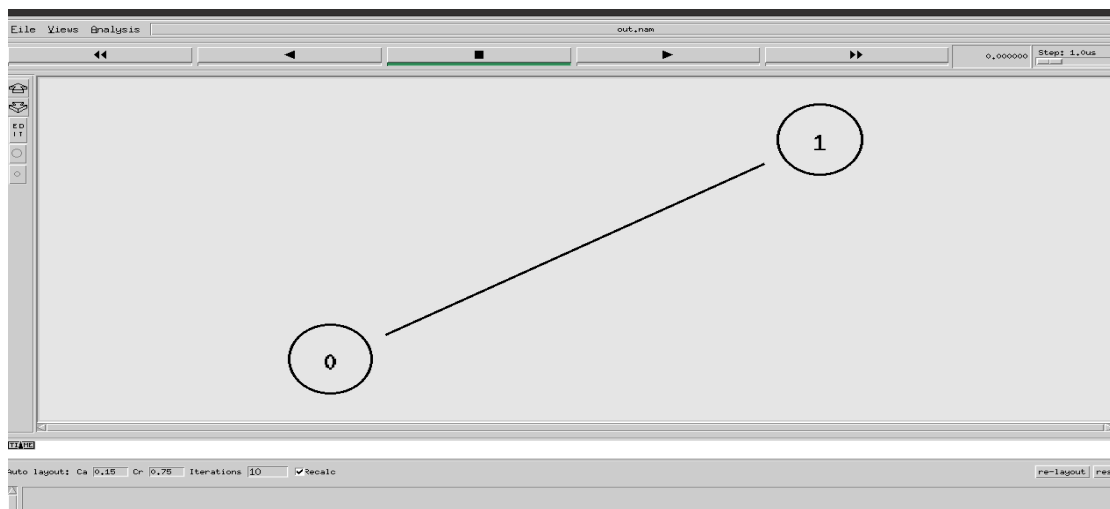


Example 01

Code:

```
#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 trace file
    close $nf
    #Execute nam on the trace file
    exec nam out.nam&
    exit 0
}
#Create two nodes
set n0 [$ns node]
set n1 [$ns node]
