

Aim

To study the Network Simulator (NS2) using a simple tcl-script generated network schedule comprising of FTP traffic using TCP packets and a constant bitrate UDP packet traffic directed to the same node serving as a TCP sink agent and as a “null” agent

Question

Implement a simple tcl-script generated network schedule consisting of FTP traffic using TCP packets and a constant bitrate UDP packet traffic directed to the same node serving as a TCP sink agent and as a “null” agent. Study the network simulator (NS2) and the Network Animator (NAM) using this script

Algorithm

- Step 1:** Instantiate a new Simulator object. Define colors for different flows for visual distinction
- Step 2:** Set the simulator to record the simulations in the NAM format. Supply the output file name to be generated at the end
- Step 3:** Define four different nodes in the network simulator — n0, n1, n2 and n3. Create duplex-links between ns2 and every other node. Define the link bandwidth, propagation delay and the type of packet-queuing procedure to follow at the packet receiver end.
- Step 4:** Create and attach a TCP agent instance to node n0 and a TCP-sink agent instance to node n3 to receive these packets and send back an acknowledgement. Setup a logical connection between the two i.e set their destination IP and port addresses. Attach an FTP agent as the application layer protocol to the TCP agent.
- Step 5:** Create and attach a UDP agent instance to node n1 and a “null” agent instance to node n3 to digest these packets, serving as a network sink. Setup a logical connection between the two nodes. Attach a CBR agent over this UDP agent to ensure constant bitrate over this connection. Also set the packet size and rate parameters for this agent
- Step 6:** Schedule the CBR connection to operate from 0.1ns time instance to 4.5ns time instance. Schedule the FTP connection to operate from 1.0ns to 4.0 time instances.

Step 7: Terminate the simulation after 5ns. Flush the simulation data collected to the NAM file object created in step-2. Run the generated output trace file using the NAM simulator.

Step 8: Display the CBR packet size and interval on the screen.

TCL Program Code

1. tcp-udp-sink.tcl - Trace file generation script for simple network simulation

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$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 NAM 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 2Mb 10ms DropTail
```

```
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$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 link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5


#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $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


#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
# Create a logic connection between the two agents
# Done by assigning respective IPs mutually
$ns connect $udp $null
$udp set fid_ 2
```

```
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

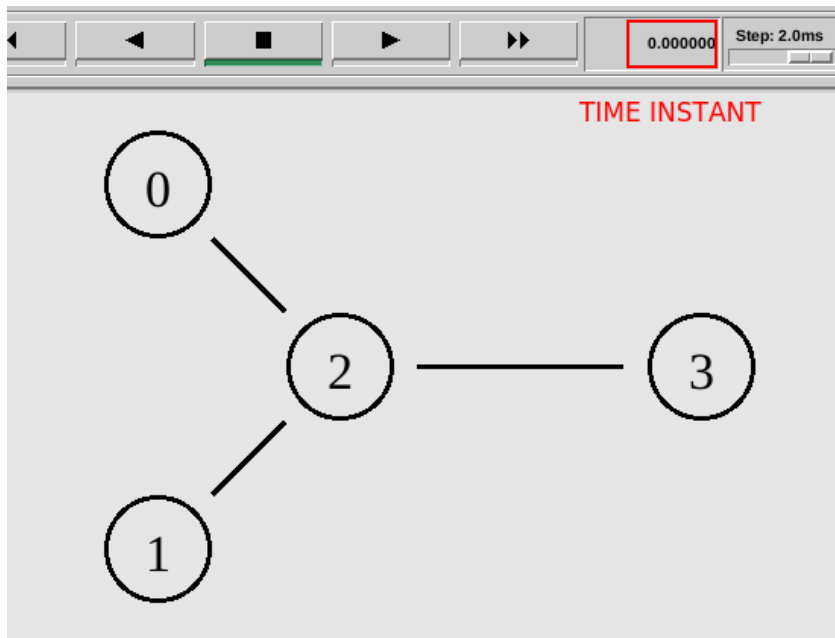
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

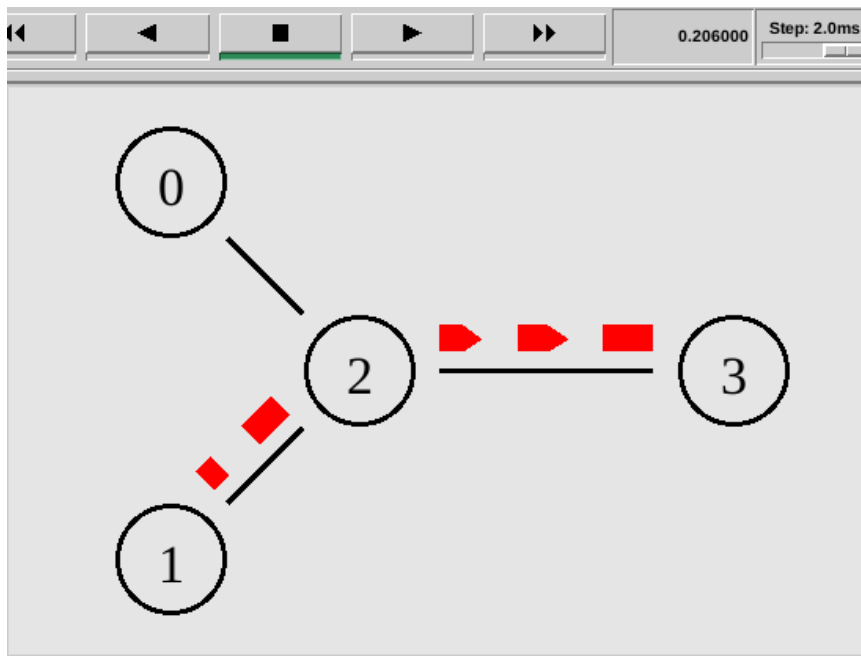
#Run the simulation
$ns run
```

Sample Output

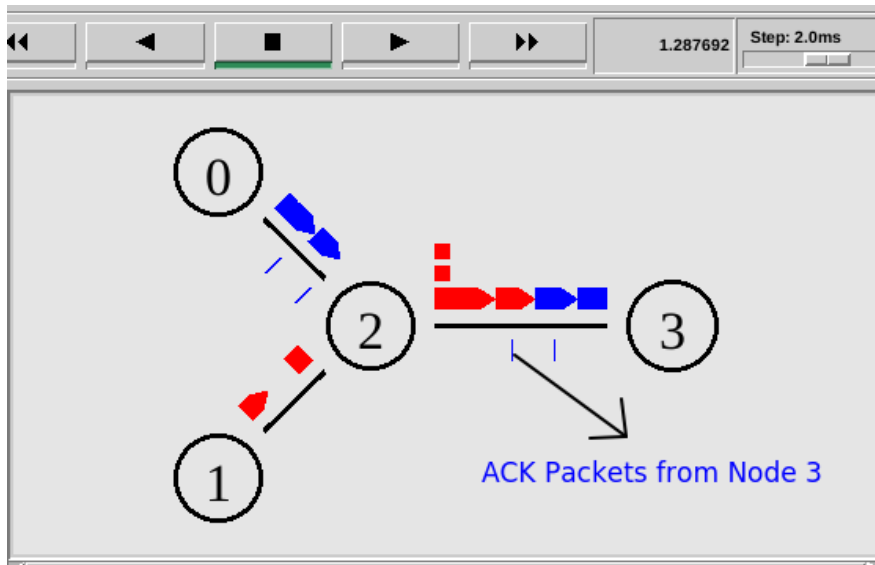
1. Initial State - No Traffic (Time 0.000)



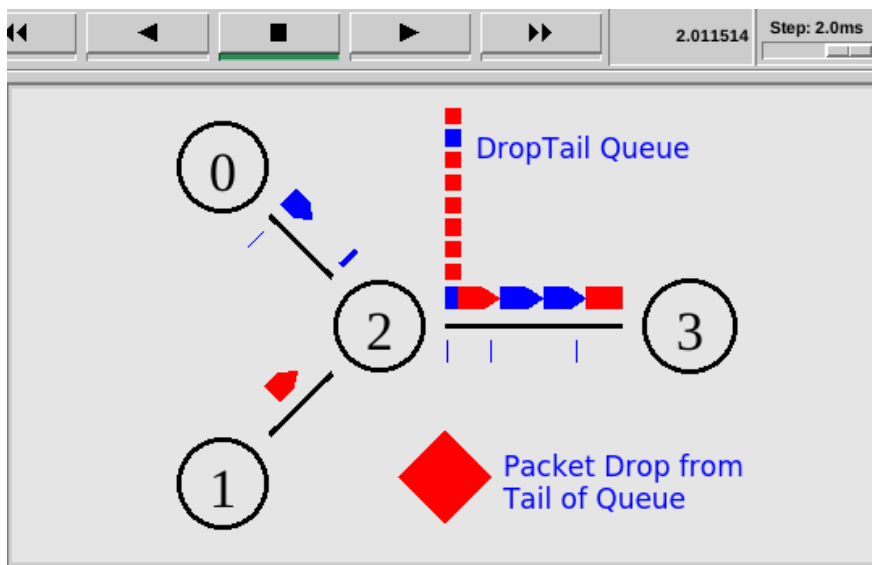
2. UDP Packet Transmission Ongoing (Time 0.206) - Started at Time 0.100



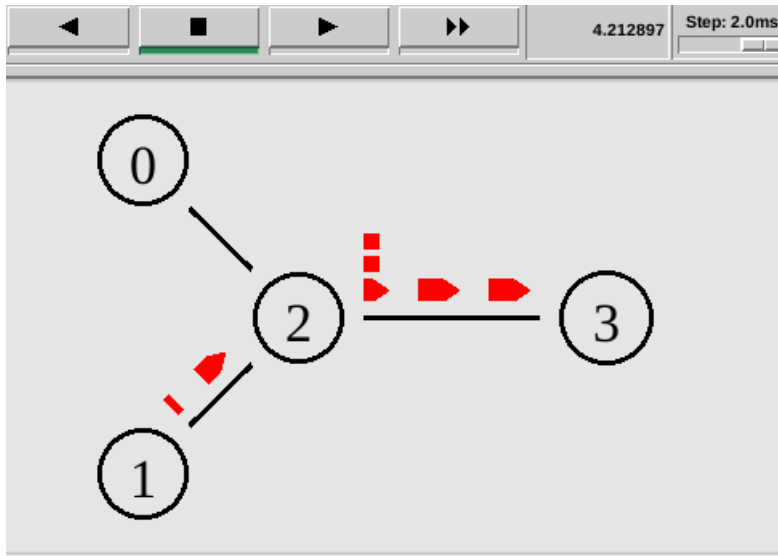
**3. TCP Packet Transmission Ongoing - ACK packets responded from Node 3
(Time 1.287) - Started at Time 1.000**



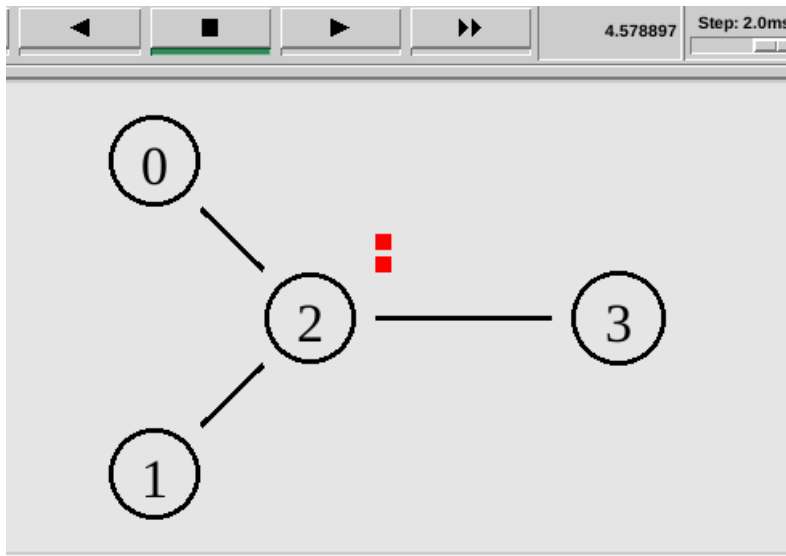
4. DropTail Queue - Packet Buffering - Dropped Packet when limit reached



5. TCP Packets Stopped (Time 4.212) - Stopped at Time 4.000



6. UDP Packets Stopped (Time 4.212) - Stopped at Time 4.500



Result

To study the Network Simulator (NS2) using a simple tcl-script generated network schedule consisting of FTP traffic using TCP packets and a constant bitrate UDP packet traffic directed to the same node serving as a TCP sink agent and as a “null” agent. Through this implementation, the following aspects were understood:

1. Basic scripting for NS and NAM using TCL language
2. Working with NS2 and NAM to observe network traffic conceptually