
Assignment 8. Using any open source Network Simulator, Implement

MANET / Wireless Sensor Network

What is NS2?

Setting up a network to do some real experiments is the best way for studying about communication in internet. However, setting a network is not easy and costly. For this reason, a virtual network provided by network simulator is used for experiment in only one computer. Specially, NS2 which is free and easy to use is the popular all over the world.

NS2 use Tcl language for creating simulation scenario file (for example, sample.tcl). Network topology, transmission time, using protocol etc... are defined in scenario file. If we execute this scenario file, the simulation result will be output to out.tr and out.nam file.out.tr all the information about communication is written in this file. We can find out the way a packet was forwarded.This file is called as trace file.

out.nam contains the data for animation of the experiment result. This file can be execute by Nam, an animation software. The state of forwarding packet in Nam is shownin Fig.2.If simulation use TCP, we can also observe the state of TCP congestion control by a trace file.

out.tcp Record the change of TCP parameters by time. Using Gnuplot software to plot a graph, it is easy to observe the appearance of congestion control. This file is called as TCP trace file.

Indicate a Network in NS2:

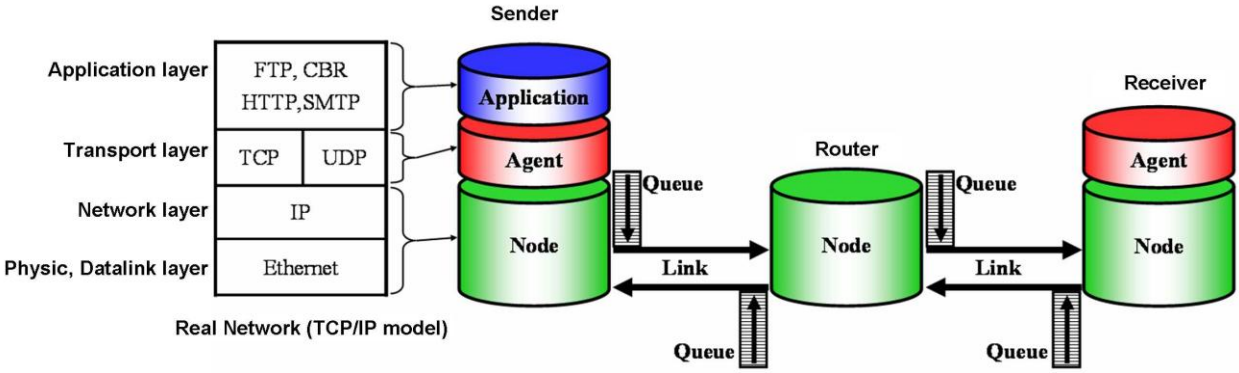


Figure 1: The network inside NS2

This section shows a simple NS simulation script and explains what each line does. Example 3 is an OTcl script that creates the simple network configuration and runs the simulation scenario in Figure 1. To run this simulation, download "ns-simple.tcl" and type "ns ns-simple.tcl" at your shell prompt.

The node (terminal, router) in real network have 4 layer using TCP/IP model .Actually, forwarding router have only 2 bottom layer. n the otherwise, 4 layers of TCP/IP model in NS2 are shown in Fig. 4. Two bottom layers are ready automatically when a node setting up. TCP or UDP in transport layer are shown as Agent. FTP, CBR are in application layers. Actually, the sender will set the application for creating data while the receive will not because it is no need to simulate the receiving data in application layer. Realistic network use cable for link between 2 node. One cable is available for biconnection. In NS2, a link is use for one connection. Two lines are needed for biconnection. Each link has a queue which similar to buffer in realistic network. Packet, which is sent from node should be queuing in queue. When queue is empty, it will be send to other node via link.

The following is steps to creating a scenario file:

Step0 Declare Simulator and setting output file

Step1 Setting Node and Link

Step2 Setting Agent

Step3 Setting Application

Step4 Setting Simulation time and schedules

Step5 Declare finish.

