THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Name: Manasi Bhagwat
Roll No: 50

Date: 01/03/2021

Lab Assignment No:5

Aim: Simulation of Network with specific routing protocols (Distance Vector, Link State).

Lab Outcome Attained:

-To understand the network simulator environment and visualize a network topology and observe its performance.

-To observe and study the traffic flow and the contents of protocol frames.

Theory:

Routing Protocols

Network routing protocols are the set of defined rules used by the routers to communicate between source and destination. They help specify way routers communicate with each other. They do not move the information to the source to a destination, but only update the routing table that contains the information. It allows the network to select routes between any two nodes on a computer network.

There are mainly two types of Network Routing Protocols -

Static - Static routing protocols are used when an administrator manually assigns the path from source to the destination network. It offers more security to the network.

Dynamic - It helps routers to add information to their routing tables from connected routers automatically. These types of protocols also send out topology updates whenever the network changes' topological structure.

Distance Vector Routing Protocol

Distance Vector Protocols advertise their routing table to every directly connected neighbour at specific time intervals using lots of bandwidths and slow converge.

In the Distance Vector routing protocol, when a route becomes unavailable, all routing tables need to be updated with new information.

Here updates of the network are exchanged periodically, and it is always broadcast. This protocol always trusts route on routing information received from neighbour routers.

As the routing information are exchanged periodically, unnecessary traffic is generated, which consumes available bandwidth.

Link State Routing Protocol

Link State Protocols take a unique approach to search the best routing path. In this protocol, the route is calculated based on the speed of the path to the destination and the cost of resources.

Routing protocol tables:

Link state routing protocol maintains below given three tables:

Neighbour table - This table contains information about the neighbours of the router only. For example, adjacency has been formed.

Topology table - This table stores information about the whole topology. For example, it contains both the best and backup routes to a particular advertised network.

Routing table - This type of table contains all the best routes to the advertised network.

This protocol maintains separate tables for both the best route and the backup routes, so it has more knowledge of the inter-network than any other distance vector routing protocol.
It uses the concept of triggered updates, so it does not consume any unnecessary bandwidth. Partial updates will be triggered when there is a topology change, so it does not need to update where the whole routing table is exchanged.

For Loop: for {start} {test} {next} {body}
For is a looping command. The start, next, and body arguments must be Tcl command strings, and test is an expression string. The for command first invokes the Tcl interpreter to execute start. It repeatedly evaluates test as an expression; if the result is non-zero it invokes the Tcl interpreter on body, then on next, then repeats the loop. The command terminates when test evaluates to 0.

- for { set i 0 } { $i < 12} { incr i 1 } {set n($i) [$ns node]} :

Here, For loop is used to create nodes. 12 nodes ranging from 0 to 11 are created.

$ns rtproto name_of_routing_protocol (DV for distance vector and LS for Link State)

This command sets a Routing Protocol. There are four in ns: Static, Session, Dynamic and Manual.

- ns rtproto DV

This command sets the routing protocol as distance vector.

$ns rtmodel-at [time] [down/up] [node_id] :

This command simulates Node Failure or Restoration at given time.

$ns rtmodel-at [time] [down/up] [node_id_from node_id_to] :

This command simulates Link Failure or Restoration at given time.

- $ns rtmodel-at 15.0 down $n(7) $n(6) :

This command simulates failure of the link between node 7 and node 6 at time 15

- $ns rtmodel-at 30.0 up $n(11) $n(5)

This command simulates restoration of the link between node 11 and node 5 at time 30.

Program –

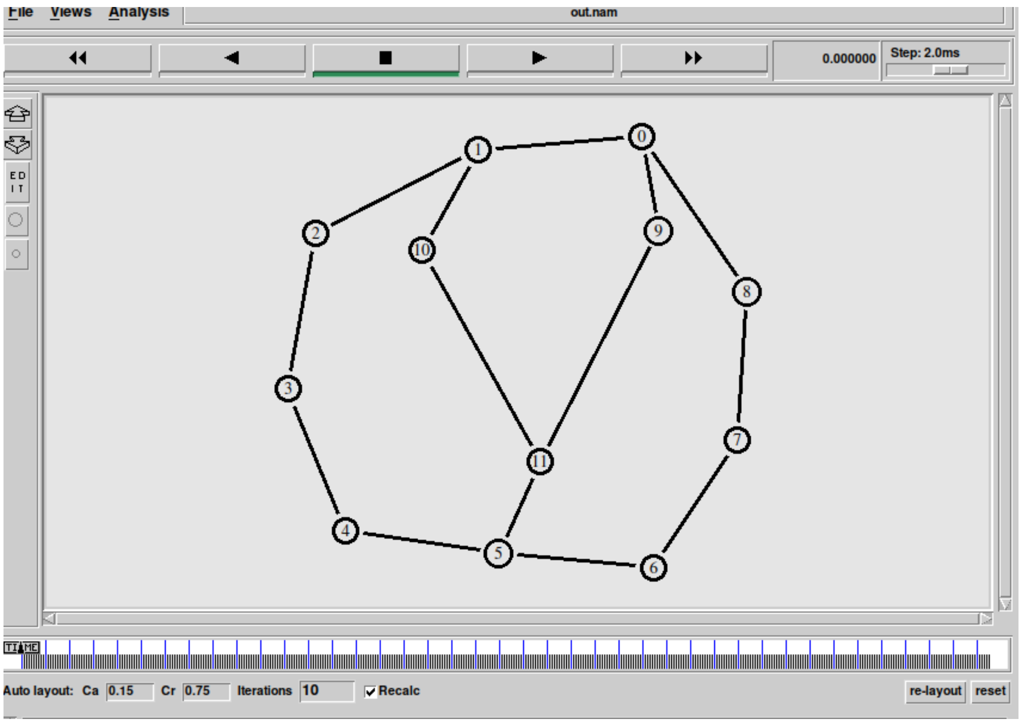
```
set ns [new Simulator]
set nr [open out.tr w]
$ns trace-all $nr
set nf [open out.nam w]

$ns namtrace-all $nf
        proc finish { } {
        global ns nr nf
        $ns flush-trace
        close $nf
        close $nr
        exec nam thro.nam &
          exit 0
        }

for { set i 0 } { $i < 12} { incr i 1 } {
set n($i) [$ns node]}

for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail
```

```
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set  interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0

set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0

$ns rtproto DV

$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)

$udp0 set fid_ 1
```

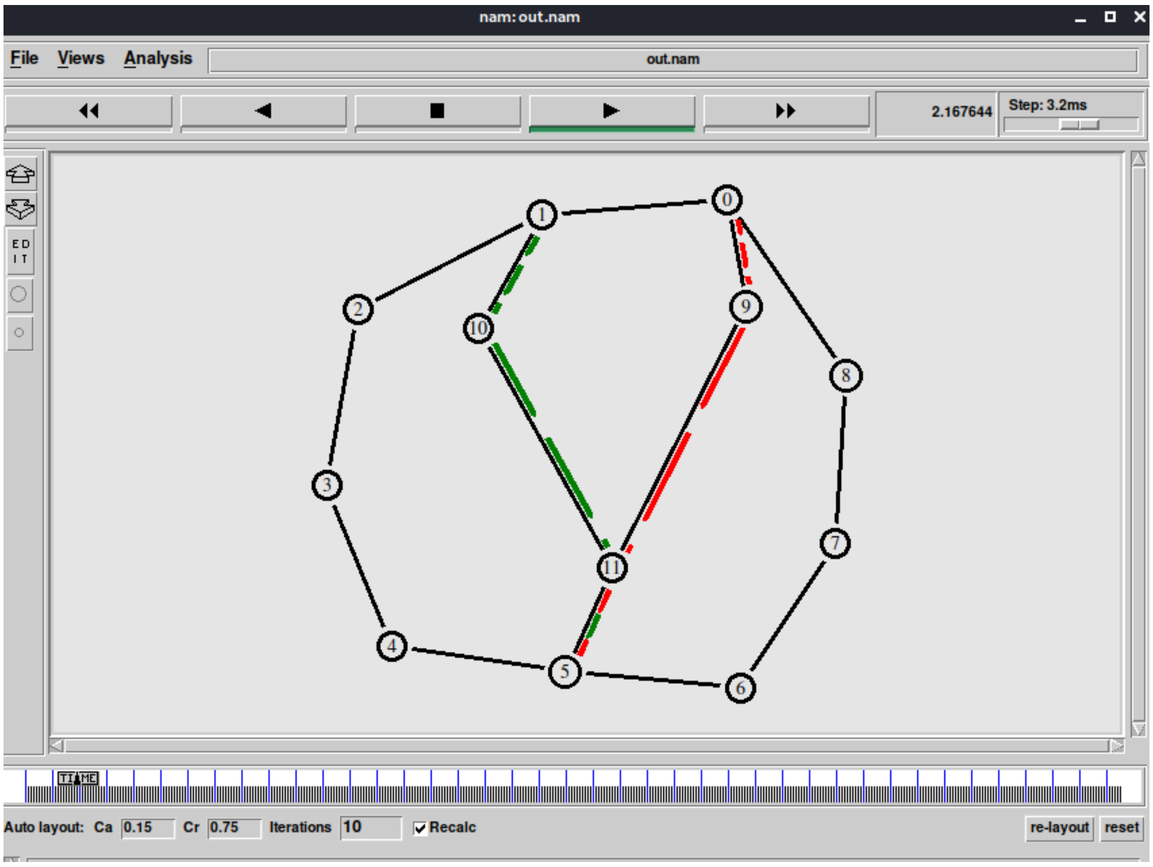
```
$udp1 set fid_ 2

$ns color 1 Red
$ns color 2 Green

$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"

$ns at 45 "finish"
$ns run
```

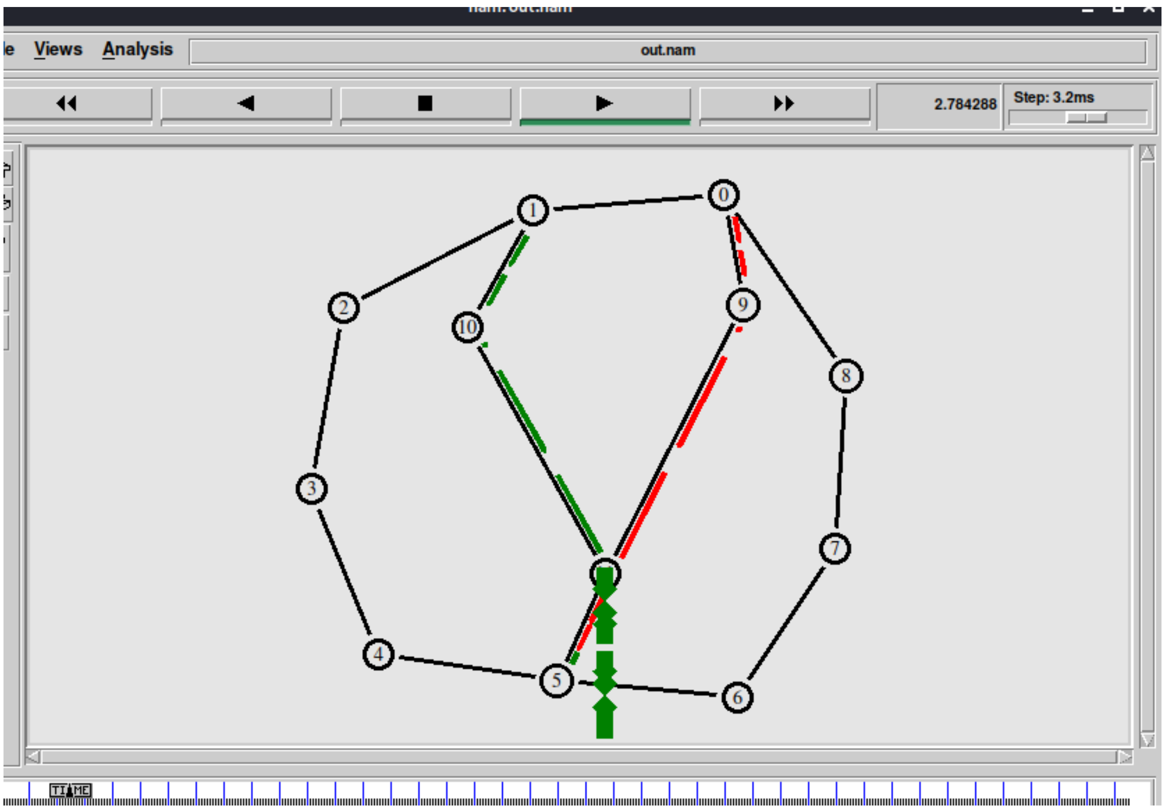
Screenshots:

Output –

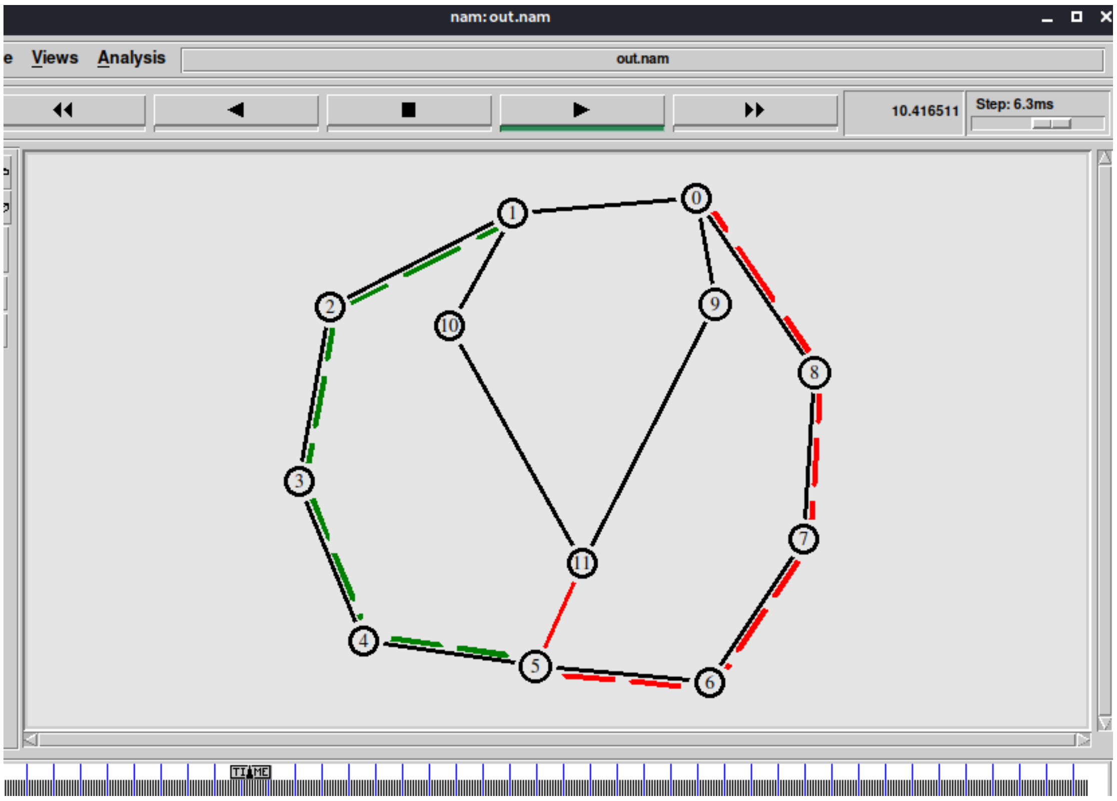


The packets are sent from Node1 to Node5 (Green) and Node0 to Node5 (Red) -

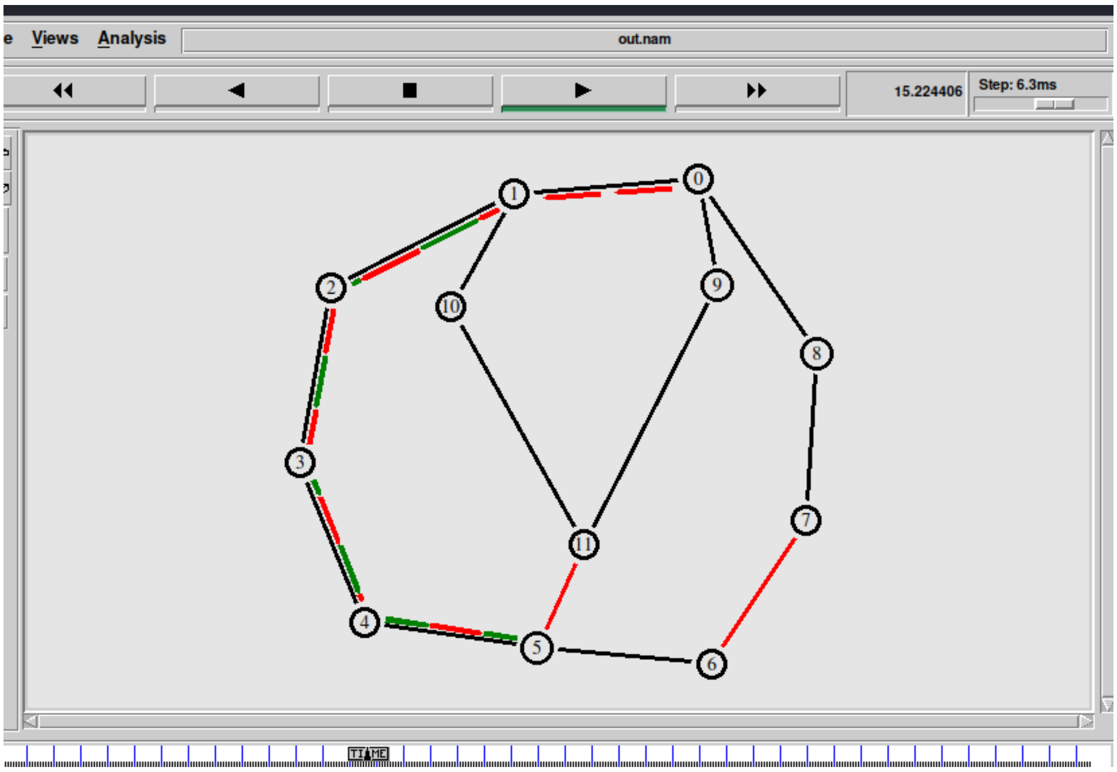




