**Date:**

## OBJECT-4

**Objective:** To Study of Network simulator (NS).and Simulation of Congestion Control Algorithms using NS.

**NET WORK SIMULATOR (NS2):-**

**NS overview**

Ns programming: A Quick start

Case study I: A simple Wireless network

Case study II: Create a new agent in Ns

**NS overview**

Ns Status

Periodical release (ns-2.26, Feb 2003)

Platform support

FreeBSD, Linux, Solaris, Windows and Mac

**NS functionalities**

Routing, Transportation, Traffic sources, Queuing disciplines, QoS

**Wireless**

Ad hoc routing, mobile IP, sensor-MAC

Tracing, visualization and various utilities

NS (Network Simulators)

Most of the commercial simulators are GUI driven, while some network simulators are CLI driven. The network model / configuration describes the state of the network (nodes, routers, switches, links) and the events (data transmissions, packet error etc.). An important output of simulations are the trace files. Trace files log every packet, every event that occurred in the simulation and are used for analysis. Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots.

**Examples of network simulators: -**

There are many both free/open-source and proprietary network simulators. Examples of notable network simulation software are, ordered after how often they are mentioned in research papers:

1. ns (open source)
2. OPNET (proprietary software)
3. NetSim (proprietary software**)**

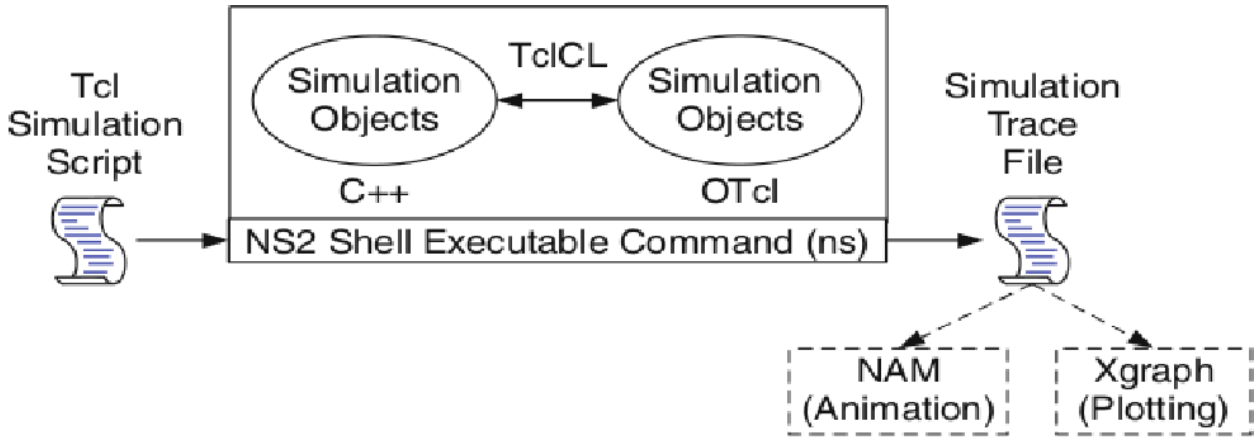


Fig 1. Basic architecture of ns-2

**Packet loss**

It occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is distinguished as one of the three main error types encountered in digital communications; the other two being bit error and spurious packets caused due to noise. Packets can be lost in a network because they may be dropped when a queue in the network node overflows. The amount of packet loss during the steady state is another important property of a congestion control scheme. The larger the value of packet loss, the more difficult it is for transport layer protocols to maintain high bandwidths, the sensitivity to loss of individual packets, as well as to frequency and patterns of loss among longer packet sequences is strongly dependent on the application itself.

**Throughput**

This is the main performance measure characteristic, and most widely used. In communication networks, such as Ethernet or packet radio, throughput or network throughput is the average rate of successful message delivery over a communication channel. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot This measure how soon the receiver is able to get a certain amount of data send by the sender. It is determined as the ratio of the total data received to the end to end delay. Throughput is an important factor which directly impacts the network performance.

**Delay**

Delay is the time elapsed while a packet travels from one point e.g., source premise or network ingress to destination premise or network degrees. The larger the value of delay, the more difficult it is for transport layer protocols to maintain high bandwidths. We will calculate end to end delay.

**Queue Length**

A queuing system in networks can be described as packets arriving for service, waiting for service if it is not immediate, and if having waited for service, leaving the system after being served. Thus queue length is very important characteristic to determine that how well the active queue management of the congestion control algorithm has been working.

**Congestion Control Mechanism in NS-2:-**

Congestion control is a state, which also occurs when a part of network message traffic is so heavy that it slows down the overall network response time. It is a global issue also which overloads a part of Network [Subnet] and leads to traffic congestion problem. Queuing algorithms can also used to overcome this problem.

**Few major congestion algorithms/protocols are:**

- Choke Algorithm
- Random Early detection Algorithm
- Random Exponential Marking also in Algorithm
- Drop tail also in algorithm
- Fair Queuing Algorithm
- Adaptive Virtual Queue also in Algorithm
- Virtual Queue Algorithm
- Rate based congestion control also in protocol
- Single Rate congestion protocol
- Window based congestion control protocol
- Multi Rate congestion also in protocol

These are also the few major congestion control algorithms used to overcome congestion problem in Network. You can also approach us for any particular code also for the above mentioned algorithms or any other codes, as per your need. We are also ready to support for all types of protocols, concepts and algorithms used in networking applications. Now, let's have also a glance over an example code also for TCP congestion control in NS2.

***Example NS-2 Simulation Code – TCP Congestion control in Wired Network [In TCL]***

**#create links between the nodes**

$ns duplex-link $node1 $node2 5Mb 15ms SFQ

$ ns duplex-link $node2 $node3 10Mb 12ms SFQ

**#Set Queue Size of link**

$ns queue-limit $node2 $node3 5

**#Setup a TCP connection**

set tcpcon [new Agent/TCP/Sack1]

$ns attach-agent $node1 $tcpcon

set sink [new Agent/TCPSink/Sack1]

$ns attach-agent $node1 $sink

$ ns connect $tcpcon $sink

$tcp set fid_ 1

$ tcp set window_ 8000

$tcp set packetSize_ 500

**#Setup a FTP over TCP connection**
set ftpcon [new Application/FTP]

$ftpcon attach-agent $tcpcon

$ ftpcon set type_ FTP

**#Setup a UDP connection**
set udpcon [new Agent/UDP]

$netsim attach-agent $node1 $udpcon

set null [new Agent/Null]

$ netsim attach-agent $n5 $null

$netsim connect $udpcon $null

$udpcon set fid_ 2

**#Setup a CBR over UDP connection**
set cbrcon [new Application/Traffic/CBR]

$cbrcon attach-agent $udpcon

$ cbrcon set type_ CBR

$cbrcon set packet_size_ 1500

$ cbrcon set rate_ 0.05mb

$cbrcon set random_ false

$ netsim at 0.1 “$cbrcon start”

$netsim at 2.0 “$ftpcon start”

$ netsim at 15.0 “$ftpcon stop”

$netsim at 16.0 “$cbrcon stop”