Jimma Institute of technology

Faculty of computing

NS2 and OMNeT++ with VEINS installation

Prepared by

Duresa teshome duressateshome@gmail.com

Getamesay Haile getamesay0923@gmail.com

Contents

1. Installation of SUMO, OMNeT++ and Veins simulator

This is a guide for installation of SUMO, OMNeT++ and VEINS simulators in Windows Systems.

1.1 Get the source code for the programs

1.1.1 Get the source code for SUMO

Download *sumo-src-0.25.0*from this website
http://sourceforge.net/projects/sumo/files/sumo/

1.1.2 Get the source code for OMNeT++

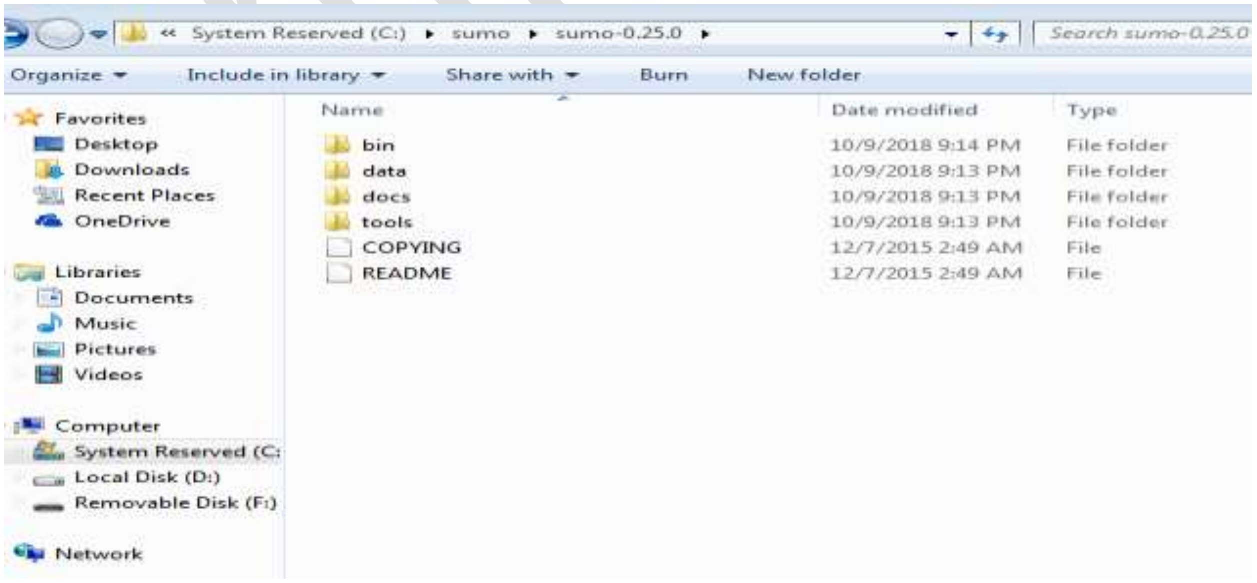
Go to the following website and download *omnetpp-5.0-src*:
http://www.omnetpp.org/omnetpp

1.1.3Get the source code for VEINS

Download *veins-4.4.zip* from the website:
http://veins.car2x.org/download/

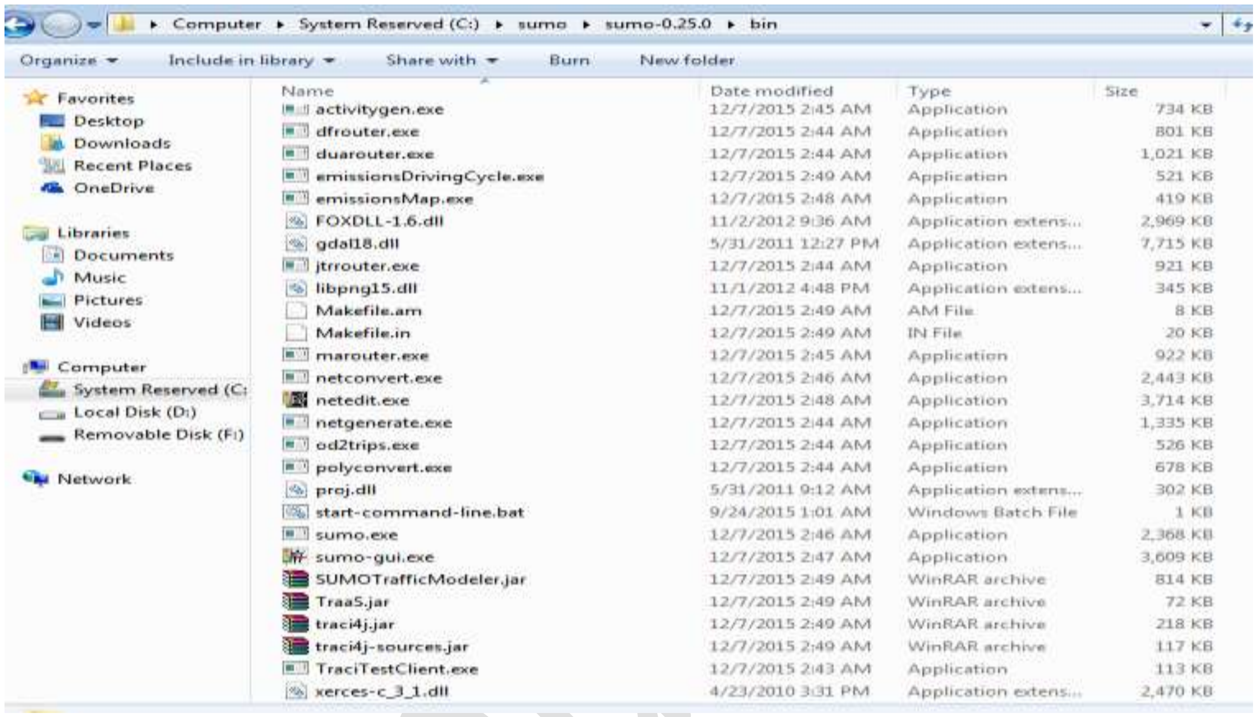
2. Installation of SUMO

- Make a folder **Sumo** in the partition **C:\** and extract there the zip file which you downloaded
- Extract the file inside your folder
- Open your folder and you will see **bin, data, docs** and **tools** folders



Name	Date modified	Type
bin	10/9/2018 9:14 PM	File folder
data	10/9/2018 9:13 PM	File folder
docs	10/9/2018 9:13 PM	File folder
tools	10/9/2018 9:13 PM	File folder
COPYING	12/7/2015 2:49 AM	File
README	12/7/2015 2:49 AM	File

➤ Open **bin** directory



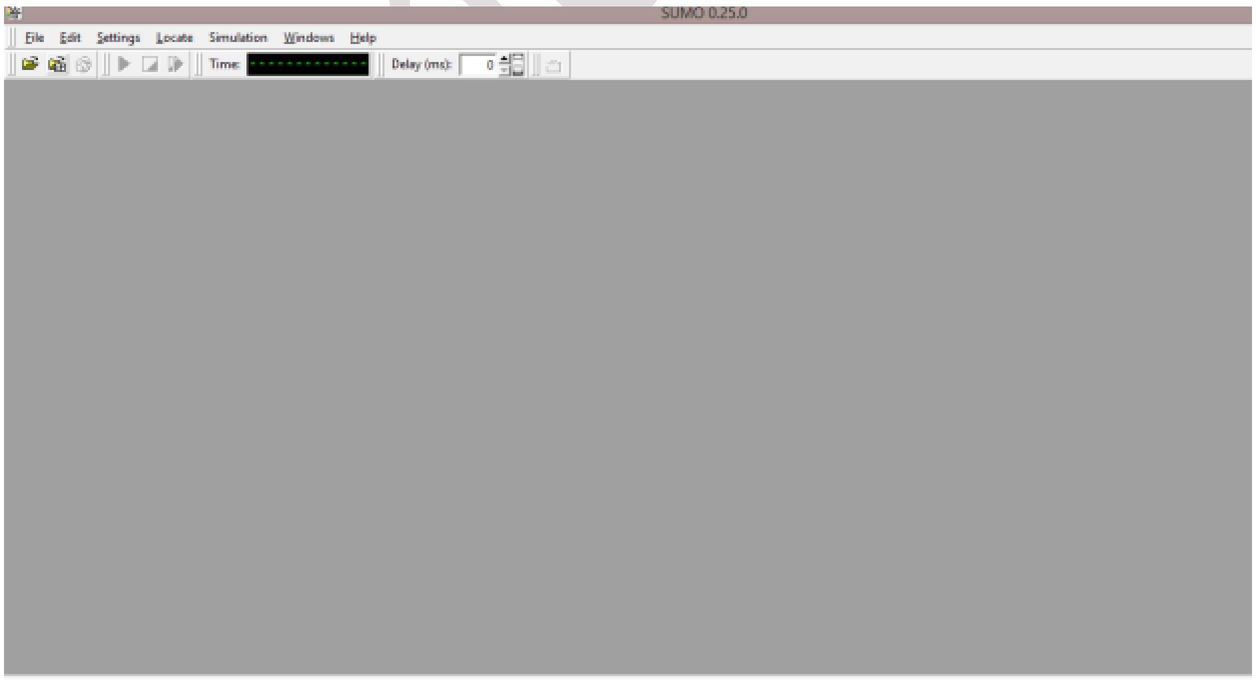
Computer ▸ System Reserved (C:) ▸ sumo ▸ sumo-0.25.0 ▸ bin

