

WEEK 7

Program -1:

AIM:

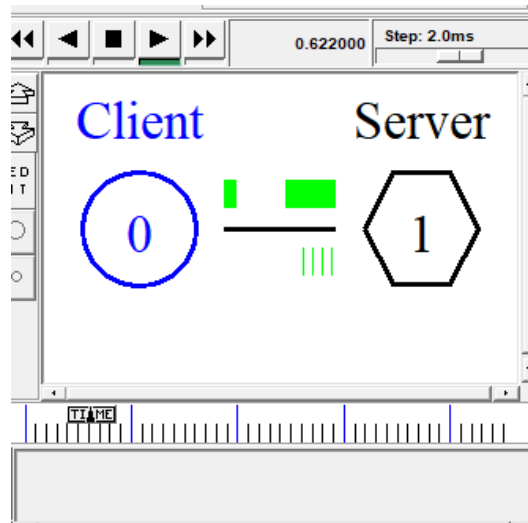
To write a TCL program to implement data transfer between client and server.

Program:

```
set cs [new Simulator]
set namf [open pg1.nam w]
$cs namtrace-all $namf
set cst [open pg1.tr w]
set client [$cs node]
$cs at 0.0 "$client label Client"
set server [$cs node]
$cs at 0.0 "$server label Server"
$cs color 1 green
$client color Blue
$server shape hexagon
$cs duplex-link $client $server 10Mbps 10ms DropTail
$cs duplex-link-op $server $client orient left
proc finish { } {
    global cs namf cst
    $cs flush-trace
    close $namf
    exec nam pg1.nam &
    exit 0
}
set tcp [new Agent/TCP]
$cs attach-agent $client $tcp
set sink [new Agent/TCPSink]
$cs attach-agent $server $sink
$cs connect $tcp $sink
$tcp set fid_ 1
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$cs at 0.1 "$ftp start"
$cs at 4.5 "$ftp stop"
$cs at 5.0 "finish"
```

\$cs run

Output:



Program -2:

AIM:

To write a TCL program to establish 3 nodes client, server, router and implement data transfer between client and server using the router.

Program:

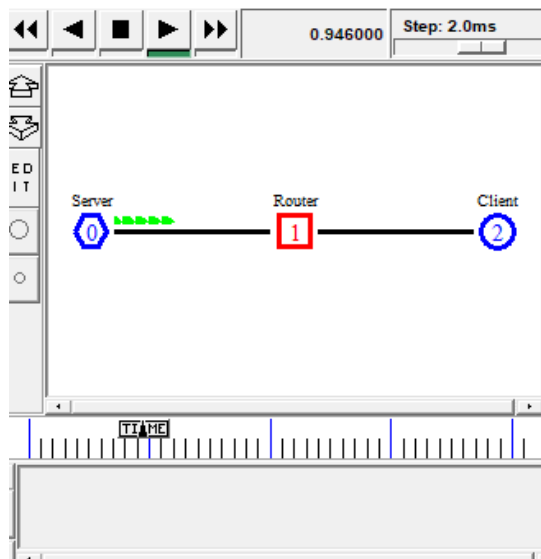
```
set ns [new Simulator]
set nt [open pg2.tr w]
$ns trace-all $nt
set nf [open pg2.nam w]
$ns namtrace-all $nf
$ns color 1 red
$ns color 2 yellow
$ns color 3 blue
$ns color 4 green
$ns color 5 black
set totalNodes 3
for {set i 0} {$i < $totalNodes} {incr i} {
    set node_($i) [$ns node]
}
set server 0
set router 1
set client 2
$ns duplex-link $node_($server) $node_($router) 2Mb 50ms DropTail
$ns duplex-link $node_($router) $node_($client) 2Mb 50ms DropTail
$ns duplex-link-op $node_($server) $node_($router) orient right
$ns duplex-link-op $node_($router) $node_($client) orient right
```

```

$ns at 0.0 "$node_($server) label Server"
$ns at 0.0 "$node_($router) label Router"
$ns at 0.0 "$node_($client) label Client"
$ns at 0.0 "$node_($server) color blue"
$ns at 0.0 "$node_($client) color blue"
$ns at 0.0 "$node_($router) color red"
$node_($server) shape hexagon
$node_($router) shape square
set tcp [new Agent/TCP]
$ns attach-agent $node_($server) $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $node_($client) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$tcp set fid_ 4
$ns at 0.1 "$ftp start"
$ns at 4.0 "$ftp stop"
proc finish {} {
    global ns nf nt
    $ns flush-trace
    close $nf
    close $nt
    exec nam pg2.nam &
    exit 0
}
$ns at 10.0 "finish"
$ns run

```

Output:



Program -3:

AIM:

To write a TCL program to implement ring topology and to transfer data between any two nodes.

Program:

```
set cs [new Simulator]
set namf [open pg3.nam w]
$cs namtrace-all $namf
set cst [open pg3.tr w]
$cs color 1 blue
$cs color 2 yellow
$cs color 3 pink
set n0 [$cs node]
set n1 [$cs node]
set n2 [$cs node]
set n3 [$cs node]
$cs at 0.0 "$n0 label client"
$cs at 0.0 "$n1 label temp1"
$cs at 0.0 "$n2 label temp2"
$cs at 0.0 "$n3 label server"
$n1 color green
$n2 color red
$cs duplex-link $n0 $n1 10Mbps 10ms DropTail
$cs duplex-link-op $n0 $n1 orient right-up
$cs duplex-link $n0 $n2 10Mbps 10ms DropTail
$cs duplex-link-op $n0 $n2 orient right-down
$cs duplex-link $n1 $n3 10Mbps 10ms DropTail
$cs duplex-link-op $n1 $n3 orient right-down
$cs duplex-link $n2 $n3 10Mbps 10ms DropTail
$cs duplex-link-op $n2 $n3 orient right-up
set udp1 [new Agent/UDP]
$cs attach-agent $n0 $udp1
$udp1 set class_ 1
set udp2 [new Agent/UDP]
$cs attach-agent $n2 $udp2
$udp2 set class_ 2
set sink1 [new Agent/Null]
set sink2 [new Agent/Null]
$cs attach-agent $n3 $sink1
$cs attach-agent $n1 $sink2
$cs connect $udp1 $sink1
```

```

$cs connect $udp2 $sink2
set ftp1 [new Application/Traffic/CBR]
$ftp1 attach-agent $udp1
set ftp2 [new Application/Traffic/CBR]
$ftp2 attach-agent $udp2
$cs color 1 Blue
$cs color 2 yellow
proc finish { } {
  global cs namf cst
  $cs flush-trace
  close $namf
  exec nam pg3.nam &
  exit 0
}
$cs at 0.1 "$ftp1 start"
$cs at 0.1 "$ftp2 start"
$cs at 2.0 "$ftp1 stop"
$cs at 2.0 "$ftp2 stop"
$cs at 5.0 "finish"
$cs run

```

Output:

