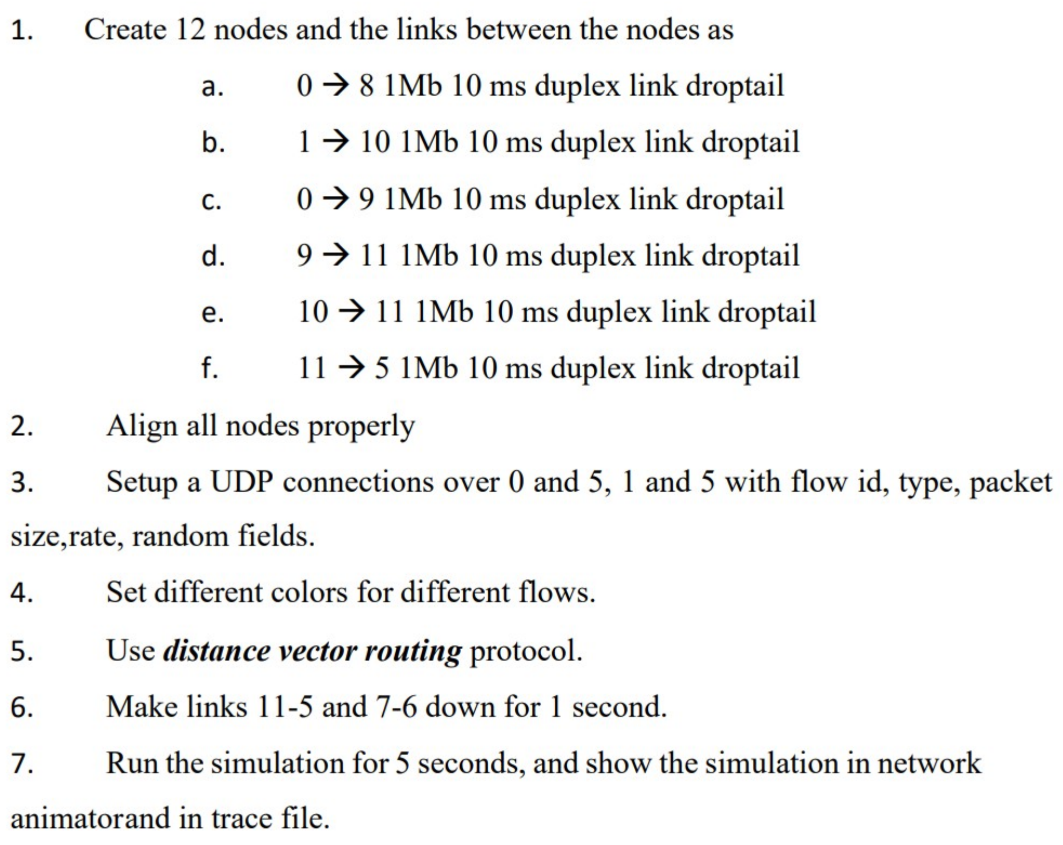
Simulation Of Routing Protocol In NS2

AIM

To create a network simulation with 12 nodes, using UDP connections over specific nodes, implementing the Distance Vector routing protocol, introducing link failures, and analyzing network performance through visualization in a Network Animator (NAM) and trace files.

Question



1. Create 12 nodes and the links between the nodes as
   a. 0 → 8 1Mb 10 ms duplex link droptail
   b. 1 → 10 1Mb 10 ms duplex link droptail
   c. 0 → 9 1Mb 10 ms duplex link droptail
   d. 9 → 11 1Mb 10 ms duplex link droptail
   e. 10 → 11 1Mb 10 ms duplex link droptail
   f. 11 → 5 1Mb 10 ms duplex link droptail
2. Align all nodes properly
3. Setup a UDP connections over 0 and 5, 1 and 5 with flow id, type, packet size,rate, random fields.
4. Set different colors for different flows.
5. Use ***distance vector routing*** protocol.
6. Make links 11-5 and 7-6 down for 1 second.
7. Run the simulation for 5 seconds, and show the simulation in network animatorand in trace file.

Algorithm:

1. Set up the simulator object and trace file.
2. Define colors for visual distinction of different flows.
3. Create 12 nodes to represent the network’s structure.
4. Connect nodes using duplex links with specific bandwidth, delay, and queue types, as specified.
5. Configure UDP agents and link them to the nodes.
6. Set flow IDs and colors for each UDP connection for easy identification in NAM.
7. Attach CBR (Constant Bit Rate) traffic to UDP agents, specifying packet size, rate, and interval.
8. Enable the Distance Vector (DV) protocol to facilitate routing.

