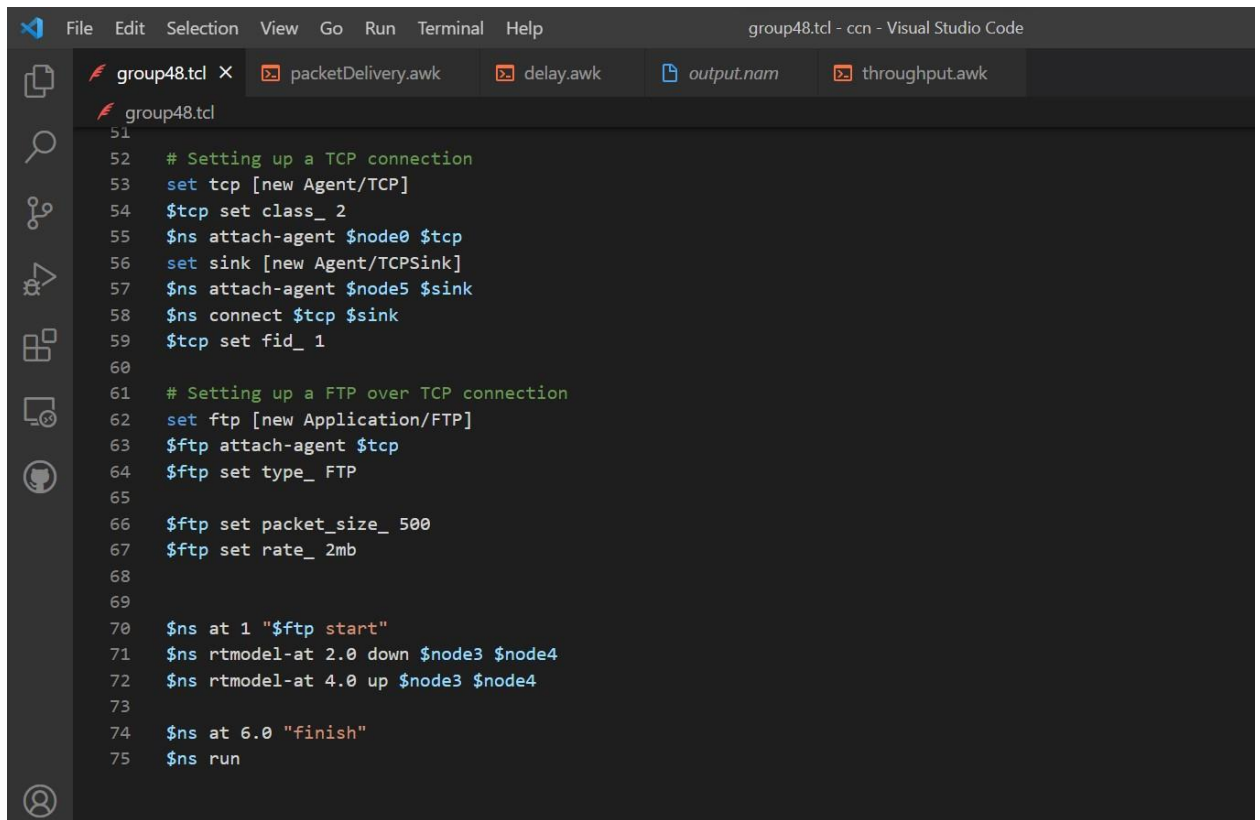
The names of the four files are as follows: -

