**EX NO 8A IMPLEMENTATION OF DISTANCE VECTOR ROUTING PROTOCOL**

**AIM:** To perform the simulation of the distance vector routing protocol using NS2.

## ALGORITHM:

* 1. Start the program
  2. Create the trace file and NAM file
  3. Setup the topology object
  4. Create mobile nodes and attach them to the channel
  5. Configure the nodes and provide initial location of mobile nodes
  6. Set up a TCP Connection between nodes
  7. Define and initialize positions for the NAM window
  8. Specify the end of simulation
  9. Stop.
  10. 10.

## PROGRAM:

set ns [new Simulator]

## #Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Yellow #Open the Trace file set file1 [open out.tr w]

$ns trace-all $file1

#Open the NAM trace file set file2 [open out.nam w]

$ns namtrace-all $file2 **#Define a 'finish' procedure** proc finish {} {

global ns file1 file2

$ns flush-trace close $file1

close $file2

exec nam out.nam & exit 0

}

## # Next line should be commented out to have the static routing

$ns rtproto DV #Create six nodes set n0 [$ns node] set n1 [$ns node] set n2 [$ns node] set n3 [$ns node] set n4 [$ns node] set n5 [$ns node]

## #Create links between the nodes

$ns duplex-link $n0 $n4 0.3Mb 10ms DropTail

$ns duplex-link $n0 $n1 0.3Mb 10ms DropTail

$ns duplex-link $n1 $n2 0.3Mb 10ms DropTail

$ns duplex-link $n2 $n3 0.3Mb 10ms DropTail

$ns duplex-link $n1 $n4 0.3Mb 10ms DropTail

$ns duplex-link $n4 $n3 0.5Mb 10ms DropTail

$ns duplex-link $n3 $n5 0.5Mb 10ms DropTail

$ns duplex-link $n4 $n5 0.5Mb 10ms DropTail

## #Give node position (for NAM)

$ns duplex-link-op $n0 $n4 orient up

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link-op $n1 $n2 orient right

$ns duplex-link-op $n2 $n3 orient up

$ns duplex-link-op $n1 $n4 orient up-left

$ns duplex-link-op $n4 $n3 orient right

$ns duplex-link-op $n3 $n5 orient left-up

$ns duplex-link-op $n4 $n5 orient right-up

## #Setup a TCP connection

set tcp [new Agent/TCP/Newreno]

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink/DelAck]

$ns attach-agent $n5 $sink

$ns connect $tcp $sink

$tcp set fid\_ 1

## #Setup a FTP over TCP connection

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type\_ FTP

$ns rtmodel-at 1.0 down $n0 $n4

$ns rtmodel-at 4.5 up $n0 $n4

$ns at 0.1 "$ftp start"

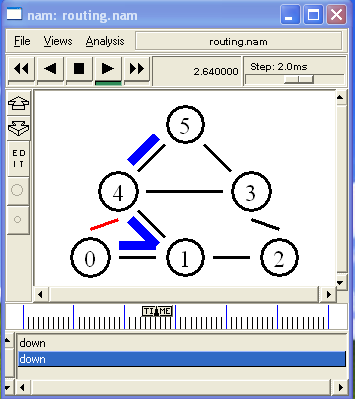
$ns at 6.0 "finish"

$ns run

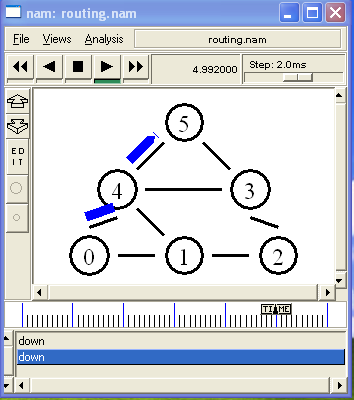
## OUTPUT:

**Up:**

## Down:



**Up:**



## RESULT:

**EX: NO: 8B IMPLEMENTATION OF LINK STATE ROUTING PROTOCOL**

**AIM:** To perform the simulation of the link state routing protocol using NS2.

## ALGORITHM:

1. Define new simulator
2. Define different colors for data flows (for NAM)
3. Define a new Trace file and open it
4. Define a new NAM Trace file and open it
5. Define a 'finish' procedure – to flush trace record in the `trace and trace output files.
6. Define the routing protocol as Link State (LS)
7. Create six nodes – n0,n1,..n5
8. Create links between the nodes with 0.3Mb and 10 ms Link with DropTail option
9. Give node position (for NAM) to place six nodes in the layout
10. Setup a TCP connection – attach TCP Source Agent to node n0 and TCP sink agent to node n5
11. Setup a FTP over TCP connection
12. Define configuration such that link between nodes n1 and n4 to be failed at 1.0 interval, and up again at 4.5 interval
13. Start the simulation

