Ex No: 11

Date:

STUDY OF NETWORK SIMULATORS AND EMULATORS

Aim:

To Study of Network simulator platform as simulator and emulator.

Theory:

Ns overview

- Ns programming: A Quick start
- Case study I: A simple Wireless network
- Case study II: Create a new agent in Ns
- 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 utilitiesNS(Network Simulators) Most of the commercial simulators are GUI driven, while some network simulators are CLI driven. The network model / configuration describe the state of the network (nodes, routers, switchesand links) and the events (data transmissions, packet error etc.). The important outputs 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.

Most network simulators use discrete event simulation, in which a list of pending "events" is stored, and those events are processed in order, with some events triggering future events such as the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node. Simulation of networks is a very complex task. For example, if congestion is high, then estimation of the average occupancy is challenging because of high variance. To estimate the likelihood of a buffer overflow in a network, the time required for an accurate answer can be extremely large. Specialized techniques such as "control variants" and "importance sampling" have been developed to speed simulation.

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:

- ns (open source)
- OPNET (proprietary software)
- NetSim (proprietary software)

Uses of network simulators

Network simulators serve a variety of needs. Compared to the cost and time involved in setting up an entire test bed containing multiple networked computers, routers and data links, network simulators are relatively fast and inexpensive. They allow engineers, researchers to test scenarios that might be particularly difficult or expensive to emulate using real hardware - for instance,simulating a scenario with several nodes or experimenting with a new protocol in the network. Network simulators are particularly useful in allowing researchers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment. A typical network simulator encompasses a wide range of networking technologies and can help the users to build complex networks from basic building blocks such as a variety of nodes and links. With the help of simulators, one can design hierarchical networks using various types of nodes like computers, hubs, bridges, routers, switches, links, mobile units etc. Various types of Wide Area Network (WAN) technologies like TCP, ATM, IP etc. and Local Area Network (LAN) technologies like Ethernet, token rings etc., can all be simulated with a typical simulator and the user can test, analyse various standard results apart from devising some novel protocol or strategy for routing etc. Network simulators are also widely used to simulate battlefield networks in Network-centric warfare There are a wide variety of network simulators, ranging from the very simple to the very complex. Minimally, a network simulator must enable a user to represent a network topology, specifying the nodes on the network, the links between those nodes and the traffic between the nodes. More complicated systems may allow the user to specify everything about the protocols used to handle traffic in a network. Graphical applications allow users to easily visualize the workings of their simulated environment. Text-based applications may provide a less intuitive interface, but may permit more advanced forms of customization.

Packet loss 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 orbps), 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

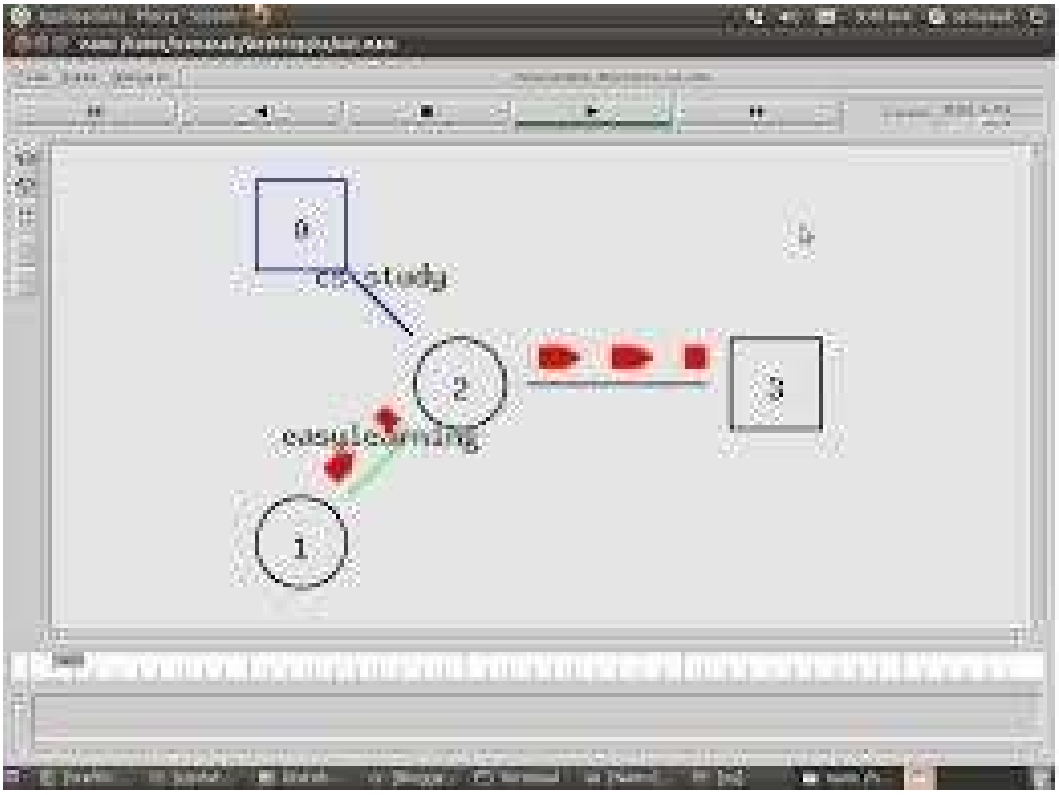
Queue Length

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.

1.Creating UDP Data traffic for wired network

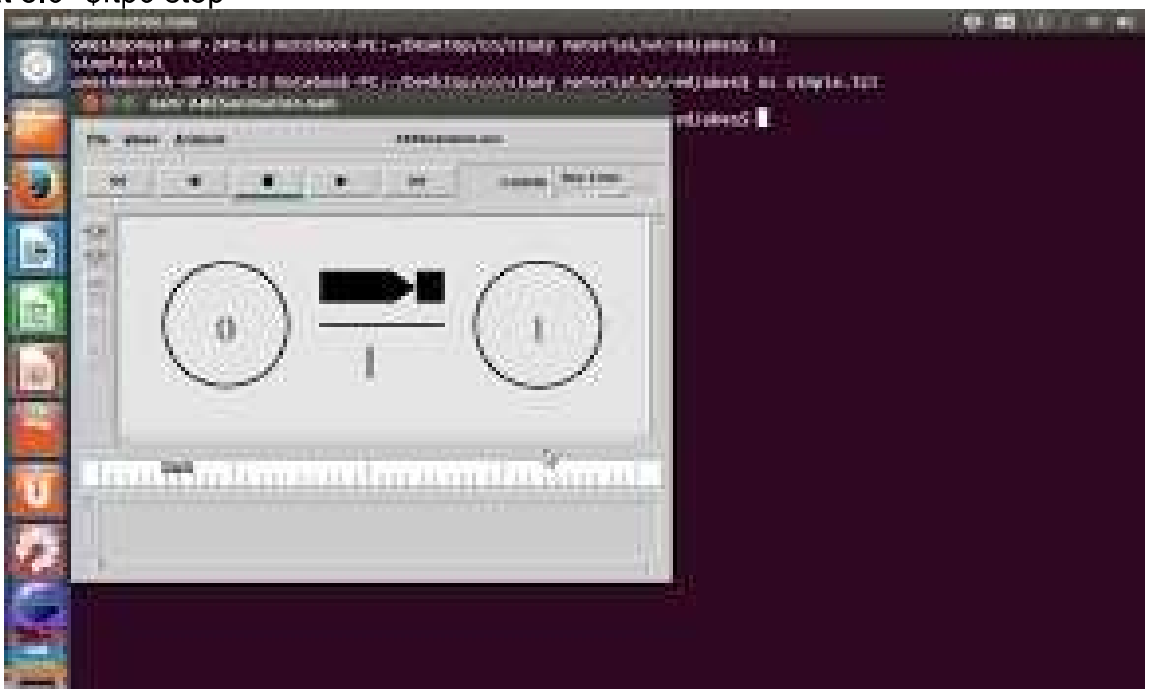
```
set ns [new Simulator]
#create file for analysis mode
set tr [open outl3.tr w]
$ns trace-all $tr
#create file for Animation Mode
set namtr [open outl3.nam w]
$ns namtrace-all $namtr
#Create Node
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#Create Link
$ns duplex-link $n0 $n1 10Mb 5ms DropTail
$ns duplex-link $n2 $n0 10Mb 5ms DropTail
$ns duplex-link $n3 $n0 10mb 5ms DropTail
#Create Orientation
```

```
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n0 $n2 orient left-up
$ns duplex-link-op $n0 $n3 orient left-down
#create UDP Source
set udp0 [new Agent/UDP]
$ns attach-agent $n3 $udp0
#create UDP Destination
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
#connecting UDP Source & Destination
$ns connect $udp0 $null0
#create application traffic
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
#Application start time
$ns at 1.0 "$cbr0 start"
#Application Stop time
$ns at 5.0 "$cbr0 stop"

$ns at 10.0 "$ns halt"
$ns run
```



2.Creating tcp data traffic for wired network

```
#create TCP Source
set tcp0 [new Agent/TCP]
$ns attach-agent $n2 $tcp0
#create TCP Destination
set sink0 [new Agent/TCPSink]
$ns attach-agent $n1 $sink0
$ns connect $tcp0 $sink0
#create application traffic
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
#Application start & stop time
$ns at 2.0 "$ftp0 start"
$ns at 5.0 "$ftp0 stop"
```



3.Wireless Network

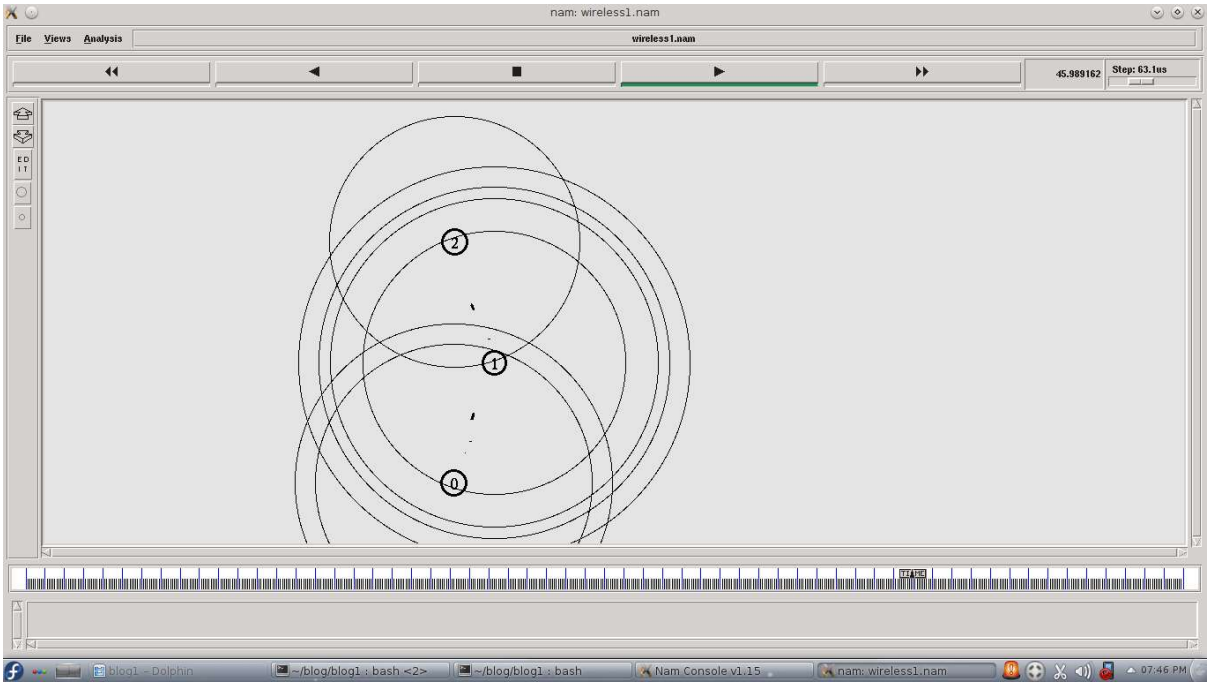
```
#parameter initialization
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/Shadowing
set val(netif)  Phy/WirelessPhy
set val(ant)    Antenna/OmniAntenna
set val(mac)  Mac/802_11
set val(ifq)     Queue/DropTail/PriQueue
set val(ll)       LL
set val(x)       300
set val(y)       300
set val(ifqlen) 100
set val(nn)     10
set val(stop) 300.0
set val(rp)     AODV
```

```
#Sheduler Creation
set ns [new Simulator]
set tracefd [open wireless.tr w]
$ns trace-all $tracefd
set namtrace [open wireless.nam w]
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

#Topography
set prop [new $val(prop)]
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

$ns node-config -adhocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -channelType $val(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON

#Creating node objects..

for {set i 0} { $i < $val(nn) } { incr i } {
        set node_($i) [$ns node]
}

for {set i 0} { $i < $val(nn) } { incr i } {
$node_($i) color darkgreen
$ns at 0.0 "$node_($i) color darkgreen"
}


##provide Initial location of wireless nodes
#Nodes:15, pause time:25.00, Max speed:3.00, Max X:100.0, MaxY:100.0
$node_(0) set X_ 271.057487973002
$node_(0) set Y_ 106.288323320442
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 96.023940800917