Theory: Simple Simulation Example

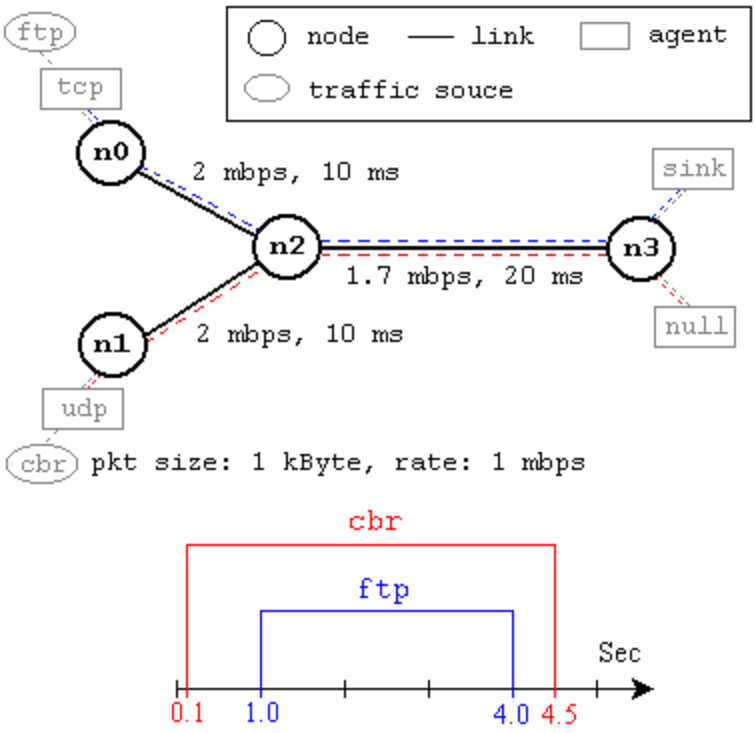


Figure 2. A Simple Network Topology and Simulation Scenario

This network consists of 4 nodes (n0, n1, n2, n3) as shown in above figure. The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay. Each node uses a DropTail queue, of which the maximum size is 10. A "tcp" agent is attached to n0, and a connection is established to a tcp "sink" agent attached to n3. As default, the maximum size of a packet that a "tcp" agent can generate is 1KByte. A tcp "sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. A "udp" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. A "ftp" and a "cbr" traffic generator are attached to "tcp" and "udp" agents respectively, and the "cbr" is configured to generate 1 KByte packets at the rate of 1 Mbps. The "cbr" is set to start at 0.1 sec and stop at 4.5 sec, and "ftp" is set to start at 1.0 sec and stop at 4.0 sec.

#Create a simulator object

set ns [new Simulator]

$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 trace file
   close $nf
   #Executenam on the trace file
   exec nam out.nam &
   exit0
}
#Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