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



```
group48.tcl
1  # generating an NS simulator object instance, and assigns it to variable ns
2  set ns [new Simulator]
3
4  $ns rtproto DV
5  $ns color 1 Blue
6  $ns color 2 Red
7
8  set nf [open output.nam w]
9  $ns namtrace-all $nf
10
11 set nt [open tra.tr w]
12 $ns trace-all $nt
13
14 # In this function, post-simulation processes are specified
15 proc finish {} {
16     global ns nf
17     $ns flush-trace
18     close $nf
19     exec nam output.nam &
20     exec awk -f throughput.awk tra.tr &
21     exec awk -f packetDelivery.awk tra.tr &
22     exec awk -f delay.awk tra.tr &
23     exit 0
24 }

group48.tcl
26 # Creating 10 nodes
27 set node0 [$ns node]
28 set node1 [$ns node]
29 set node2 [$ns node]
30 set node3 [$ns node]
31 set node4 [$ns node]
32 set node5 [$ns node]
33 set node6 [$ns node]
34 set node7 [$ns node]
35 set node8 [$ns node]
36 set node9 [$ns node]
37 set node10 [$ns node]
38
39 # creating 11 simplex links of specified bandwidth and delay, and connects the two specified nodes
40 $ns duplex-link $node0 $node1 2Mb 10ms DropTail
41 $ns duplex-link $node1 $node2 2Mb 10ms DropTail
42 $ns duplex-link $node2 $node3 1.7Mb 4.9ms DropTail
