COMPUTER AND COMMUNICATION NETWORKS (CCN)

LAB REPORT

Topic :- Simulation of Routing Information Protocol in NS2

Group Number :- 24

Group Members :-

- Akshat Saxena (S20200020243)
- Shardul Kurdukar (S20200010193)
- Ashwani Kumar (S20200020249)
- Akula Alekhya (S20200020244)
- Mohd Hamza (S20200010132)

Software Used :

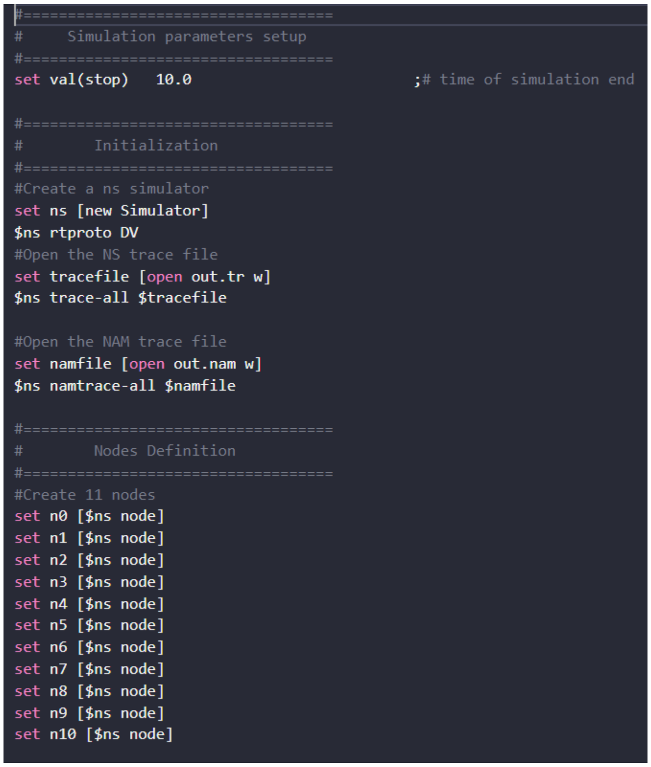
NS2

Theory :

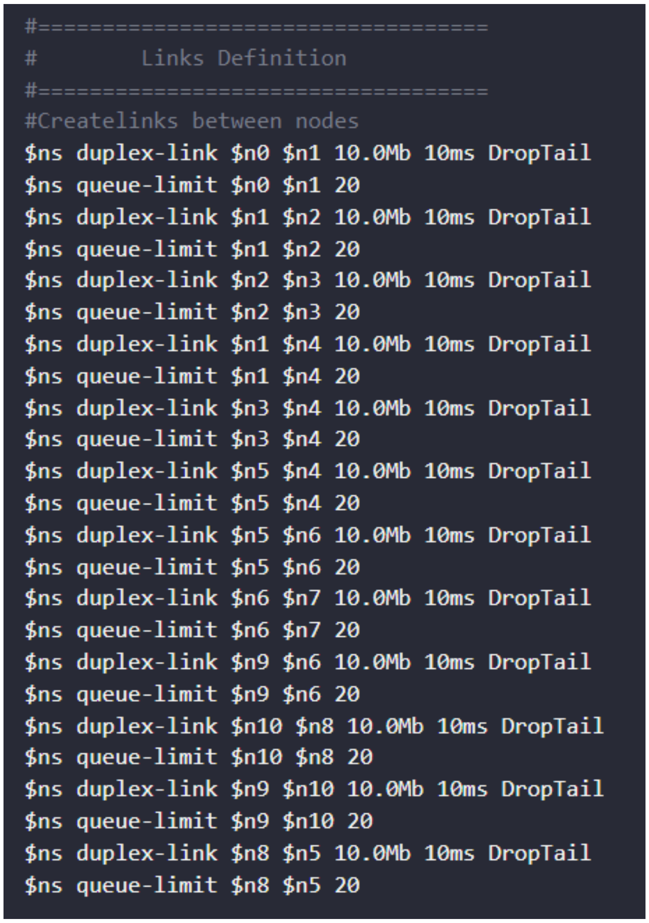
Routing Information Protocol (RIP) is a distance vector protocol that uses hop count as its primary metric. RIP defines how routers should share information when moving traffic among an interconnected group of local area networks (LAN).

It uses a distance vector algorithm to decide which path to put a packet on to get to its destination. Each RIP router maintains a routing table, which is a list of all the destinations the router knows how to reach. RIP is an intra-domain routing protocol used within an autonomous system.

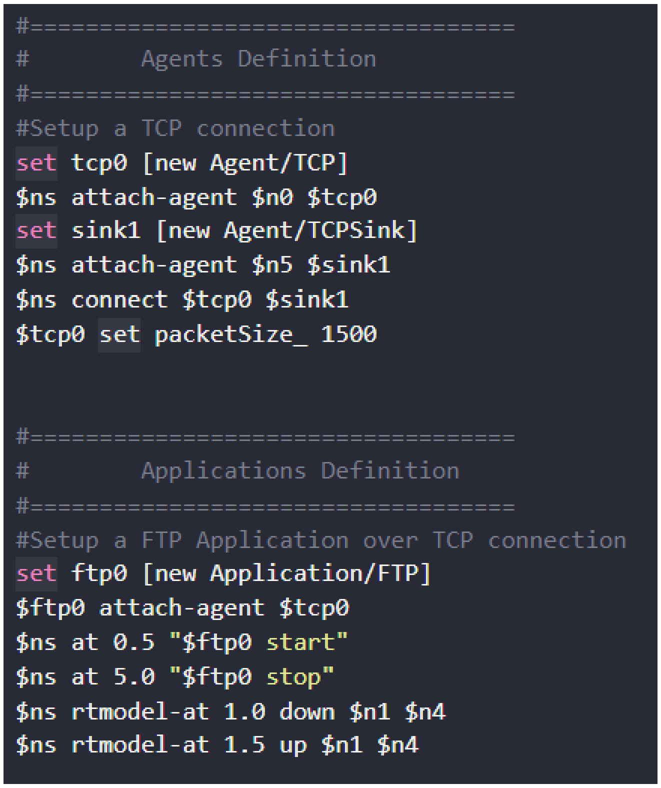
Implementation :-



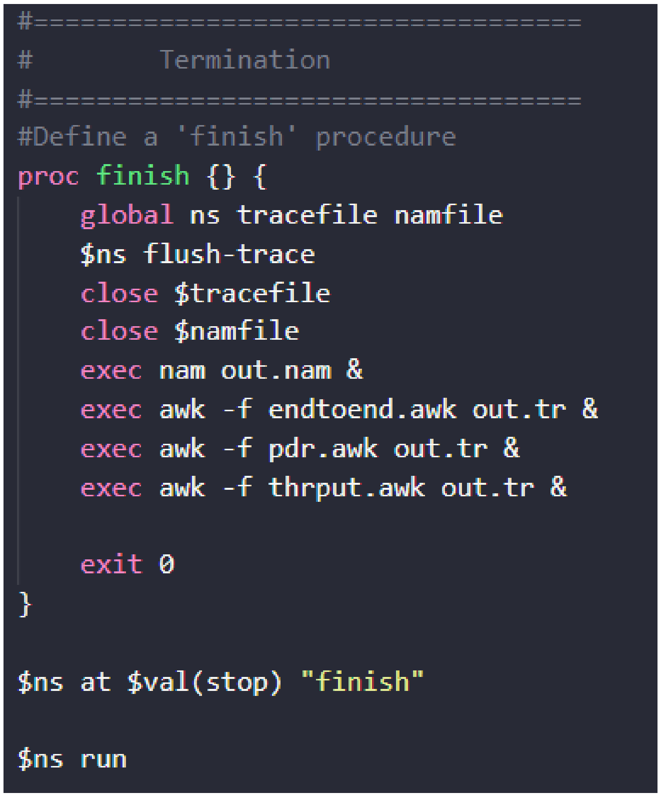
```
#===================================
#     Simulation parameters setup
#===================================
set val(stop)   10.0                         ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]
$ns rtproto DV
#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 11 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]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
set n9 [$ns node]
set n10 [$ns node]
```



```
#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 10.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 20
$ns duplex-link $n1 $n2 10.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 20
$ns duplex-link $n2 $n3 10.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 20
$ns duplex-link $n1 $n4 10.0Mb 10ms DropTail
$ns queue-limit $n1 $n4 20
$ns duplex-link $n3 $n4 10.0Mb 10ms DropTail
$ns queue-limit $n3 $n4 20
$ns duplex-link $n5 $n4 10.0Mb 10ms DropTail
$ns queue-limit $n5 $n4 20
$ns duplex-link $n5 $n6 10.0Mb 10ms DropTail
$ns queue-limit $n5 $n6 20
$ns duplex-link $n6 $n7 10.0Mb 10ms DropTail
$ns queue-limit $n6 $n7 20
$ns duplex-link $n9 $n6 10.0Mb 10ms DropTail
$ns queue-limit $n9 $n6 20
$ns duplex-link $n10 $n8 10.0Mb 10ms DropTail
$ns queue-limit $n10 $n8 20
$ns duplex-link $n9 $n10 10.0Mb 10ms DropTail
$ns queue-limit $n9 $n10 20
$ns duplex-link $n8 $n5 10.0Mb 10ms DropTail
$ns queue-limit $n8 $n5 20
```

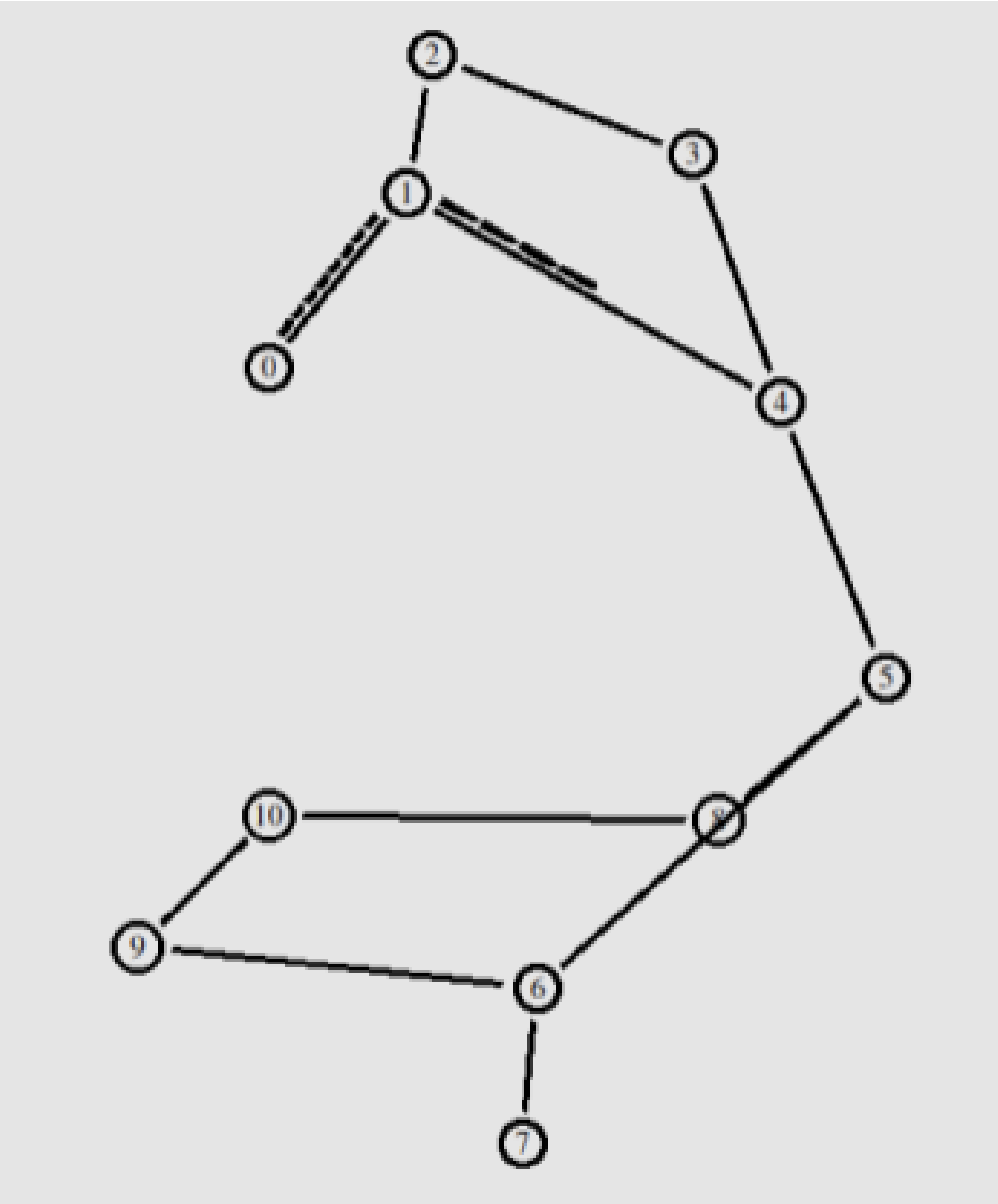


```
#===================================
#        Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n5 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#===================================
#        Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 0.5 "$ftp0 start"
$ns at 5.0 "$ftp0 stop"
$ns rtmodel-at 1.0 down $n1 $n4
$ns rtmodel-at 1.5 up $n1 $n4
```



