**Q. Write a TCL script for connecting two nodes using UDP protocol**

**Solution :**

***#1 Create a simulator object***

set ns [new Simulator]

***#2 Open trace file for event trace***

set nf [open out.tr w]

$ns trace-all $nf

***#3 Open NAM file for visualizing event***

set f [open out.nam w]

$ns namtrace-all $f

***#4 Create two nodes***

set n0 [$ns node]

set n1 [$ns node]

***#5 Create duplex-link of 10Mb bandwidth, 10ms propagation delay and Droptail queueing sequence(FIFO)***

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

***#6 Create and attach UDP agent at n0***

set udp [new Agent/UDP]

$ns attach-agent $n0 $udp

***#7 Create and attach null agent at n1***

set null [new Agent/Null]

$ns attach-agent $n1 $null

***#8 Connect agents***

$ns connect $udp $null

***#9 Create and attach CBR(cost and bit rate) traffic with udp***

set cbr [new Application/Traffic/CBR]

$cbr set packetSize\_ 100

$cbr set interval\_ 0.005

$cbr attach-agent $udp

***#10 Define finish process***

proc finish {} {

global ns nf f

$ns flush-trace

close $nf

close $f

exec nam out.nam &

exit 0

}

***#11 Schedule events for CBR***

$ns at 0.1 "$cbr start"

$ns at 3.5 "$cbr stop"

***#12 Call the finish procedure after 5 seconds of simulation time***

$ns at 5.0 "finish"

***#13 Run the simulation***

$ns run

**Q. Write a TCL script implementing ring topology using UDP protocol**

**Solution :**

***#1 Create a simulator object***

set ns [new Simulator]

***#2 Tell the simulator to use dynamic routing***

$ns rtproto DV

***#3 Open the nam trace file***

set nf [open out.nam w]

$ns namtrace-all $nf

***#4 Define a 'finish' procedure***

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

***#5 Create seven nodes***

for {set i 0} {$i < 7} {incr i} {

set n($i) [$ns node]

}

***#6 Create duplex-links between the nodes***

for {set i 0} {$i < 7} {incr i} {

$ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail

}

***#7 Create a UDP agent and attach it to node n(0)***

set udp0 [new Agent/UDP]

$ns attach-agent $n(0) $udp0

***#8 Create a CBR traffic source and attach it to udp0***

set cbr0 [new Application/Traffic/CBR]

$cbr0 set packetSize\_ 500

$cbr0 set interval\_ 0.005

$cbr0 attach-agent $udp0

***#9 Create a Null agent (a traffic sink) and attach it to node n(3)***

set null0 [new Agent/Null]

$ns attach-agent $n(3) $null0

***#10 Connect the traffic source with the traffic sink***

$ns connect $udp0 $null0

***#11 Schedule events for the CBR agent and the network dynamics***

$ns at 0.5 "$cbr0 start"

$ns rtmodel-at 1.0 down $n(1) $n(2)

$ns rtmodel-at 2.0 up $n(1) $n(2)

$ns at 4.5 "$cbr0 stop"

***#12 Call the finish procedure after 5 seconds of simulation time***

$ns at 5.0 "finish"

***#13 Run the simulation***

$ns run

**Q. Write a TCL script implementing star topology using UDP protocol**

**Solution :**

***#1 Create a simulator object***

set ns [new Simulator]

$ns color 1 Red

$ns color 2 Blue

***#2 Open trace file for event trace***

set f [open out.tr w]

$ns trace-all $f

***#3 Open NAM file for visualizing event***

set nf [open out.nam w]

$ns namtrace-all $nf

***#4 Create four nodes***

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

***#5 Create duplex-links***

$ns duplex-link $n0 $n2 1Mb 10ms DropTail

$ns duplex-link $n1 $n2 1Mb 10ms DropTail

$ns duplex-link $n2 $n3 1Mb 10ms SFQ

$ns duplex-link-op $n0 $n2 orient right-down

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

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

***#6 Create and attach UDP agent at n0***

set udp0 [new Agent/UDP]

$udp0 set class\_ 1

$ns attach-agent $n0 $udp0

***#7 Create and attach CBR(cost and bit rate) traffic with udp0***

set cbr [new Application/Traffic/CBR]

$cbr set packetSize\_ 500

$cbr set interval\_ 0.005

$cbr attach-agent $udp0

***#8 Create and attach UDP agent at n1***

set udp1 [new Agent/UDP]

$udp1 set class\_ 2

$ns attach-agent $n1 $udp1

***#9 Create and attach CBR(cost and bit rate) traffic with udp1***

set cbr1 [new Application/Traffic/CBR]

$cbr1 set packetSize\_ 500

$cbr1 set interval\_ 0.005

$cbr1 attach-agent $udp1

***#10 Create and attach UDP agent at n2***

set udp2 [new Agent/UDP]

$ns attach-agent $n2 $udp2

***#11 Create and attach CBR(cost and bit rate) traffic with udp2***

set cbr2 [new Application/Traffic/CBR]

$cbr2 set packetSize\_ 500

$cbr2 set interval\_ 0.005

$cbr2 attach-agent $udp2

***#12 Create and attach null agent at n3***

set null0 [new Agent/Null]

$ns attach-agent $n3 $null0

***#13 Connect agents***

$ns connect $udp0 $null0

$ns connect $udp1 $null0

$ns connect $udp2 $null0

***#14 Define finish process***

proc finish {} {

global ns nf f

$ns flush-trace

close $nf

close $f

exec nam out.nam &

exit 0

}

***#15 Schedule events for CBRs***

$ns at 0.5 "$cbr start"

$ns at 3.0 "$cbr stop"

$ns at 0.5 "$cbr1 start"

$ns at 3.0 "$cbr1 stop"

$ns at 0.5 "$cbr2 start"

$ns at 3.0 "$cbr2 stop"

***#16 Call the finish procedure after 5 seconds of simulation time***

$ns at 5.0 "finish"

***#17 Run the simulation***

$ns run

**Q. Write a TCL script for connecting two nodes using TCP protocol**

**Solution** :

***#1 Create a simulator object***

set ns [new Simulator]

***#2 Open trace file for event trace***

set nf [open out.tr w]

$ns trace-all $nf

***#3 Open NAM file for visualizing event***

set f [open out.nam w]

$ns namtrace-all $f

***#4 Create four nodes***

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

***#5 Create duplex-links***

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

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

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

$ns duplex-link $n3 $n1 10Mb 10ms DropTail

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

***#6 Create and attach TCP sender agent at n0***

set tcp [new Agent/TCP]

$ns attach-agent $n0 $tcp

***#7 Create and attach TCP receiver agent at n3***

set sink [new Agent/TCP]

$ns attach-agent $n3 $sink

***#8 Create and attach CBR(cost and bit rate) traffic with tcp***

set cbr [new Application/Traffic/CBR]

$cbr set packetSize\_ 500

$cbr set interval\_ 0.005

$cbr attach-agent $tcp

***#9 Connect agents***

$ns connect $tcp $sink

***#10 Define finish process***

proc finish {} {

global ns nf f

$ns flush-trace

close $nf

close $f

exec nam out.nam &

exit 0

}

***#11 Schedule events for CBRs***

$ns at 0.5 "$cbr start"

$ns at 3.0 "$cbr stop"

***#12 Call the finish procedure after 5 seconds of simulation time***

$ns at 5.00 "finish"

***#13 Run the simulation***

$ns run