

## Experiment No: 3

**Aim:** Write TCL scripts to create topologies. Create and run traffics and analyze the result using NS2

**LO no:** 3,5

**LO statement:** Demonstrate and measure different network scenarios and their performance behavior. Analyze the traffic flow of different protocols

### Theory:

Simulation is the process of *learning by doing*. Whenever there is something new in the world, we try to analyze it first by examining it and in the process get to learn a lot of things. This entire course is called Simulation.

To create a node we can simply use the simulator method node. The following two lines create two nodes and assign them to the handles 'n0' and 'n1'.

```
set n0 [$ns node] set n1  
        [$ns node]
```

We can then either use the simulator method simplex-link or the method duplex-link to connect the nodes with a link:

```
$ns simplex-link $n0 $n1 1Mb 10ms DropTail $ns duplex-link  
$n0 $n1 1Mb 10ms DropTail
```

The first line creates a unidirectional link between n0 and n1 with bandwidth 1Mbps, a propagation delay of 10ms and a DropTail queue. The second line creates a bidirectional link with the same parameters.

Traffic generation in ns is based on objects of two classes, the Agent and the Application. Agents represent endpoints where network-layer packets are constructed or consumed. Every node in the network that needs to send or receive traffic must have an agent attached to it. These agents can be thought of as the implementation of the transport protocol. On top of that an agent runs an application. The application determines the kind of traffic source that is simulated (e.g. ftp or telnet). Applications represent the application layer in an ns-simulation.

### A. Creating Agents

Corresponding to the two most popular transport protocols used in the Internet there are also two types of agents in ns: UDP agents and TCP agents. The following code shows an example of attaching a UDP agent to nodes n0 and n1:

```
set udp0 [new Agent/UDP]
```

```
$ns attach-agent $n0 $udp0 set null0  
[new Agent/Null] $ns attach-agent $n1  
$null0 ns connect udp0 null0
```

This code first creates a UDP agent and attaches it to n0 using the attach-agent procedure. It then creates a Null agent which will act as a traffic sink and attaches it to n1. Finally, the two agents are connected using the simulator method connect. In the next section the UDP agent will be used by an application to send data. (c) Read the ns manual to find out how to write similar code to set up a TCP connection.

### **B. Creating Applications -**

In the previous section we have set up the agents implementing the transport layer. We will now create applications that we attach to the transport agents and that will actually generate traffic. In ns there are two basic types of applications: simulated applications and traffic generators. Traffic generators generate On/Off traffic: during On-periods, packets are generated at a constant burst rate and during Off-periods no packets are generated. ns provides three different classes of traffic generators which differ in how the lengths of the On and Off-periods are modeled:

1. A traffic generator of the type Application/Traffic/Exponential takes the length of the On and Off periods from an Exponential distribution.
2. A Application/Traffic/Pareto source generates the lengths of these periods from a Pareto distribution.
3. Finally, the class Application/Traffic/CBR has no off periods and generates packets at a constant bit rate. The following code generates one traffic generator of each class.

```
set exp [new set Application/Traffic/Exponential]  
par [new Application/Traffic/Pareto]  
set cbr0 [new Application/Traffic/CBR]
```

See the ns-manual for how to configure these traffic generators. All traffic generators run on top of a UDP agent. Therefore, we have to attach a traffic generator to a UDP agent before we can use it to send data. The following example illustrates the use of the CBR traffic generator that we created above.

```
$cbr0 set packetSize 500 $cbr0 set  
interval 0.005 $cbr0 attach-agent  
$udp0 $ns at 1.0 "$cbr start"
```

The simulated applications currently implemented in ns are Application/FTP and Application/Telnet. These try to simulate the corresponding applications in the real world: FTP and Telnet. Like the real applications the ns applications can run only on TCP. They therefore have to be attached to a TCP agent.

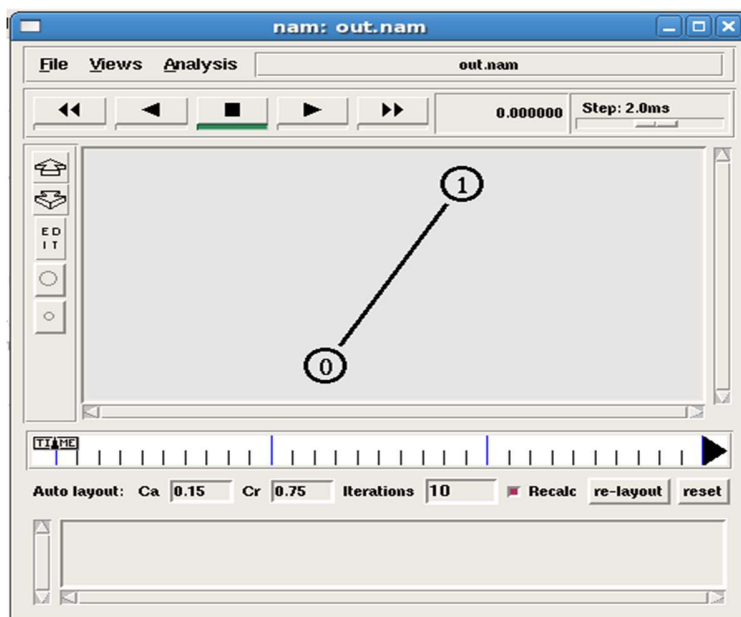
Name: Soham Desai  
XIE ID: 202003021

Roll no:11  
Batch: B

### Code:

```
set ns [new Simulator]
set f [open out.tr w]
$ns trace-all $f
set nr [open out.nam w]
$ns namtrace-all $nr
proc finish {} {
    global ns f nr
    $ns flush-trace
    close $f
    close $nr
    exec nam out.nam &
    exit 0
}
set n0 [$ns node]
set n1 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns at 5.0 "finish"
$ns run
```

### Output:



**Conclusion:** From the experiment we have learned to write TCL scripts and we can create a network topology and traffic and we can analyze the result using NS2.