



CCN PROJECT

GROUP-23



Group members :

CV BHANU PRAKASH – S20200020255

P JAGAN RAM JASWANTH – S20200020295

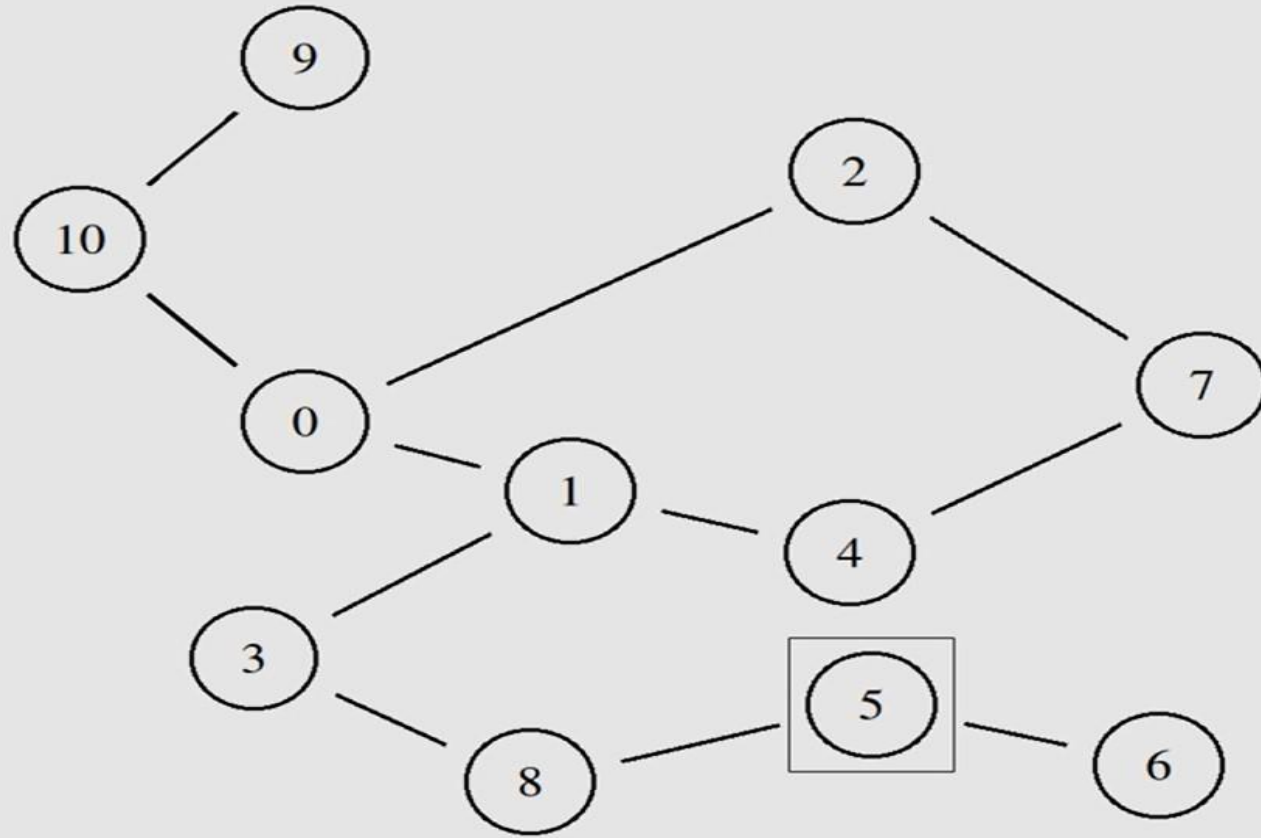
K MANOJ KUMAR – S20200020097



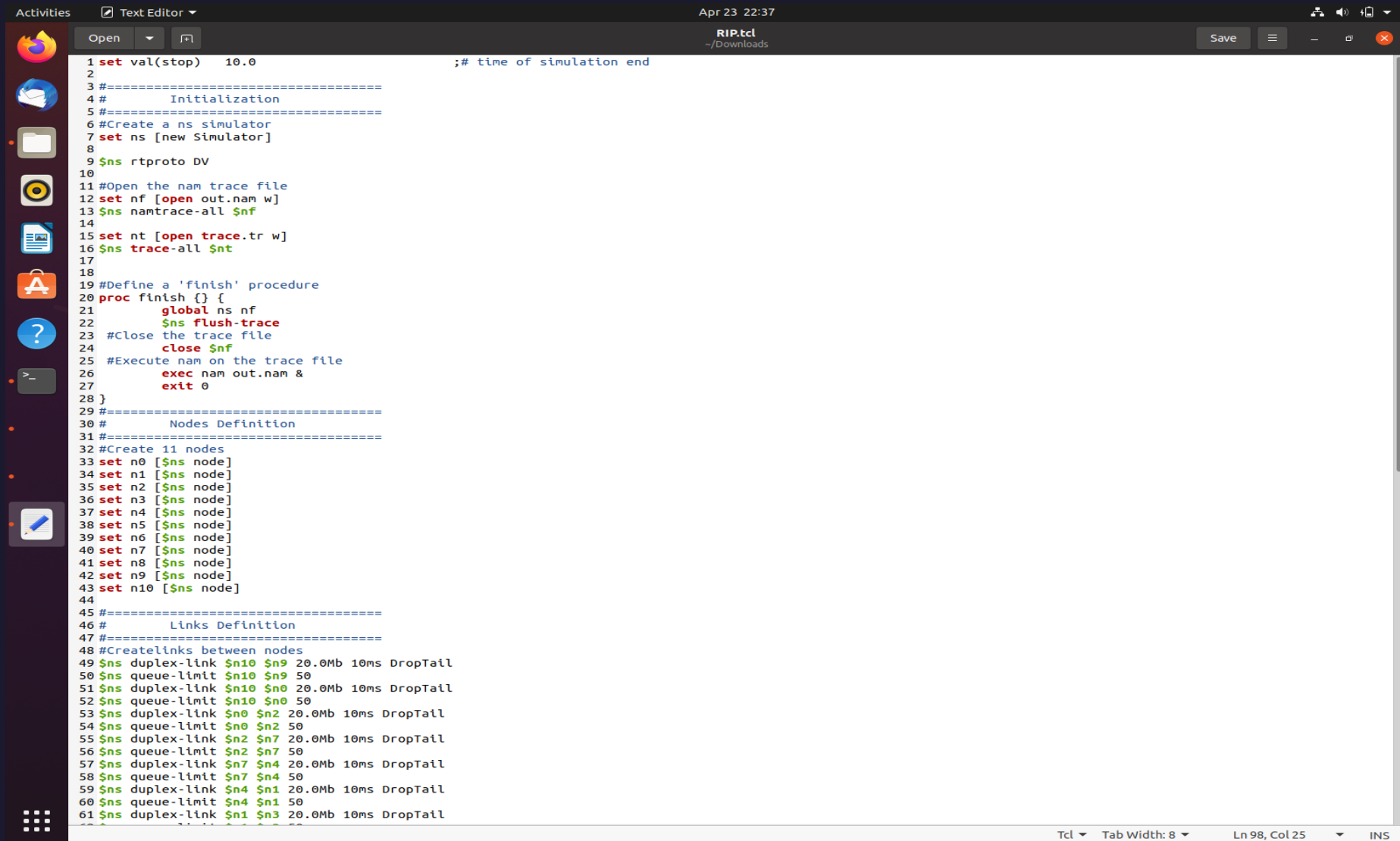
ROUTING INFORMATION PROTOCOL -RIP

- Routing Information Protocol (RIP) is a distance vector protocol that uses hop count as its primary metric.
- RIP sends out periodic routing updates (every 30 seconds)
- RIP sends out the full routing table every periodic update
- RIP uses a form of distance as its metric (in this case, hopcount)
- RIP uses the Bellman-Ford Distance Vector algorithm to determine the best “path” to a particular destination

