# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 5

Date Of Performance: 05/03/2025

Aim: Implement Distance Vector Routing Protocol in NS2.

Lab Objectives:

The main objective of this lab is to implement Distance Vector (DV) Routing Protocol in Network Simulator 2 (NS2) and understand how dynamic routing works when there is a link failure and recovery in the network.

Lab Outcomes:

This lab demonstrates the implementation of the Distance Vector (DV) Routing Protocol in NS2, enabling dynamic routing in a simulated network. It shows how packets are rerouted automatically during link failures and recovery, ensuring continuous data transmission. Students learn to analyse network resilience using trace files and NAM visualization. This experiment enhances understanding of adaptive routing protocols in real-world networking scenarios.

CODE:

# Define the simulation environment

set ns [new Simulator]

# Open the trace and NAM files

set tracefile [open out.tr w]

$ns trace-all $tracefile

set namfile [open out.nam w]

$ns namtrace-all $namfile

# Define the finish procedure

proc finish {} {

global ns tracefile namfile

$ns flush-trace

close $tracefile

close $namfile

exec nam out.nam &

exit 0

}

# Create network nodes

set node1 [$ns node]

set node2 [$ns node]

set node3 [$ns node]

set node4 [$ns node]

set node5 [$ns node]

# Assign different colors to nodes

$node1 color "red"

$node2 color "blue"

$node3 color "green"

$node4 color "orange"

$node5 color "purple"

# Create duplex links between nodes with different delays and bandwidths

$ns duplex-link $node1 $node2 10Mb 10ms DropTail

$ns duplex-link $node2 $node3 10Mb 10ms DropTail

$ns duplex-link $node3 $node4 10Mb 10ms DropTail

$ns duplex-link $node4 $node5 10Mb 10ms DropTail

$ns duplex-link $node1 $node5 5Mb 50ms DropTail

$ns duplex-link $node2 $node4 5Mb 30ms DropTail

# Assign different colors to links

$ns duplex-link-op $node1 $node2 color "red"

$ns duplex-link-op $node2 $node3 color "blue"

$ns duplex-link-op $node3 $node4 color "green"

$ns duplex-link-op $node4 $node5 color "orange"

$ns duplex-link-op $node1 $node5 color "purple"

$ns duplex-link-op $node2 $node4 color "yellow"

# Set initial positions of nodes for NAM visualization

$node1 label "N1"

$node2 label "N2"

$node3 label "N3"

$node4 label "N4"

$node5 label "N5"

# Enable Distance Vector (DV) routing protocol

$ns rtproto DV

# Attach TCP agents for traffic generation

set tcp [new Agent/TCP]

$ns attach-agent $node1 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $node5 $sink

# Connect the TCP agent and sink

$ns connect $tcp $sink

# Create an FTP application and attach it to TCP

set ftp [new Application/FTP]

$ftp attach-agent $tcp

# Start the FTP application at 1 second

$ns at 1.0 "$ftp start"