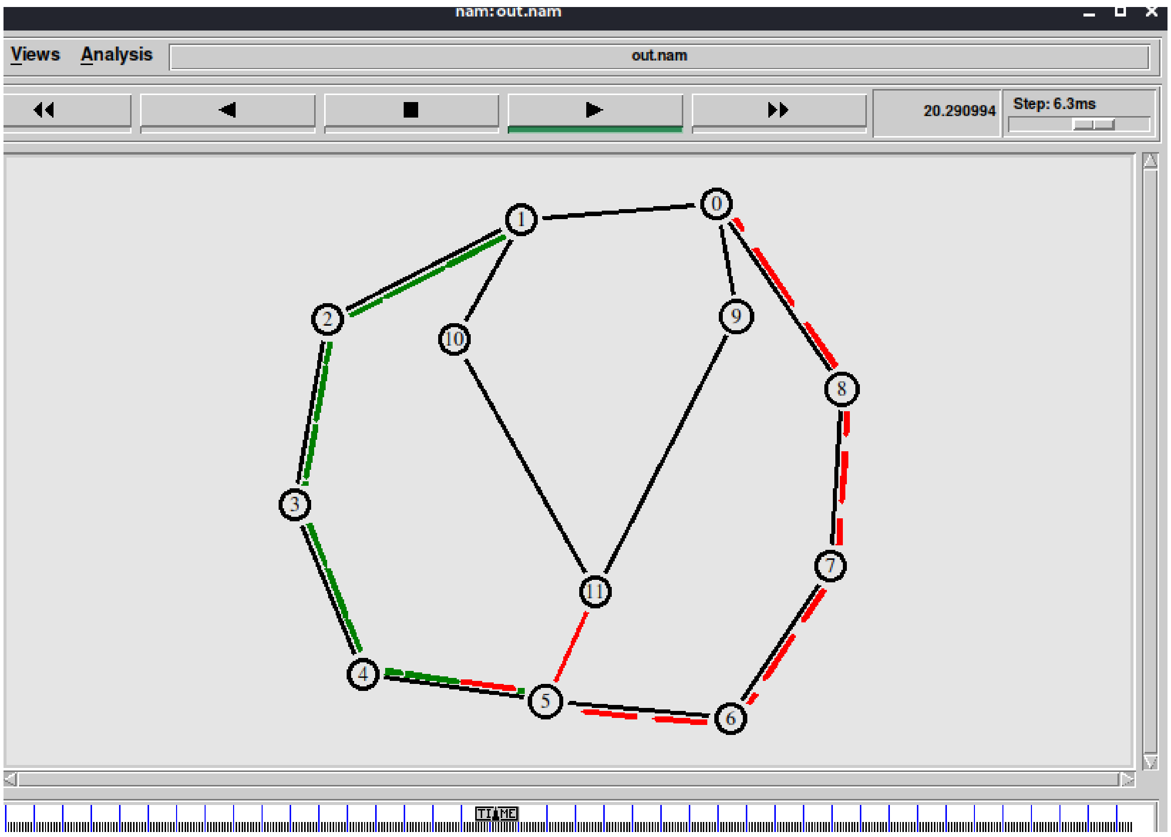
At time 10 , Link between Node 11 and Node 5 goes down -



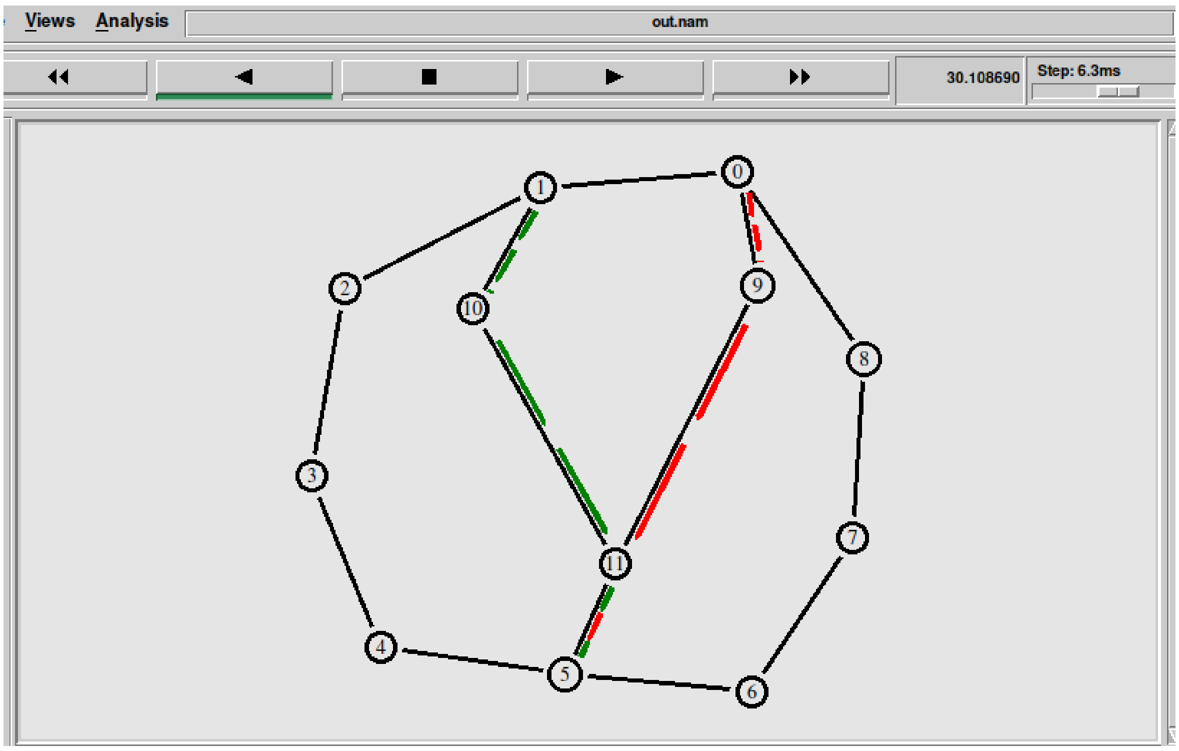
At time 10 , Link between Node 7 and Node 6 goes down –



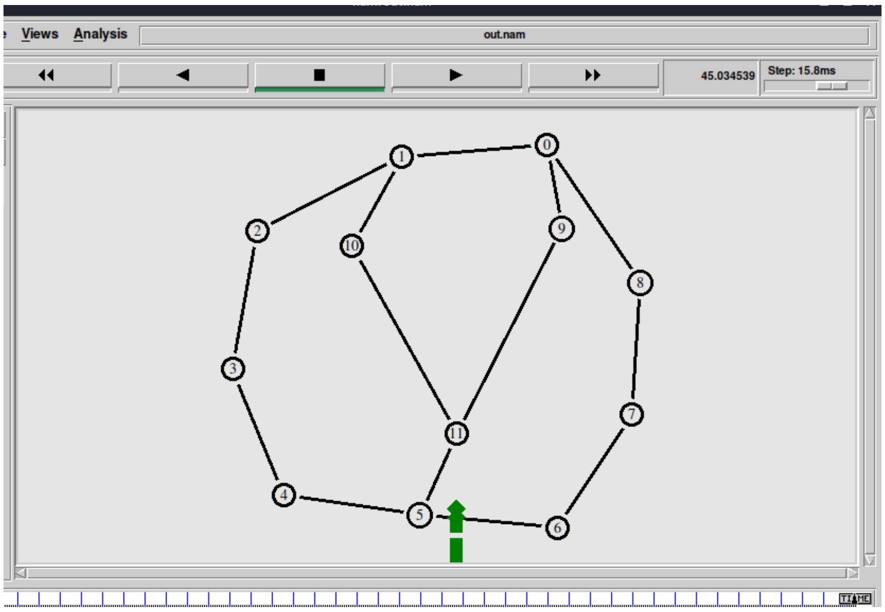
At time 20 , Link between Node 7 and Node 6 is restored -



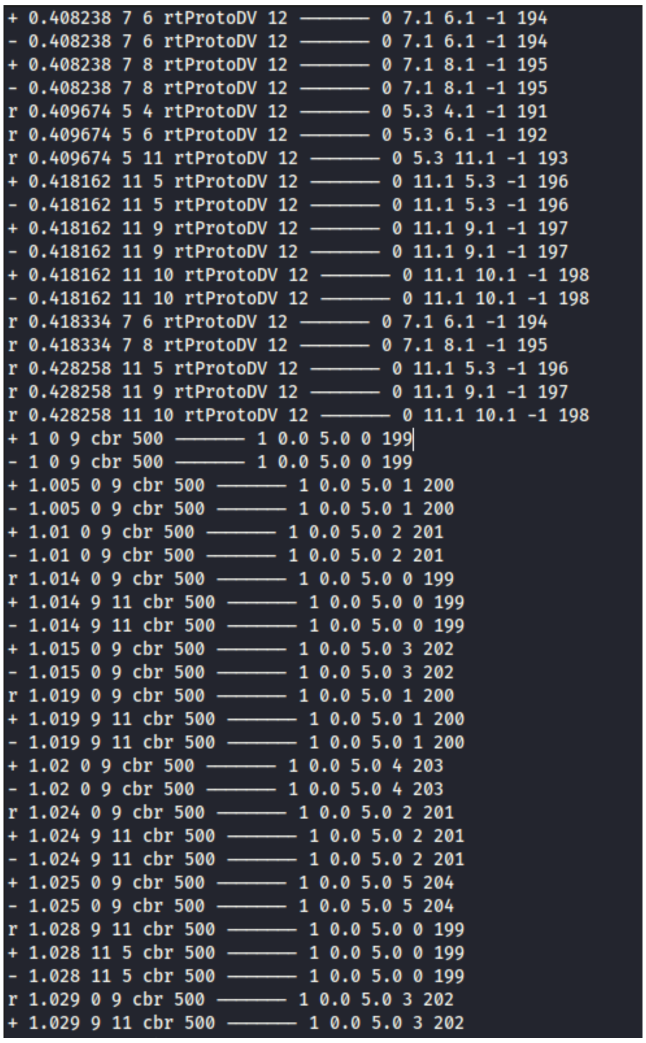
At time 30 , Link between Node 11 and Node 5 is restored -



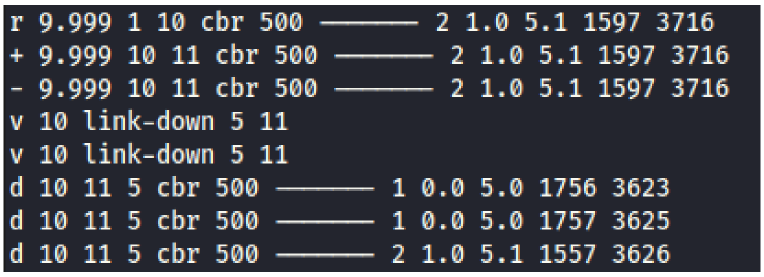
The simulation stops at 45 –



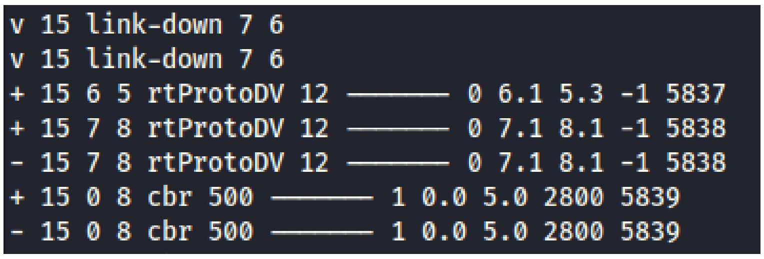
Out.tr file –



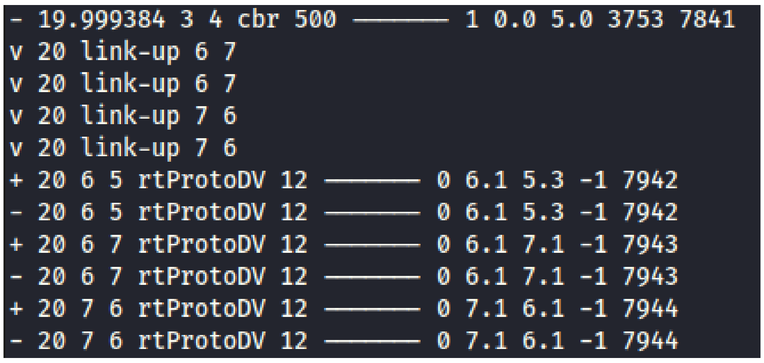
```
+ 0.408238 7 6 rtProtoDV 12 ------- 0 7.1 6.1 -1 194
- 0.408238 7 6 rtProtoDV 12 ------- 0 7.1 6.1 -1 194
+ 0.408238 7 8 rtProtoDV 12 ------- 0 7.1 8.1 -1 195
- 0.408238 7 8 rtProtoDV 12 ------- 0 7.1 8.1 -1 195
r 0.409674 5 4 rtProtoDV 12 ------- 0 5.3 4.1 -1 191
r 0.409674 5 6 rtProtoDV 12 ------- 0 5.3 6.1 -1 192
r 0.409674 5 11 rtProtoDV 12 ------- 0 5.3 11.1 -1 193
+ 0.418162 11 5 rtProtoDV 12 ------- 0 11.1 5.3 -1 196
- 0.418162 11 5 rtProtoDV 12 ------- 0 11.1 5.3 -1 196
+ 0.418162 11 9 rtProtoDV 12 ------- 0 11.1 9.1 -1 197
- 0.418162 11 9 rtProtoDV 12 ------- 0 11.1 9.1 -1 197
+ 0.418162 11 10 rtProtoDV 12 ------- 0 11.1 10.1 -1 198
- 0.418162 11 10 rtProtoDV 12 ------- 0 11.1 10.1 -1 198
r 0.418334 7 6 rtProtoDV 12 ------- 0 7.1 6.1 -1 194
r 0.418334 7 8 rtProtoDV 12 ------- 0 7.1 8.1 -1 195
r 0.428258 11 5 rtProtoDV 12 ------- 0 11.1 5.3 -1 196
r 0.428258 11 9 rtProtoDV 12 ------- 0 11.1 9.1 -1 197
r 0.428258 11 10 rtProtoDV 12 ------- 0 11.1 10.1 -1 198
+ 1 0 9 cbr 500 ------- 1 0.0 5.0 0 199
- 1 0 9 cbr 500 ------- 1 0.0 5.0 0 199
+ 1.005 0 9 cbr 500 ------- 1 0.0 5.0 1 200
- 1.005 0 9 cbr 500 ------- 1 0.0 5.0 1 200
+ 1.01 0 9 cbr 500 ------- 1 0.0 5.0 2 201
- 1.01 0 9 cbr 500 ------- 1 0.0 5.0 2 201
r 1.014 0 9 cbr 500 ------- 1 0.0 5.0 0 199
+ 1.014 9 11 cbr 500 ------- 1 0.0 5.0 0 199
- 1.014 9 11 cbr 500 ------- 1 0.0 5.0 0 199
+ 1.015 0 9 cbr 500 ------- 1 0.0 5.0 3 202
- 1.015 0 9 cbr 500 ------- 1 0.0 5.0 3 202
r 1.019 0 9 cbr 500 ------- 1 0.0 5.0 1 200
+ 1.019 9 11 cbr 500 ------- 1 0.0 5.0 1 200
- 1.019 9 11 cbr 500 ------- 1 0.0 5.0 1 200
+ 1.02 0 9 cbr 500 ------- 1 0.0 5.0 4 203
- 1.02 0 9 cbr 500 ------- 1 0.0 5.0 4 203
r 1.024 0 9 cbr 500 ------- 1 0.0 5.0 2 201
+ 1.024 9 11 cbr 500 ------- 1 0.0 5.0 2 201
- 1.024 9 11 cbr 500 ------- 1 0.0 5.0 2 201
+ 1.025 0 9 cbr 500 ------- 1 0.0 5.0 5 204
- 1.025 0 9 cbr 500 ------- 1 0.0 5.0 5 204
r 1.028 9 11 cbr 500 ------- 1 0.0 5.0 0 199
+ 1.028 11 5 cbr 500 ------- 1 0.0 5.0 0 199
- 1.028 11 5 cbr 500 ------- 1 0.0 5.0 0 199
r 1.029 0 9 cbr 500 ------- 1 0.0 5.0 3 202
+ 1.029 9 11 cbr 500 ------- 1 0.0 5.0 3 202
```