Topology



RIP .tcl Code

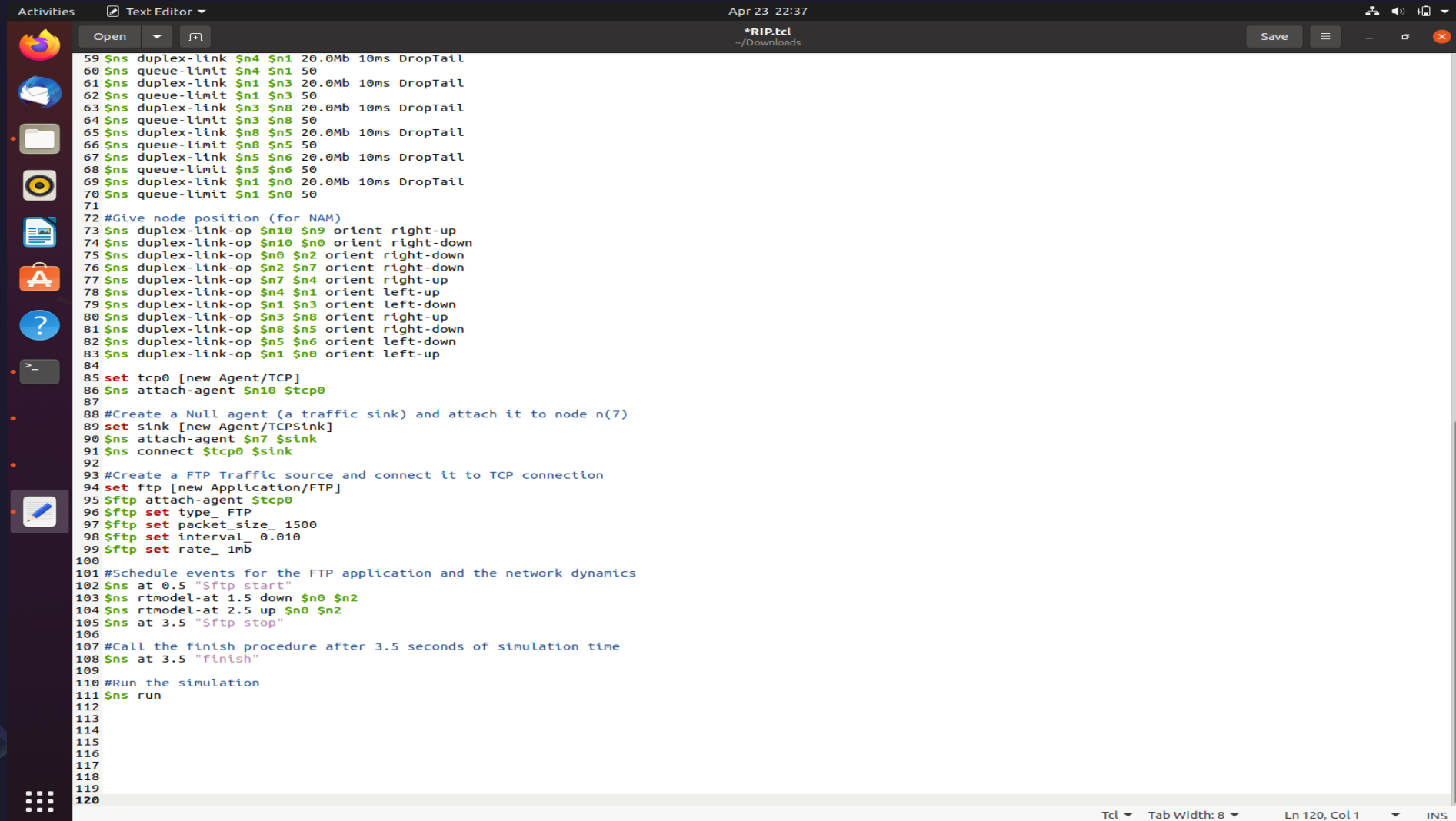


The screenshot shows a Linux desktop with a dark theme. On the left is a vertical dock with icons for Firefox, a mail client, a file manager, a terminal, a help icon, and a text editor. The text editor window is titled 'RIP.tcl' and shows a script for simulating a network topology using ns-3. The script includes comments for simulation time, initialization, node creation, link creation, and a finish procedure. The script is currently at line 98, column 25.

```
1 set val(stop) 10.0 ;# time of simulation end
2
3 #=====
4 # Initialization
5 #=====
6 #Create a ns simulator
7 set ns [new Simulator]
8
9 $ns rtproto DV
10
11 #Open the nam trace file
12 set nf [open out.nam w]
13 $ns namtrace-all $nf
14
15 set nt [open trace.tr w]
16 $ns trace-all $nt
17
18
19 #Define a 'finish' procedure
20 proc finish {} {
21     global ns nf
22     $ns flush-trace
23     #Close the trace file
24     close $nf
25     #Execute nam on the trace file
26     exec nam out.nam &
27     exit 0
28 }
29 #=====
30 # Nodes Definition
31 #=====
32 #Create 11 nodes
33 set n0 [$ns node]
34 set n1 [$ns node]
35 set n2 [$ns node]
36 set n3 [$ns node]
37 set n4 [$ns node]
38 set n5 [$ns node]
39 set n6 [$ns node]
40 set n7 [$ns node]
41 set n8 [$ns node]
42 set n9 [$ns node]
43 set n10 [$ns node]
44
45 #=====
46 # Links Definition
47 #=====
48 #Createlinks between nodes
49 $ns duplex-link $n10 $n9 20.0Mb 10ms DropTail
50 $ns queue-limit $n10 $n9 50
51 $ns duplex-link $n10 $n0 20.0Mb 10ms DropTail
52 $ns queue-limit $n10 $n0 50
53 $ns duplex-link $n0 $n2 20.0Mb 10ms DropTail
54 $ns queue-limit $n0 $n2 50
55 $ns duplex-link $n2 $n7 20.0Mb 10ms DropTail
56 $ns queue-limit $n2 $n7 50
57 $ns duplex-link $n7 $n4 20.0Mb 10ms DropTail
58 $ns queue-limit $n7 $n4 50
59 $ns duplex-link $n4 $n1 20.0Mb 10ms DropTail
60 $ns queue-limit $n4 $n1 50
61 $ns duplex-link $n1 $n3 20.0Mb 10ms DropTail
