

**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT  
YELAHANKA, BENGALURU - 560064**

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

**Report on P1**

<b>Name</b>	Raghavendra K M
<b>USN</b>	1BY18IS093
<b>Semester/Section</b>	5B
<b>Course Code</b>	18CSL57
<b>Course Name</b>	Computer Network Laboratory
<b>Faculty</b>	Prof. Gireesh babu C N
<b>Title</b>	P1
<b>Date</b>	16-11-2020

**Signature of a Student**

**Signature of a Faculty**

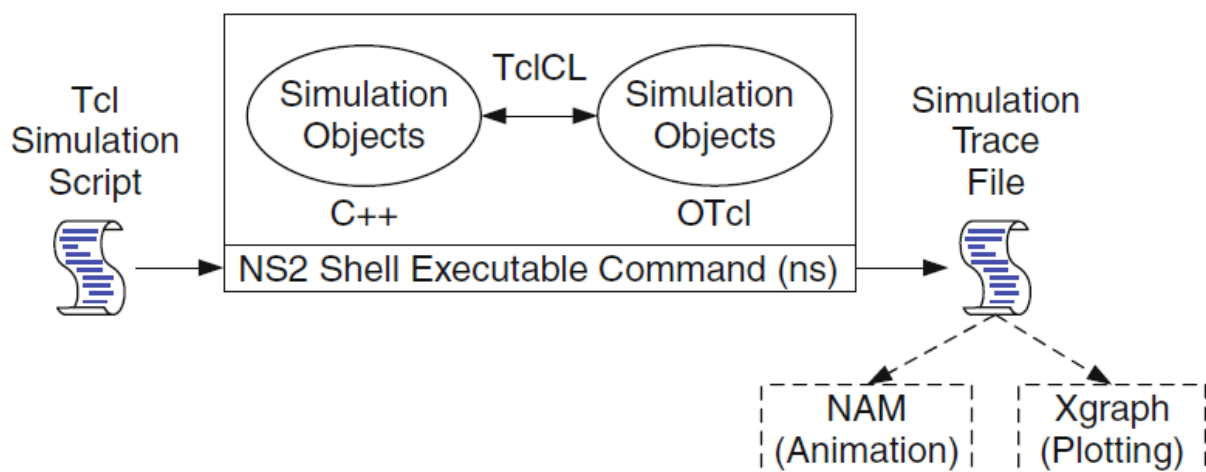
### Aim:

Implement three nodes point – to – point networks with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.

### Theory:

NS2 is simply an event driven simulation tool useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. Ns2 is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

### BASIC ARCHITECTURE OF NS2



- Tcl is a general-purpose scripting language. [Interpreter]
- Tcl runs on most of the platforms such as Unix, Windows, and Mac.
- The strength of Tcl is its simplicity.
- It is not necessary to declare a data type for variables prior to the usage.

### Steps:

- Open gedit editor and type program. Program name should have the extension “.tcl”
- [root@localhost ~] # gedit lab 1.tcl
- Save the program and close the file.
- Open gedit editor and type awk program. Program name should have the extension “.awk”
- [root@localhost ~]# gedit lab 1.awk

- Save the program and close the file.
- Run the simulation program.
- [root@localhost~]# ns lab 1.tcl
- Here “ns” indicates network simulator. We get the topology shown in the snapshot.
- Now press the play button in the simulation window and the simulation will begin.
- After simulation is completed run awk file to see the output ,
- [root@localhost~]# awk -f lab1.awk lab1.tr
- To see the trace file contents open the file as ,
- [root@localhost~]# gedit lab1.tr
- Trace file contains 12 columns:
- Event type, Event time, From Node, To Node, Packet Type, Packet Size, Flags (indicated by -----), Flow ID, Source address, Destination address, Sequence ID, Packet ID.

### Source code:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#=====
#      Simulation parameters setup
#=====
set val(stop)    10.0                                ;# time of simulation end

#=====
#      Initialization
#=====
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open p1.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open p1.nam w]
$ns namtrace-all $namfile

#=====
#      Nodes Definition
#=====
#Create 3 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
```

```

#=====
#           Links Definition
#=====
#Createlinks between nodes
$ns duplex-link $n0 $n1 10.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 1.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up

#=====
#           Agents Definition
#=====
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

#=====
#           Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"