Organize ▾ Include in library ▾ Share with ▾ Burn New folder

Favorites: Desktop, Downloads, Recent Places, OneDrive
Libraries: Documents, Music, Pictures, Videos
Computer: System Reserved (C:), Local Disk (D:), Removable Disk (F:)
Network

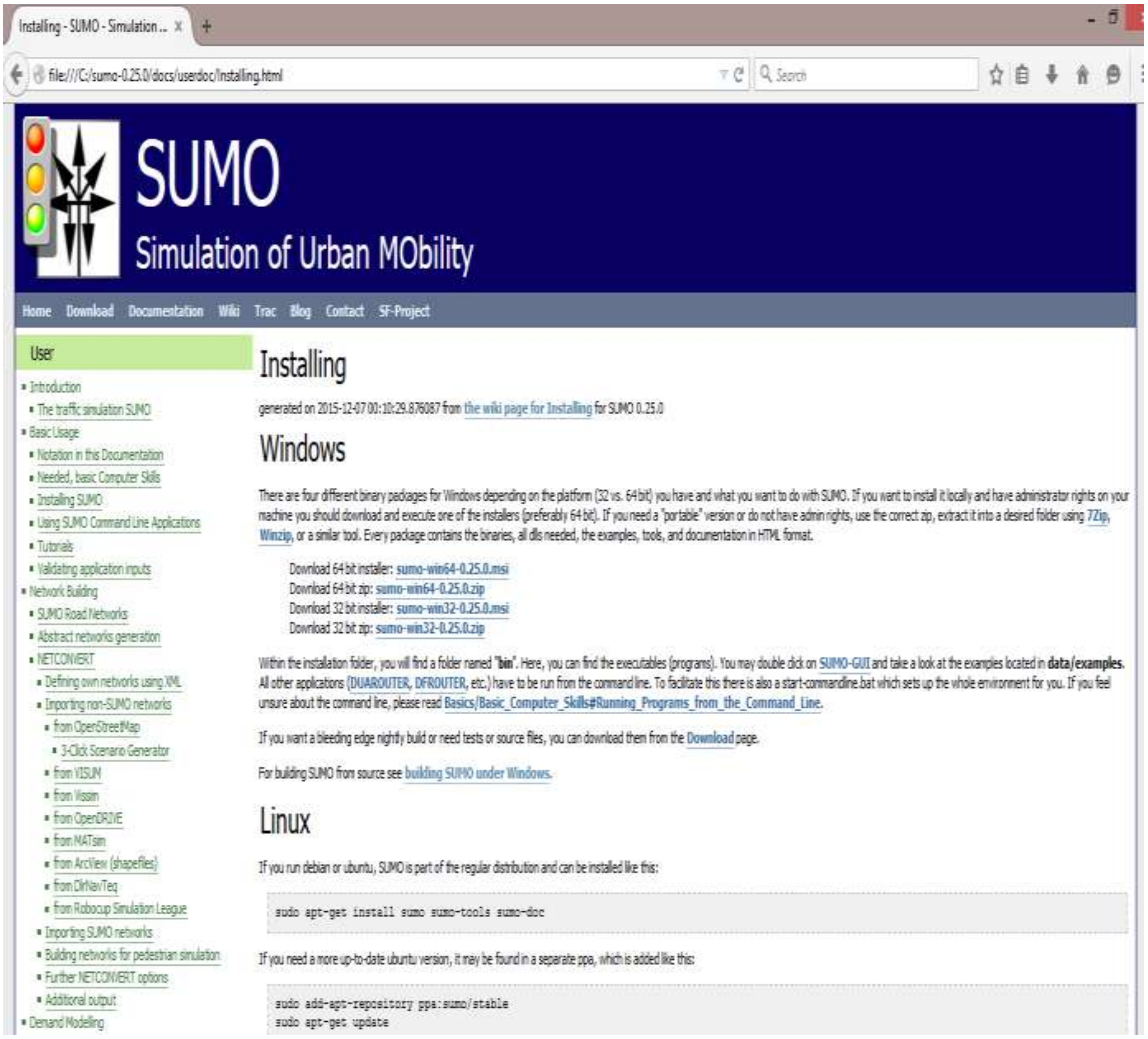
Name	Date modified	Type	Size
activitygen.exe	12/7/2015 2:45 AM	Application	734 KB
dfrouter.exe	12/7/2015 2:44 AM	Application	801 KB
duarouter.exe	12/7/2015 2:44 AM	Application	1,021 KB
emissionsDrivingCycle.exe	12/7/2015 2:49 AM	Application	521 KB
emissionsMap.exe	12/7/2015 2:48 AM	Application	419 KB
FOXDLL-1.6.dll	11/2/2012 9:36 AM	Application extens...	2,969 KB
gdal18.dll	5/31/2011 12:27 PM	Application extens...	7,715 KB
jtrrouter.exe	12/7/2015 2:44 AM	Application	921 KB
libpng15.dll	11/1/2012 4:48 PM	Application extens...	345 KB
Makefile.am	12/7/2015 2:49 AM	AM File	8 KB
Makefile.in	12/7/2015 2:49 AM	IN File	20 KB
marouter.exe	12/7/2015 2:45 AM	Application	922 KB
netconvert.exe	12/7/2015 2:46 AM	Application	2,443 KB
netedit.exe	12/7/2015 2:48 AM	Application	3,714 KB
netgenerate.exe	12/7/2015 2:44 AM	Application	1,335 KB
od2trips.exe	12/7/2015 2:44 AM	Application	526 KB
polyconvert.exe	12/7/2015 2:44 AM	Application	678 KB
proj.dll	5/31/2011 9:12 AM	Application extens...	302 KB
start-command-line.bat	9/24/2015 1:01 AM	Windows Batch File	1 KB
sumo.exe	12/7/2015 2:46 AM	Application	2,368 KB
sumo-gui.exe	12/7/2015 2:47 AM	Application	3,609 KB
SUMOTrafficModeler.jar	12/7/2015 2:49 AM	WinRAR archive	814 KB
TraaS.jar	12/7/2015 2:49 AM	WinRAR archive	72 KB
traci4j.jar	12/7/2015 2:49 AM	WinRAR archive	218 KB
traci4j-sources.jar	12/7/2015 2:49 AM	WinRAR archive	117 KB
TraciTestClient.exe	12/7/2015 2:43 AM	Application	113 KB
xerces-c_3_1.dll	4/23/2010 3:31 PM	Application extens...	2,470 KB

Then, go to folder ***C:/sumo/sumo-0.25.0/bin*** and double click on ***sumo-gui***to verify that the program runs correctly, as it is shown in Figure below



For more information

- ➢ Open **docs** directory
- ➢ Open **userdoc** directory and
- ➢ Open **index.html** in your browser and you can learn more about SUMO