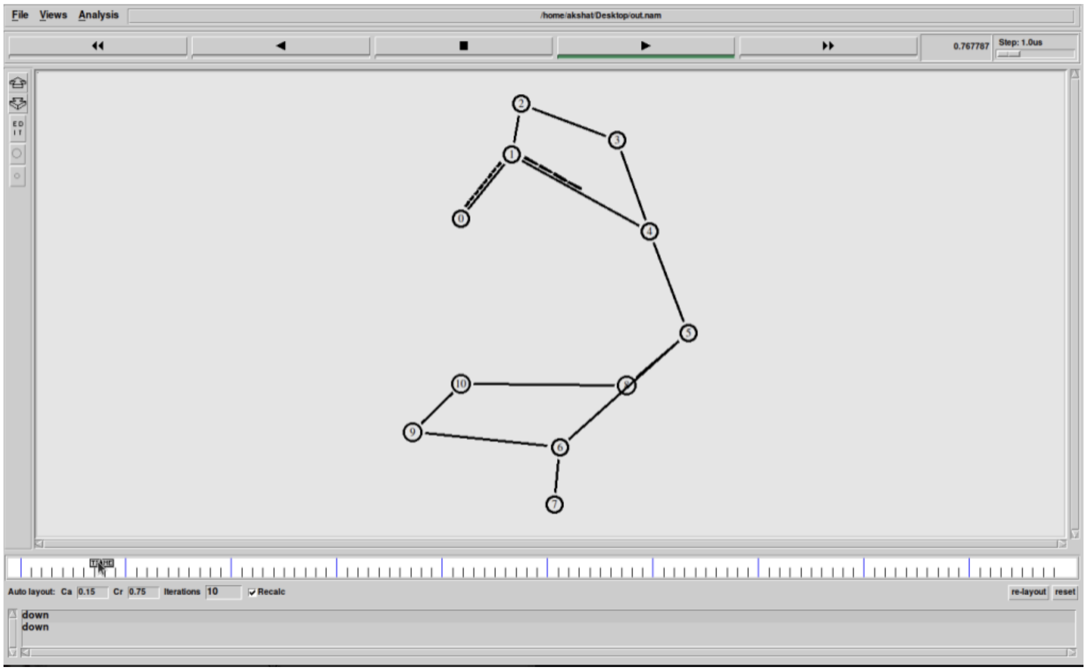
```
#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exec awk -f endtoend.awk out.tr &
    exec awk -f pdr.awk out.tr &
    exec awk -f thrput.awk out.tr &

    exit 0
}

$ns at $val(stop) "finish"

$ns run
```

