Network Simulator Usage

Amrith M

April 9, 2018

1 About NS2

ns (from network simulator) is a name for a series of discrete event network simulators, specifically ns-1, and ns-2. All of them are discrete-event computer network simulators, primarily used in research and teaching.Network simulators are tools used to simulate discrete events in a network and which helps to predict the behaviours of a computer network. Generally the simulated networks have entities like links, switches, hubs, applications, etc. Once the simulation model is complete, it is executed to analyse the performance. Administrators can then customize the simulator to suit their needs. Network simulators typically come with support for the most popular protocols and networks in use today, such as WLAN,UDP,TCP,IP, WAN, etc.

2 Installation

- Download the all in one package for ns2
- All the files will be extracted into a folder called "ns-allinone-2.35".
- Ns2 requires a few packages to be pre installed. It also requires the GCC-version 4.3 to work correctly. So install all of them by using the following command
- Now we are ready to install ns2. To do so we first require root privileges and then we can run the install script. Use the following two commands: sudo su cd /ns-allinone-2.35/./install
- The final step is to tell the system, where the files for ns2 are installed or present. To do that, we have to set the environment path using the ".bashrc" file
- Once the system has restarted, open a terminal and start ns2 by using the following command: ns

3 Wired Communication Simulation

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

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

#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
        #Close the NAM trace file
        close $nf
        #Execute NAM on the trace file
        exec nam out.nam &
        exit 0
}

#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail

#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right

#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

```
#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP


#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false


#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
```

```
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

#Run the simulation
$ns run
```

4 Wireless Communication Simulation

```
# Simulator Instance Creation
set ns [new Simulator]

#Fixing the co-ordinate of simutaion area
set val(x) 500
set val(y) 500
# Define options
set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model

set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 2 ;# number of mobilenodes
set val(rp) AODV ;# routing protocol
set val(x) 500 ;# X dimension of topography
set val(y) 400 ;# Y dimension of topography
set val(stop) 10.0 ;# time of simulation end

# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

#Nam File Creation nam - network animator
set namfile [open sample1.nam w]

#Tracing all the events and cofiguration
$ns namtrace-all-wireless $namfile $val(x) $val(y)

#Trace File creation
set tracefile [open sample1.tr w]

#Tracing all the events and cofiguration
$ns trace-all $tracefile
```

```
# general operational descriptor- storing the hop details in the network
create-god $val(nn)

# configure the nodes
$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 OFF
-movementTrace ON

# Node Creation
set node1 [$ns node]
# Initial color of the node
$node1 color black

#Location fixing for a single node
$node1 set X_ 200
$node1 set Y_ 100
$node1 set Z_ 0

set node2 [$ns node]
$node2 color black

$node2 set X_ 200
$node2 set Y_ 300
$node2 set Z_ 0
# Label and coloring
$ns at 0.1 "$node1 color blue"
$ns at 0.1 "$node1 label Node1"
$ns at 0.1 "$node2 label Node2"
#Size of the node
$ns initial_node_pos $node1 30
$ns initial_node_pos $node2 30
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
```

```
#Stopping the scheduler
$ns at 10.01 "puts \"end simulation\" ; $ns halt"
#$ns at 10.01 "$ns halt"
proc stop {} {
global namfile tracefile ns
$ns flush-trace
close $namfile
close $tracefile
#executing nam file
exec nam sample1.nam &
}

#Starting scheduler
$ns run

#################################################################
Execution:
ns sample1.tcl
```