#Create a duplex link between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

Output:



Example 02

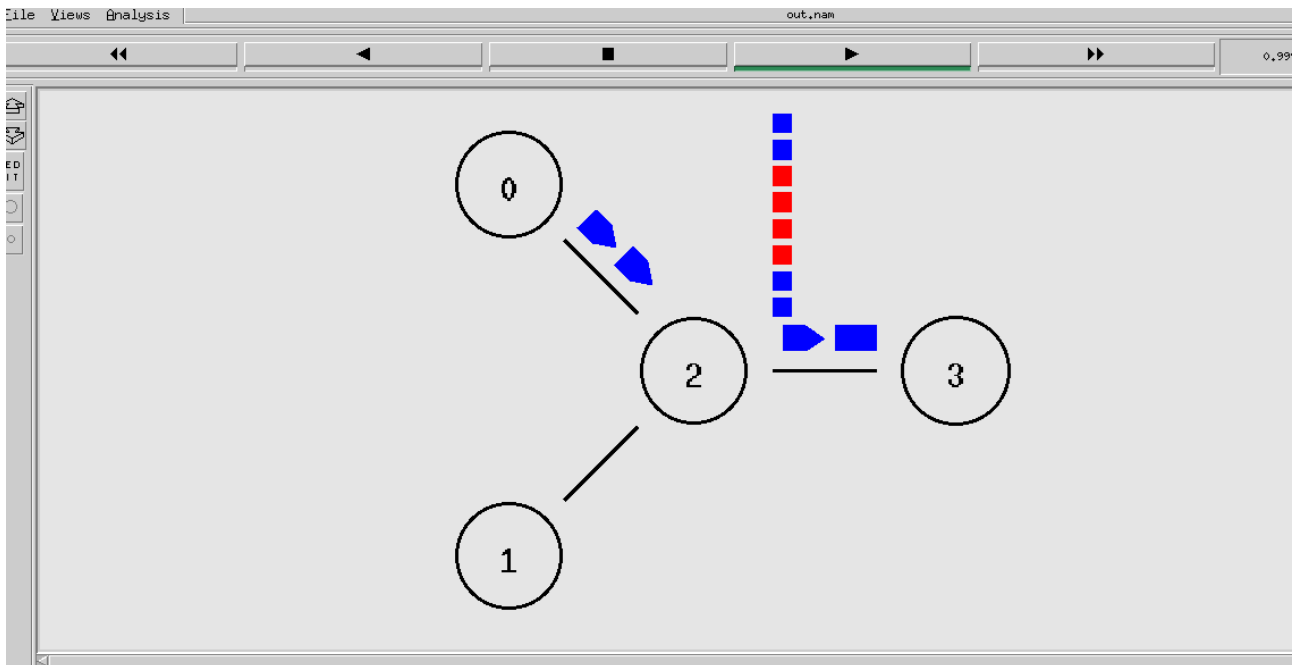
Build a script to simulate following network simulator scenario:

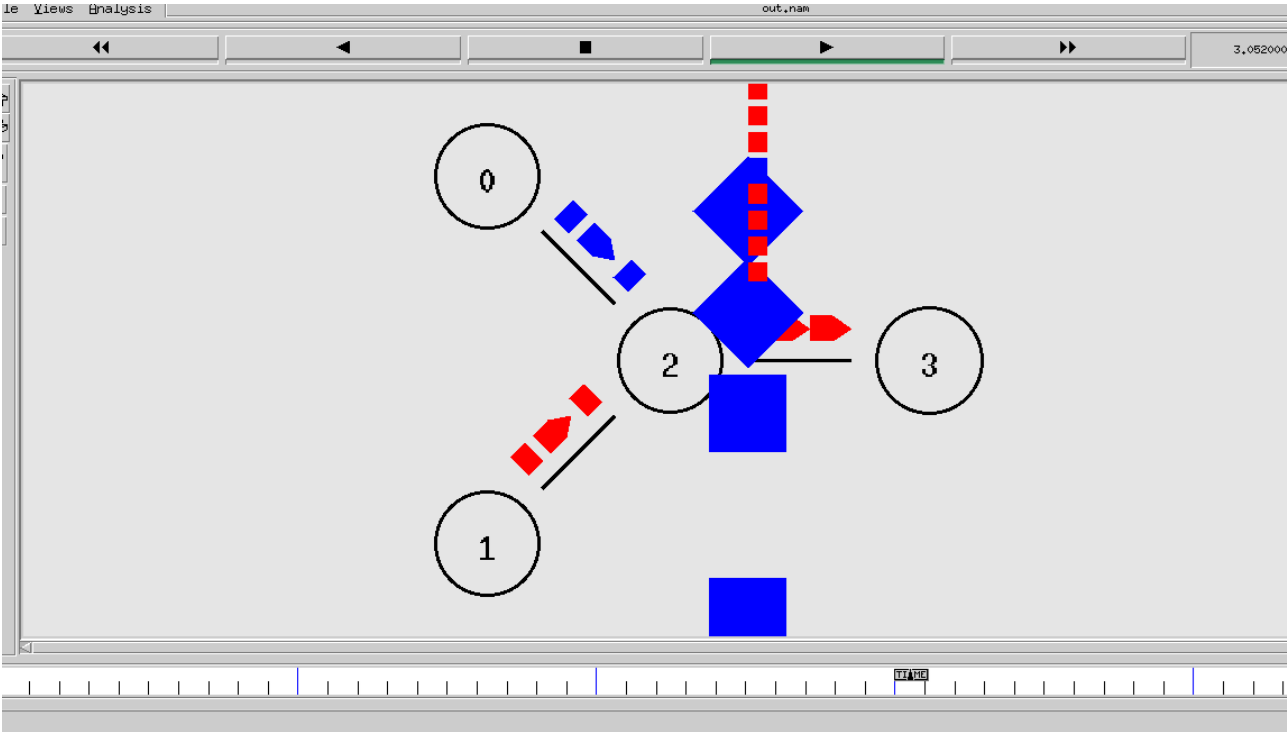
Code:

```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows
$ns color 1 Blue
$ns color 2 Red
#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 trace file
    close $nf
    #Execute nam on the trace file
    exec nam out.nam&
    exit 0
}
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#Create links between the nodes
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail
$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
#Monitor the queue for the link between node 2 and node 3
$ns duplex-link-op $n2 $n3 queuePos 0.5
#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0
# 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
#Create a UDP agent and attach it to node n1
set udp1 [new Agent/UDP]
$udp1 set class_ 2
$ns attach-agent $n1 $udp1
# Create a CBR traffic source and attach it to udp1
set cbr1 [new Application/Traffic/CBR]
```

```
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
#Create a Null agent (a traffic sink) and attach it to node n3
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
#Connect the traffic sources with the traffic sink
$ns connect $udp0 $null0
$ns connect $udp1 $null0
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

Output





Exercise 1:-

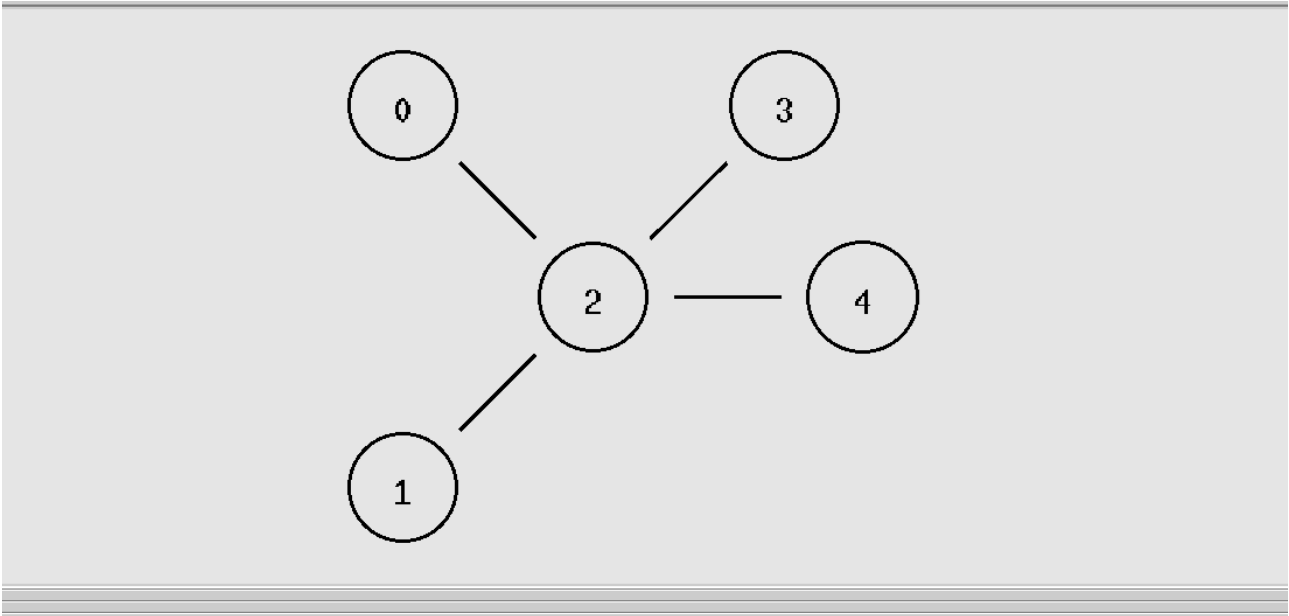
Build a script to simulate following network simulator scenario. Apply TCP/FTP at node 1, 3 and UDP/CBR at node 5.