# Simulate link failure between node2 and node3 at 3.0 seconds

$ns rtmodel-at 3.0 down $node2 $node3

# Restore the link between node2 and node3 at 6.0 seconds

$ns rtmodel-at 6.0 up $node2 $node3

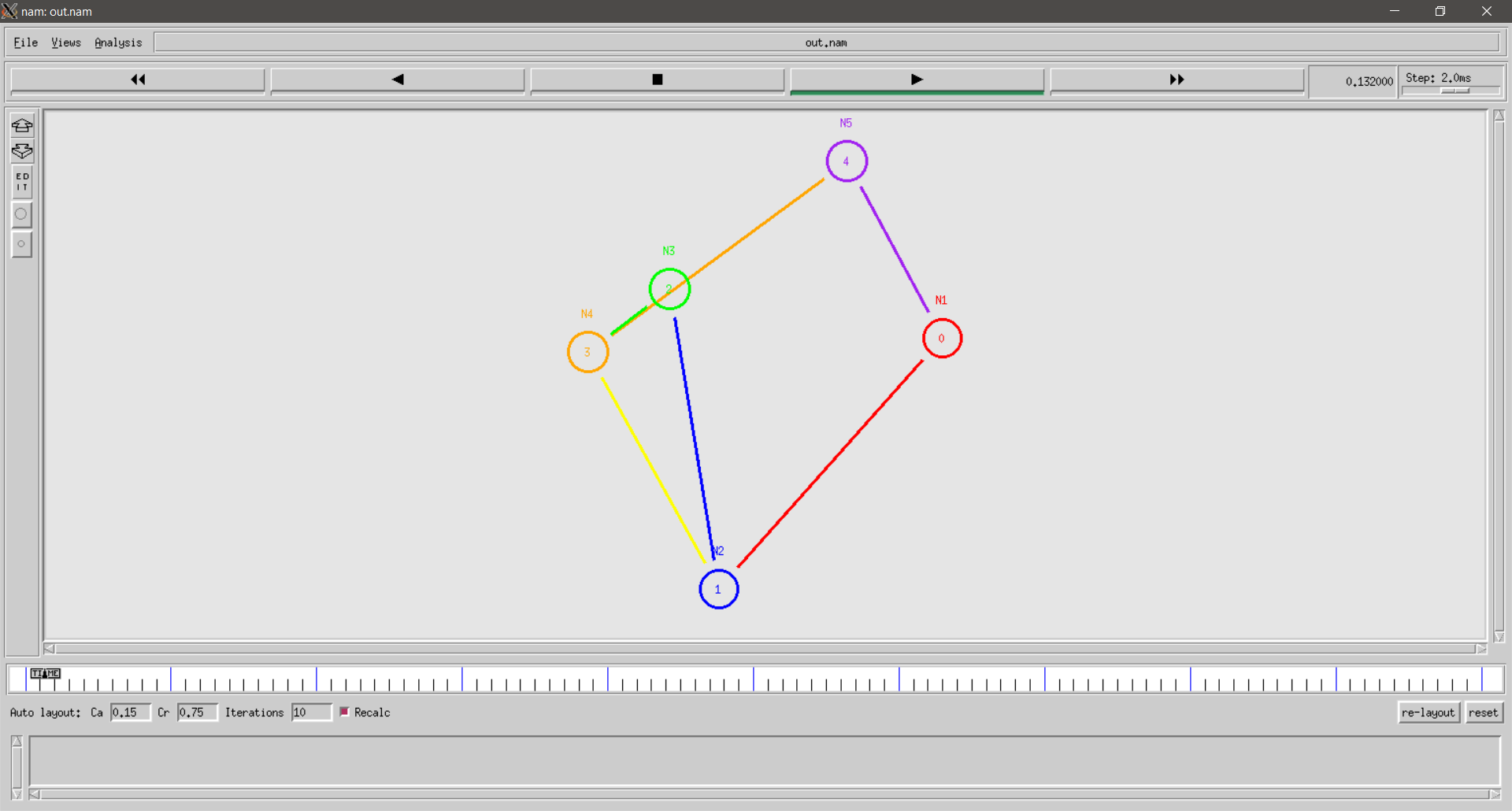
# Stop the simulation at 10.0 seconds

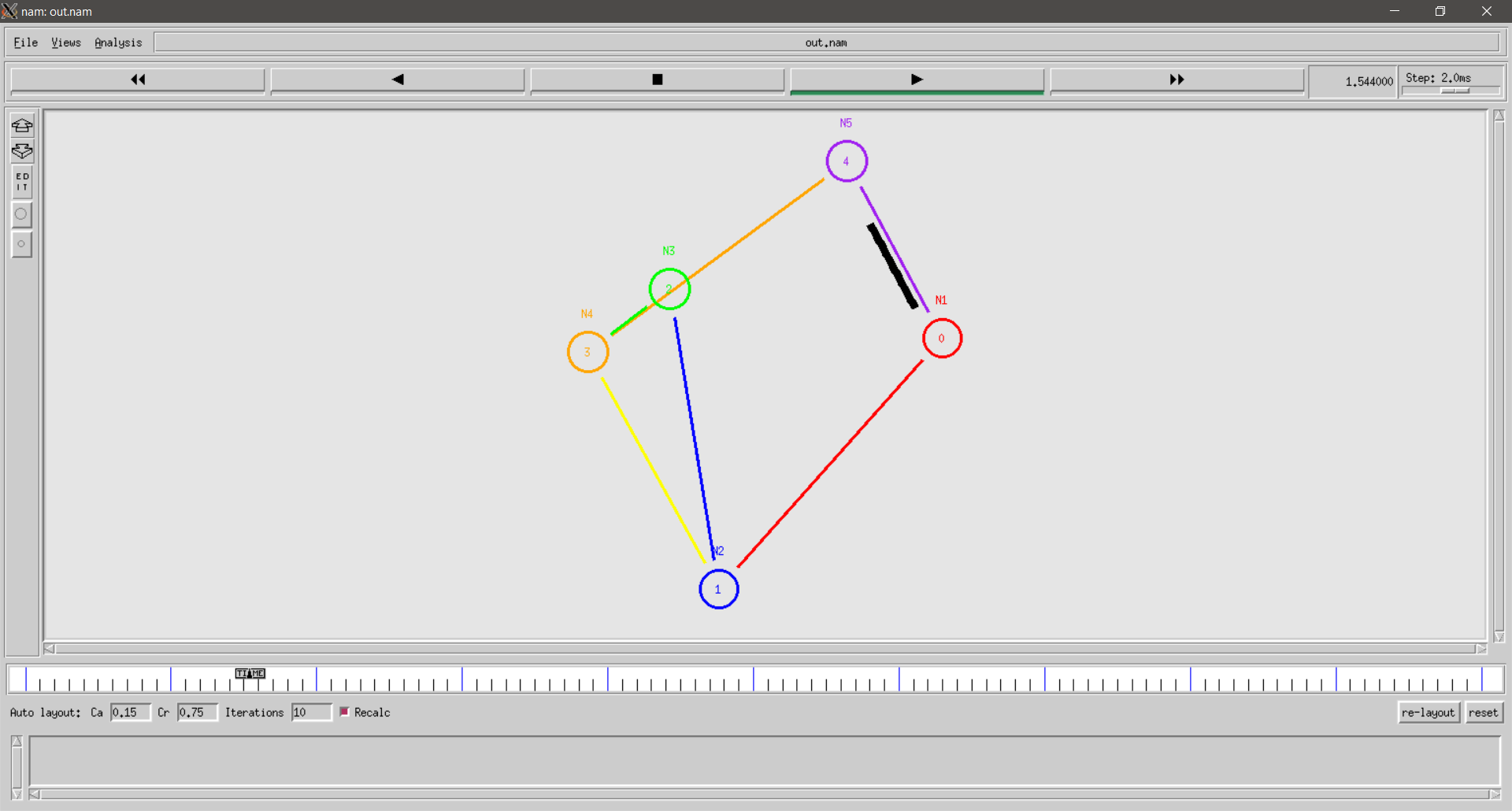
$ns at 10.0 "finish"

# Run the simulation

$ns run

OUTPUT:





Conclusion:

From this experiment, we successfully implemented Distance Vector Routing Protocol in NS2. The simulation demonstrated how DV dynamically reroutes packets during link failures and restores the communication when the link is recovered. This proves that Distance Vector Routing is a dynamic protocol that automatically updates the routing table based on network topology changes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Performance  (7M) | Journal  (3M) | Lab Ethics  (2M) | Attendance  (3M) | Total  (15M) | Faculty Signature |
|  |  |  |  |  |  |