## PROGRAM:

#routing2.tcl

set ns [new Simulator]

#Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Red

#Open the Trace file

set file1 [open routing2.tr w]

$ns trace-all $file1

#Open the NAM trace file

set file2 [open routing2.nam w]

$ns namtrace-all $file2

#Define a 'finish' procedure proc finish {}

{

global ns file1 file2

$ns flush-trace close $file1

close $file2

exec nam routing2.nam & exit 0

}

# Next line should be commented out to have the static routing

$ns rtproto LS #Create six nodes set n0 [$ns node] set n1 [$ns node] set n2 [$ns node] set n3 [$ns node] set n4 [$ns node] set n5 [$ns node]

#Create links between the nodes

$ns duplex-link $n0 $n1 0.3Mb 10ms DropTail

$ns duplex-link $n1 $n2 0.3Mb 10ms DropTail

$ns duplex-link $n2 $n3 0.3Mb 10ms DropTail

$ns duplex-link $n1 $n4 0.3Mb 10ms DropTail

$ns duplex-link $n3 $n5 0.5Mb 10ms DropTail

$ns duplex-link $n4 $n5 0.5Mb 10ms DropTail

#Give node position (for NAM)

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link-op $n1 $n2 orient right

$ns duplex-link-op $n2 $n3 orient up-down

$ns duplex-link-op $n1 $n4 orient up-left

$ns duplex-link-op $n3 $n5 orient left-up

$ns duplex-link-op $n4 $n5 orient right-up

#Setup a TCP connection

set tcp [new Agent/TCP/Newreno]

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink/DelAck]

$ns attach-agent $n5 $sink

$ns connect $tcp $sink

$tcp set fid\_ 1

#Setup a FTP over TCP connection set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type\_ FTP

$ns rtmodel-at 1.0 down $n1 $n4

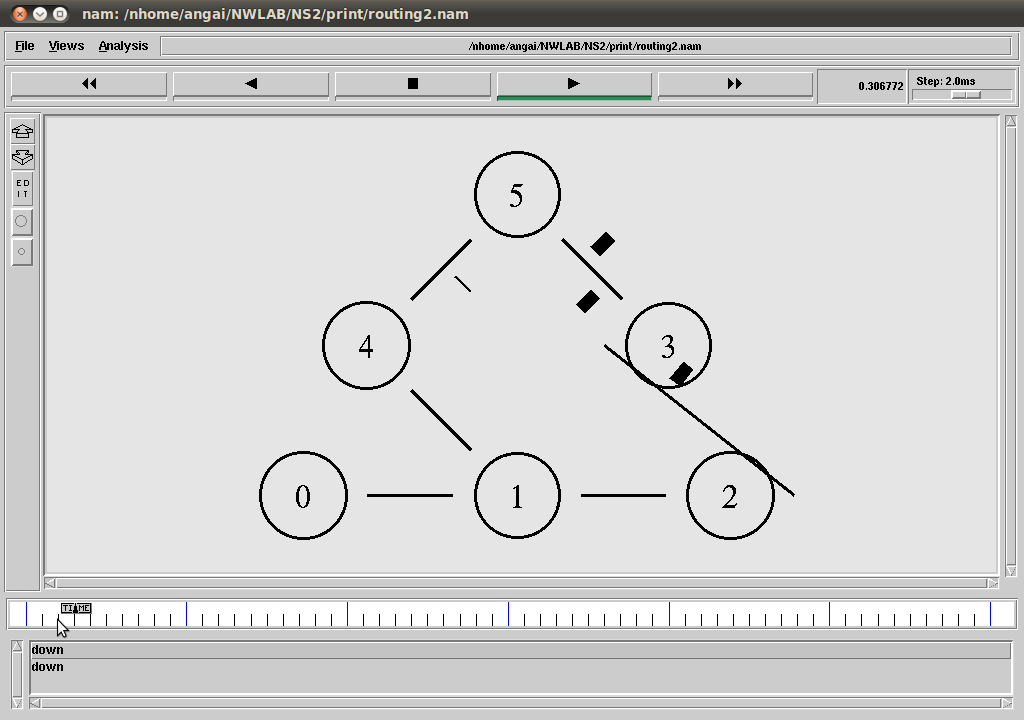
$ns rtmodel-at 3.0 up $n1 $n4

$ns at 0.1 "$ftp start"

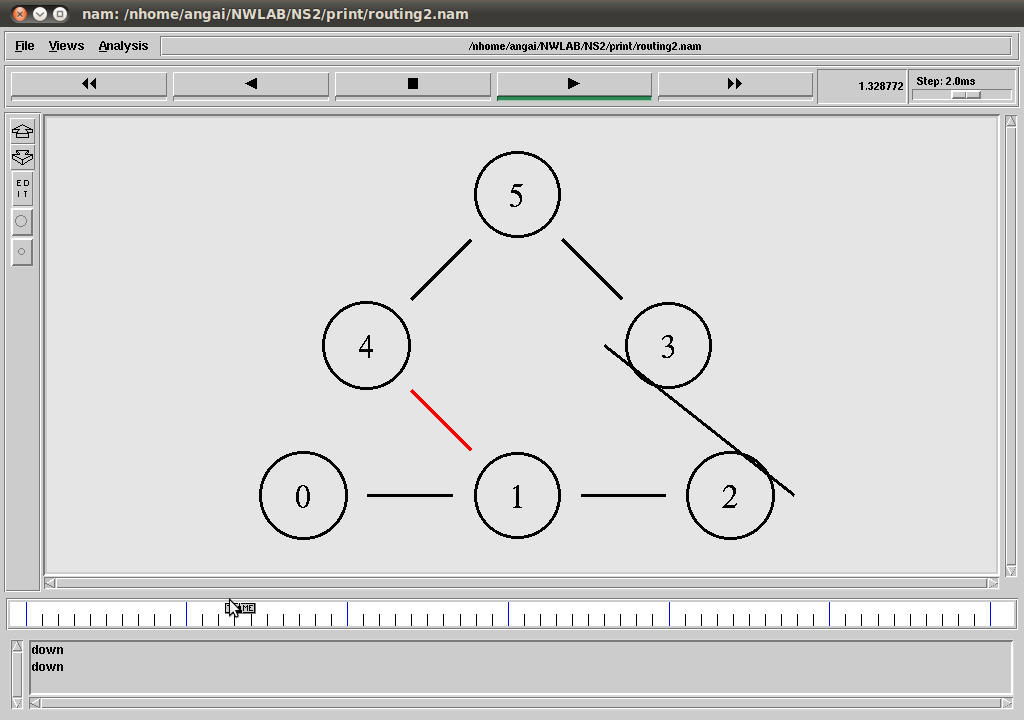
$ns at 6.0 "finish"

$ns run

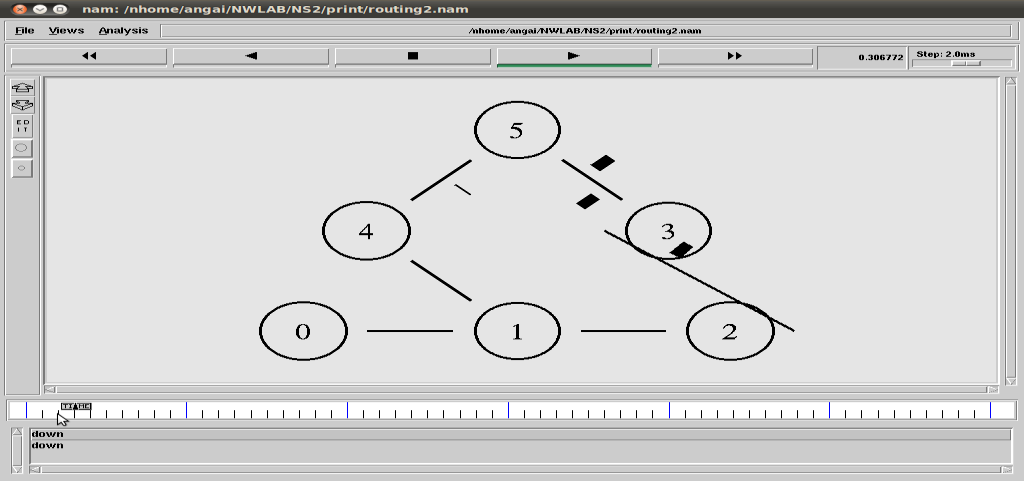
## OUTPUT:



**Before Link failure between Nodes n1 and n4**



## While Link failure between Nodes n1 and n4



**After failed link between Nodes n1 and n4 up**

## RESULT:

**EX NO 9 PERFORMANCE EVALUATION OF ROUTING PROTOCOLS USING SIMULATION TOOL**

## AIM:

To perform the simulation of the routing protocols using NS2.

## ALGORITHM:

Step 1: Start the program

Step 2: Create the trace file and NAM file Step 3: Setup the topology object

Step 4: Create mobile nodes and attach them to the channel

Step 5: Configure the nodes and provide initial location of mobile nodes Step 6: Set up a TCP Connection between nodes

Step 7: Define and initialize positions for the NAM window Step 8: Specify the end of simulation

Step 9: Stop.

## PROGRAM

set ns [new Simulator]

#Define different colors for data flows (for NAM)

$ns color 1 Blue

$ns color 2 Yellow #Open the Trace file set file1 [open out.tr w]

$ns trace-all $file1

#Open the NAM trace file set file2 [open out.nam w]

$ns namtrace-all $file2 #Define a 'finish' procedure proc finish {} {

global ns file1 file2

$ns flush-trace close $file1

close $file2

exec nam out.nam &

exit 0

}

# Next line should be commented out to have the static routing

$ns rtproto DV #Create six nodes set n0 [$ns node] set n1 [$ns node] set n2 [$ns node] set n3 [$ns node] set n4 [$ns node] set n5 [$ns node]

#Create links between the nodes

$ns duplex-link $n0 $n4 0.3Mb 10ms DropTail

$ns duplex-link $n0 $n1 0.3Mb 10ms DropTail

$ns duplex-link $n1 $n2 0.3Mb 10ms DropTail

$ns duplex-link $n2 $n3 0.3Mb 10ms DropTail

$ns duplex-link $n1 $n4 0.3Mb 10ms DropTail

$ns duplex-link $n4 $n3 0.5Mb 10ms DropTail

$ns duplex-link $n3 $n5 0.5Mb 10ms DropTail

$ns duplex-link $n4 $n5 0.5Mb 10ms DropTail #Give node position (for NAM)

$ns duplex-link-op $n0 $n4 orient up

$ns duplex-link-op $n0 $n1 orient right

$ns duplex-link-op $n1 $n2 orient right

$ns duplex-link-op $n2 $n3 orient up

$ns duplex-link-op $n1 $n4 orient up-left

$ns duplex-link-op $n4 $n3 orient right

$ns duplex-link-op $n3 $n5 orient left-up

$ns duplex-link-op $n4 $n5 orient right-up

#Setup a TCP connection

set tcp [new Agent/TCP/Newreno]

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink/DelAck]

$ns attach-agent $n5 $sink

$ns connect $tcp $sink

$tcp set fid\_ 1

#Setup a FTP over TCP connection set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type\_ FTP

$ns rtmodel-at 1.0 down $n0 $n4

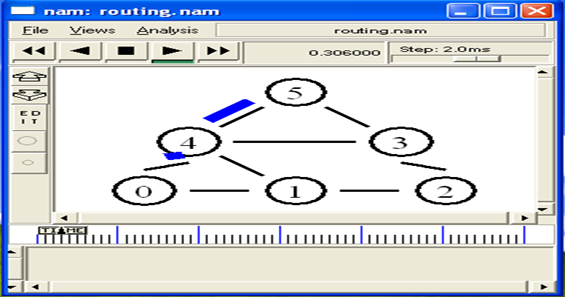
$ns rtmodel-at 4.5 up $n0 $n4

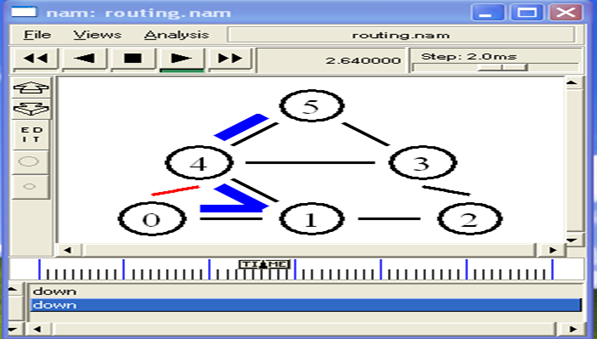
$ns at 0.1 "$ftp start"

$ns at 6.0 "finish"

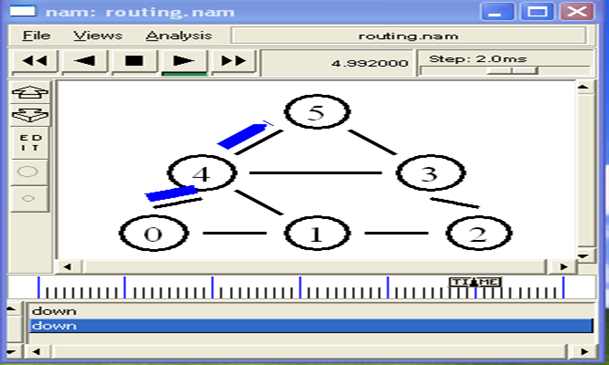
$ns run

## Output: UP:



**Down:**

## UP:



**Result:**.