9. Bring down specific links temporarily to observe the impact of link failures on routing and data delivery.
10. Run the simulation for a specified duration.
11. Capture outputs in trace files and visualize the network in NAM.

TCL Code:

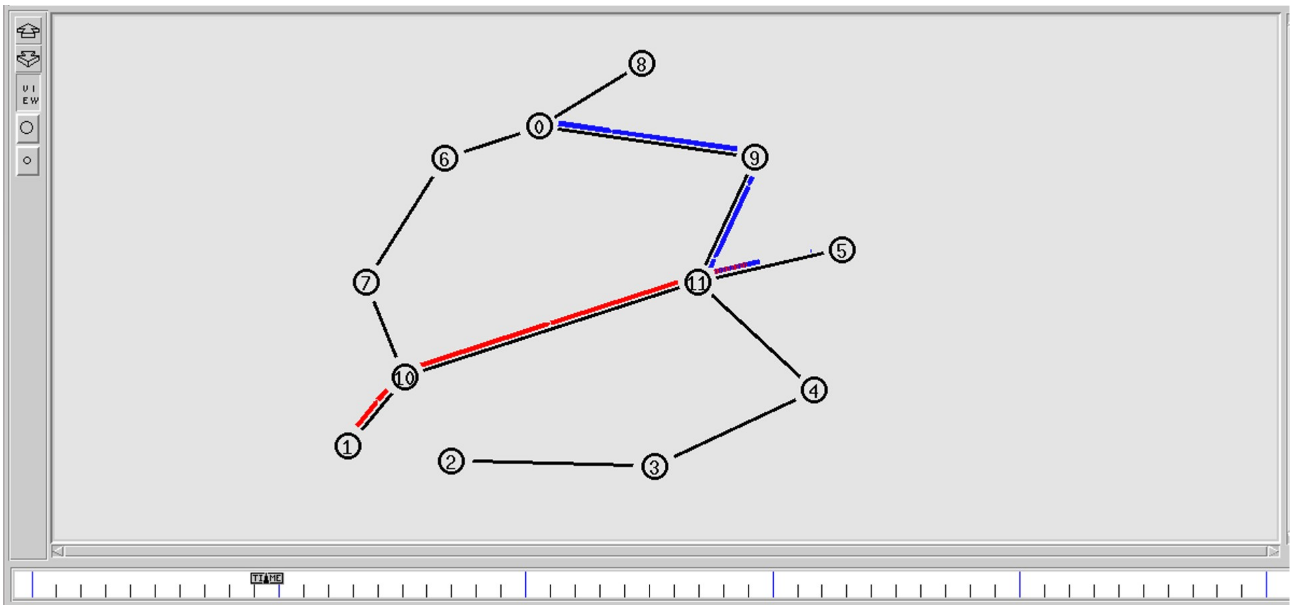
```
set ns [new Simulator]

$ns color 1 Blue
$ns color 2 Red

set nf [open out.nam w]
$ns namtrace-all $nf

proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}

# Create 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]

# Create duplex 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 1000ms DropTail
$ns duplex-link $n7 $n6 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
```

```
$ns duplex-link $n4 $n11 1Mb 10ms DropTail
$ns duplex-link $n10 $n7 1Mb 10ms DropTail
$ns duplex-link $n6 $n0 1Mb 10ms DropTail


# Create and attach UDP agents
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 1
$ns color $udp Blue

set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null1 [new Agent/Null]
$ns attach-agent $n5 $null1
$ns connect $udp1 $null1
$udp1 set fid_ 2
$ns color $udp1 Red

# Set up the CBR traffic generators
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp  ;# Attach CBR to the first UDP agent
$cbr set packetSize_ 1000
$cbr set interval 0.1
$cbr set rate_ 1Mb
$cbr set random_ false

set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1 ;# Attach CBR1 to the second UDP agent
$cbr1 set packetSize_ 1000
$cbr1 set interval 0.1
$cbr1 set rate_ 1Mb
$cbr1 set random_ false

# Start and stop CBR traffic
$ns at 0.5 "$cbr start"
$ns at 0.6 "$cbr1 start"

$ns at 5.0 "finish"

# Enabling Distance Vector Routing Protocol (DV)
$ns rtproto DV
# Network failure and recovery events
$ns rtmodel-at 3.5 down $n11 $n5 ;# Bringing down the link 11-5
$ns rtmodel-at 4.5 up $n11 $n5 ;# Bringing up the link 11-5 after
1 second
$ns rtmodel-at 3.0 down $n7 $n6 ;# Bringing down the link 7-6
```

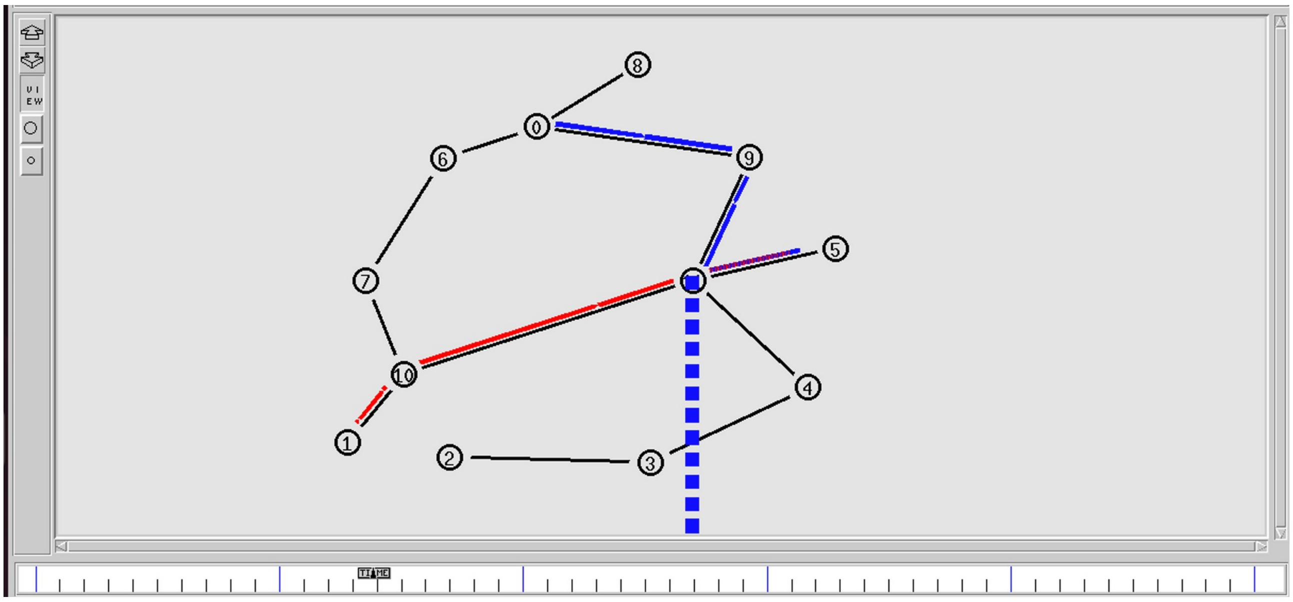
```
$ns rtmodel-at 4.0 up $n7 $n6 ;# Bringing up the link 7-6 after 1
second

puts "CBR packet size = [$cbr set packetSize_]"
puts "CBR interval = [$cbr set interval_]"

$ns run
```

EXECUTION SNAPSHOTS:

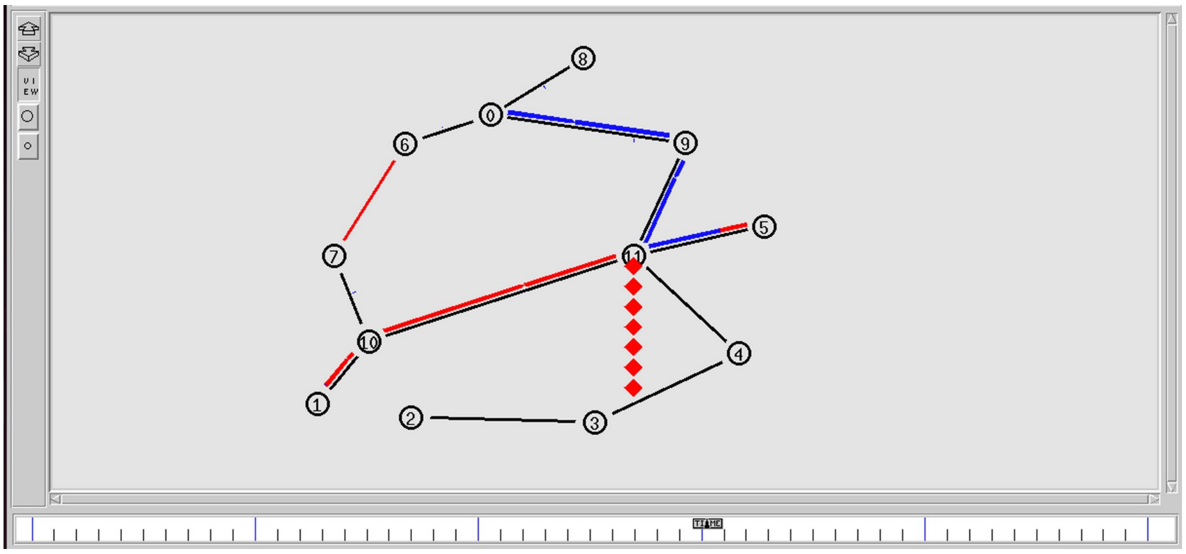
i. Start of simulation



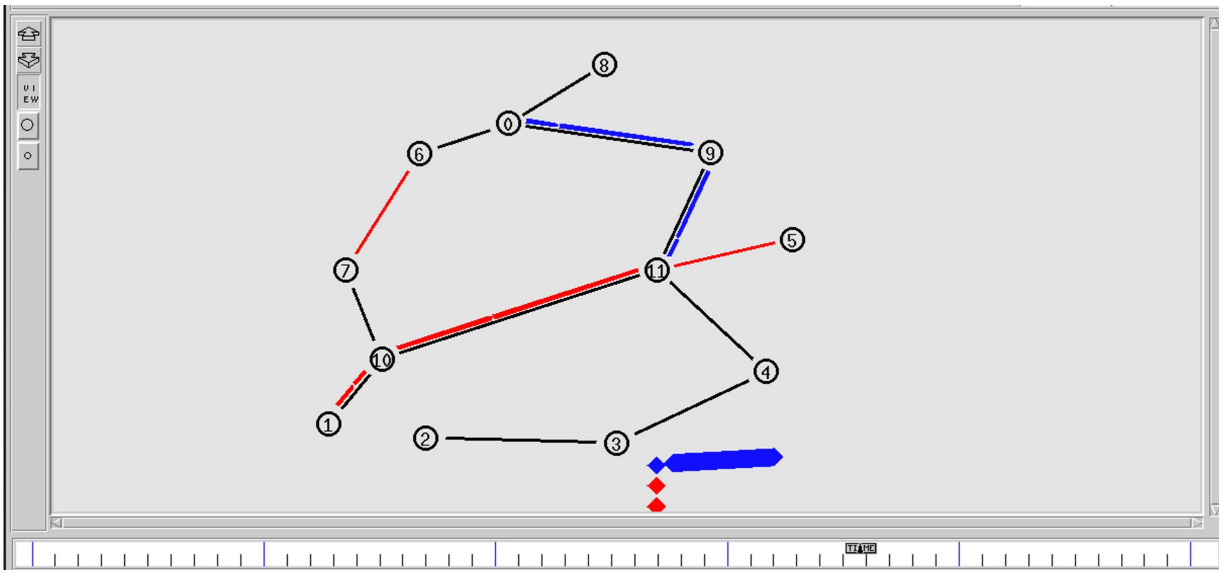
ii. Packets dropped



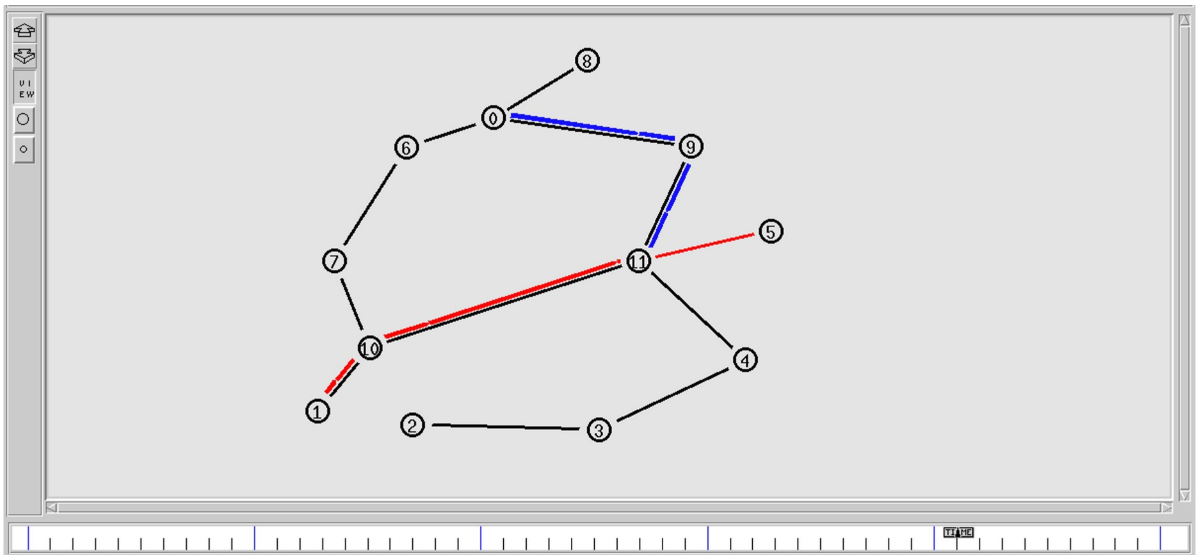
iii. 7-6 link down



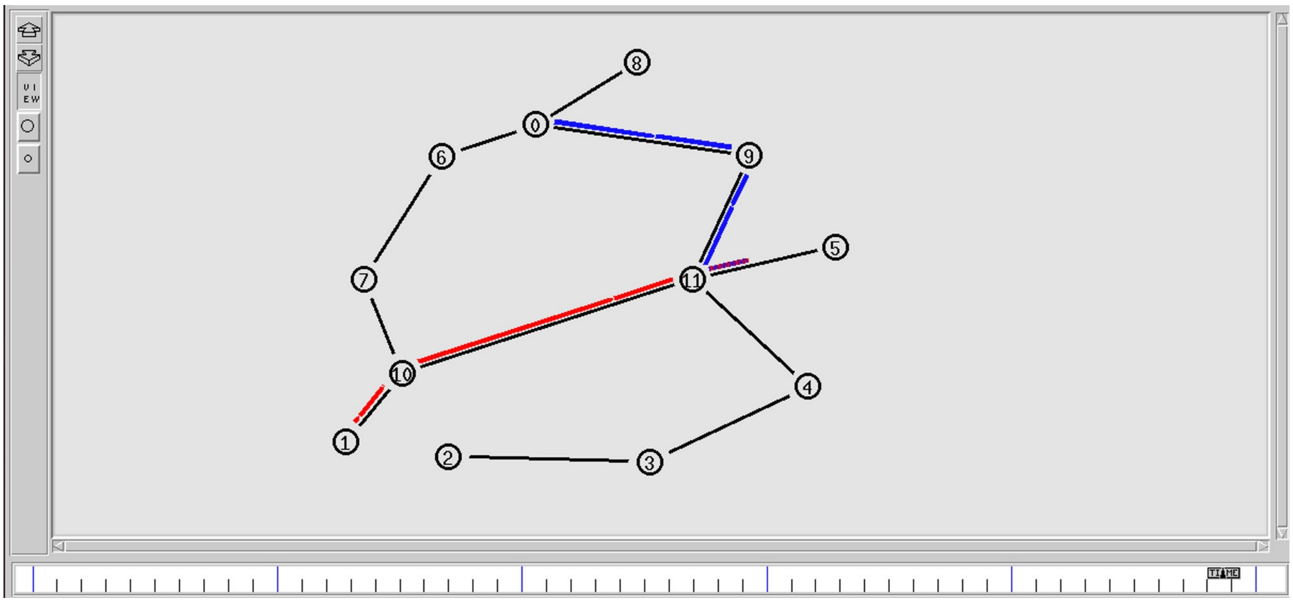
iv. Both 7-6 and 11-5 link down



v. 11-5 link down



vi. All links up



Learning Outcomes:

1. Understood to design and configure network topologies with multiple nodes and links.
2. Learnt to set up UDP agents and generating traffic with specific parameters for performance analysis.
3. Learnt to implement Distance Vector routing protocol in network simulations.

Best Practices:

1. Ensure nodes and links are appropriately labelled for easy identification.
2. Ensure that links and nodes are properly connected before starting traffic to avoid runtime errors.
3. Use comments and logical code blocks to improve readability and maintenance.