

[2CEIT503 COMPUTER NETWORKS]

Practical: 6

AIM- Study and installation of Network Simulator.



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**

Department of Computer Engineering/Information Technology

Introduction

The network simulator is discrete event packet level simulator. The network simulator covers a very large number of application of different kind of protocols of different network types consisting of different network elements and traffic models. Network simulator is a package of tools that simulates behavior of networks such as creating network topologies, log events that happen under any load, analyze the events and understand the network. Well the main aim of our first experiment is to learn how to use network simulator and to get acquainted with the simulated objects and understand the operations of network simulation and we also need to analyze the behavior of the simulation object using network simulation.

Platform required to run network simulator

Linux and UNIX like systems

Linux (Use Fedora or Ubuntu versions)

Free BSD

SunOS/Solaris

Windows 95/98/NT/2000/XP

Backend Environment of Network Simulator

Network Simulator is mainly based on two languages. They are C++ and OTcl. OTcl is the object oriented version of Tool Command language. The network simulator is a bank of different network and protocol objects. C++ helps in the following way:

- It helps to increase the efficiency of simulation.
- It is used to provide details of the protocols and their operation.
- OTcl helps in the following way:
 - With the help of OTcl we can describe different network topologies
 - It helps us to specify the protocols and their applications
 - It allows fast development
 - Tcl is compatible with many platforms and it is flexible for integration
 - Tcl is very easy to use and it is available in free
 - Developed by UC Berkeley
 - Maintained by USC
 - Popular simulator in scientific environment
 - Other popular network simulators – Glomosim: UCLA, CMU; ParseC, Mobile

Practical: 6

Simulation mostly – OPNET: commercial software, graphical interface, not free; –

- Others: commercial ones, not free, e.g. IBM TPNS

NS2 Goals

- To support networking research and education
 - Protocol design, traffic studies, etc.
 - Protocol comparison;
- New architecture designs are also supported.
- To provide collaborative environment
- Freely distributed, open source;
- Increase confidence in result

Two Languages:

C++, OTclOTcl: short for MIT Object Tcl, an extension to Tcl/Tk for object-oriented programming.

- Used to build the network structure and topology which is just the surface of your simulation;
- Easily to configure your network parameters;
- Not enough for research schemes and protocol architecture adaption.
- Two Languages (Con't) C++: Most important and kernel part of the NS2
- To implement the kernel of the architecture of the protocol designs; □ From the packet flow view, the processes run on a single node;
- To change or “comment out” the existing protocols running in NS2; □ Details of your research scheme.

Why 2 Languages?

- 2 requirements of the simulator – Detailed simulation of Protocol: Run-time speed; – Varying parameters or configuration: easy to use.
- C++ is fast to run but slower to code and change;
- OTcl is easy to code but runs slowly

Protocols/Models supported by NS2

- Wired Networking – Routing: Unicast, Multicast, and Hierarchical Routing, etc. – Transportation: TCP, UDP, others; – Traffic sources: web, ftp, telnet, cbr, etc. – Queuing

Practical: 6

- disciplines: drop-tail, RED, etc. – QoS: IntServ and Diffserv Wireless Networking • Ad hoc routing and mobile IP
- Sensor Networks(hmmm) – SensorSim: built up on NS2, additional features, for TinyOS

NS2 Models

- Traffic models and applications: Web, FTP, telnet, constant-bit rate(CBR) • Transport protocols: Unicast: TCP (Reno,Vegas), UDP Multicast • Routing and queuing: Wired routing, Ad Hoc routing. • Queuing protocols: RED(Random Early Drop), drop-tail • Physical media: Wired (point-to-point, LANs), wireless, satellite

Researches based on NS2

- Intserv/Diffserv (QoS) • Multicast: Routing, Reliable multicast • Transport: TCP Congestion control • Application: Web caching Multimedia • Sensor Networks: LEACH, Directed Diffusion, etc. Most are routing protocols.

