**Ex.no.11 UNICAST ROUTING PROTOCOL**

Aim:

To write a ns2 program for implementing unicast routing protocol.

Algorithm:

Step 1: start the program.

Step 2: declare the global variables ns for creating a new simulator.

Step 3: set the color for packets.

Step 4: open the network animator file in the name of file2 in the write mode.

Step 5: open the trace file in the name of file 1 in the write mode.

Step 6: set the unicast routing protocol to transfer the packets in network.

Step 7: create the required no of nodes.

Step 8: create the duplex-link between the nodes including the delay time,bandwidth and dropping

. queue mechanism.

Step 9: give the position for the links between the nodes.

Step 10: set a tcp reno connection for source node.

Step 11: set the destination node using tcp sink.

Step 12: setup a ftp connection over the tcp connection.

Step 13: down the connection between any nodes at a particular time.

Step 14: reconnect the downed connection at a particular time.

Step 15: define the finish procedure.

Step 16: in the definition of the finish procedure declare the global variables ns,file1,file2.

Step 17: close the trace file and namefile and execute the network animation file.

Step 18: at the particular time call the finish procedure.

Step 19: stop the program.

**Program:**

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 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 3

}

# 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 n4 [$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

$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 4.5 up $n1 $n4

$ns at 0.1 "$ftp start"

$ns at 6.0 "finish"

$ns run