#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile

```

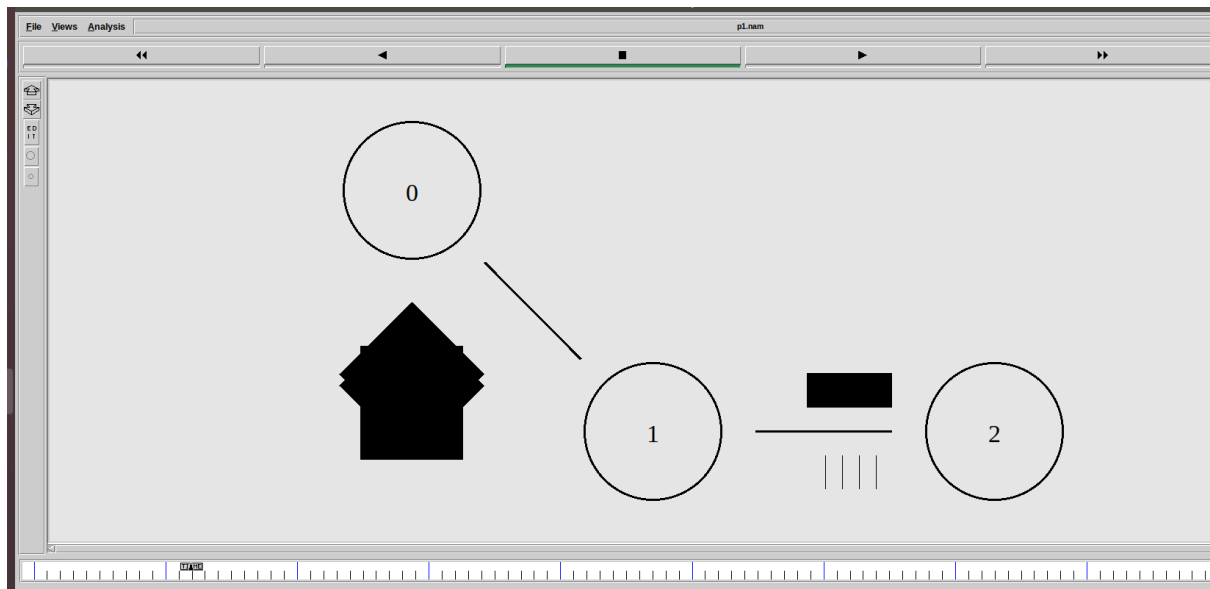
```

    close $namfile
    exec nam p1.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\\\" ; $ns halt"
$ns run

```

## Output:

1) Capacity = 10 mbps, Propagation delay = 10ms, Queue size =5 (both n0 to n1 and n1 to n2)

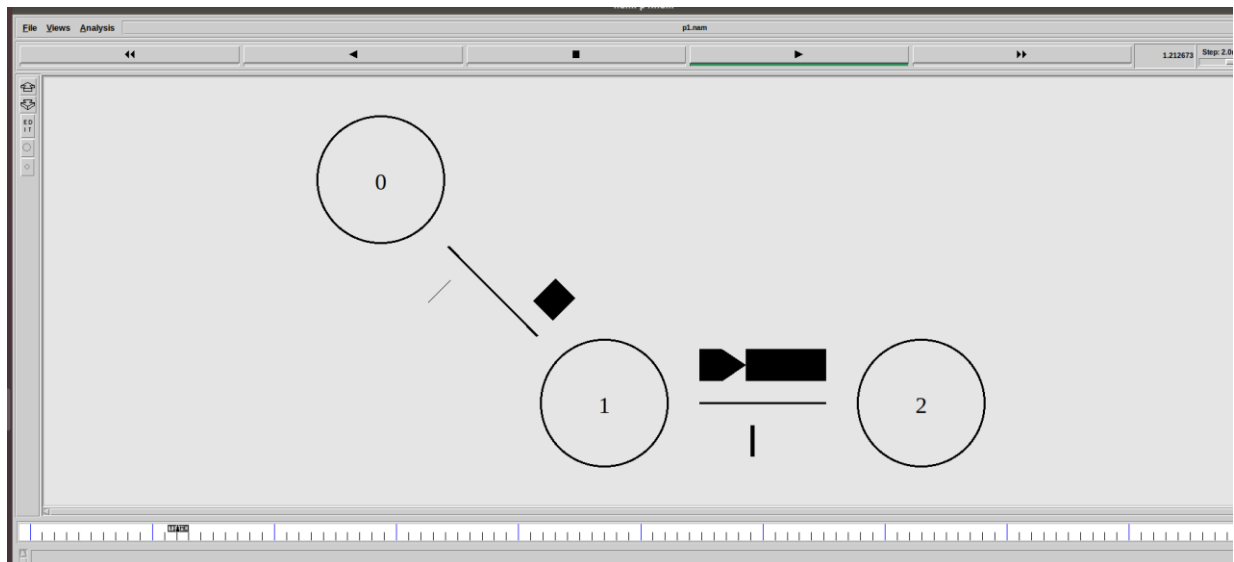


```

Activities  Terminal
rkm@rkm-VirtualBox:~/Downloads/cns/p1$ grep -c "^d" p1.tr
5
rkm@rkm-VirtualBox:~/Downloads/cns/p1$

```

2) Capacity = 10 mbps, Propagation delay = 10ms, Queue size = 50(both n0 to n1 and n1 to n2)



```
rkm@rkm-VirtualBox:~/Downloads/cns/p1$ grep -c "^d" p1.tr
0
rkm@rkm-VirtualBox:~/Downloads/cns/p1$ ns p1.tcl
```

3) Capacity = 100mbps, Propagation delay = 10 ms, Queue size = 5 (n0 to n1)  
 Capacity = 1mbps, Propagation delay = 10 ms, Queue size = 50 (n1 to n2)

```
rkm@rkm-VirtualBox:~/Downloads/cns/p1$ grep -c "^d" p1.tr
0
```