43 $ns duplex-link $node3 $node4 2Mb 9ms DropTail
44 $ns duplex-link $node4 $node5 1.5Mb 7ms DropTail
45 $ns duplex-link $node5 $node6 2Mb 10ms DropTail
46 $ns duplex-link $node6 $node7 1.9Mb 8.7ms DropTail
47 $ns duplex-link $node7 $node8 2Mb 10ms DropTail
48 $ns duplex-link $node8 $node9 1Mb 9.8ms DropTail
49 $ns duplex-link $node9 $node10 2Mb 10ms DropTail
50 $ns duplex-link $node10 $node0 2Mb 8ms DropTail
51
```

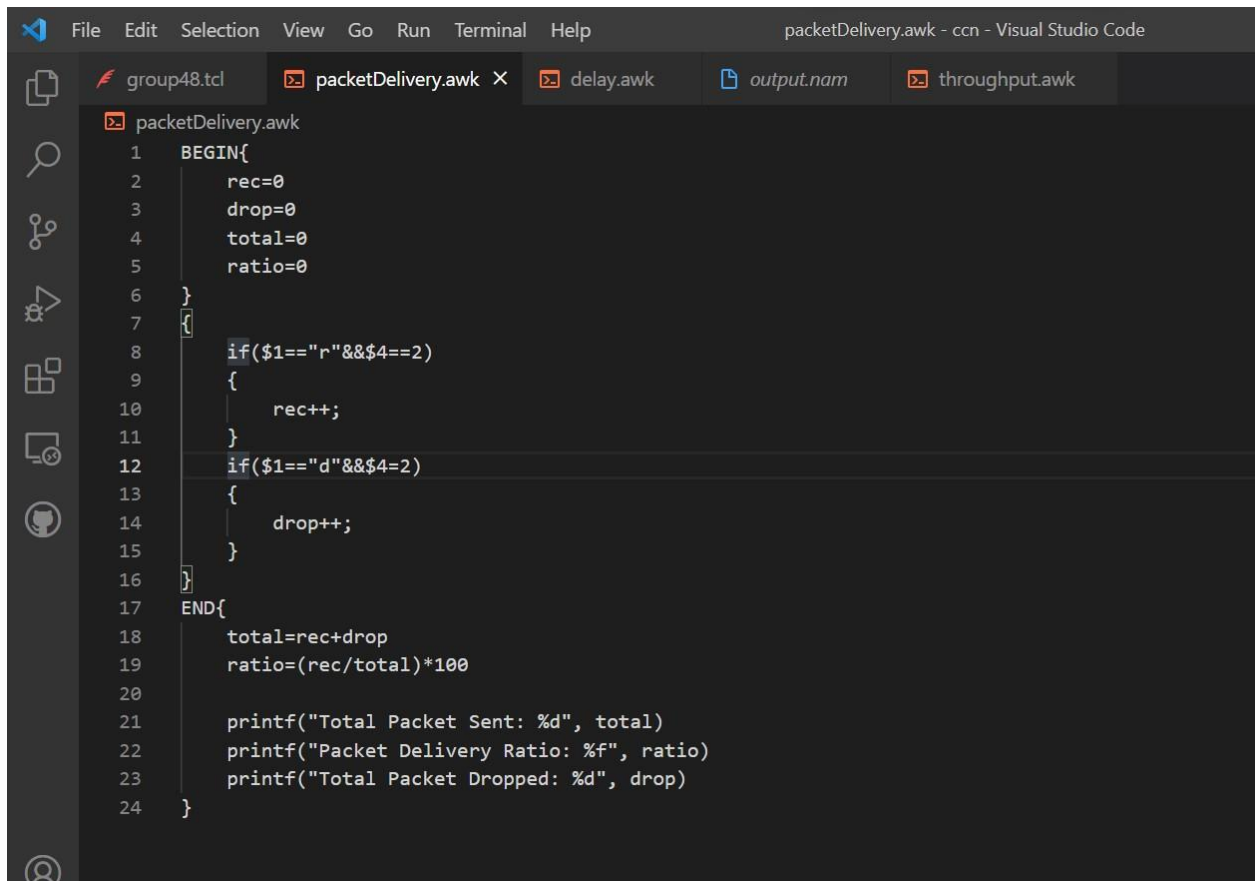


```
File Edit Selection View Go Run Terminal Help group48.tcl - ccn - Visual Studio Code

group48.tcl X packetDelivery.awk delay.awk output.nam throughput.awk

group48.tcl
51
52 # Setting up a TCP connection
53 set tcp [new Agent/TCP]
54 $tcp set class_ 2
55 $ns attach-agent $node0 $tcp
56 set sink [new Agent/TCPSink]
57 $ns attach-agent $node5 $sink
58 $ns connect $tcp $sink
59 $tcp set fid_ 1
60
61 # Setting up a FTP over TCP connection
62 set ftp [new Application/FTP]
63 $ftp attach-agent $tcp
64 $ftp set type_ FTP
65
66 $ftp set packet_size_ 500
67 $ftp set rate_ 2mb
68
69
70 $ns at 1 "$ftp start"
71 $ns rtmodel-at 2.0 down $node3 $node4
72 $ns rtmodel-at 4.0 up $node3 $node4
73
74 $ns at 6.0 "finish"
75 $ns run
```