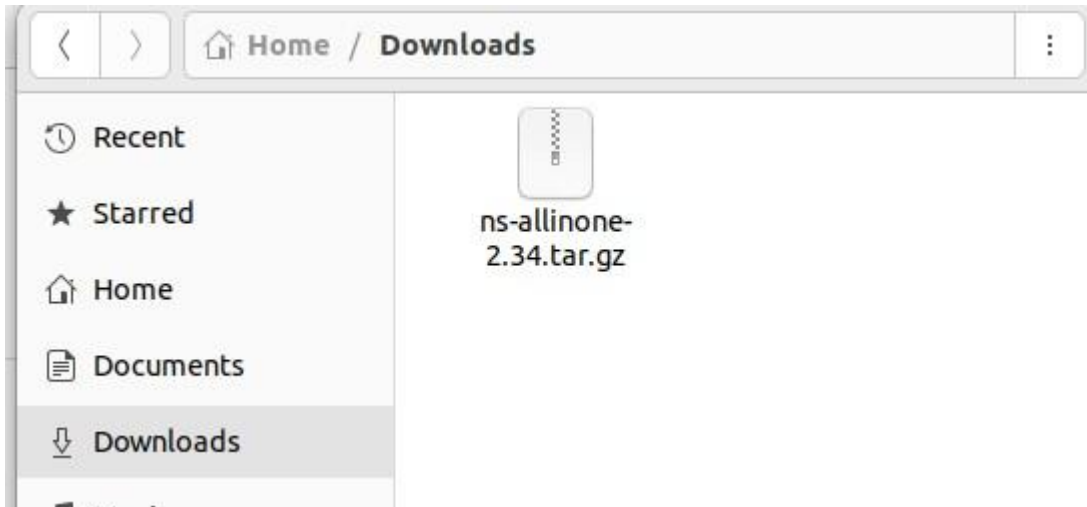
NS2 Components • NS2: the simulator itself, now version: ns-2.26 We will work with the part mostly. • NAM: Network animator. Visualized trace tool(not really). Nam editor: GUI interface to generate ns scripts Just for presentation now, not useful for research tracing. • Preprocessing: Traffic and topology generators • Post-processing: Simple trace analysis, often in Awk, Perl(mostly), or Tcl

Installation of NS2.

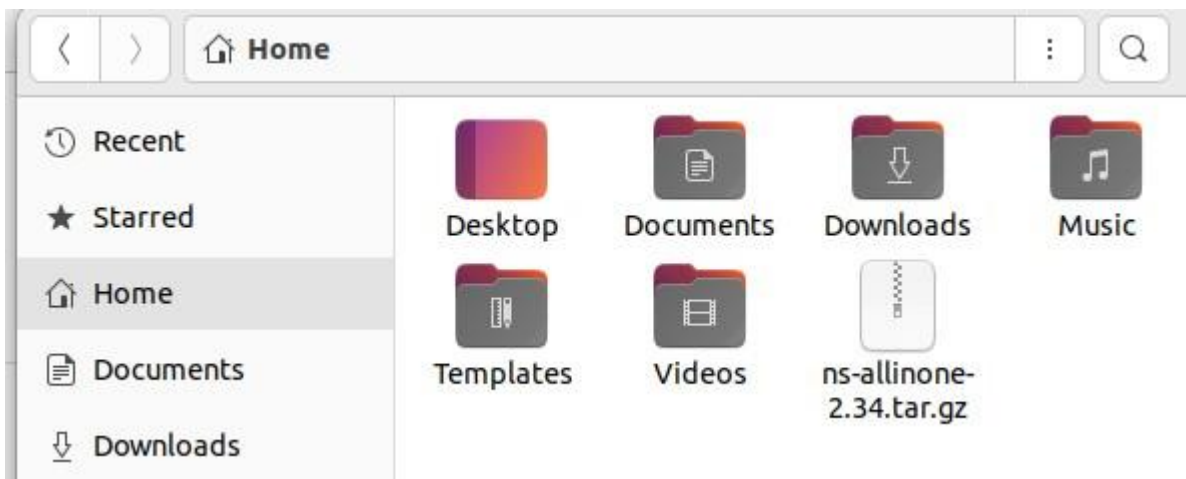
Installation Steps for ns-2.34 on ubuntu 10.04

Step 1: Download the ns-allinone-2.35.