```

Tcl Tab Width: 8 Ln 98, Col 25 INS

RIP .tcl Code



```
59 $ns duplex-link $n4 $n1 20.0Mb 10ms DropTail
60 $ns queue-limit $n4 $n1 50
61 $ns duplex-link $n1 $n3 20.0Mb 10ms DropTail
62 $ns queue-limit $n1 $n3 50
63 $ns duplex-link $n3 $n8 20.0Mb 10ms DropTail
64 $ns queue-limit $n3 $n8 50
65 $ns duplex-link $n8 $n5 20.0Mb 10ms DropTail
66 $ns queue-limit $n8 $n5 50
67 $ns duplex-link $n5 $n6 20.0Mb 10ms DropTail
68 $ns queue-limit $n5 $n6 50
69 $ns duplex-link $n1 $n0 20.0Mb 10ms DropTail
70 $ns queue-limit $n1 $n0 50
71
72 #Give node position (for NAM)
73 $ns duplex-link-op $n10 $n9 orient right-up
74 $ns duplex-link-op $n10 $n0 orient right-down
75 $ns duplex-link-op $n0 $n2 orient right-down
76 $ns duplex-link-op $n2 $n7 orient right-down
77 $ns duplex-link-op $n7 $n4 orient right-up
78 $ns duplex-link-op $n4 $n1 orient left-up
79 $ns duplex-link-op $n1 $n3 orient left-down
80 $ns duplex-link-op $n3 $n8 orient right-up
81 $ns duplex-link-op $n8 $n5 orient right-down
82 $ns duplex-link-op $n5 $n6 orient left-down
83 $ns duplex-link-op $n1 $n0 orient left-up
84
85 set tcp0 [new Agent/TCP]
86 $ns attach-agent $n10 $tcp0
87
88 #Create a Null agent (a traffic sink) and attach it to node n(7)
89 set sink [new Agent/TCPSink]
90 $ns attach-agent $n7 $sink
91 $ns connect $tcp0 $sink
92
93 #Create a FTP Traffic source and connect it to TCP connection
94 set ftp [new Application/FTP]
95 $ftp attach-agent $tcp0
96 $ftp set type_ FTP
97 $ftp set packet_size_ 1500
98 $ftp set interval_ 0.010
99 $ftp set rate_ 1mb
100
101 #Schedule events for the FTP application and the network dynamics
102 $ns at 0.5 "$ftp start"
103 $ns rtmodel-at 1.5 down $n0 $n2
104 $ns rtmodel-at 2.5 up $n0 $n2
105 $ns at 3.5 "$ftp stop"
106
107 #Call the finish procedure after 3.5 seconds of simulation time
108 $ns at 3.5 "finish"
109
110 #Run the simulation
111 $ns run
112
113
114
115
116
117
118
119
120
```

RIP Delay Code

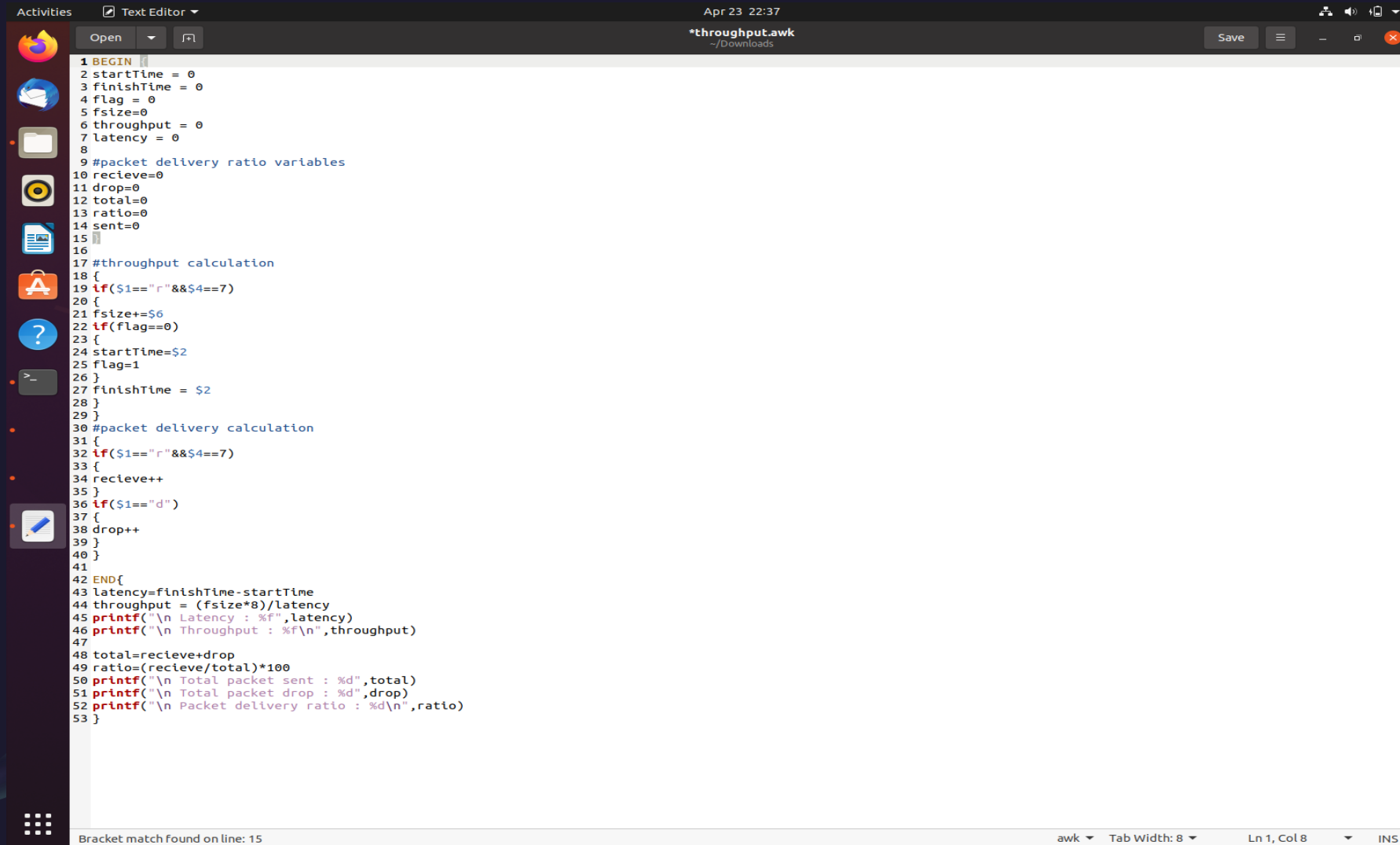


The screenshot shows a Linux desktop environment with a dark theme. On the left is a vertical dock with icons for Firefox, a mail client, a file manager, a terminal, and a help icon. The main window is a text editor titled "Text Editor" with a tab labeled "*RIPDelay.awk". The editor contains the following AWK script:

```
1 BEGIN {
2   highest_packet_id = 0;
3   total_recieved_packet
4 }
5
6 {
7   action = $1;
8   time = $2;
9   node_1 = $3;
10  node_2 = $4;
11  src = $5;
12  flow_id = $8;
13  node_1_address = $9;
14  node_2_address = $10;
15  seq_no = $11;
16  packet_id = $12;
17  if ( packet_id > highest_packet_id ) highest_packet_id = packet_id;
18  if ( start_time[packet_id] == 0 ) start_time[packet_id] = time;
19  if ( action == "d" ) {
20    if ( action == "r" ) {
21      total_recieved_packet++;
22      end_time[packet_id] = time;
23    }
24    }else {
25      end_time[packet_id] = -1;
26    }
27  }
28  END {
29    total_delay;
30    for ( packet_id = 0; packet_id <= highest_packet_id; packet_id++ ) {
31      start = start_time[packet_id];
32      end = end_time[packet_id];
33      packet_duration = end - start;
34      total_delay+=packet_duration;
35    }
36    printf("Average delay = %f\n",total_delay/total_recieved_packet);
37  }
```

The status bar at the bottom of the text editor shows "awk", "Tab Width: 8", "Ln 21, Col 26", and "INS".

Throughput Code



```
1 BEGIN {
2   startTime = 0
3   finishTime = 0
4   flag = 0
5   fsize=0
6   throughput = 0
7   latency = 0
8 }
9 #packet delivery ratio variables
10 recieve=0
11 drop=0
12 total=0
13 ratio=0
14 sent=0
15 }
16
17 #throughput calculation
18 {
19   if($1=="r" && $4==7)
20   {
21     fsize+=56
22     if(flag==0)
23     {
24       startTime=$2
25       flag=1
26     }
27     finishTime = $2
28   }
29 }
30 #packet delivery calculation
31 {
32   if($1=="r" && $4==7)
33   {
34     recieve++
35   }
36   if($1=="d")