Topology (Mesh) :-



Source: Node 0
Destination: Node 5

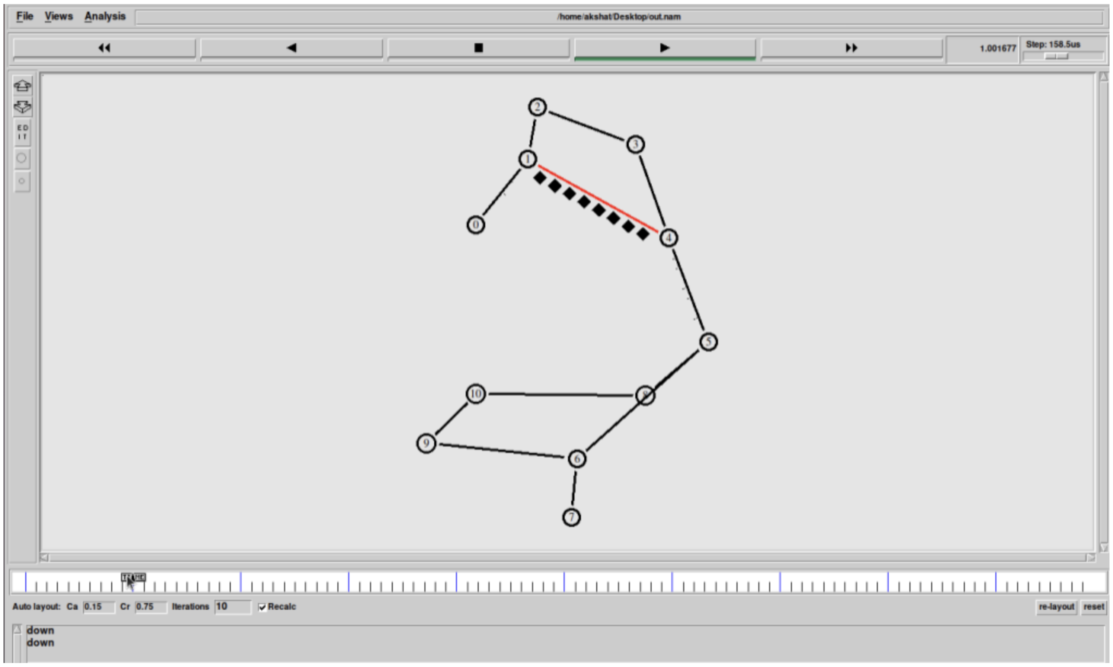
Simulation :-



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

```
down
down
```

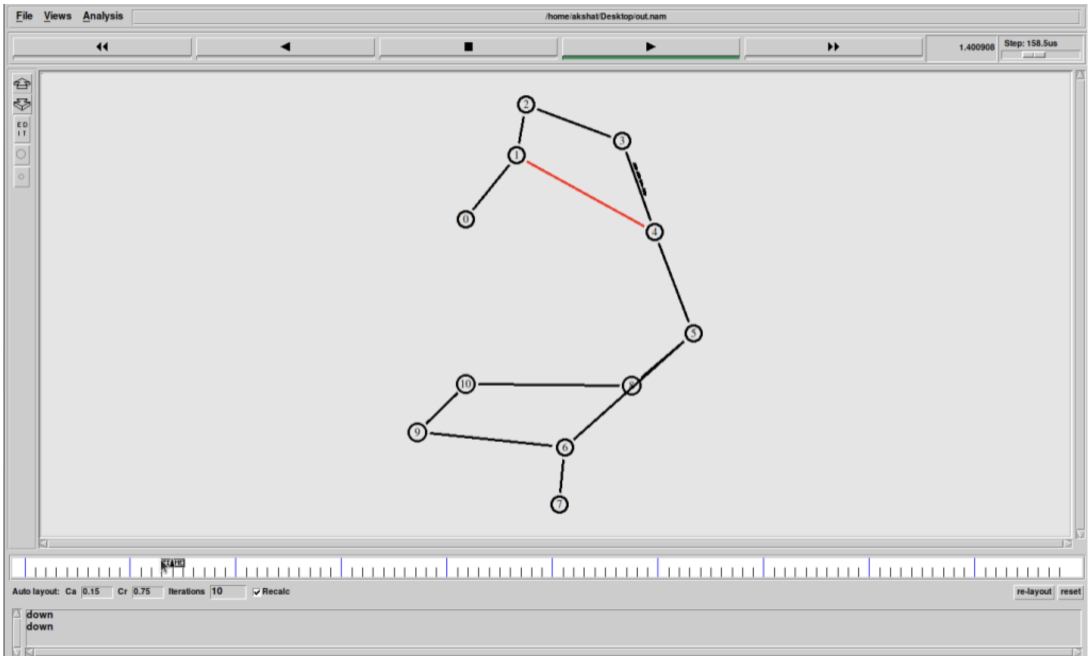
- Traffic being transmitted from node 0 to node 5 through node 1 and node 4



