# Department of Computing

# EE353: Computer Networks

# Class: BESE-13AB

**CLO 3,CLO4**

# Lab 09: Installation/Introduction to NS2



**Date: 15-11-2023**

# Time: 10:00 to 1:00 and 2:00 to 05:00

# Instructor: Dr. Humma Gafoor

# Lab Engineer: Syed Muhammad Ali Musa

**Name:** Aimen Munawar

**Class:** BESE-13-A

**CMS ID:** 415867



# Lab 09: Installation and Introduction of NS2

**Introduction**

Network Simulator 2 (ns2) is a very popular network simulation software for linux operating system. It works really good on the Ubuntu operating system.

**Objectives**

The main target of this lab is to understand and explore NS2

**Tools/Software Requirement**

NS2 or NS3

**Description**

Network Simulator 2 (ns2) is a very popular network simulation software for linux operating system.

Diagram

Description automatically generated

Nam File

It works really good on the Ubuntu operating system. Presently, I am using Ubuntu 12.04 LTS. In this Lab, we will see how to install ns2.35 on Ubuntu 12.04 LTS with simple and easy way.

Installing ns2 on Ubuntu is very easy and simple. First you need to [download NS2.35](http://www.isi.edu/nsnam/ns/) and copy it to the home folder. Then follow the steps below:

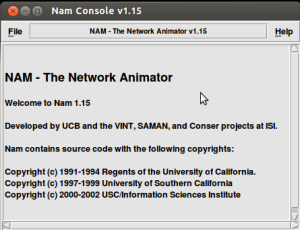
First open the Terminal (Ctl+Alt+T) and write “**sudo apt-get update**” for necessary updates.

Then write “**sudo apt-get install build essential autoconf automake libxmu-dev**” on terminal for necessary library.

The downloaded file is a tar file. You need to untar it. So, write “**tar zxvf ns-allinone-2.35.tar.gz**“.

Then run the commands below:

* **cd ns-allinone-2.35**
* **./install**
* **cd**
* **sudo apt-get install ns2**



**Simulation Example:**

# Filename: test1.tcl

#——-Event scheduler object creation——–#  
set ns [new Simulator]

#———-creating trace objects—————-#

set nt [open test1.tr w]  
$ns trace-all $nt  
#———-creating nam objects—————-#

set nf [open test1.nam w]  
$ns namtrace-all $nf  
#———-Setting color ID—————-#  
$ns color 1 darkmagenta  
$ns color 2 yellow  
$ns color 3 blue  
$ns color 4 green  
$ns color 5 black

#———- Creating Network—————-#

set totalNodes 3

for {set i 0} {$i < $totalNodes} {incr i} {  
set node\_($i) [$ns node]  
}

set server 0  
set router 1  
set client 2

#———- Creating Duplex Link—————-#  
$ns duplex-link $node\_($server) $node\_($router) 2Mb 50ms DropTail  
$ns duplex-link $node\_($router) $node\_($client) 2Mb 50ms DropTail

$ns duplex-link-op $node\_($server) $node\_($router) orient right  
$ns duplex-link-op $node\_($router) $node\_($client) orient right

#————Labelling—————-#

$ns at 0.0 "$node\_($server) label Server"  
$ns at 0.0 "$node\_($router) label Router"  
$ns at 0.0 "$node\_($client) label Client"

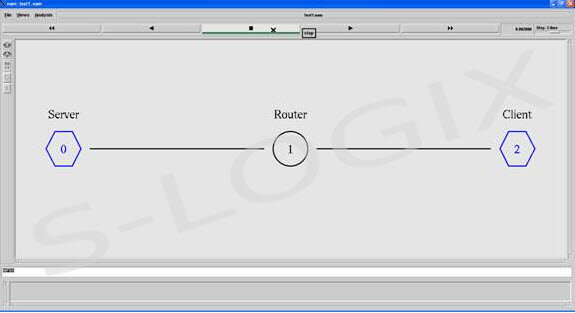
$ns at 0.0 "$node\_($server) color blue"  
$ns at 0.0 "$node\_($client) color blue"

$node\_($server) shape hexagon  
$node\_($client) shape hexagon  
#———finish procedure——–#

proc finish {} {  
global ns nf nt  
$ns flush-trace  
close $nf  
close $nt  
puts "running nam…"  
exec nam test1.nam &  
exit 0  
}

#Calling finish procedure  
$ns at 10.0 "finish"  
$ns run

**NAM:**



Execute the script using the ns command.