37   {
38     drop++
39   }
40 }
41
42 END{
43   latency=finishTime-startTime
44   throughput = (fsize*8)/latency
45   printf("\n Latency : %f",latency)
46   printf("\n Throughput : %f\n",throughput)
47 }
48 total=recieve+drop
49 ratio=(recieve/total)*100
50 printf("\n Total packet sent : %d",total)
51 printf("\n Total packet drop : %d",drop)
52 printf("\n Packet delivery ratio : %d\n",ratio)
53 }
```

Bracket match found on line: 15

awk Tab Width: 8 Ln 1, Col 8 INS

Parameters

```
jaswanth@jaswanth-VirtualBox: ~/Downloads
Nam syntax has changed: v -t 2.5 link-up 2.5 2 0
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2.5 link-up 2.5 0 2
Please use this format in the future.
v -t <time> -e <tcl expression>

Nam syntax has changed: v -t 2.5 link-up 2.5 0 2
Please use this format in the future.
v -t <time> -e <tcl expression>

jaswanth@jaswanth-VirtualBox:~/Downloads$ gawk -f throughput.awk trace.tr

Latency : 3.446205
Throughput : 1569891.518351

Total packet sent : 692
Total packet drop : 20
Packet delivery ratio : 97
jaswanth@jaswanth-VirtualBox:~/Downloads$ gawk -f RIPDelay.awk trace.tr
Average delay = -0.009296
jaswanth@jaswanth-VirtualBox:~/Downloads$
```

Methodology



- NSG2 is used to create topology using that we created RIP.tcl file.
- Now a TCP connection is established between Node10 and Node7.
- RIP.tcl file is executed and simulated.
- After that we observed latency, delay, end to end delay, throughput.
- After the simulation we can observe that the packets are being transmitted from node10 to node 7 using RIP protocol after the link is down now it takes a different path from node 10 to node 7 via node1 and node 4 which is the next shortest path, which is the path with next minimum hops.

Observations:

- RIP Protocol is successfully implemented and performed simulation.
- Latency = 3.4462045
- Throughput = 1569891.518351
- Avg Delay = 0.009296

Challenges

- NSG2 Installation
- Few errors while running

Thank You