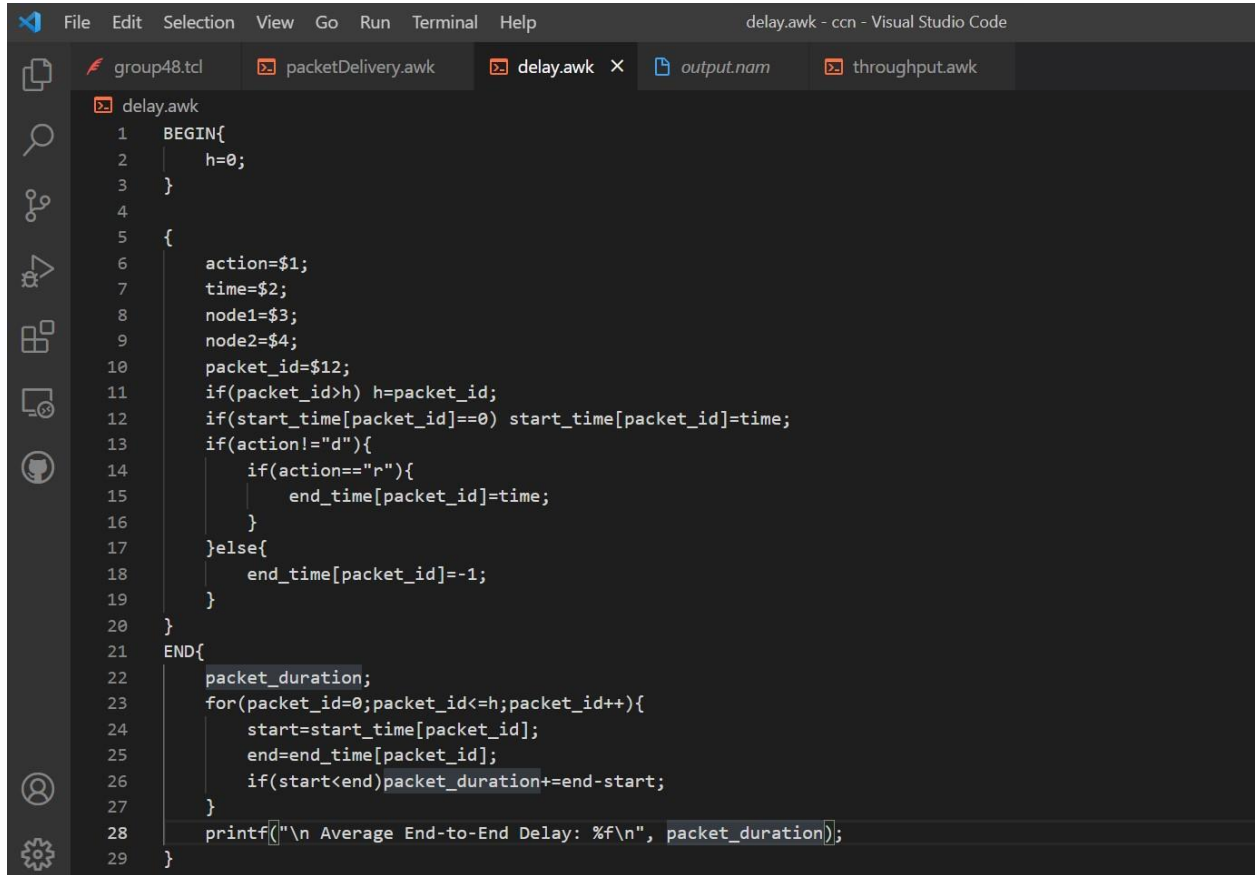
2. packetDelivery.awk



The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "packetDelivery.awk - ccn - Visual Studio Code". The menu bar includes "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". The Explorer sidebar on the left shows a project structure with files "group48.tcl", "packetDelivery.awk", "delay.awk", "output.nam", and "throughput.awk". The "packetDelivery.awk" file is open in the editor, displaying the following AWK script:

```
1 BEGIN{
2     rec=0
3     drop=0
4     total=0
5     ratio=0
6 }
7 {
8     if($1=="r"&&$4==2)
9     {
10         rec++;
11     }
12     if($1=="d"&&$4==2)
13     {
14         drop++;
15     }
16 }
17 END{
18     total=rec+drop
19     ratio=(rec/total)*100
20
21     printf("Total Packet Sent: %d", total)
22     printf("Packet Delivery Ratio: %f", ratio)
23     printf("Total Packet Dropped: %d", drop)
24 }
```

