Simulation of Ethernet Lan

Experiment Specific Instructions

1. To analyze the given problem you have to write a Tcl script and simulate with ns2

2. Begin by specifying the trace files and the nam files to be created

3. Define a finish procedure

4. Determine and create the nodes that will be used to create the topology. Here in our

experiment we are selecting 6 nodes namely 0, 1, 2, 3, 4, 5

5. Create the links to connect the nodes

6. Set up the LAN by specifying the nodes, and assign values for bandwidth, delay,

queue type and channel to it

7. Optionally you can position and orient the nodes and links to view a nice video

output with Nam

8. Set up the TCP and/or UDP connection(s) and the FTP/CBR (or any other

application) that will run over it

9. Schedule the different events like simulation start and stop, data transmission start

and stop

10. Call the finish procedure and mention the time at what time your simulation will

end

11. Execute the script with ns

Simulation Script:
```
#Lan simulation
set ns [new Simulator]
#define color for data flows
$ns color 1 Blue
$ns color 2 Red
#open tracefiles
set tracefile1 [open out.tr w]
set winfile [open winfile w]
```

```
$ns trace-all $tracefile1
#open nam file
set namfile [open out.nam w]
$ns namtrace-all $namfile
#define the finish procedure
proc finish {} {
global ns tracefile1 namfile
$ns flush-trace
close $tracefile1
close $namfile
exec nam out.nam &
exit 0
} #create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color Red
$n1 shape box
#create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail
MAC/Csma/Cd Channel]
#Give node position
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
#set queue size of link(n2-n3) to 20
$ns queue-limit $n2 $n3 20
#setup TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set packet_size_ 552
#set ftp over tcp connection
```

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
#setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$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_ 0.01Mb
$cbr set random_ false
#scheduling the events
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 125.5 "$cbr stop"
proc plotWindow {tcpSource file} {
global ns
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 125.0 "finish"
$ns run
```