SSN College of Engineering, Kalavakkam

Department of Computer Science and Engineering

UCS1511 NETWORKS LAB

Exercise 9: SIMULATION OF ROUTING PROTOCOLS

Name: Rahul Ram M

Roll No.: 185001121

Date: 05/11/2020

LEARNING OBJECTIVES:

To write tcl script to simulate the routing protocols in wired networks.

a. Distance Vector Routing

ALGORITHM:

1. Create a new object ns with the class Simulator.
2. Open the nam trace file.
3. The namtrace is set for enabling animation to simulate the environment.
4. The color field here is used to discriminate the different data packets travelling across the nodes.
5. Declare the nodes namely n0, n1, ... to be used in the simulation.
6. The duplex links between the nodes is set appropriately.
7. Set the orientation of the nodes appropriately for proper representation.
8. The queue limit is set to determine the capacity of the queue for any communication.
9. A UDP connection is set up between the node n0 and n5 and n1 and n5.
10. The CBR (constant bit rate) here facilitates this UDP connection.
11. Protocol to be used is specified by '$ns rtproto DV' where DV – Distance Vector Protocol.
12. Making the links 11-5 and 7-6 down for 1 second starting from 2.0 to 3.0 in the course of simulation for a total of 5.0 seconds.
13. The run command is used to execute the simulation.

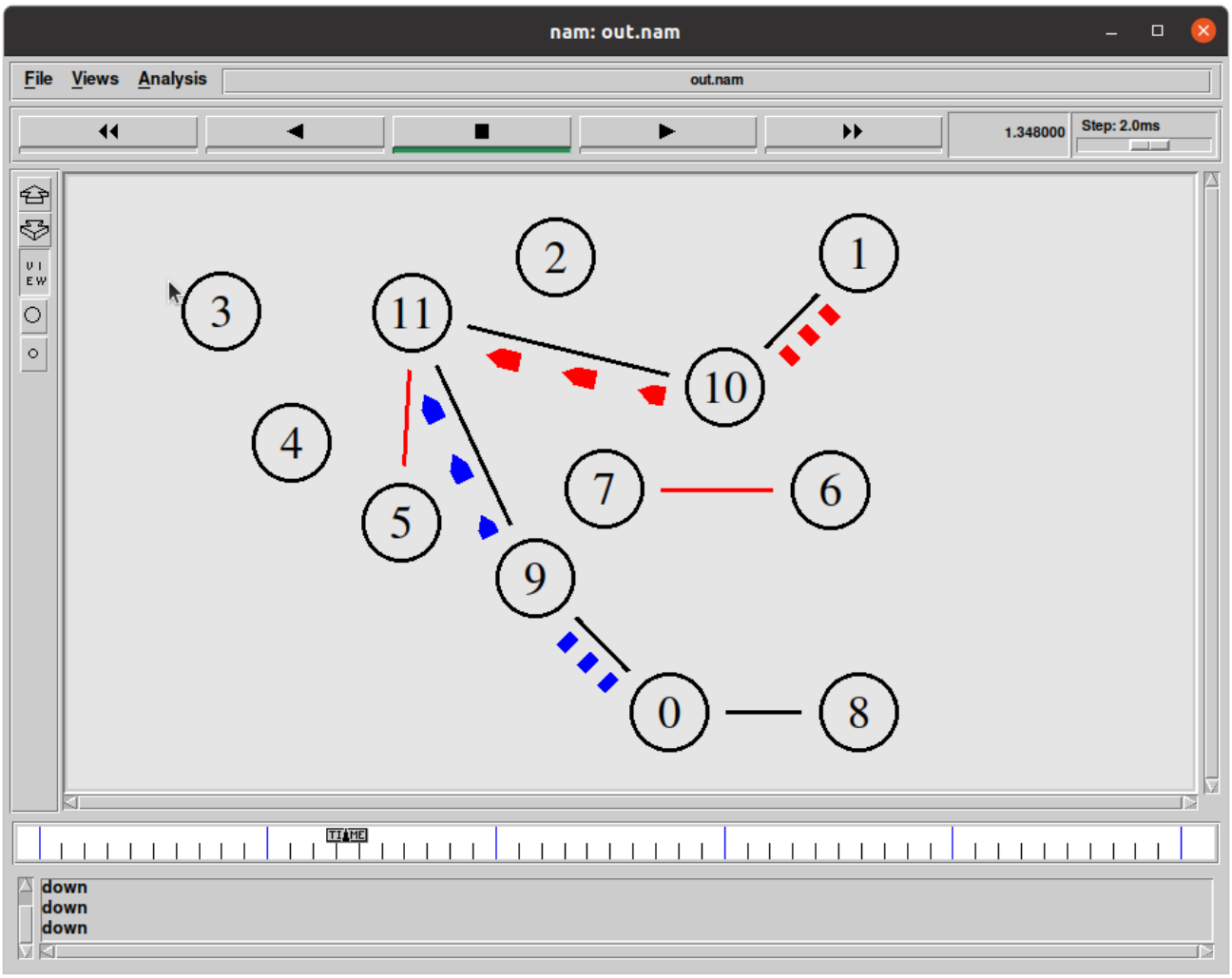
CODE:

```
#Create a simulator object
set ns [new Simulator]
```

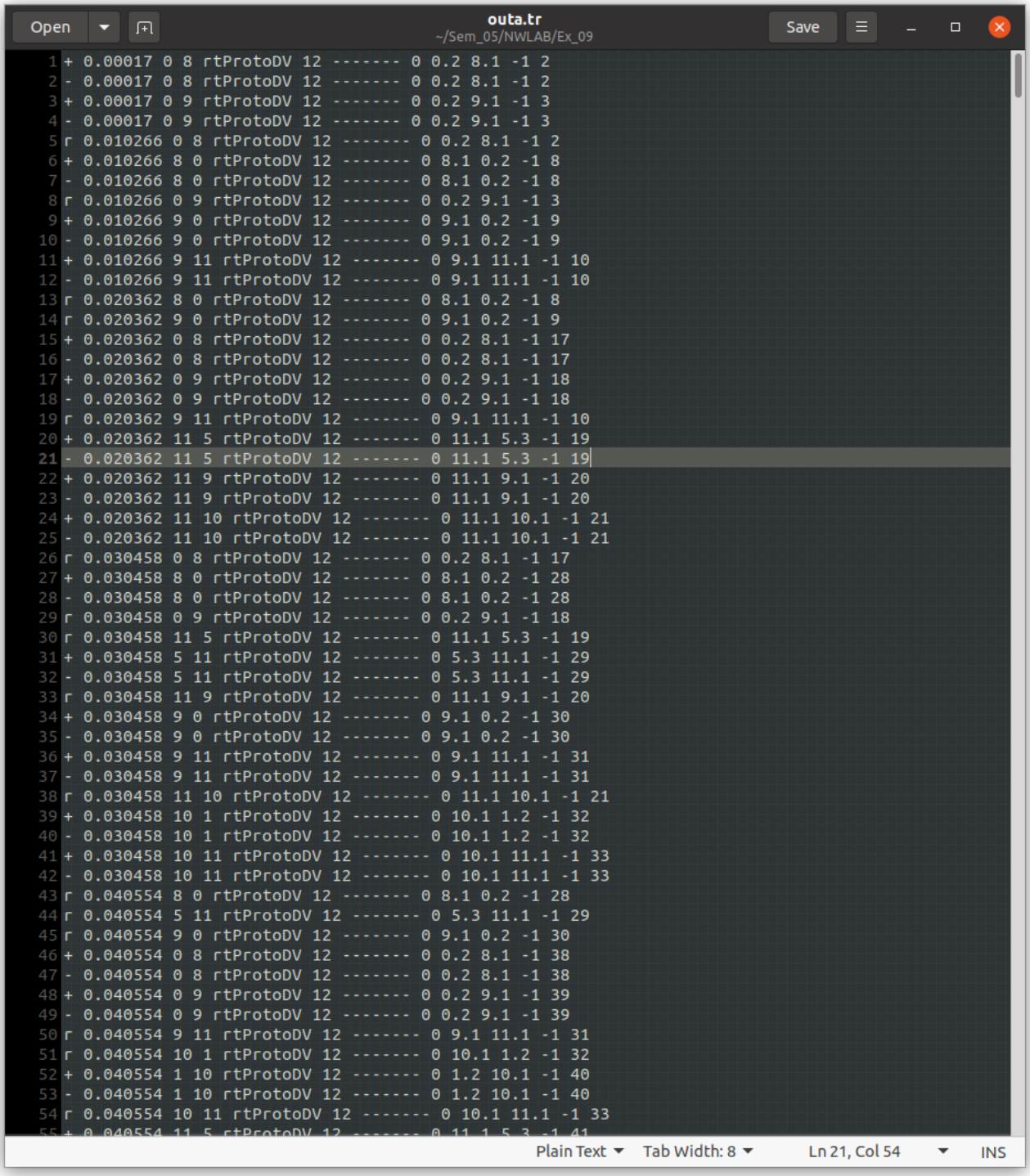
```
#Open the nam trace file
set nf [open outa.nam w]
$ns namtrace-all $nf

$ns color 0 blue
$ns color 1 red

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

#Define a 'finish' procedure
proc finish {} {
    global nf ns tr
    $ns flush-trace
     close $tr
    exec nam out.nam &
     exit 0
}

#Creating 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 [$ns node]
set n11 [$ns node]

#Setting Links
$ns duplex-link $n0 $n8 1Mb 10ms DropTail
$ns duplex-link $n1 $n10 1Mb 10ms DropTail
$ns duplex-link $n0 $n9 1Mb 10ms DropTail
$ns duplex-link $n9 $n11 1Mb 10ms DropTail
$ns duplex-link $n10 $n11 1Mb 10ms DropTail
$ns duplex-link $n11 $n5 1Mb 10ms DropTail
$ns duplex-link $n7 $n6 1Mb 10ms DropTail

#Setting Topology
$ns duplex-link-op $n0 $n8 orient right
$ns duplex-link-op $n1 $n10 orient left-down
$ns duplex-link-op $n0 $n9 orient left-up
$ns duplex-link-op $n9 $n11 orient up
$ns duplex-link-op $n10 $n11 orient left
```

```
$ns duplex-link-op $n11 $n5 orient left-down
$ns duplex-link-op $n7 $n6 orient right


set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0

set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1

set null0 [new Agent/Null]
$ns attach-agent $n5 $null0
set null1 [new Agent/Null]
$ns attach-agent $n5 $null1
$ns connect $udp0 $null0
$ns connect $udp1 $null1

$udp0 set class_ 0
$udp1 set class_ 1

$udp0 set fid_ 0
$udp0 set window_ 8000
$udp0 set packetSize_ 552
$udp1 set fid_ 1
$udp1 set window_ 8000
$udp1 set packetSize_ 552

$ns rtproto DV
$ns rtmodel-at 1.0 down $n11 $n5
$ns rtmodel-at 1.0 down $n7 $n6

$ns rtmodel-at 2.0 up $n11 $n5
$ns rtmodel-at 2.0 up $n7 $n6

$ns at 0.0 "$cbr0 start"
$ns at 0.0 "$cbr1 start"
$ns at 5.0 "$cbr0 stop"
$ns at 5.0 "$cbr1 stop"
$ns at 5.0 "finish"
$ns run
```

SCREENSHOT:


nam: out.nam
File
Views
Analysis
out.nam
1.348000
Step: 2.0ms
VIEW
TIME
down
down
down
0
1
2
3
4
5
6
7
8
9
10
11

Trace file:



b) Link State Routing

ALGORITHM:

1. Create a new object ns with the class Simulator.
2. Open the nam trace file.
3. The namtrace is set for enabling animation to simulate the environment.
4. The color field here is used to discriminate the different data packets travelling across the nodes.

5. Declare the nodes namely n0, n1, ... to be used in the simulation.
6. The duplex links between the nodes is set appropriately.
7. Set the orientation of the nodes appropriately for proper representation.
8. The queue limit is set to determine the capacity of the queue for any communication.
9. A UDP connection is set up between the node n0 and n5 and n1 and n5.
10. The CBR (constant bit rate) here facilitates this UDP connection.
11. Protocol to be used is specified by '$ns rtproto LS' where DV – Link State Protocol.
12. Making the links 11-5 and 7-6 down for 1 second starting from 2.0 to 3.0 in the course of simulation for a total of 5.0 seconds.
13. The run command is used to execute the simulation.

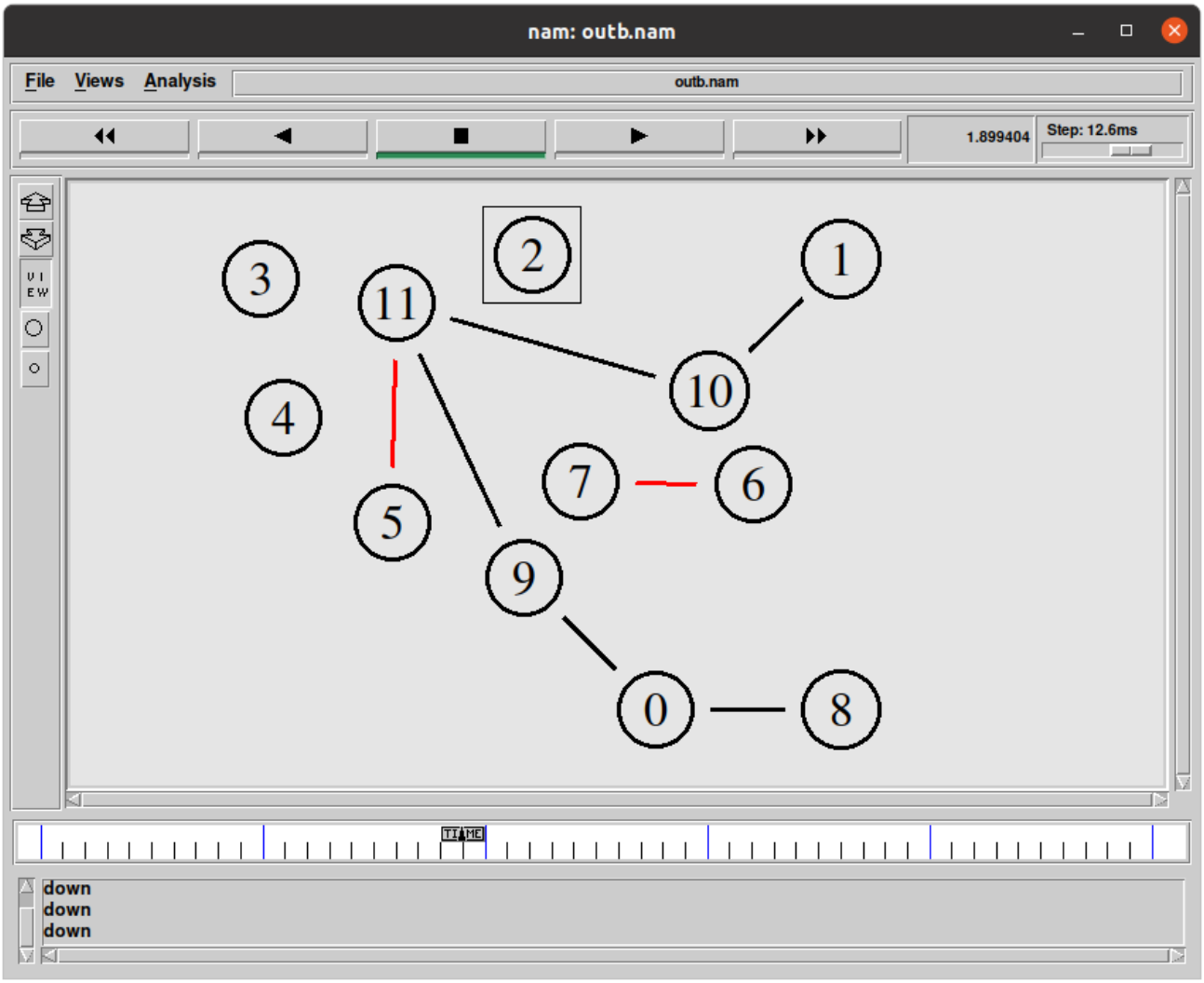
CODE:

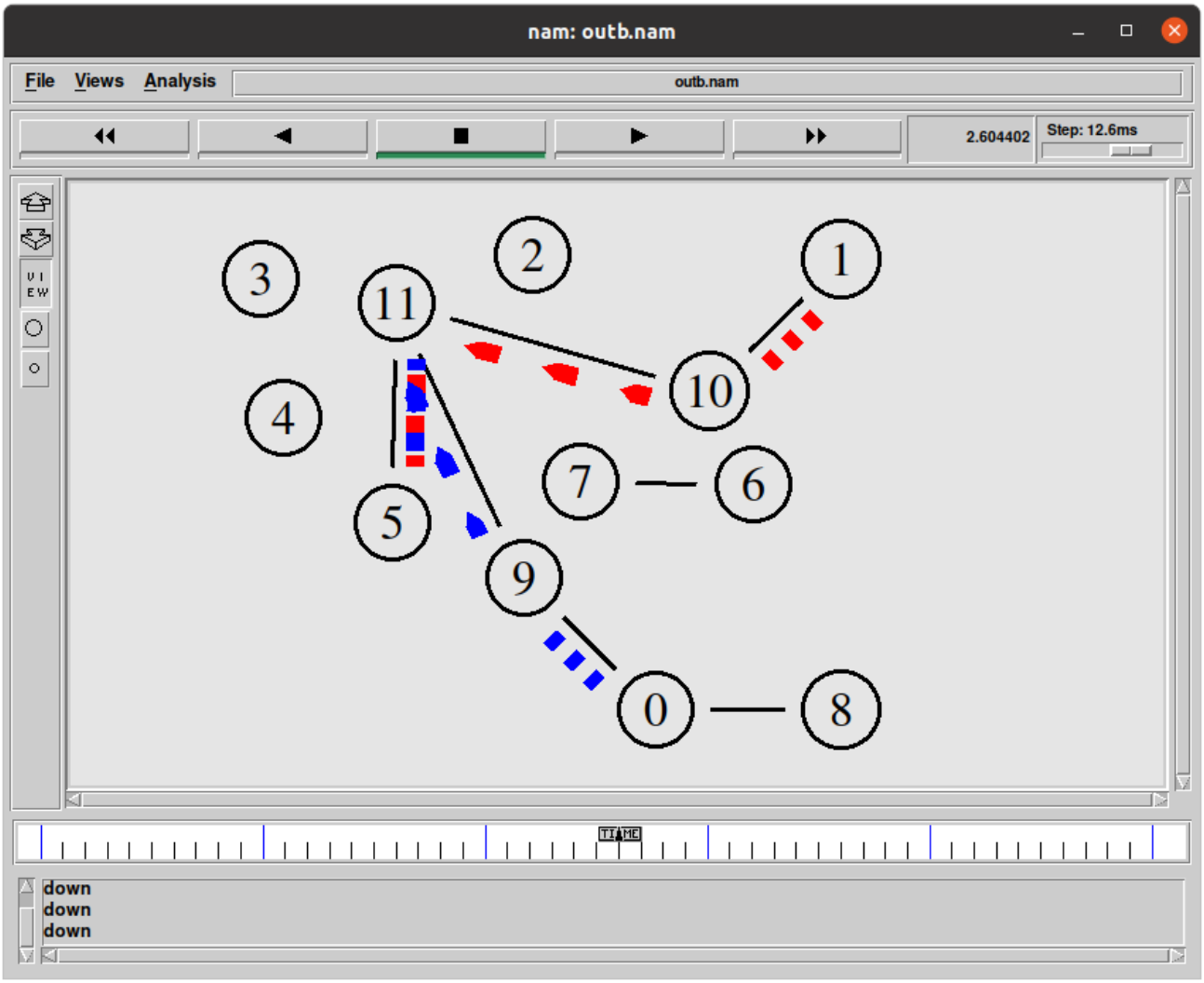
```
#Create a simulator object
set ns [new Simulator]

#Open the nam trace file
set nf [open outb.nam w]
$ns namtrace-all $nf

$ns color 0 blue
$ns color 1 red

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

#Define a 'finish' procedure
proc finish {} {
    global nf ns tr
    $ns flush-trace
     close $tr
    exec nam outb.nam &
     exit 0
}

#Creating 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]
set n11 [$ns node]

#Setting Links
$ns duplex-link $n0 $n8 1Mb 10ms DropTail
$ns duplex-link $n1 $n10 1Mb 10ms DropTail
$ns duplex-link $n0 $n9 1Mb 10ms DropTail
$ns duplex-link $n9 $n11 1Mb 10ms DropTail
$ns duplex-link $n10 $n11 1Mb 10ms DropTail
$ns duplex-link $n11 $n5 1Mb 10ms DropTail
$ns duplex-link $n7 $n6 1Mb 10ms DropTail

#Setting Topology
$ns duplex-link-op $n0 $n8 orient right
$ns duplex-link-op $n1 $n10 orient left-down
$ns duplex-link-op $n0 $n9 orient left-up
$ns duplex-link-op $n9 $n11 orient up
$ns duplex-link-op $n10 $n11 orient left
$ns duplex-link-op $n11 $n5 orient left-down
$ns duplex-link-op $n7 $n6 orient right


set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0

set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1

set null0 [new Agent/Null]
$ns attach-agent $n5 $null0
set null1 [new Agent/Null]
$ns attach-agent $n5 $null1
$ns connect $udp0 $null0
$ns connect $udp1 $null1

$udp0 set class_ 0
$udp1 set class_ 1

$udp0 set fid_ 0
$udp0 set window_ 8000
$udp0 set packetSize_ 552
$udp1 set fid_ 1
$udp1 set window_ 8000
$udp1 set packetSize_ 552

$ns rtproto LS
$ns rtmodel-at 1.0 down $n11 $n5
```

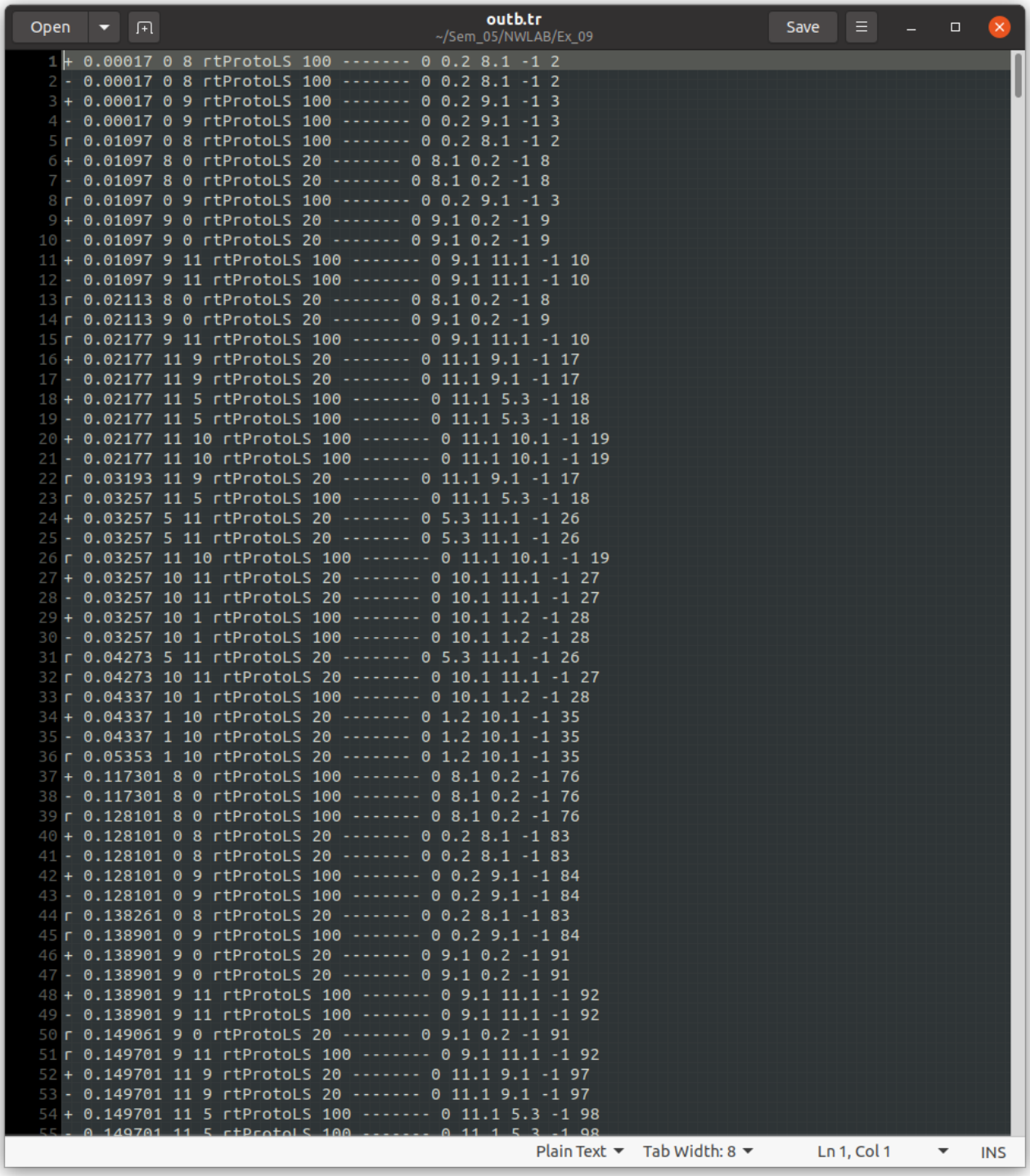
```
$ns rtmodel-at 1.0 down $n7 $n6

$ns rtmodel-at 2.0 up $n11 $n5
$ns rtmodel-at 2.0 up $n7 $n6

$ns at 0.0 "$cbr0 start"
$ns at 0.0 "$cbr1 start"
$ns at 5.0 "$cbr0 stop"
$ns at 5.0 "$cbr1 stop"
$ns at 5.0 "finish"
$ns run
```

SCREENSHOT:




nam: outb.nam
File
Views
Analysis
outb.nam
2.604402
Step: 12.6ms
3
11
2
1
10
4
7
6
5
9
0
8
down
down
down

Trace file:



Learning Outcomes:

This exercise helped me

- To write code for the network using Distance vector protocol and Link State protocol.
- To understand the difference between the Distance Vector and Link State protocol.
- To understand the flooding in Link State protocol.