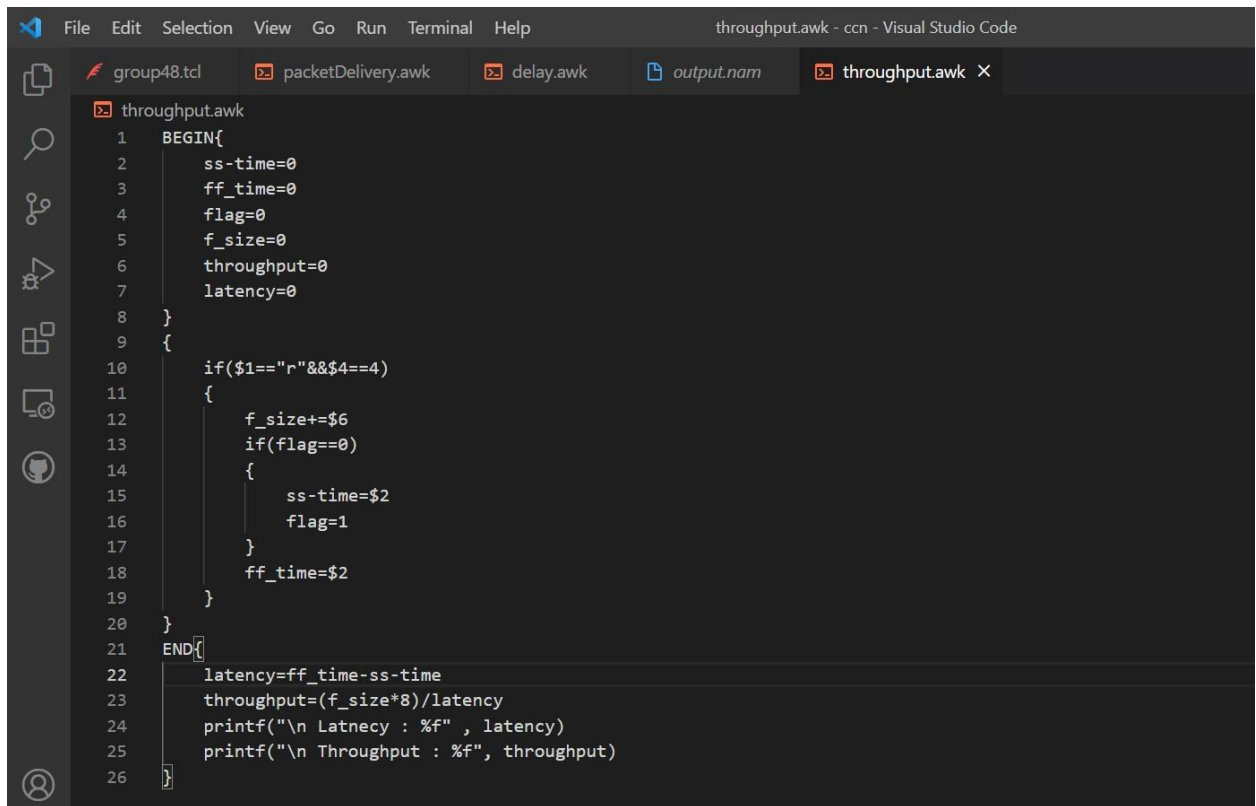
3. delay.awk



The screenshot shows the Visual Studio Code editor with the file `delay.awk` open. The editor has a dark theme and a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Docker, and Settings. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the file is `delay.awk - ccn - Visual Studio Code`. The script content is as follows:

```
1 BEGIN{
2     h=0;
3 }
4
5 {
6     action=$1;
7     time=$2;
8     node1=$3;
9     node2=$4;
10    packet_id=$12;
11    if(packet_id>h) h=packet_id;
12    if(start_time[packet_id]==0) start_time[packet_id]=time;
13    if(action!="d"){
14        if(action=="r"){
15            end_time[packet_id]=time;
16        }
17    }else{
18        end_time[packet_id]=-1;
19    }
20 }
21 END{
22     packet_duration;
23     for(packet_id=0;packet_id<=h;packet_id++){
24         start=start_time[packet_id];
25         end=end_time[packet_id];
26         if(start<end)packet_duration+=end-start;
27     }
28     printf("\n Average End-to-End Delay: %f\n", packet_duration);
29 }
```

4. Throughput.awk

A screenshot of the Visual Studio Code editor interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows 'throughput.awk - ccn - Visual Studio Code'. The file explorer on the left shows a project with files: group48.tcl, packetDelivery.awk, delay.awk, output.nam, and throughput.awk. The throughput.awk file is open in the editor, showing the following code:

```
1 BEGIN{
2     ss-time=0
3     ff_time=0
4     flag=0
5     f_size=0
6     throughput=0
7     latency=0
8 }
9 {
10    if($1=="r"&&$4==4)
11    {
12        f_size+=$6
13        if(flag==0)
14        {
15            ss-time=$2
16            flag=1
17        }
18        ff_time=$2
19    }
20 }
21 END{
22     latency=ff_time-ss-time
23     throughput=(f_size*8)/latency
24     printf("\n Latnecy : %f" , latency)
25     printf("\n Throughput : %f", throughput)
26 }
```

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>
```