ns test2.tcl

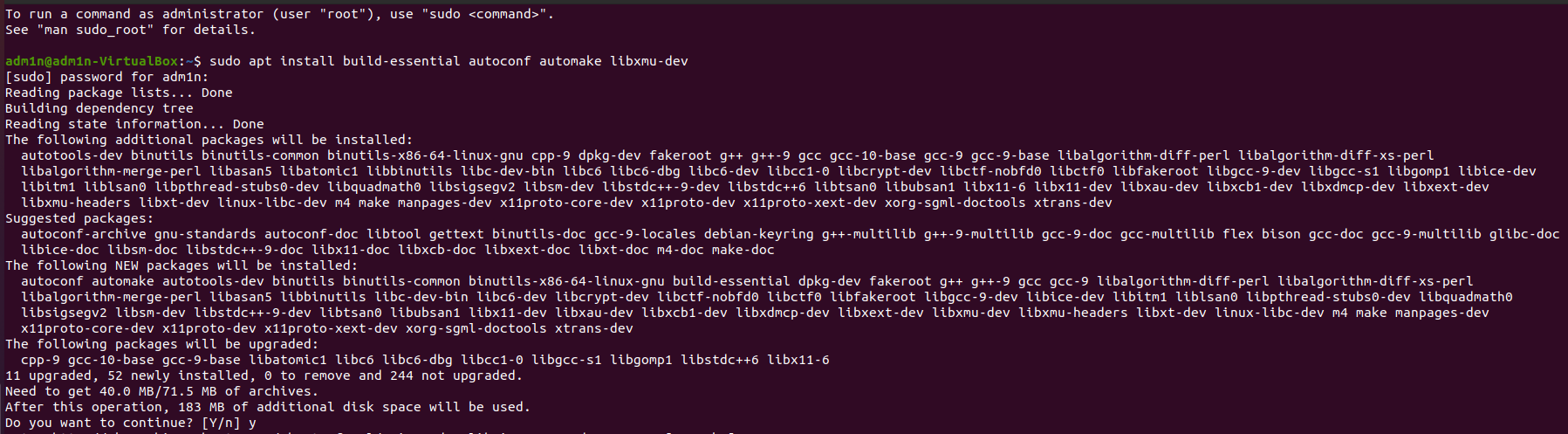
#### Starting nam Execution part with the following command

nam test2.tcl

**Lab Task:**

* Install NS2 in your system.

**Step 1: Installing basic libraries.**

**Commands**: sudo apt install build-essential autoconf automake libxmu-dev

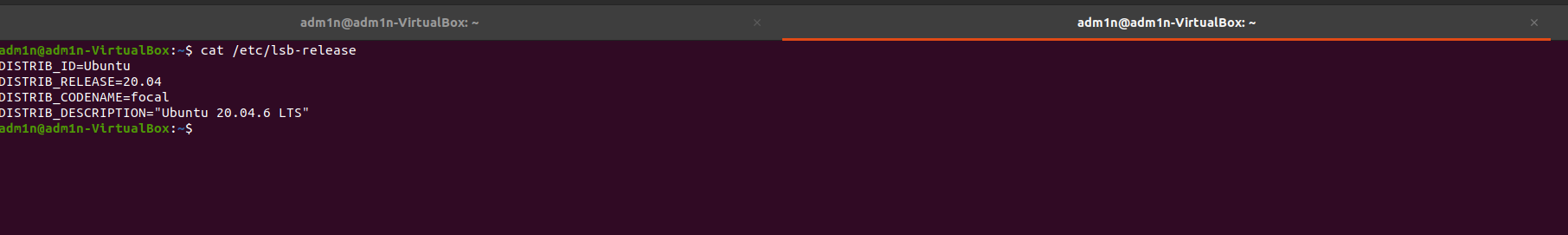
**Step 2: Installing GCC-4.8 and G++- 4.8**

**Commands:**

**A screenshot of a computer program

Description automatically generated**sudo nano /etc/apt/sources.list

**Commands:** cat /etc/lsb-release

****

**Commands:**

**A screen shot of a computer

Description automatically generated**deb http://in.archive.ubuntu.com/ubuntu bionic main universe

A blurry image of a building

Description automatically generated**Commands:** sudo apt update

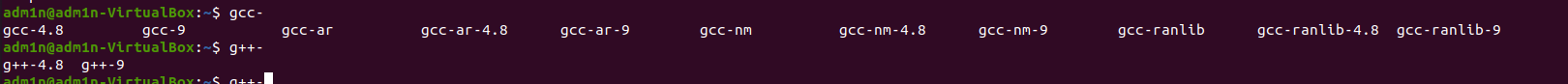
A computer screen shot of a computer screen