Installing - SUMO - Simulation ...
file:///C:/sumo-0.25.0/docs/userdoc/Installing.html
Search
SUMO
Simulation of Urban MObility
Home Download Documentation Wiki Trac Blog Contact SF-Project
User
Introduction
The traffic simulation SUMO
Basic Usage
Notation in this Documentation
Needed, basic Computer Skills
Installing SUMO
Using SUMO Command Line Applications
Tutorials
Validating application inputs
Network Building
SUMO Road Networks
Abstract networks generation
NETCONVERT
Defining own networks using XML
Importing non-SUMO networks
from OpenStreetMap
3-Click Scenario Generator
from VISUM
from Vissim
from OpenDRIVE
from MATsim
from ArcView (shapefiles)
from DlrNavTeq
from Robocup Simulation League
Importing SUMO networks
Building networks for pedestrian simulation
Further NETCONVERT options
Additional output
Demand Modelling

Installing

generated on 2015-12-07 00:10:29.876087 from the wiki page for Installing for SUMO 0.25.0

Windows

There are four different binary packages for Windows depending on the platform (32 vs. 64bit) you have and what you want to do with SUMO. If you want to install it locally and have administrator rights on your machine you should download and execute one of the installers (preferably 64 bit). If you need a "portable" version or do not have admin rights, use the correct zip, extract it into a desired folder using 7Zip, Winzip, or a similar tool. Every package contains the binaries, all dlls needed, the examples, tools, and documentation in HTML format.

Download 64 bit installer: sumo-win64-0.25.0.msi
Download 64 bit zip: sumo-win64-0.25.0.zip
Download 32 bit installer: sumo-win32-0.25.0.msi
Download 32 bit zip: sumo-win32-0.25.0.zip

Within the installation folder, you will find a folder named "**bin**". Here, you can find the executables (programs). You may double click on SUMO-GUI and take a look at the examples located in **data/examples**. All other applications (DUAROUTER, DFROUTER, etc.) have to be run from the command line. To facilitate this there is also a start-commandline.bat which sets up the whole environment for you. If you feel unsure about the command line, please read Basics/Basic_Computer_Skills#Running_Programs_from_the_Command_Line.

If you want a bleeding edge nightly build or need tests or source files, you can download them from the Download page.

For building SUMO from source see building SUMO under Windows.

Linux

If you run debian or ubuntu, SUMO is part of the regular distribution and can be installed like this:

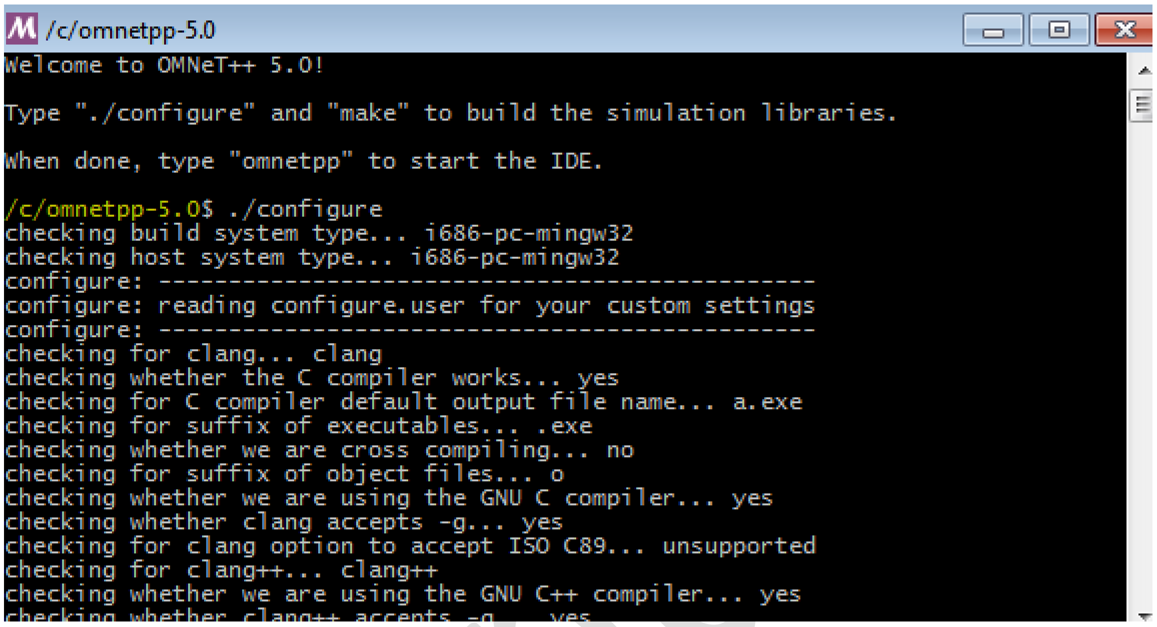
```
sudo apt-get install sumo sumo-tools sumo-doc
```

If you need a more up-to-date ubuntu version, it may be found in a separate ppa, which is added like this:

```
sudo add-apt-repository ppa:sumo/stable
sudo apt-get update
```

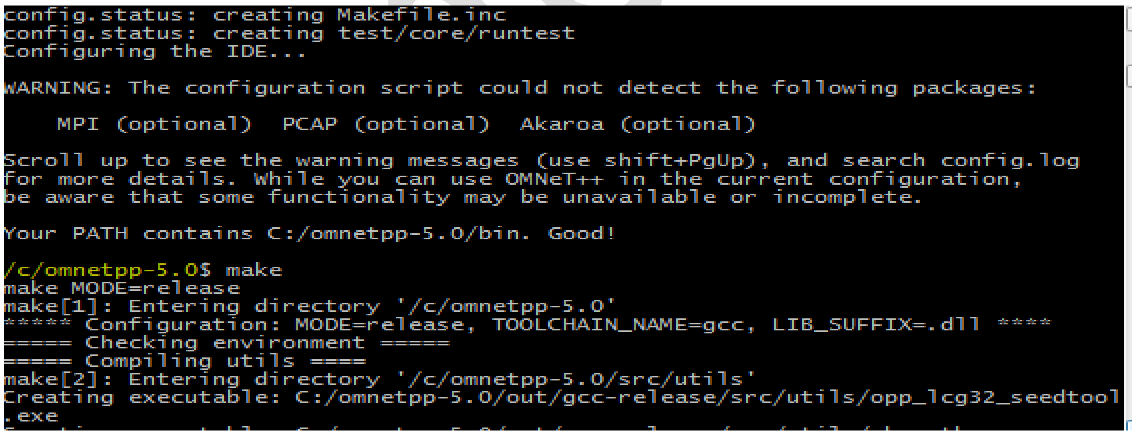
3. Installation of OMNeT

- Extract the zip file *omnet-5.0-src-windows* which you have downloaded in **C:\**
- Open **omnetpp-5.0** folder
- Click on **mingwenv.cmd** and wait for some time
- Write **./configure** command as the figure below



```
/c/omnetpp-5.0
Welcome to OMNeT++ 5.0!

Type "./configure" and "make" to build the simulation libraries.

When done, type "omnetpp" to start the IDE.

/c/omnetpp-5.0$ ./configure
checking build system type... i686-pc-mingw32
checking host system type... i686-pc-mingw32
configure: ---------------------------------------------
configure: reading configure.user for your custom settings
configure: ---------------------------------------------
checking for clang... clang
checking whether the C compiler works... yes
checking for C compiler default output file name... a.exe
checking for suffix of executables... .exe
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether clang accepts -g... yes
checking for clang option to accept ISO C89... unsupported
checking for clang++... clang++
checking whether we are using the GNU C++ compiler... yes
checking whether clang++ accepts -g... yes
```

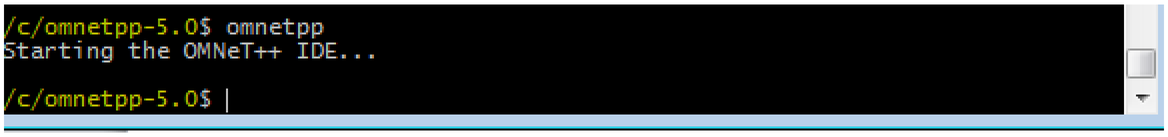
- Write **make** command



```
config.status: creating Makefile.inc
config.status: creating test/core/runtest
Configuring the IDE...

WARNING: The configuration script could not detect the following packages:

    MPI (optional)  PCAP (optional)  Akaroa (optional)

Scroll up to see the warning messages (use shift+PgUp), and search config.log
for more details. While you can use OMNeT++ in the current configuration,
be aware that some functionality may be unavailable or incomplete.

Your PATH contains C:/omnetpp-5.0/bin. Good!

/c/omnetpp-5.0$ make
make MODE=release
make[1]: Entering directory '/c/omnetpp-5.0'
***** Configuration: MODE=release, TOOLCHAIN_NAME=gcc, LIB_SUFFIX=.dll ****
===== Checking environment =====
===== Compiling utils ====
make[2]: Entering directory '/c/omnetpp-5.0/src/utils'
Creating executable: C:/omnetpp-5.0/out/gcc-release/src/utils/opp_lcg32_seedtool
.exe
```

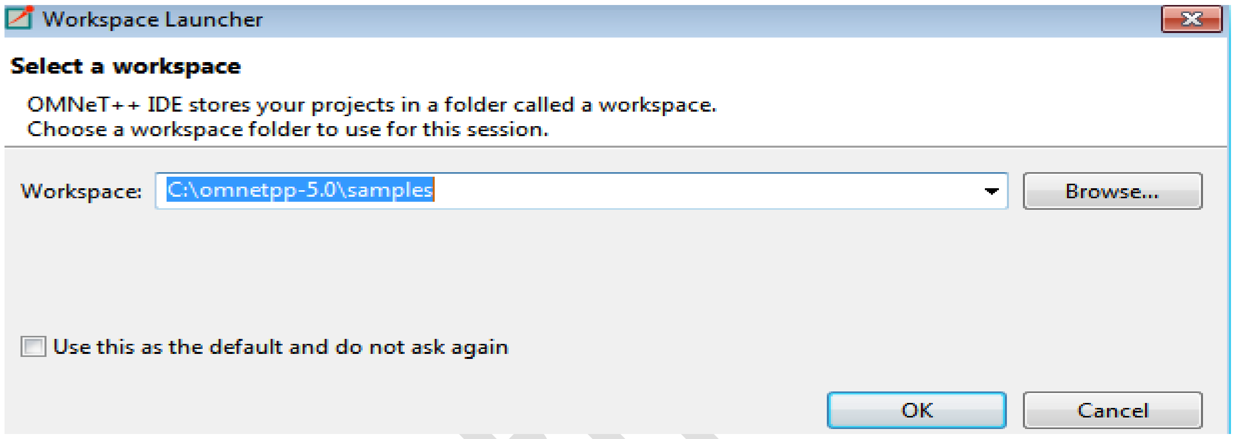
- Now, you can launch the IDE by typing the next command in the console ***mingwenv.cmd***

Omnetpp



```
/c/omnetpp-5.0$ omnetpp
Starting the OMNeT++ IDE...

/c/omnetpp-5.0$ |
```

- *Then you can create your workspace and click* ***ok***



Workspace Launcher

Select a workspace

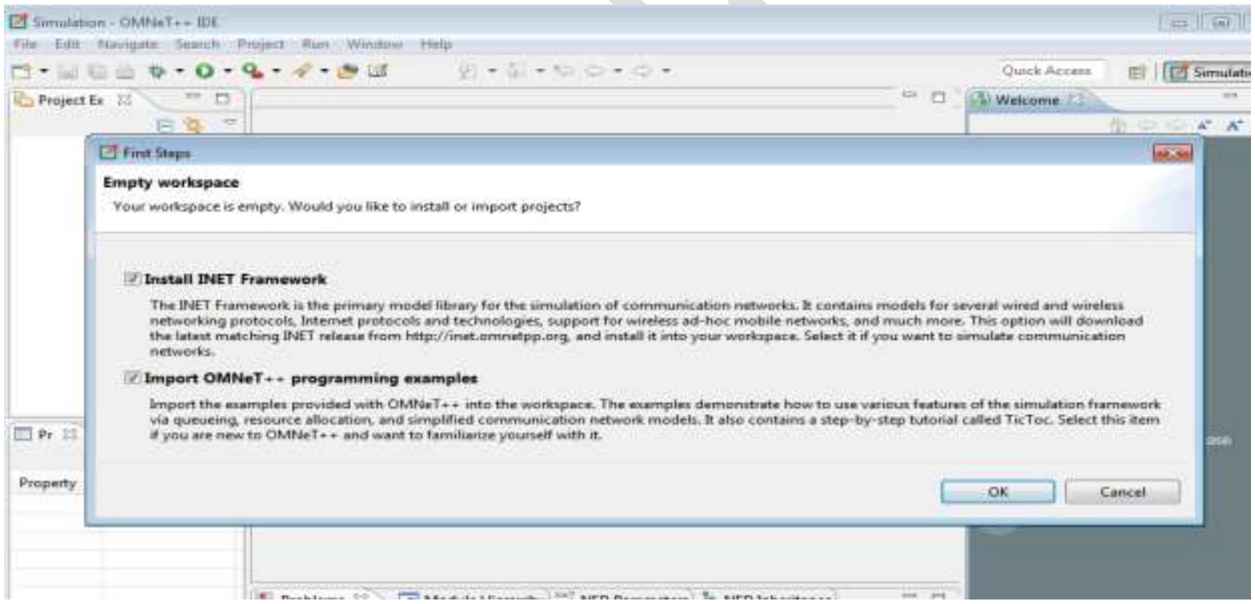
OMNeT++ IDE stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\omnetpp-5.0\samples

[ ] Use this as the default and do not ask again

-

Then, a window will appear that will allow you to install the INET framework and import the OMNeT++ examples. So, press ***Ok*** and the program will download and install, as it is shown in Figure B.23. The OMNeT++ was installed successful



First Steps

Empty workspace

Your workspace is empty. Would you like to install or import projects?

[✓] **Install INET Framework**

The INET Framework is the primary model library for the simulation of communication networks. It contains models for several wired and wireless networking protocols, Internet protocols and technologies, support for wireless ad-hoc mobile networks, and much more. This option will download the latest matching INET release from http://inet.omnetpp.org, and install it into your workspace. Select it if you want to simulate communication networks.

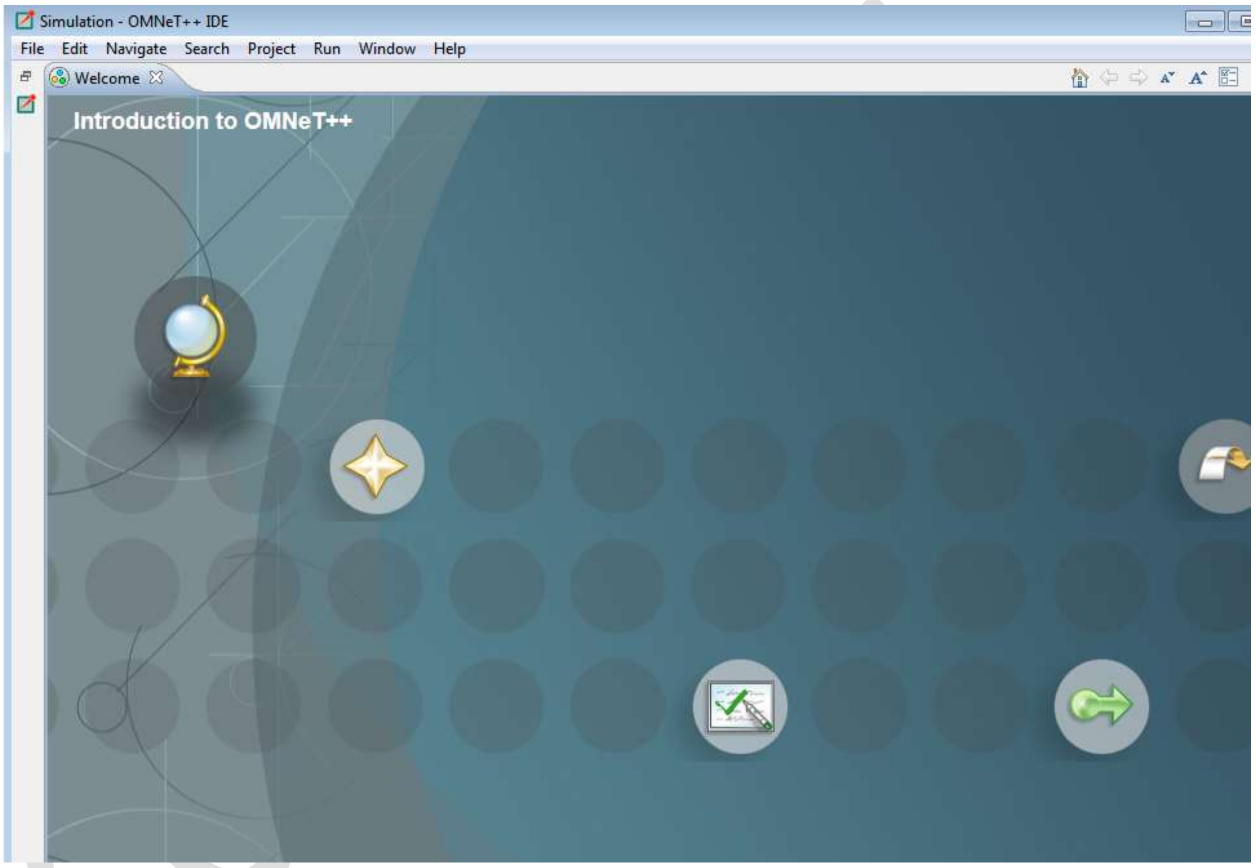
[✓] **Import OMNeT++ programming examples**