$n0 shape box
$n0 color green
$n4 color red
```

$n2 color red

$n1 color blue

$n3 color blue

#Create links between the nodes

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

$ns duplex-link $n2 $n0 1Mb 10ms DropTail

$ns duplex-link $n3 $n0 1Mb 10ms DropTail

$ns duplex-link $n4 $n0 1Mb 10ms DropTail

$ns duplex-link $n5 $n0 1Mb 10ms DropTail

$ns duplex-link-op $n0 $n1 orient left-up

$ns duplex-link-op $n2 $n0 orient left-down

$ns duplex-link-op $n0 $n3 orient up

$ns duplex-link-op $n0 $n4 orient left-down

$ns duplex-link-op $n0 $n5 orient right-down

#Create a TCP agent and attach it to node n0

set tcp0 [new Agent/TCP]

$ns attach-agent $n1 $tcp0

#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3

set sink0 [new Agent/TCPSink]

$ns attach-agent $n3 $sink0

#Connect the traffic sources with the traffic sink

$ns connect $tcp0 $sink0

#Create a TCP agent and attach it to node n0

set tcp1 [new Agent/TCP]

$ns attach-agent $n4 $tcp1

#Create a TCP Sink agent (a traffic sink) for TCP and attach it to node n3

set sink1 [new Agent/TCPSink]

$ns attach-agent $n2 $sink1

#Connect the traffic sources with the traffic sink

$ns connect $tcp1 $sink1

$tcp0 set fid_ 1

$tcp1 set fid_ 2

 # Create a FTP and attach it to tcp0

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp1

 # Create a CBR traffic source and attach it to tcp0

set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $tcp0

#Schedule events for the CBR agents

$ns at 0.5 "$cbr0 start"

$ns at 0.5 "$ftp0 start"

$ns at 3.5 "$ftp0 stop"

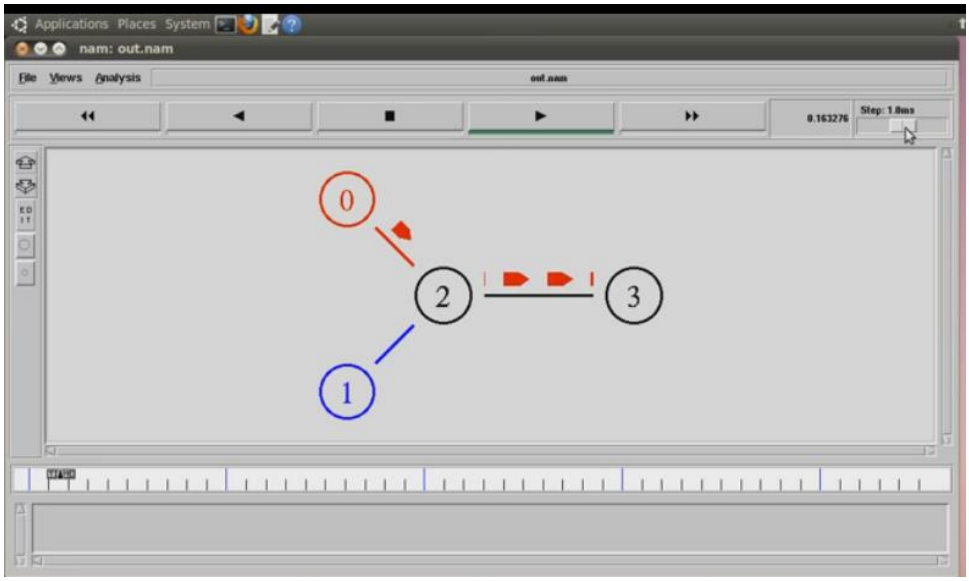
$ns at 4.5 "$cbr0 stop"

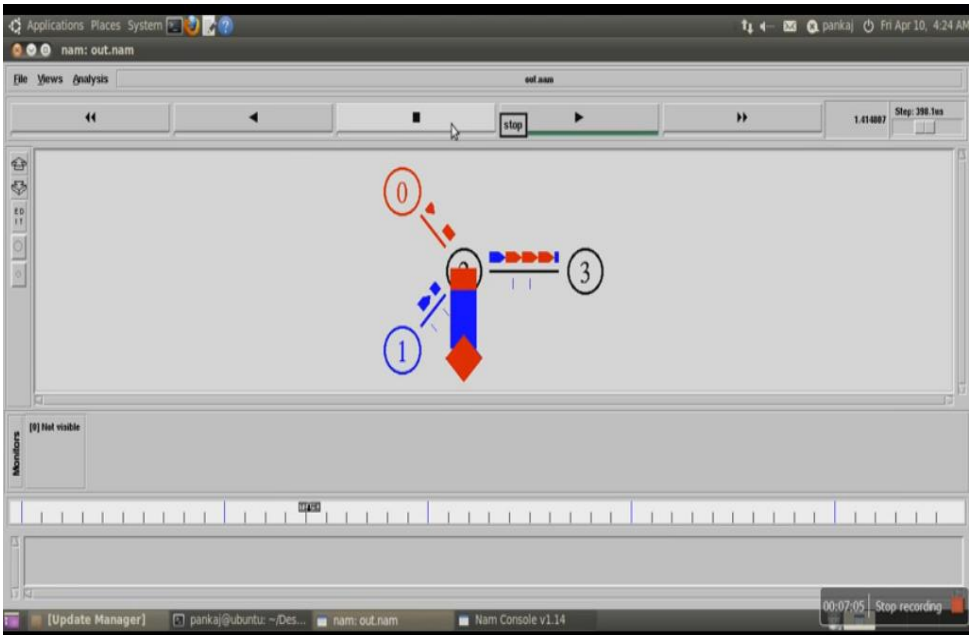
#Call the finish procedure after 5 seconds of simulation time

$ns at 5.0 "finish"

#Run the simulation

$ns run


Applications Places System
nam: out.nam
File Views Analysis
0
1
2
3


Applications Places System
nam: out.nam
File Views Analysis
stop
0
1
2
3
Monitors
[Update Manager]
Nam Console v1.14
Stop recording