Practical: 6



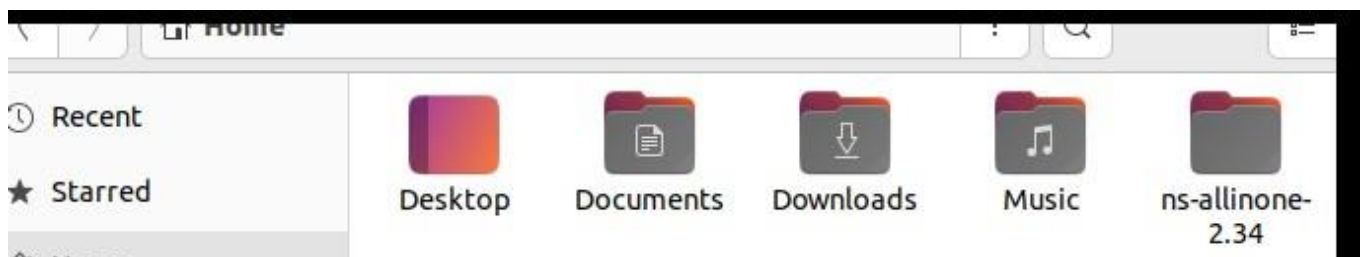
Step 2: Place the ns-allinone-2.34.tar.gz package in your home folder.



Step 3: Next, open the Terminal (Applications --> Accessories --> Terminal) or (ctrl+alt+t)

Step 4: extract the contents in the home folder by typing following command in terminal

```
tar -xzf ns-allinone-2.34.tar.gz
```



Step 5: Next, Change to ns-allinone2.34 directory

Practical: 6

cd ns-allinone2.34

```
ubuntu@ubuntu-virtual-machine:~$ tar -xzf ns-allinone-2.34.tar.gz
ubuntu@ubuntu-virtual-machine:~$ tar -xzf ns-allinone-2.34.tar.gz
ubuntu@ubuntu-virtual-machine:~$ cd ns-allinone-2.34/
ubuntu@ubuntu-virtual-machine:~/ns-allinone-2.34$
```

Step 6: for administrative rights

sudo su

```
ubuntu@ubuntu-virtual-machine:~$ cd ns-allinone-2.34/
ubuntu@ubuntu-virtual-machine:~/ns-allinone-2.34$ sudo su
[sudo] password for ubuntu:
root@ubuntu-virtual-machine:/home/ubuntu/ns-allinone-2.34#
```

Step 7: First install all the dependencies

sudo apt-get install build-essential

autoconf automake libxmu-dev gcc-4.3

```
root@ubuntu-virtual-machine:/home/ubuntu/ns-allinone-2.34# sudo apt-get install
build-essential autoconf automake libxmy-dev dcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package libxmy-dev
E: Unable to locate package dcc
root@ubuntu-virtual-machine:/home/ubuntu/ns-allinone-2.34#
```