Import the examples provided with OMNeT++ into the workspace. The examples demonstrate how to use various features of the simulation framework via queueing, resource allocation, and simplified communication network models. It also contains a step-by-step tutorial called TicToc. Select this item if you are new to OMNeT++ and want to familiarize yourself with it.

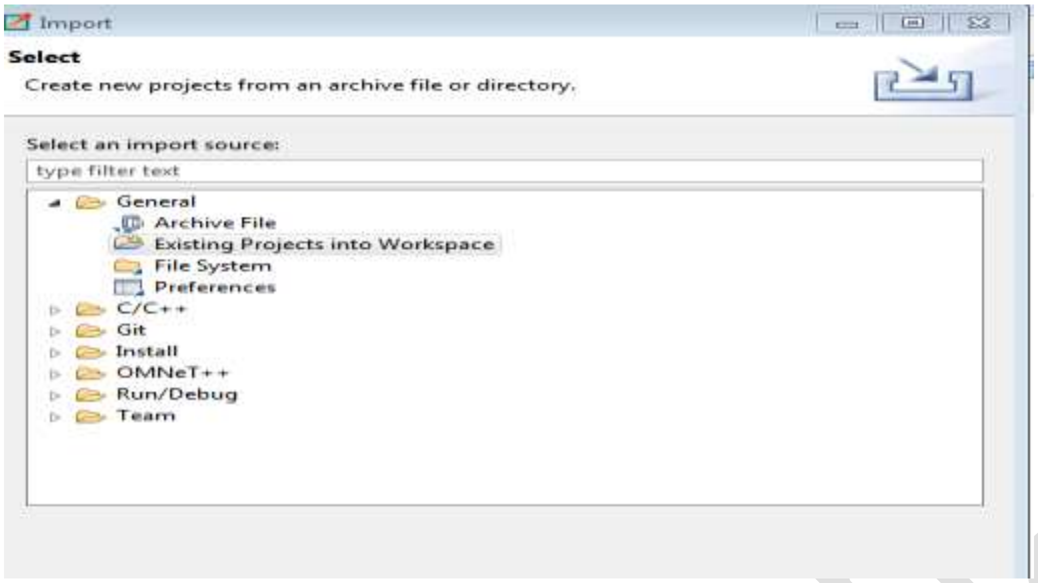
4. Installation of Veins

4.1 Import VEINS in OMNeT++ IDE

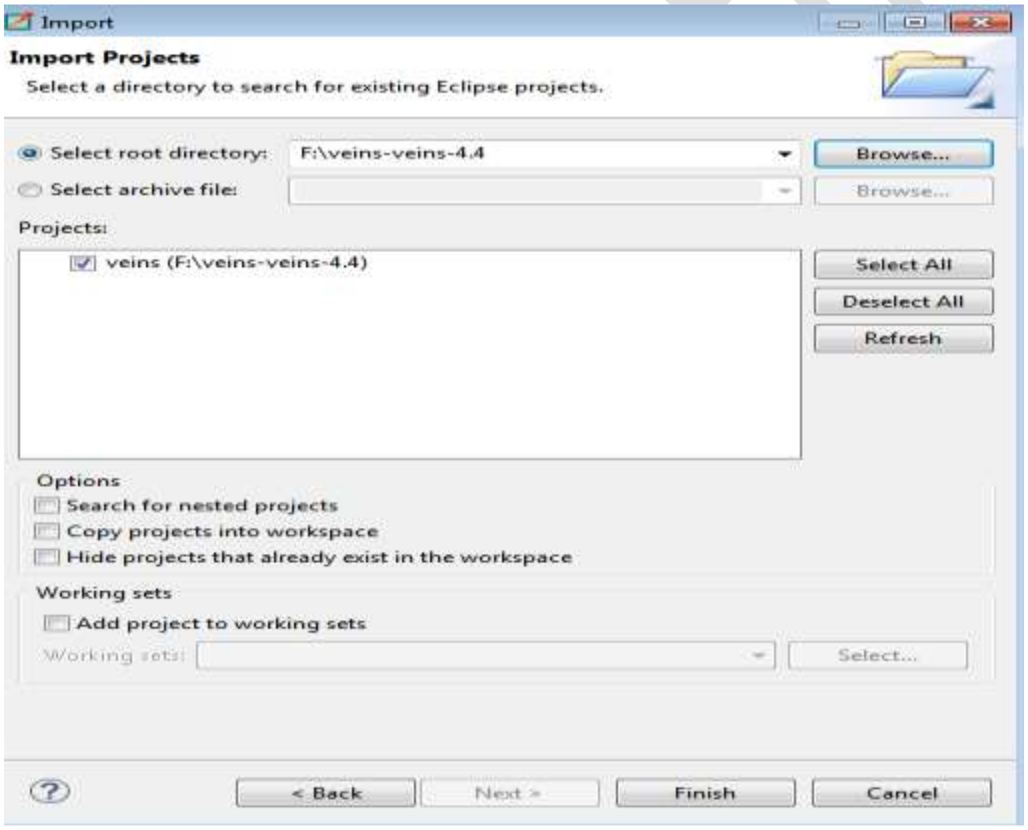
- Extract the zip file ***veins-4.4***in the **C:\sumo**folder that you created before
- Then, launch OMNeT++ IDE by typing the command ***omnetpp***in the terminal as you have done
  before during installation of OMNeT++.
- import the project into your OMNeT++ IDE workspace by clicking: ***File*** > ***Import***



A window will appear and you must select ***General: Existing Projects into Workspace*** and press ***Next*** >



Then, press ***Finish*** and VEINS will be imported successfully, as it is shown in Figure below



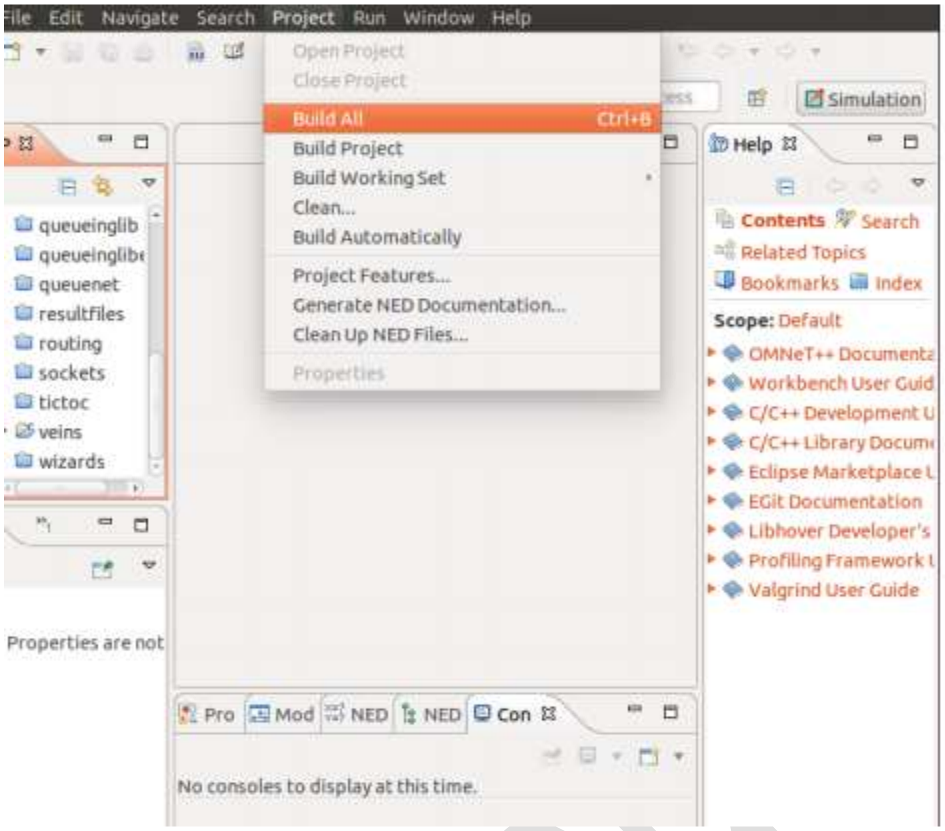
Select root directory: F:\veins-veins-4.4

Projects: veins (F:\veins-veins-4.4) [✓]

Options
- Search for nested projects [ ]
- Copy projects into workspace [ ]
- Hide projects that already exist in the workspace [ ]

Working sets
- Add project to working sets [ ]

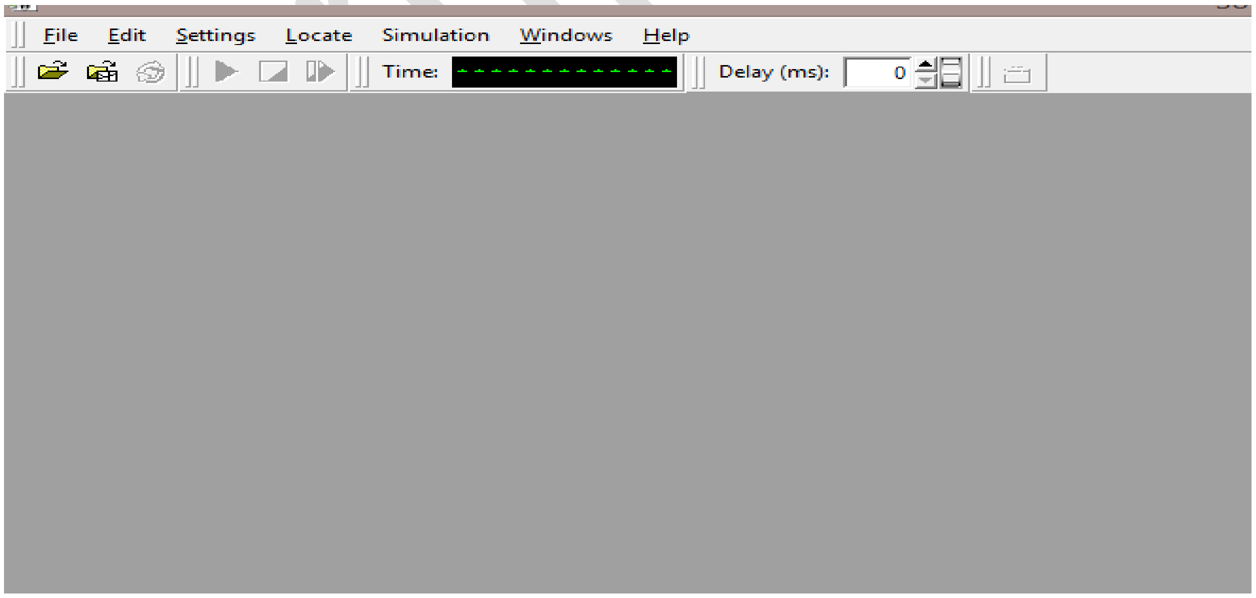
Finally, you must build the project by clicking ***Project > Build All*** in OMNeT++



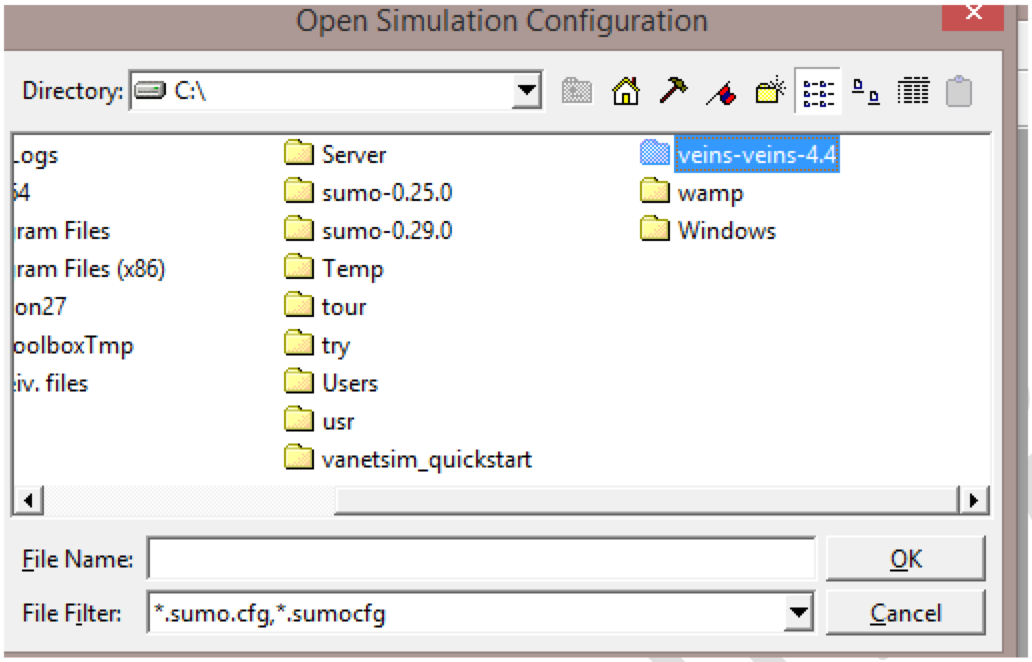
No consoles to display at this time.

4.2 Run the Veins demo scenario

First, you should be sure that SUMO is working correctly. so go to **C:\sumo-0.25.0\bin** and click **Sumo-GUI.exe.**



Then click on file->**open**

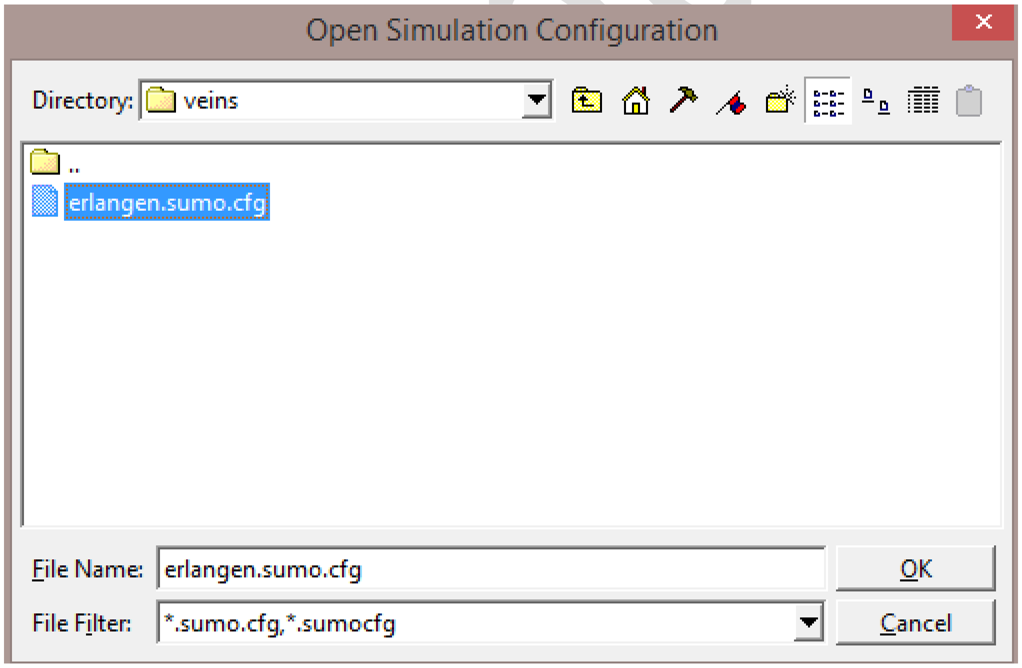

Open Simulation Configuration

Directory: C:\

Logs
ram Files
ram Files (x86)
on27
oolboxTmp
iv. files
Server
sumo-0.25.0
sumo-0.29.0
Temp
tour
try
Users
usr
vanetsim_quickstart
veins-veins-4.4
wamp
Windows

File Name:
File Filter: *.sumo.cfg,*.sumocfg

Then click **veins-veins-4.4->examples->veins**

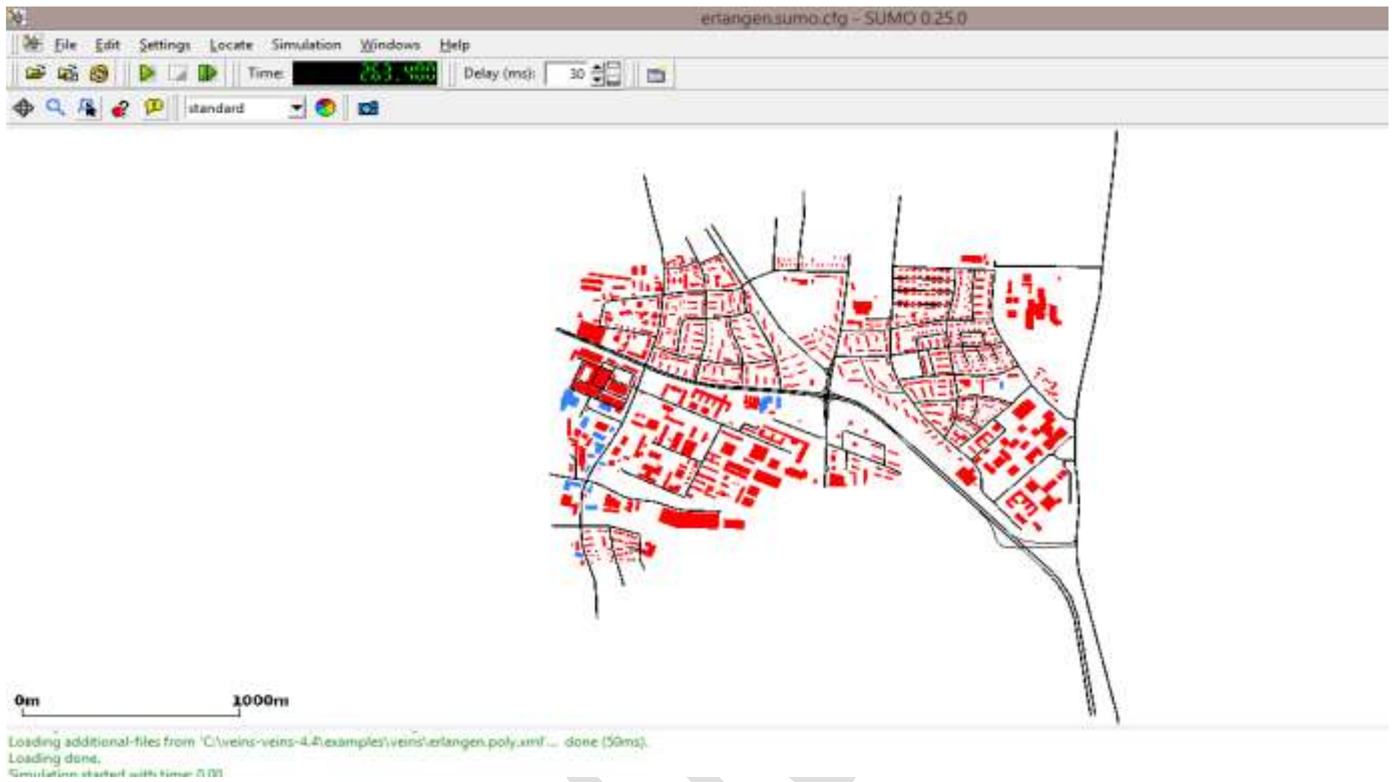

Open Simulation Configuration

Directory: veins

..
erlangen.sumo.cfg

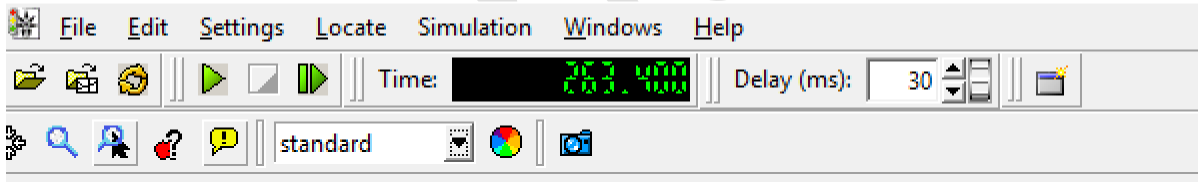
File Name: erlangen.sumo.cfg
File Filter: *.sumo.cfg,*.sumocfg

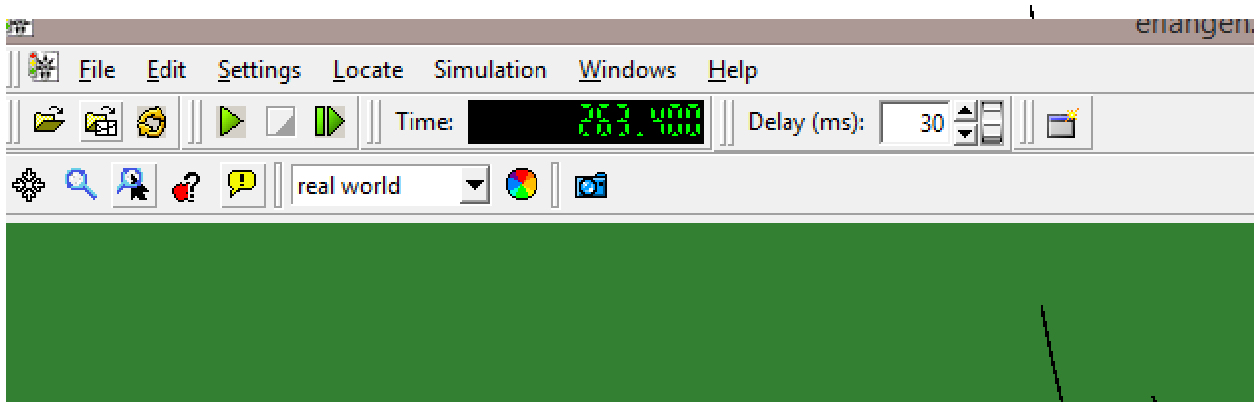
Then you can make the **delay** as you want in our case 30ms