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

```
down
down
```

- We break the link between node1 and node4 by the packets start to drop.



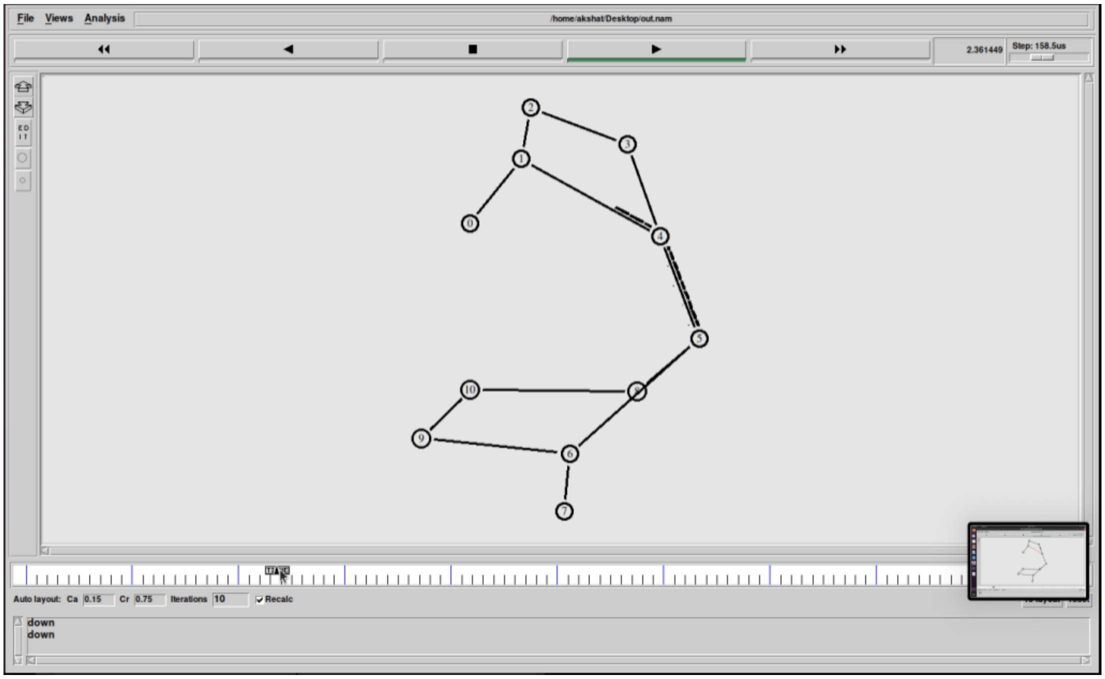
/home/akshat/Desktop/out.nam

1.400908 Step: 158.5us

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

```
down
down
```

- Using the distance vector algorithm that uses hop count as its primary metric, it chooses the next least hop count path to reach node 5.