Code:

```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows
$ns color 1 Blue
$ns color 2 Red
#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 trace file
    close $nf
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
#Create links between the nodes
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail
$ns duplex-link $n4 $n2 1Mb 10ms DropTail
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n4 orient right
$ns duplex-link-op $n3 $n2 orient left-down
#Monitor the queue for the link between node 2 and node 3
$ns duplex-link-op $n2 $n4 queuePos 0.5
#Create a UDP agent and attach it to node n3
set udp [new Agent/UDP]
$udp set class_ 1
$ns attach-agent $n3 $udp
# Createa CBR traffic source and attach it to udp
set cbr [new Application/Traffic/CBR]
$cbr set packetSize_ 500
$cbr set interval_ 0.005
$cbr attach-agent $udp
# Setup a TCP connection
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n0 $tcp0
```

```
set sink [new Agent/TCPSink]
$ns attach-agent $n0 $sink
$ns connect $tcp0 $sink
$tcp0 set fid_ 1
# Setup a FTP over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set type_ FTP
# Setup a TCP connection
set tcp1 [new Agent/TCP]
$tcp1 set class_ 1
$ns attach-agent $n1 $tcp1
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp1 $sink
$tcp1 set fid_ 1
# Setup a FTP over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set type_ FTP
#Create a Null agent (a traffic sink) and attach it to node n4
set null0 [new Agent/Null]
$ns attach-agent $n4 $null0
#Connect the traffic sources with the traffic sink
$ns connect $tcp0 $null0
$ns connect $tcp1 $null0
$ns connect $udp $null0
#Schedule events for the CBR agents
#$ns at 0.5 "$cbr start"
$ns at 0.5 "$ftp0 start"
$ns at 1.0 "$ftp1 start"
$ns at 1.5 "$ftp1 stop"
$ns at 2.0 "$ftp0 stop"
#$ns at 5.5 "$cbr stop"
#Call the finish procedure after 5 seconds of simulation time
$ns at 3.0 "finish"
#Run the simulation
$ns run
```

Output:



Exercise 2:-**Build a script to simulate following network simulator scenario*****Code:***

```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green
#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 trace file
    close $nf
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
#Create links between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n0 $n6 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n6 $n5 1Mb 10ms DropTail
$ns duplex-link $n5 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n3 1Mb 10ms DropTail
$ns duplex-link-op $n0 $n1 orient right-up
$ns duplex-link-op $n0 $n6 orient left-up
$ns duplex-link-op $n1 $n2 orient down
$ns duplex-link-op $n2 $n3 orient down
$ns duplex-link-op $n6 $n5 orient down
$ns duplex-link-op $n5 $n4 orient down
$ns duplex-link-op $n4 $n3 orient right
#Monitor the queue for the link between
$ns duplex-link-op $n0 $n1 queuePos 0.5
$ns duplex-link-op $n0 $n6 queuePos 0.5
$ns duplex-link-op $n1 $n2 queuePos 0.5
```



```
$ns duplex-link-op $n2 $n3 queuePos 0.5
$ns duplex-link-op $n6 $n5 queuePos 0.5
$ns duplex-link-op $n5 $n4 queuePos 0.5
$ns duplex-link-op $n4 $n3 queuePos 0.5
#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$udp0 set class_ 1
$ns attach-agent $n0 $udp0
# Createa 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
#Create a Null agent (a traffic sink) and attach it to node n3
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
#Connect the traffic sources with the traffic sink
$ns connect $udp0 $null0
#Schedule events for the CBR agents
$ns at 0.5 "$cbr0 start"
#Call the finish procedure after 5 seconds of simulation time
$ns at 3.0 "finish"
#Run the simulation
$ns run
```

Output:

