# Exercise 11: Link State Routing Protocol

## lsrp.tcl

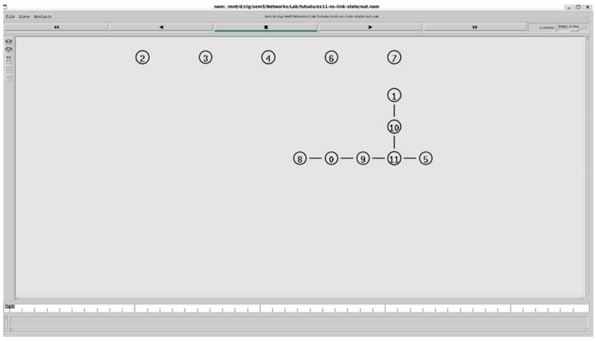
```
# Create a new simulator instance
set ns [new Simulator]

# Enable multicast routing
set multicast_on 1

# Create trace files
set tf [open out.tr w]
$ns trace-all $tf
set nf [open out.nam w]
$ns namtrace-all $nf

# Define different colors for different flows
$ns color 1 Blue
$ns color 2 Red

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

# Set node positions for better visualization  # Connected nodes (part of the network)
$n(0) set X_ 50
$n(0) set Y_ 50
$n(0) set Z_ 0

$n(1) set X_ 50
$n(1) set Y_ 150
$n(1) set Z_ 0

$n(5) set X_ 350
$n(5) set Y_ 100
$n(5) set Z_ 0

$n(8) set X_ 150
$n(8) set Y_ 100
$n(8) set Z_ 0

$n(9) set X_ 150
$n(9) set Y_ 50
$n(9) set Z_ 0

$n(10) set X_ 150
$n(10) set Y_ 150
$n(10) set Z_ 0
```

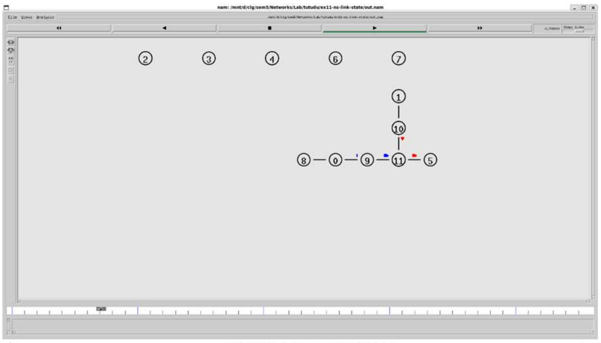
```
$n(11) set X_ 250
$n(11) set Y_ 100
$n(11) set Z_ 0

# Unused nodes (positioned away from the main network)
$n(2) set X_ 50
$n(2) set Y_ 180
$n(2) set Z_ 0

$n(3) set X_ 100
$n(3) set Y_ 180
$n(3) set Z_ 0

$n(4) set X_ 150
$n(4) set Y_ 180
$n(4) set Z_ 0

$n(6) set X_ 200
$n(6) set Y_ 180
$n(6) set Z_ 0

$n(7) set X_ 250
$n(7) set Y_ 180
$n(7) set Z_ 0

# Create links between nodes
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 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 link orientations
$ns duplex-link-op $n(0) $n(8) orient left
$ns duplex-link-op $n(0) $n(9) orient right
$ns duplex-link-op $n(1) $n(10) orient down
$ns duplex-link-op $n(9) $n(11) orient right
$ns duplex-link-op $n(10) $n(11) orient down
$ns duplex-link-op $n(11) $n(5) orient right

# Setup UDP connections
# First UDP connection (0 to 5)
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0
$udp0 set fid_ 1

# Second UDP connection (1 to 5)
```

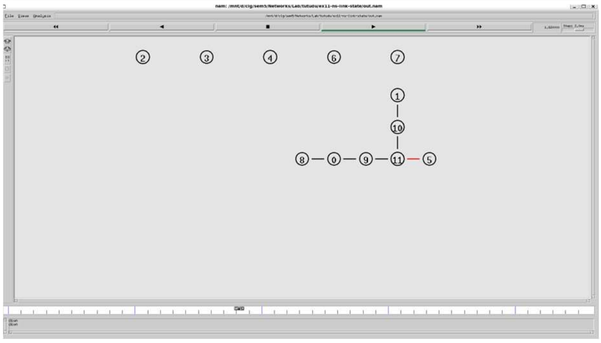
```
set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set null1 [new Agent/Null]
$ns attach-agent $n(5) $null1
$ns connect $udp1 $null1
$udp1 set fid_ 2

# Create CBR traffic for both connections
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set rate_ 200kb
$cbr0 set random_ 1
$cbr0 attach-agent $udp0

set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set rate_ 200kb
$cbr1 set random_ 1
$cbr1 attach-agent $udp1

# Use Link State Routing (instead of DV)
$ns rtproto LS

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

# Schedule events
$ns at 0.1 "$cbr0 start"
$ns at 0.2 "$cbr1 start"

# Schedule link failure for only link 11-5
$ns rtmodel-at 1.0 down $n(11) $n(5)
$ns rtmodel-at 2.0 up $n(11) $n(5)

# Stop the traffic
$ns at 4.5 "$cbr0 stop"
$ns at 4.5 "$cbr1 stop"

# Call finish procedure after 5 seconds
$ns at 5.0 "finish"

# Run the simulation
$ns run
```

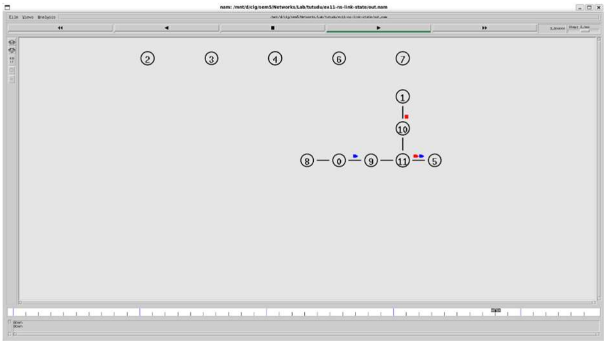
Pradeep KM
3122225001092

# Output


nam: /mnt/d/clg/sem5/Networks/Lab/tutudu/ex11-ns-link-state/out.nam
File
Views
Analysis
2
3
4
6
7
1
10
8
0
9
11
5


nam: /mnt/d/clg/sem5/Networks/Lab/tutudu/ex11-ns-link-state/out.nam
File
Views
Analysis
2
3
4
6
7
1
10
8
0
9
11
5


nam: /mnt/d/clg/sem5/Networks/Lab/tutudu/ex11-ns-link-state/out.nam
File
Views
Analysis
2
3
4
6
7
1
10
8
0
9
11
5


nam: /mnt/d/clg/sem5/Networks/Lab/tutudu/ex11-ns-link-state/out.nam
File
Views
Analysis
2
3
4
6
7
1
10
8
0
9
11
5