/home/akshat/Desktop/out.nam

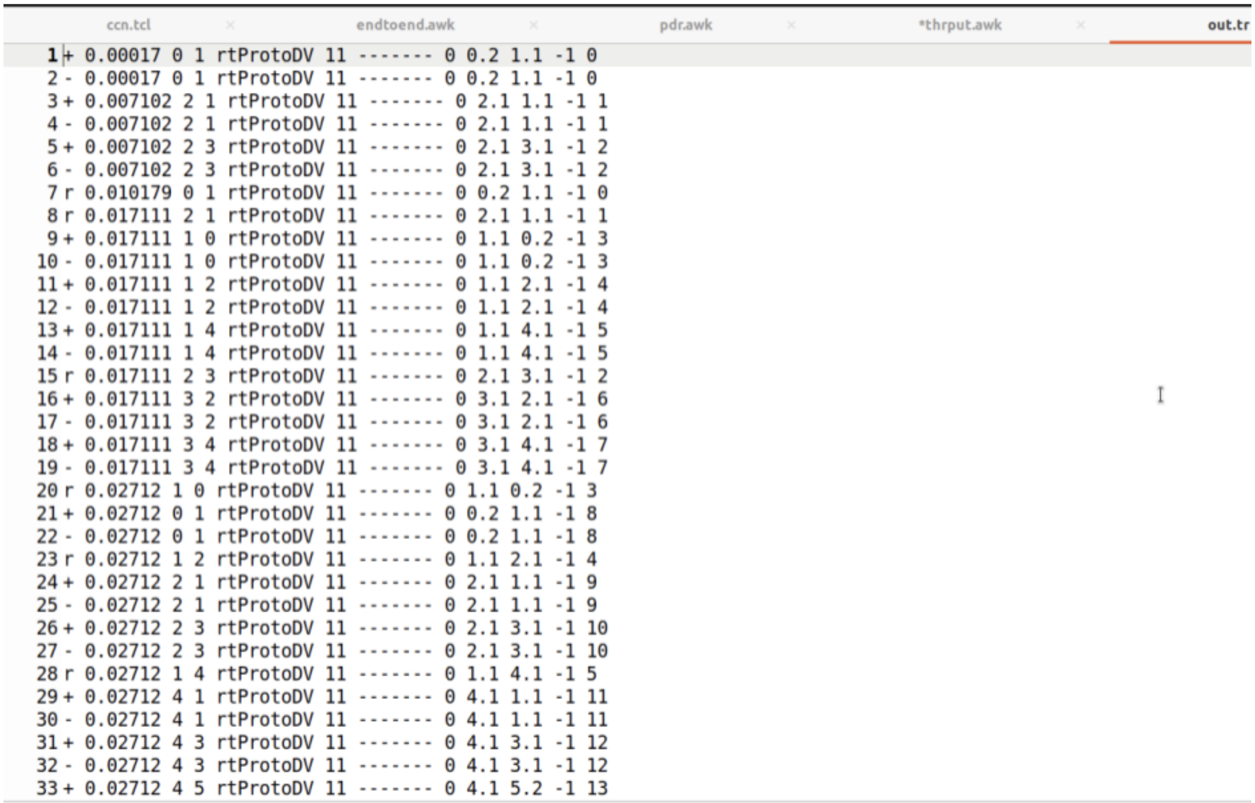
2.361449 Step: 158.5us

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 ☑ Recalc

```
down
down
```

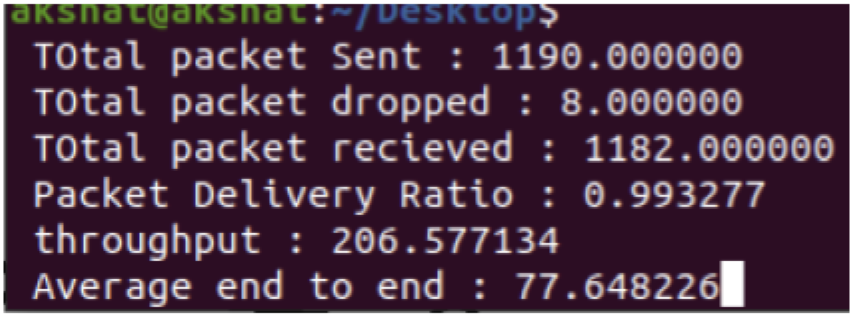
- Link (node 1 - 4) becomes active at 1.5sec mark, and the RIP protocol recalculates the most optimal path for transferring the packets (i.e through node 1 - 4.)