Description automatically generated

Commands: sudo apt install gcc-4.8 g++-4.8

A computer screen with many colorful lights

Description automatically generated with medium confidence

**Step 3:** unzipping the NS2 packages to home folderA screenshot of a computer program

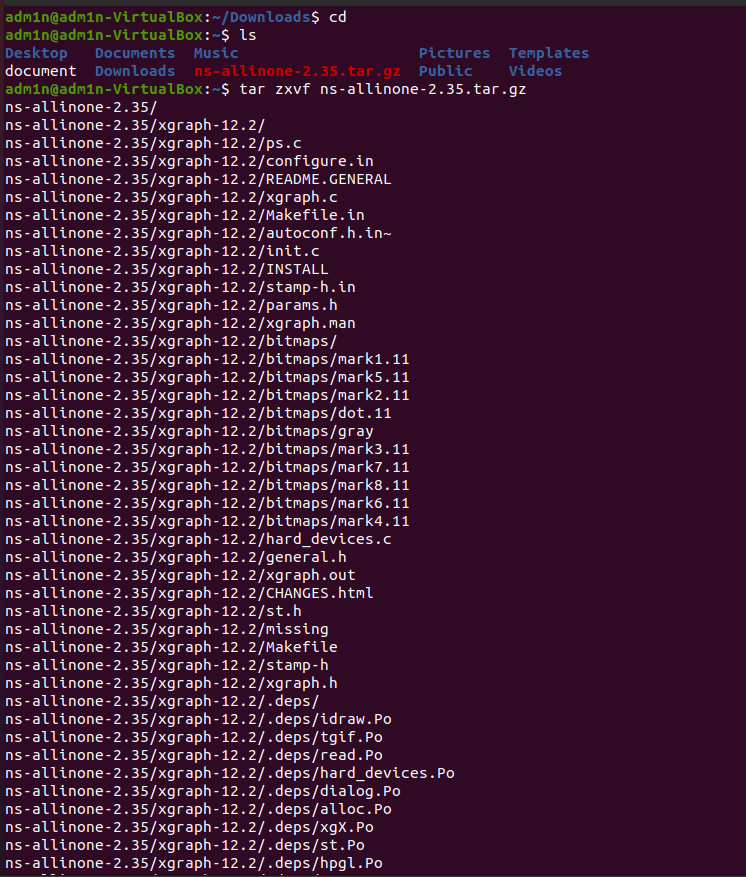
Description automatically generated

**Commands:** cp ns-allinone-2.35.tar.gz ../

**A screen shot of a computer

Description automatically generated**

**Commands:** tar zxvf ns-allinone-2.35.tar.gz

****

Commands:

cd ns-allinone-2.35/

cd ns-2.35

getit Makefile.in

Changing @CC@ to gcc-4.8 and @CXX@ to g++-4.8 in make file(inside ns-2.35)

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Commands: Changing @CC@ to gcc-4.8 and @CXX@ to g++-4.8 in make file(inside nam-1.15)

**A screenshot of a computer program

Description automatically generatedA screenshot of a computer screen

Description automatically generated**

Commands:

cd otcl-1.14/

gedit Makefile.in

Changing @CC@ to gcc-4.8 in make file

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Commands:

cd xgraph-12.2/

gedit Makefile.in

Changing @CC@ to gcc-4.8 and @cpp@ to g++-4.8 in Makefile

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Commands:**

Opening file ns-2.35/linkstate.ls.h

Change at line no 137

**Before Changing:**

**A screenshot of a computer program

Description automatically generated**

**After Changing:**

**A screenshot of a computer program

Description automatically generated**

**Step 4:**

cd ns-allinone-2.35/

. /install

**A screenshot of a computer program

Description automatically generated**

Commands:

gedit ./bashrc

**A screenshot of a computer program

Description automatically generated**

**Step 5:**

set the path

gedit ./.bashrc

enter the path and LD\_library\_path

**A computer screen shot of a computer screen

Description automatically generated**

**Command :**

**source .bashrc**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Part 2:**

**Installing nam using command sudo apt-get install nam**

**A screenshot of a computer program

Description automatically generated**

* Implement basic wired network in ns2 with 4Clients.
* Generate NAM and trace files.

**Running the given piece of code:**

**Code:**

# Filename: test1.tcl

#——-Event scheduler object creation——–#

set ns [new Simulator]

#———-creating trace objects—————-#

set nt [open test1.tr w]

$ns trace-all $nt

#———-creating nam objects—————-#

set nf [open test1.nam w]

$ns namtrace-all $nf

#———-Setting color ID—————-#

$ns color 1 darkmagenta

$ns color 2 yellow

$ns color 3 blue

$ns color 4 green

$ns color 5 black

#———- Creating Network—————-#

set totalNodes 3

for {set i 0} {$i < $totalNodes} {incr i} {

set node\_($i) [$ns node]

}

set server 0

set router 1

set client 2

#———- Creating Duplex Link—————-#

$ns duplex-link $node\_($server) $node\_($router) 2Mb 50ms DropTail

$ns duplex-link $node\_($router) $node\_($client) 2Mb 50ms DropTail

$ns duplex-link-op $node\_($server) $node\_($router) orient right

$ns duplex-link-op $node\_($router) $node\_($client) orient right

#————Labelling—————-#

$ns at 0.0 "$node\_($server) label Server"

$ns at 0.0 "$node\_($router) label Router"

$ns at 0.0 "$node\_($client) label Client"

$ns at 0.0 "$node\_($server) color blue"

$ns at 0.0 "$node\_($client) color blue"

$node\_($server) shape hexagon

$node\_($client) shape hexagon

#———finish procedure——–#

proc finish {} {

global ns nf nt

$ns flush-trace

close $nf

close $nt

puts "running nam…"

exec nam test1.nam &

exit 0

}

#Calling finish procedure

$ns at 10.0 "finish"

$ns run

**Output:**

**A screenshot of a video editing

Description automatically generated**

**Writing Code to Implement basic wired network in ns2 with 4 Clients.**

**Code:**

# Filename: test2.tcl

# Event scheduler object creation

set mySimulator [new Simulator]

# Creating trace objects

set myTraceFile [open test2.tr w]

$mySimulator trace-all $myTraceFile

# Creating nam objects

set myNamFile [open test2.nam w]

$mySimulator namtrace-all $myNamFile

# Setting color ID

$mySimulator color 1 darkmagenta

$mySimulator color 2 yellow

$mySimulator color 3 blue

$mySimulator color 4 green

$mySimulator color 5 black

# Creating Network

set totalNodes 6

for {set i 0} {$i < $totalNodes} {incr i} {

set myNode\_($i) [$mySimulator node]

}

set myServer 0

set myRouter 1

set myClientStart 2

set myClientEnd 6

# Setting Colors

$myNode\_(0) color green

$myNode\_(1) color blue

$myNode\_(2) color red

$myNode\_(3) color red

$myNode\_(4) color red

$myNode\_(5) color red

# Setting Labels and Shapes

$mySimulator at 0.0 "$myNode\_(0) label \"Server\""

$mySimulator at 0.0 "$myNode\_(1) label \"Router\""

for {set i $myClientStart} {$i < $myClientEnd} {incr i} {

$mySimulator at 0.0 "$myNode\_($i) label \"Client$i\""

$mySimulator at 0.0 "$myNode\_($i) color blue"

$myNode\_($i) shape square

}

# Creating Duplex Links

for {set i $myClientStart} {$i < $myClientEnd} {incr i} {

$mySimulator duplex-link $myNode\_($myServer) $myNode\_($myRouter) 2Mb 50ms DropTail

$mySimulator duplex-link $myNode\_($myRouter) $myNode\_($i) 2Mb 50ms DropTail

$mySimulator duplex-link-op $myNode\_($myServer) $myNode\_($myRouter) orient right

$mySimulator duplex-link-op $myNode\_($myRouter) $myNode\_($i) orient right

}

# Finish Procedure

proc myFinish {} {

global mySimulator myNamFile myTraceFile

$mySimulator flush-trace

close $myNamFile

close $myTraceFile

puts "Running nam…"

exec nam test2.nam &

exit 0

}

# Calling finish procedure

$mySimulator at 10.0 "myFinish"

# Run the simulation

$mySimulator run

**Output:**

A computer screen shot of a diagram

Description automatically generated

**Deliverables**

Upload TCL file with trace file and snap of NAM file.

**Grade Criteria**

This lab is graded. Min marks: 0. Max marks: 10.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Minimum** | **Maximum** |
| Documentation with clearly defined understanding of the lab task and approach | Fail | Pass |
| Lab Tasks | 0 | 10 |