$node_(1) set Y_ 223.736201025809
$node_(1) set Z_ 0.000000000000
…..
$node_(9) set X_ 75.928619711928
```

```
$node_(9) set Y_ 203.805071579166
$node_(9) set Z_ 0.000000000000

#Assigning Mobility to Wireless Nodes

#Assigning Traffic -TCP
#Wireless Nodes:15, max connection:5, Send rate;0.0, seed:0
$ns at 50.000000000000 "$node_(0) setdest 234.760870186207 250.399813866979
1.818754605553"
$ns at 50.000000000000 "$node_(1) setdest 14.519853731648 84.887787293796
0.343160886173"
…..
$ns at 295.147357982603 "$node_(7) setdest 122.314673415723 126.467685356103
0.000000000000"


#0 connecting to 1 at time 133.84
set tcp_(0) [$ns create-connection TCP $node_(0) TCPSink $node_(1) 0]
$tcp_(0) set window_ 32
$tcp_(0) set packetSize_ 512
set ftp_(0) [$tcp_(0) attach-source FTP]
$ns at 133.8466 "$ftp_(0) start"

#0 connecting to 2 at time 33.84
set tcp_(1) [$ns create-connection TCP $node_(0) TCPSink $node_(2) 0]
$tcp_(1) set window_ 32
$tcp_(1) set packetSize_ 512
set ftp_(1) [$tcp_(1) attach-source FTP]
$ns at 33.8466 "$ftp_(1) start"

……

#Define node initial position in nam..
for {set i 0} { $i < $val(nn) } { incr i } {
#2 defines the node size for nam..
$ns initial_node_pos $node_($i) 2
}

#Tellin nodes when the simulation ends.
for {set i 0} {$i < $val(nn) } { incr i } {
$ns at $val(stop) "$node_($i) reset";
}

#Ending nam and the simulation..
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 299.01 "puts\"end simulation\"";#$ns halt
```

#stop procedure.

```
proc stop {} {
global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
exec nam 50nodes.nam &
exit 0
}

$ns run
```



Result:

Thus the Network Simulator 3 was studied for specific implementations as simulator and emulator.