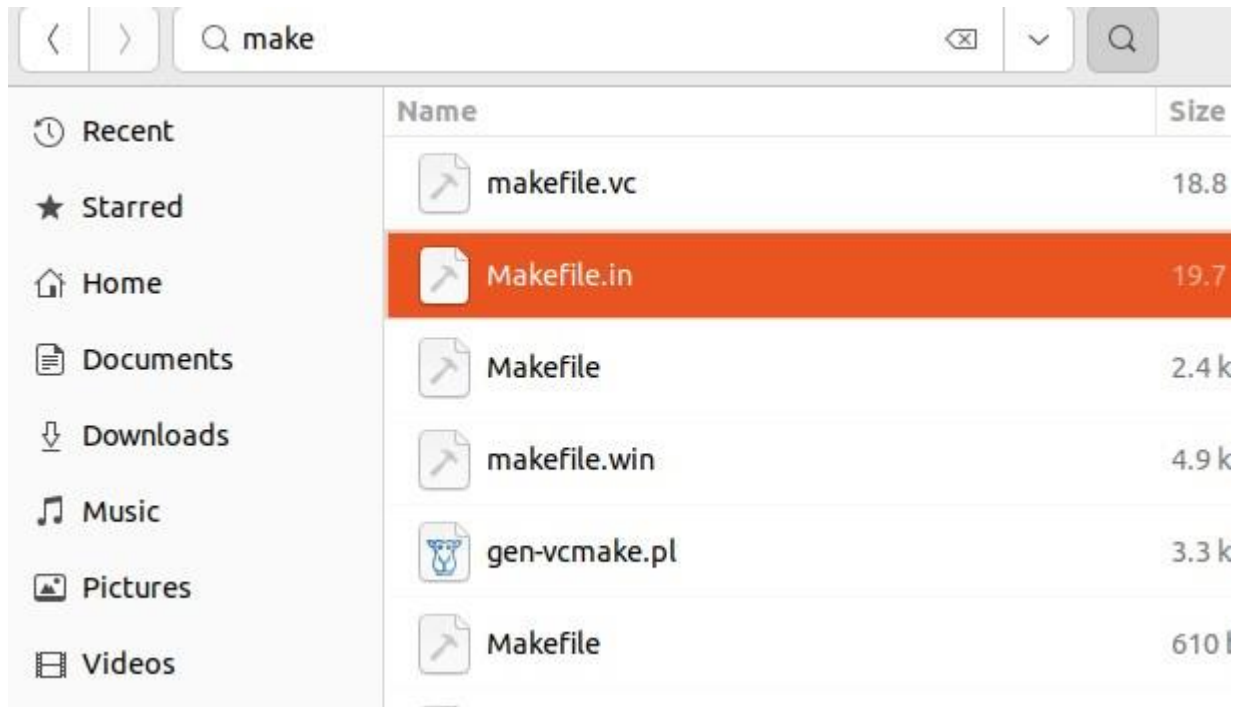
Step 8: Edit Makefile.in found at this location ns-allinone-2.34/otcl-1.13/Makefile.in as follows: Find the line that says:

CC= @CC@

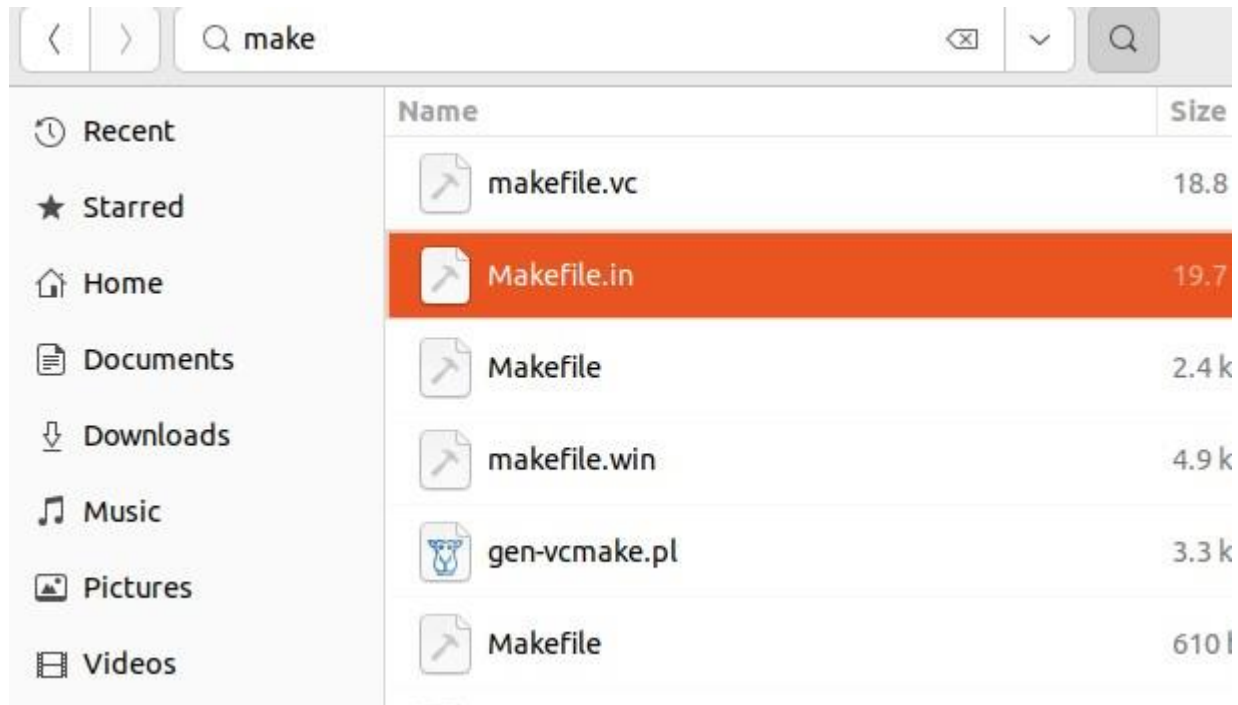
and change it to:

CC= gcc-4.3

Practical: 6



Practical: 6



```
33 BLANK = # make a blank space. DO NOT ac
34
35 # The following will be redefined under Wi
36 CC = gcc-4.3
37 CPP = @CXX@
38 LINK = $(CPP)
39 LINK_SHLIB = @SHLIB_LD@
40 MKDEP = ./conf/mkdep
41 TCLSH = @V_TCLSH@
42 TCL2C = @V_TCL2CPP@
43 AR = ar rc $(BLANK)
44
45 RANLIB = @V_RANLIB@
46 INSTALL = @INSTALL@
47 IN = ln
```


Practical: 6

```
ubuntu@ubuntu-virtual-machine:~$ cd ns-allinone-2.34/
ubuntu@ubuntu-virtual-machine:~/ns-allinone-2.34$ ./install
=====
* Testing for Darwin (OS X) environment
=====
* Testing for Cygwin environment
=====
Cygwin not detected, proceeding with regular install.
=====
* Testing for FreeBSD environment
=====
FreeBSD not detected
=====
* Build XGraph-12.1
=====
./configure: 423: cannot create ./config.log: Permission denied
./install: 297: make: not found
Can not create xgraph; But xgraph is an optional package, continue
=====
* Build CWeb
=====
Making cweb
```

Step 10: For the changes to take effect immediately, do the following:

source ~/.bashrc

ns

% ctrl+c

```
Setting up libc6:amd64 (2.35-0ubuntu3.1) ...
Setting up libtclcl1:amd64 (1.20-9.1) ...
Setting up ns2 (2.35+dfsg-3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
ubuntu@ubuntu-virtual-machine:~$ ns
%
```

Step 11: install nam

Practical: 6

```
ubuntu@ubuntu-virtual-machine:~/Downloads$ sudo apt install nam
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nam
0 upgraded, 1 newly installed, 0 to remove and 52 not upgraded.
Need to get 201 kB of archives.
After this operation, 724 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 nam amd64 1.15-5.2 [201 kB]
Fetched 201 kB in 2s (130 kB/s)
Selecting previously unselected package nam.
(Reading database ... 206445 files and directories currently installed.)
Preparing to unpack .../nam_1.15-5.2_amd64.deb ...
Unpacking nam (1.15-5.2) ...
Setting up nam (1.15-5.2) ...
Processing triggers for man-db (2.10.2-1) ...
ubuntu@ubuntu-virtual-machine:~/Downloads$
```