Tracefile :-



```
+ 0.00017 0 1 rtProtoDV 11 ------- 0 0.2 1.1 -1 0
- 0.00017 0 1 rtProtoDV 11 ------- 0 0.2 1.1 -1 0
+ 0.007102 2 1 rtProtoDV 11 ------- 0 2.1 1.1 -1 1
- 0.007102 2 1 rtProtoDV 11 ------- 0 2.1 1.1 -1 1
+ 0.007102 2 3 rtProtoDV 11 ------- 0 2.1 3.1 -1 2
- 0.007102 2 3 rtProtoDV 11 ------- 0 2.1 3.1 -1 2
r 0.010179 0 1 rtProtoDV 11 ------- 0 0.2 1.1 -1 0
r 0.017111 2 1 rtProtoDV 11 ------- 0 2.1 1.1 -1 1
+ 0.017111 1 0 rtProtoDV 11 ------- 0 1.1 0.2 -1 3
- 0.017111 1 0 rtProtoDV 11 ------- 0 1.1 0.2 -1 3
+ 0.017111 1 2 rtProtoDV 11 ------- 0 1.1 2.1 -1 4
- 0.017111 1 2 rtProtoDV 11 ------- 0 1.1 2.1 -1 4
+ 0.017111 1 4 rtProtoDV 11 ------- 0 1.1 4.1 -1 5
- 0.017111 1 4 rtProtoDV 11 ------- 0 1.1 4.1 -1 5
r 0.017111 2 3 rtProtoDV 11 ------- 0 2.1 3.1 -1 2
+ 0.017111 3 2 rtProtoDV 11 ------- 0 3.1 2.1 -1 6
- 0.017111 3 2 rtProtoDV 11 ------- 0 3.1 2.1 -1 6
+ 0.017111 3 4 rtProtoDV 11 ------- 0 3.1 4.1 -1 7
- 0.017111 3 4 rtProtoDV 11 ------- 0 3.1 4.1 -1 7
r 0.02712 1 0 rtProtoDV 11 ------- 0 1.1 0.2 -1 3
+ 0.02712 0 1 rtProtoDV 11 ------- 0 0.2 1.1 -1 8
- 0.02712 0 1 rtProtoDV 11 ------- 0 0.2 1.1 -1 8
r 0.02712 1 2 rtProtoDV 11 ------- 0 1.1 2.1 -1 4
+ 0.02712 2 1 rtProtoDV 11 ------- 0 2.1 1.1 -1 9
- 0.02712 2 1 rtProtoDV 11 ------- 0 2.1 1.1 -1 9
+ 0.02712 2 3 rtProtoDV 11 ------- 0 2.1 3.1 -1 10
- 0.02712 2 3 rtProtoDV 11 ------- 0 2.1 3.1 -1 10
r 0.02712 1 4 rtProtoDV 11 ------- 0 1.1 4.1 -1 5
+ 0.02712 4 1 rtProtoDV 11 ------- 0 4.1 1.1 -1 11
- 0.02712 4 1 rtProtoDV 11 ------- 0 4.1 1.1 -1 11
+ 0.02712 4 3 rtProtoDV 11 ------- 0 4.1 3.1 -1 12
- 0.02712 4 3 rtProtoDV 11 ------- 0 4.1 3.1 -1 12
+ 0.02712 4 5 rtProtoDV 11 ------- 0 4.1 5.2 -1 13
```

Output :-



```
akshat@akshat:~/Desktop$
 TOtal packet Sent : 1190.000000
 TOtal packet dropped : 8.000000
 TOtal packet recieved : 1182.000000
 Packet Delivery Ratio : 0.993277
 throughput : 206.577134
 Average end to end : 77.648226
```

Observation :-

- From this simulation it is observed that the path taken by packets from source node 0 to destination node 5 is optimal as per RIP routing protocol.
- The RIP works even after any link between nodes drops and it recovers back to the old route when the link starts working again.