**SOLUTION 1**

**#Create a simulator object**

set ns [new Simulator]

**#Open the NAM trace file**

set nf [open out.nam w]

$ns namtrace-all $nf

**#Define a 'finish' procedure**

proc finish {} {

global ns nf

$ns flush-trace

#Close the NAM trace file

close $nf

#Execute NAM on the trace file

exec nam out.nam &

exit 0

}

**# Set routing protocol**

$ns rtproto DV

**#Creation of Nodes**  
for {set i 0} {$i <7} {incr i}{

set n($i) [$ns node] }

**#Setting Parameters of Nodes**

$n3 shape box

$n2 shape box

**#Creation of Links**  
$ns duplex-link $n0 $n2 2Mb 20ms DropTail  
$ns duplex-link $n5 $n2 2Mb 20ms DropTail  
$ns duplex-link $n1 $n2 2Mb 20ms DropTail  
$ns duplex-link $n2 $n3 10Mb 40ms DropTail  
$ns duplex-link $n3 $n6 5Mb 15ms DropTail  
$ns duplex-link $n3 $n4 5Mb 15ms DropTail  
$ns duplex-link $n3 $n7 5Mb 15ms DropTail

**#Orientation**  
$ns duplex-link-op $n0 $n2 orient right-down  
$ns duplex-link-op $n5 $n2 orient right  
$ns duplex-link-op $n1 $n2 orient right-up  
$ns duplex-link-op $n2 $n3 orient center  
$ns duplex-link-op $n3 $n6 orient right-up  
$ns duplex-link-op $n3 $n4 orient right  
$ns duplex-link-op $n3 $n7 orient right-down

**#Limiting the queue to only 30 packets**

$ns queue-limit $n2 $n3 30

#**Connection and Agents**  
set tcp0 [new Agent/TCP]  
$ns attach-agent $n0 $tcp0  
set ftp [new Application/FTP]  
$ftp attach-agent $tcp0

set tcp1 [new Agent/TCP]  
$ns attach-agent $n1 $tcp1  
set ftp1 [new Application/FTP]  
$ftp1 attach-agent $tcp1

set udp5 [new Agent/UDP]  
$ns attach-agent $n5 $udp5  
set cbr5 [new Application/Traffic/CBR]  
$cbr5 attach-agent $udp5

$cbr set type\_ CBR

$cbr set packet\_size\_

$cbr set rate\_

**#Attach Sink and Null nodes**

set sink6 [new Agent/TCPSink]  
$ns attach-agent $n6 $sink6

set sink7 [new Agent/TCPSink]  
$ns attach-agent $n7 $sink7

set null4 [new Agent/Null]  
$ns attach-agent $n4 $null4

**#connections**

$ns connect $tcp0 $sink7  
$ns connect $tcp1 $sink6

$ns connect $udp5 $null4

**#Schedule events for the CBR and FTP agents**  
$ns at 1.5 “$ftp start”  
$ns at 4.3 “$ftp stop”  
$ns at 2.5 “$ftp1 start”  
$ns at 7.3 “$ftp1 stop”  
$ns at 2.0 “$cb1 start”  
$ns at 6.5 “$cb1 stop”

**#Link down and up**

$ns rtmodel-at 1.3 down $n3 $n4

$ns rtmodel-at 2.5 up $n3 $n4

$ns at 12.0 “finish”  
$ns run

**SOLUTION 2 - MCQS**

1. \_\_\_\_\_\_\_\_ tool that allows visualization of nodes.
2. Network animator
3. Trace Files
4. Logs
5. None
6. To take input in a variable **\_\_\_\_\_\_** is used.
7. puts stdin
8. set
9. gets stdin
10. execute
11. \_\_\_\_\_\_\_ procedure is call at the end of the simulation.
12. proc
13. finish
14. ns
15. none
16. \_\_\_\_\_\_\_ sets up simulation by assembling and configuring the objects as well as scheduling discrete events.
17. C++
18. C
19. OTcl
20. Nam
21. In given statement what does 10ms and 10Mb represents.

**$ns duplex-link $n0 $n2 10Mb 10ms DropTail**

1. Capacity, Propagation Delay
2. Propagation delay, Capacity
3. Queue size, Transmission Delay
4. Transmission delay, Capacity
5. Is it possible to send different packets to different nodes through one single node using Ns2?
6. Yes
7. No
8. \_\_\_\_\_\_ queuing mechanism uses hashing algorithm to divides the traffic over a limited number of queues.
9. Fair queue
10. Drop Tail
11. SFQ
12. Round Robin
13. In distance vector routing, each router must know the whole network topology.
14. True
15. False
16. Data flow in a computer ring network topology is
17. Uni-directional
18. Bi-directional
19. Simplex
20. Duplex
21. A router using distance-vector routing protocol sends only new information to other routers on the network.
22. True
23. False