CNL Assignment B1 (NS2) **(31139-Durvesh)**

// Write-ups

Page

Assignment No. B1

- Aim: Study of any network simulation tools - To create a network with three nodes & establish a TCP connection between node 0 of node 1 such that node 0 will send TCP packet to node 2 via node 1.
- Objective: To use Network Simulator and demonstrate packet transfer of TCP.
- Outcomes: Students will be able to use Network Simulator to simulate network flows in various situations.
- H/W & S/W Requirements: Network Simulator 2, 64-bit OS, 8 GB RAM, 1 TB HDD.
- Theory:

Network Simulator (Version 2) widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communications networks. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool.

Concepts Overview: NS uses two languages because simulator has two different kinds of things it needs to do. On one hand, detailed simulations of protocols requires a systems programming language which can efficiently manipulate bytes. For these tasks run-time speed is important & turn around time is less important. On the other hand network research involves slightly varying parameters or configurations. Here, Iteration time is more important.

NS meets both the needs with two languages, C++ & OTcl.

Tcl scripting - Tcl is a general purpose scripting language.

Basics of TCL

Syntax : command arg1 arg2 arg3

Hello World!

puts stdout {Hello, World!} {Hello, World!}

Initialization & Termination of TCL script in NS-2

```
set ns [new Simulator]
# Open the Trace file
set tracefile1 [open out.tr w]
$ns trace-all $tracefile1
# Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
```

Define a "finish" procedure

```
Proc finish {} {
global ns tracefile1 namfile
$ns flush-trace
close $tracefile1
close $namfile
Exec nam out.nam &
Exit 0
}
```

Definition of a network of links and nodes

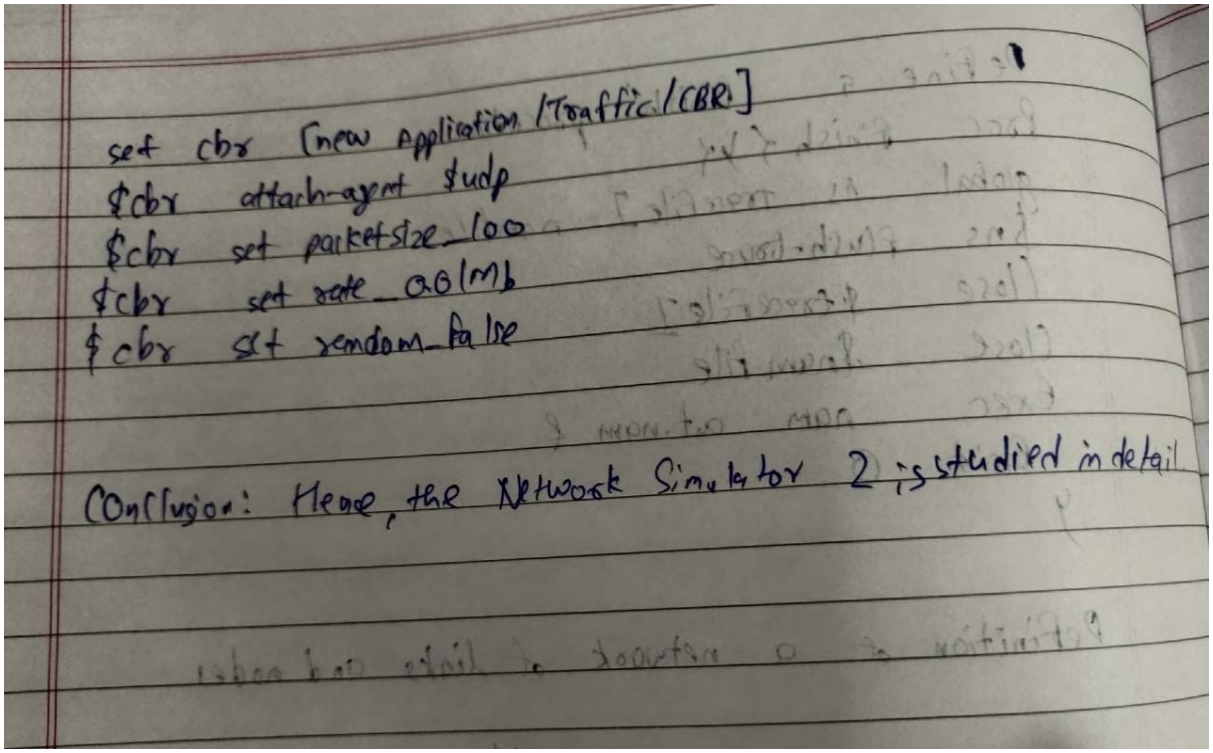
```
set n0 [$ns node]
$ns duplex-link $n0 $n2 10Mb 10ms DropTail

# set Queue Size of link (n0-n2) to 20
$ns queue-limit $n0 $n2 20
```

FTP over TCP

```
set tcp [new 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
```



// Sample Code

------b1.tcl

```
#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
$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
```