**SSN College of Engineering, Kalavakkam**

**Department of Computer Science and Engineering**

**UCS1511 NETWORKS LAB**

**Exercise 9: SIMULATION OF ROUTING PROTOCOLS**

**Name :** Kshitij Sharma **Roll No. :** 185001080 **Date :** 22/10/2020

**LEARNING OBJECTIVES:** Write tcl script to simulate the routing protocols in wired networks

1. **Distance Vector Routing**

**ALGORITHM:**

1. The Simulator class is used to create a new variable ns.
2. Set namtrace for enabling animation to simulate the environment.
3. The color field here is used to discriminate the different data packets travelling across the nodes.
4. Then the various nodes n0,n1..etc are declared accordingly.
5. The duplex links between the nodes is set appropriately.
6. Following this the orientation of these nodes in the simulator is decided upon.
7. A UDP connection is set up between the node n0 and n5 and also between the nodes n1 and n5.
8. The CBR here facilitates this UDP connection.
9. The rtproto command is used for determining the routing protocol whether DV-distance vector or LS-link state routing is to be used.
10. The rtmodel down command facilitates in keeping the required links down for some time.

**CODE:**

set ns [new Simulator]

$ns color 0 blue

$ns color 1 red

set nf [open out.nam w]

$ns namtrace-all $nf

set tr [open out.tr w]

$ns trace-all $tr

proc finish {} {

global nf ns tr

$ns flush-trace

close $tr

exec nam out.nam &

exit 0

}

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]

$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

$ns duplex-link-op $n0 $n8 orient right

$ns duplex-link-op $n1 $n10 orient right-up

$ns duplex-link-op $n0 $n9 orient left-up

$ns duplex-link-op $n9 $n11 orient left

$ns duplex-link-op $n10 $n11 orient up

$ns duplex-link-op $n11 $n5 orient right-up

$ns duplex-link-op $n7 $n6 orient right

set udp [new Agent/UDP]

$ns attach-agent $n0 $udp

set udp1 [new Agent/UDP]

$ns attach-agent $n1 $udp1

$udp set class\_ 0

$udp1 set class\_ 1

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

set cbr1 [new Application/Traffic/CBR]

$cbr1 attach-agent $udp1

set null [new Agent/Null]

$ns attach-agent $n5 $null

$ns attach-agent $n6 $null

$ns connect $udp $null

$ns connect $udp1 $null

$udp set fid\_ 2

$udp set window\_ 8000

$udp set packetSize\_ 552

$udp1 set fid\_ 2

$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

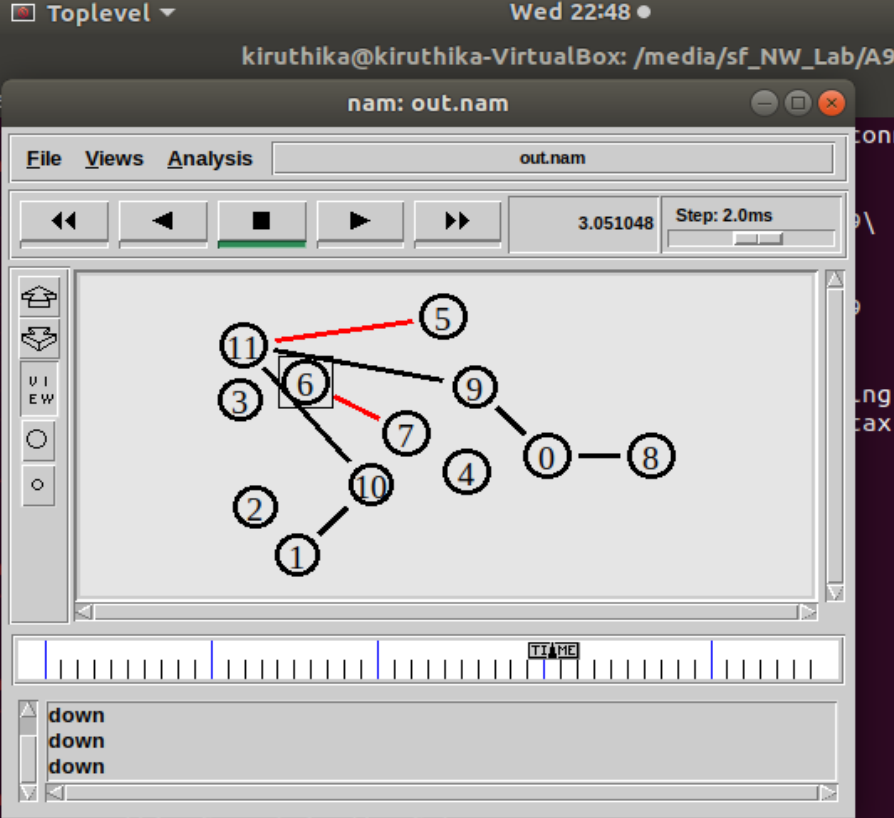
$ns at 0.0 "$cbr start"

$ns at 0.0 "$cbr1 start"

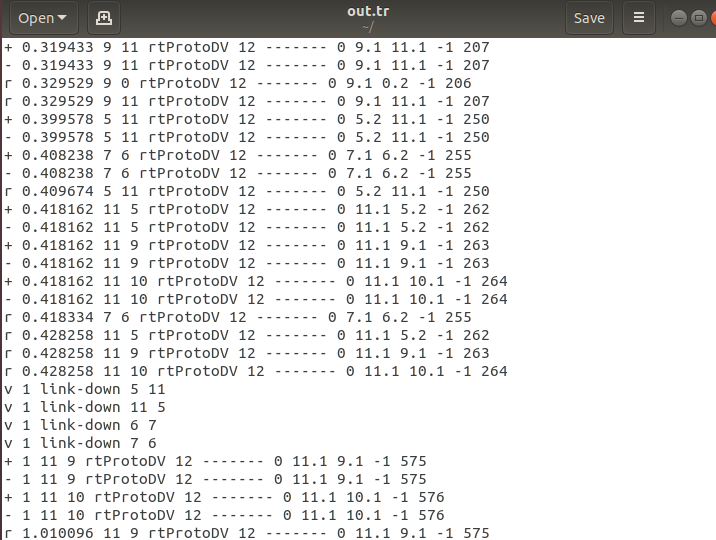
$ns at 5.0 "finish"

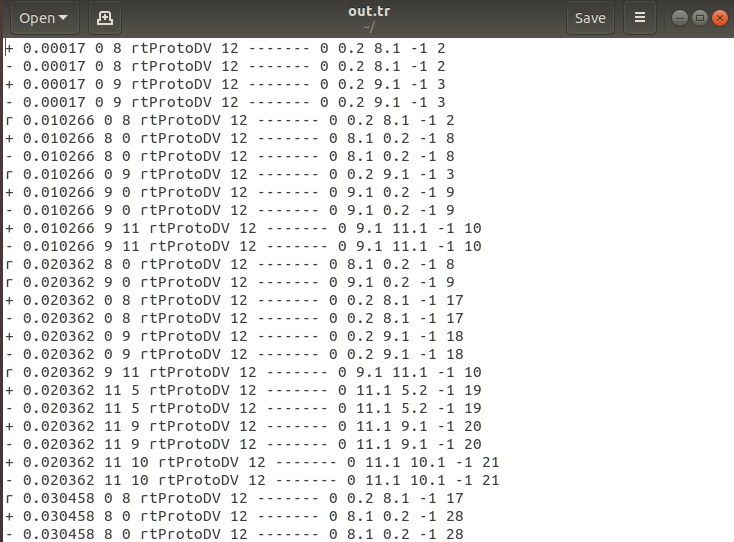
$ns run

**SCREENSHOT:**



**Trace file:**





**b) Link State Routing**

**Code:**

set ns [new Simulator]

$ns color 0 blue

$ns color 1 red

set nf [open out.nam w]

$ns namtrace-all $nf

set tr [open out.tr w]

$ns trace-all $tr

proc finish {} {

global nf ns tr

$ns flush-trace

close $tr

exec nam out.nam &

exit 0

}

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]

$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

$ns duplex-link-op $n0 $n8 orient right

$ns duplex-link-op $n1 $n10 orient right-up

$ns duplex-link-op $n0 $n9 orient left-up

$ns duplex-link-op $n9 $n11 orient left

$ns duplex-link-op $n10 $n11 orient up

$ns duplex-link-op $n11 $n5 orient right-up

$ns duplex-link-op $n7 $n6 orient right-up

set udp [new Agent/UDP]

$ns attach-agent $n0 $udp

set udp1 [new Agent/UDP]

$ns attach-agent $n1 $udp1

$udp set class\_ 0

$udp1 set class\_ 1

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

set cbr1 [new Application/Traffic/CBR]

$cbr1 attach-agent $udp1

set null [new Agent/Null]

$ns attach-agent $n5 $null

$ns attach-agent $n6 $null

$ns connect $udp $null

$ns connect $udp1 $null

$udp set fid\_ 2

$udp set window\_ 8000

$udp set packetSize\_ 552

$udp1 set fid\_ 2

$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

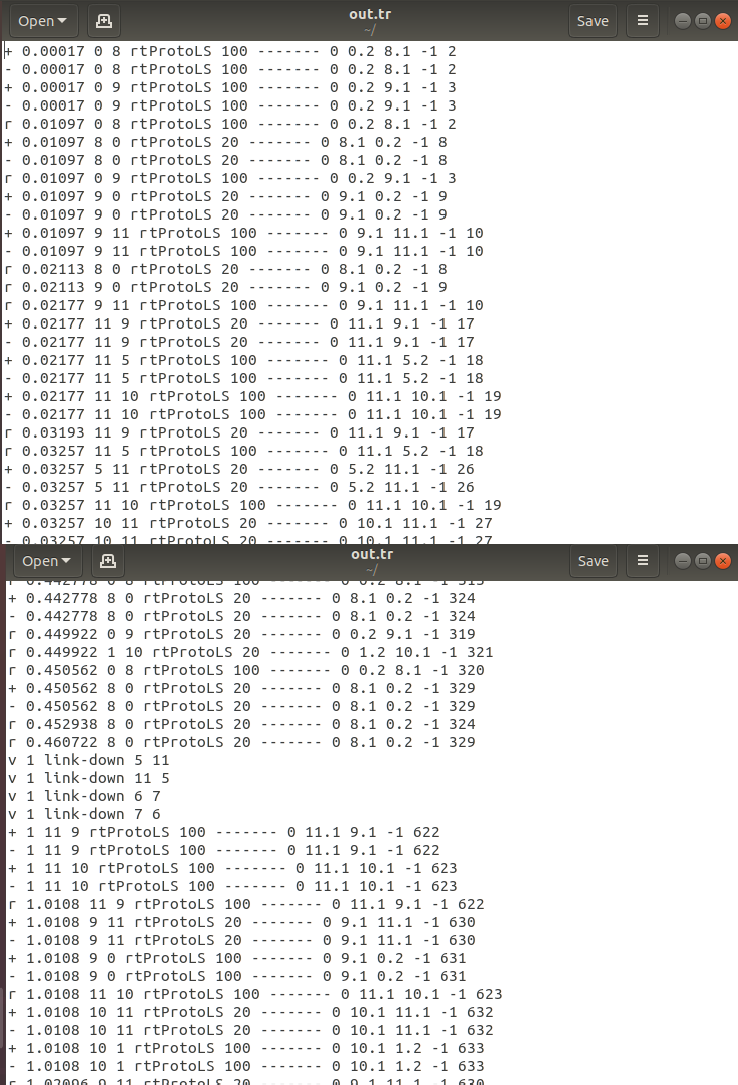
$ns at 0.0 "$cbr start"

$ns at 0.0 "$cbr1 start"

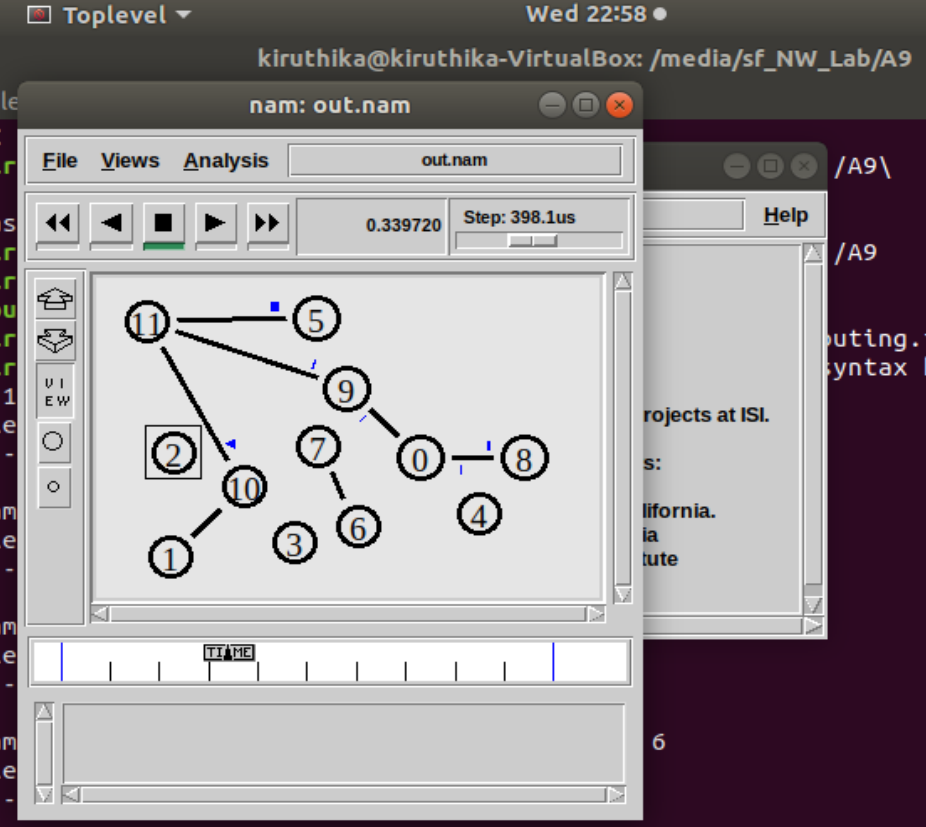
$ns at 5.0 "finish"

$ns run

**Trace file:**



**Screenshots:**



**LEARNING OUTCOMES:**

* I learnt to implement distance vector routing protocol and link state routing protocol using ns2.
* I learnt to analyse the simulation.