Introduction to NS2

- *Network Simulator 2* (NS2) is an object-oriented, discrete event network simulator developed at UC Berkely
- Written in C++ and Otcl (Tcl script language with Object-oriented extensions developed at MIT)
- Network Simulator : A package of tools that simulates behaviour of networks.
- NS2 Creates Network Topologies
- It analyzes events to understand the network behavior
- It is used to implement
- - Network protocols such as TCP and UPD
- - Traffic source behavior such as FTP, Telnet and CBR,
- - Router queue management mechanism such as Drop Tail, RED and CBQ,
- - Routing algorithms such as AODV, AOMDV, Link State...etc.,

Network Animator (NAM):

A visual aid showing how packets flow along the network .

Apart from having a trace file, it comes with a Network animator tool that allows visualization of nodes.

Simulators:

NS2-Network Simulator-2

NS3-Network Simulator-3

QualNet

1. To open a page and write tcl script

syntax : vi filename.tcl

2. To run the script

syntax: ns filename.tcl

3. NS2 script start with creating simulator object

syntax: set ns [new Simulator]

4. To open a nam trace file

syntax: set nf [open o.nam w]

$ns namtrace-all $nf

To open a trace file

syntax: set nf1 [open o.tr w]

$ns trace-all $nf

5. To create two nodes

syntax: set n0 [$ns node]

set n1 [$ns node]

6. To create a link: syntax: $ns duplex-link $n0 $n1 2Mb 10ms DropTail

7. Calling finish procedure

syntax:$ns at 1.0 “finish”

8. Defining finish procedure

syntax:

```
proc finish {} {                                                    global
ns nf
$ns flush-trace
close $nf  #to close the tracefile
exec nam o.nam &          #to execute nam on the trace file
exit 0
}
```

9. To run the simulation

syntax: $ns run

10. ***set*** - assign a value to a variable

syntax: **set** a 43

11. [**expr** ...] is to make the interpreter calculate the value of expression within the bracket after the keyword.

syntax: set c [expr $a + $b]

12. ***puts*** prints out the following string within double quotation marks

syntax: puts "Welcome to OTcl"

- ***proc*** - define a procedure
- Ex: proc test {}
- {.....}
- ***set*** - assign a value to a variable
- Ex: **set** a 43
- [**expr** ...] is to make the interpreter calculate the value of expression within the bracket after the keyword.
- Ex: set c [expr $a + $b]

set <variable_name> <variable_value>

```
    set x 10
 set name "john"
    set price 12.2
set y $x
puts "hello world"
puts "value of y is $y"
expr $x + $y
set z [expr $x + $y]
puts "result of addition: $z"
```

- Throughput is the number of successfully received packets in a unit time and it is represented in bps.
- Packet Delivery Ratio: the ratio between the recived packets by the destination and the generated packets at the source.
- Delay: difference between the packet generated time and packet recived time

TraceFile Format