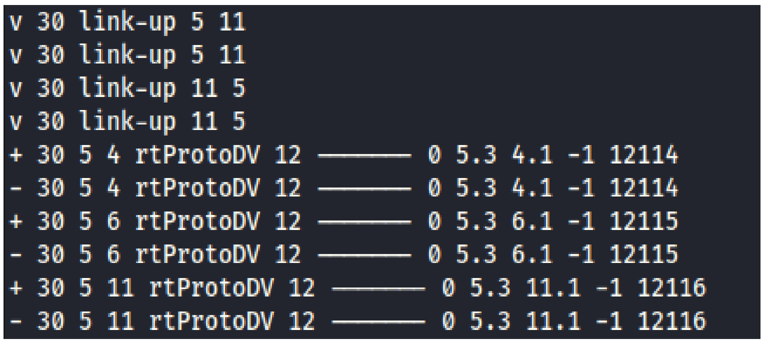
```
r 9.999 1 10 cbr 500 ------- 2 1.0 5.1 1597 3716
+ 9.999 10 11 cbr 500 ------- 2 1.0 5.1 1597 3716
- 9.999 10 11 cbr 500 ------- 2 1.0 5.1 1597 3716
v 10 link-down 5 11
v 10 link-down 5 11
d 10 11 5 cbr 500 ------- 1 0.0 5.0 1756 3623
d 10 11 5 cbr 500 ------- 1 0.0 5.0 1757 3625
d 10 11 5 cbr 500 ------- 2 1.0 5.1 1557 3626
```



```
v 15 link-down 7 6
v 15 link-down 7 6
+ 15 6 5 rtProtoDV 12 ------- 0 6.1 5.3 -1 5837
+ 15 7 8 rtProtoDV 12 ------- 0 7.1 8.1 -1 5838
- 15 7 8 rtProtoDV 12 ------- 0 7.1 8.1 -1 5838
+ 15 0 8 cbr 500 ------- 1 0.0 5.0 2800 5839
- 15 0 8 cbr 500 ------- 1 0.0 5.0 2800 5839
```



```
- 19.999384 3 4 cbr 500 ------- 1 0.0 5.0 3753 7841
v 20 link-up 6 7
v 20 link-up 6 7
v 20 link-up 7 6
v 20 link-up 7 6
+ 20 6 5 rtProtoDV 12 ------- 0 6.1 5.3 -1 7942
- 20 6 5 rtProtoDV 12 ------- 0 6.1 5.3 -1 7942
+ 20 6 7 rtProtoDV 12 ------- 0 6.1 7.1 -1 7943
- 20 6 7 rtProtoDV 12 ------- 0 6.1 7.1 -1 7943
+ 20 7 6 rtProtoDV 12 ------- 0 7.1 6.1 -1 7944
- 20 7 6 rtProtoDV 12 ------- 0 7.1 6.1 -1 7944
```



```
v 30 link-up 5 11
v 30 link-up 5 11
v 30 link-up 11 5
v 30 link-up 11 5
+ 30 5 4 rtProtoDV 12 ------- 0 5.3 4.1 -1 12114
- 30 5 4 rtProtoDV 12 ------- 0 5.3 4.1 -1 12114
+ 30 5 6 rtProtoDV 12 ------- 0 5.3 6.1 -1 12115
- 30 5 6 rtProtoDV 12 ------- 0 5.3 6.1 -1 12115
+ 30 5 11 rtProtoDV 12 ------- 0 5.3 11.1 -1 12116
- 30 5 11 rtProtoDV 12 ------- 0 5.3 11.1 -1 12116
```

Conclusion

We have successfully implemented network simulations with specific routing protocols (Distance Vector, Link State).