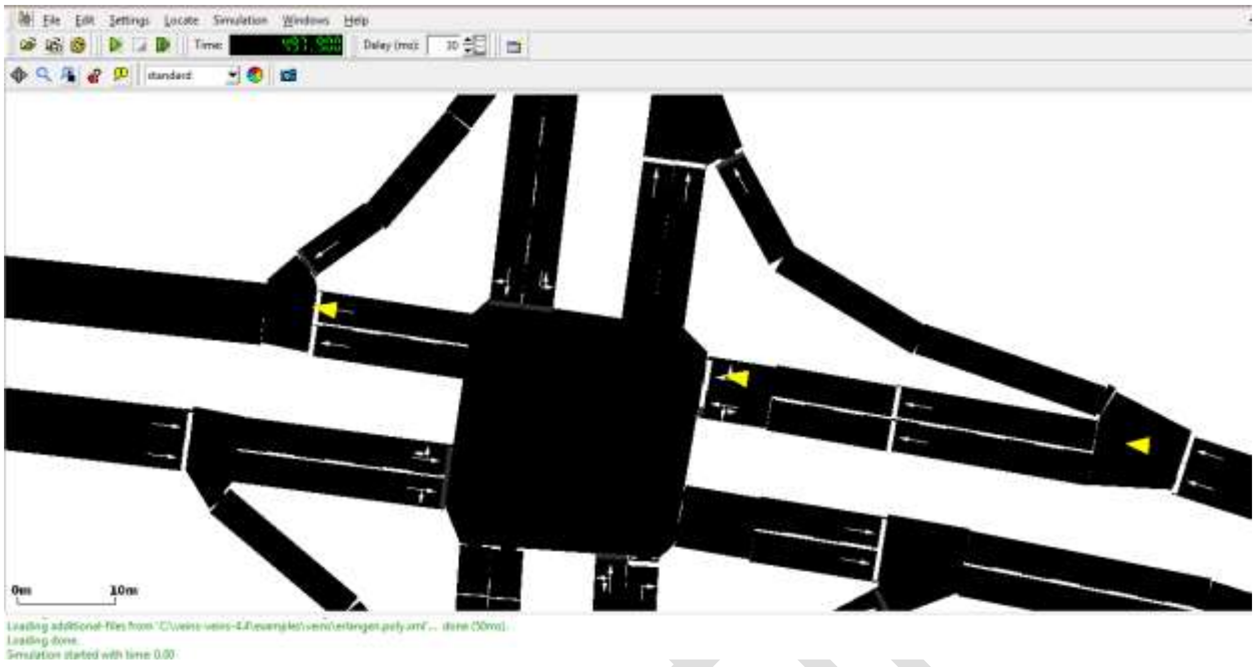
```
Loading additional-files from 'C:\veins-veins-4.4\examples\veins\erlangen.poly.xml' ... done (50ms).
Loading done.
Simulation started with time: 0.00
```

For more clarity change **standard** in to **real world** in the tab





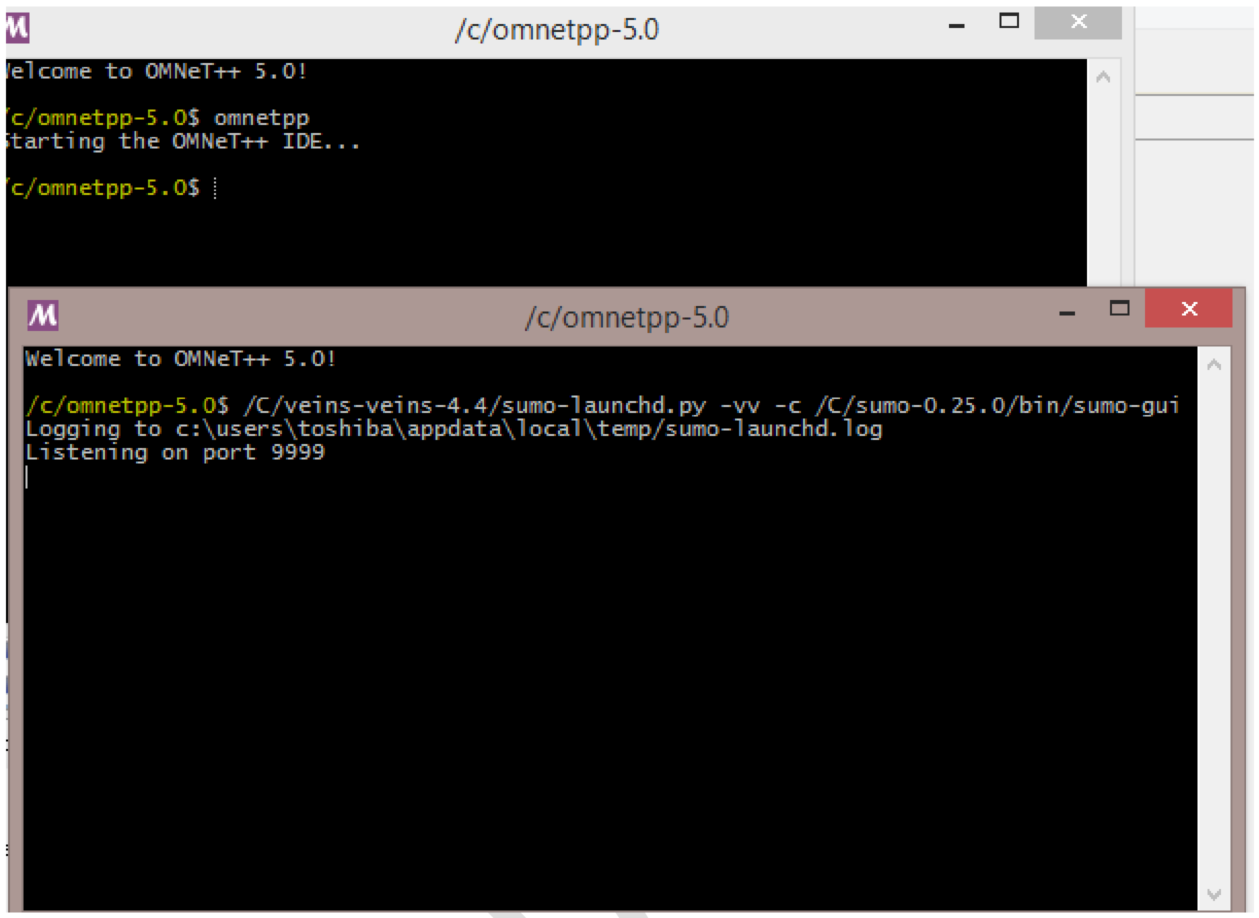
Then you will see like the following



```
Loading additional-files from 'C:\veins-veins-4.4\examples\veins\erlangen.poly.xml'... done (50ms).
Loading done.
Simulation started with time: 0.00
```

Then, you get an impression of example scenario looks like, as it is illustrated in Figure above. As a result, you can conclude that SUMO is working correctly.

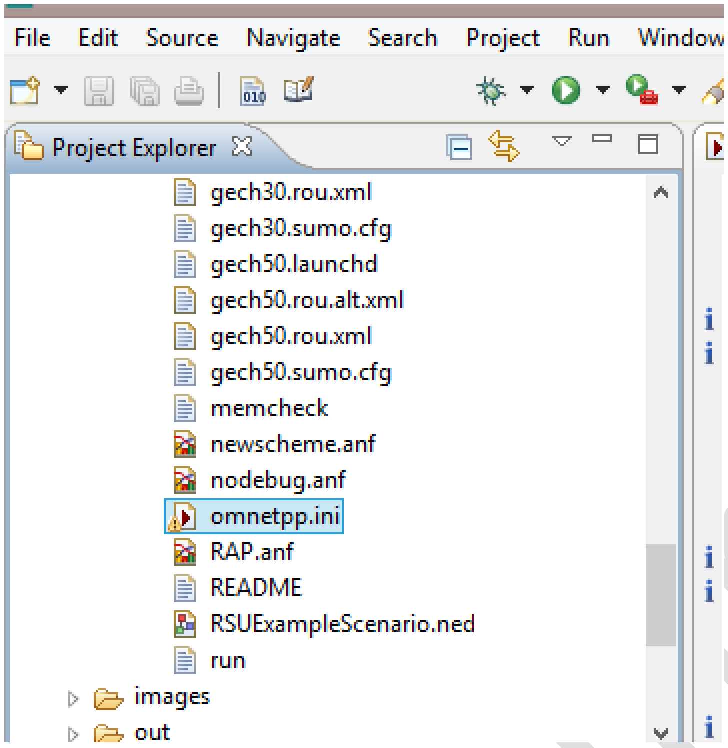
- ➢ The next step is to run SUMO and OMNeT simultaneously. VEINS comes with a small python script wills proxy TCP connections between OMNeT++ and SUMO. To do that this script starts a new copy of the SUMO simulation for every OMNeT++ simulation connecting.
    - ❖ Go to **C:\omnetpp-5.0** and open **mingwenv**
    - ❖ Then start OMNET++ ide by typing **omnetpp** as follows
    - ❖ Write the command in **mingwenv** terminal /C/veins-veins-4.4/sumo-launchd.py -vv -c /C/sumo-0.25.0/bin/sumo-gui
    - ❖ The script will print Listening on port 9999 as figure below



```
/c/omnetpp-5.0
Welcome to OMNeT++ 5.0!

/c/omnetpp-5.0$ omnetpp
Starting the OMNeT++ IDE...

/c/omnetpp-5.0$
```

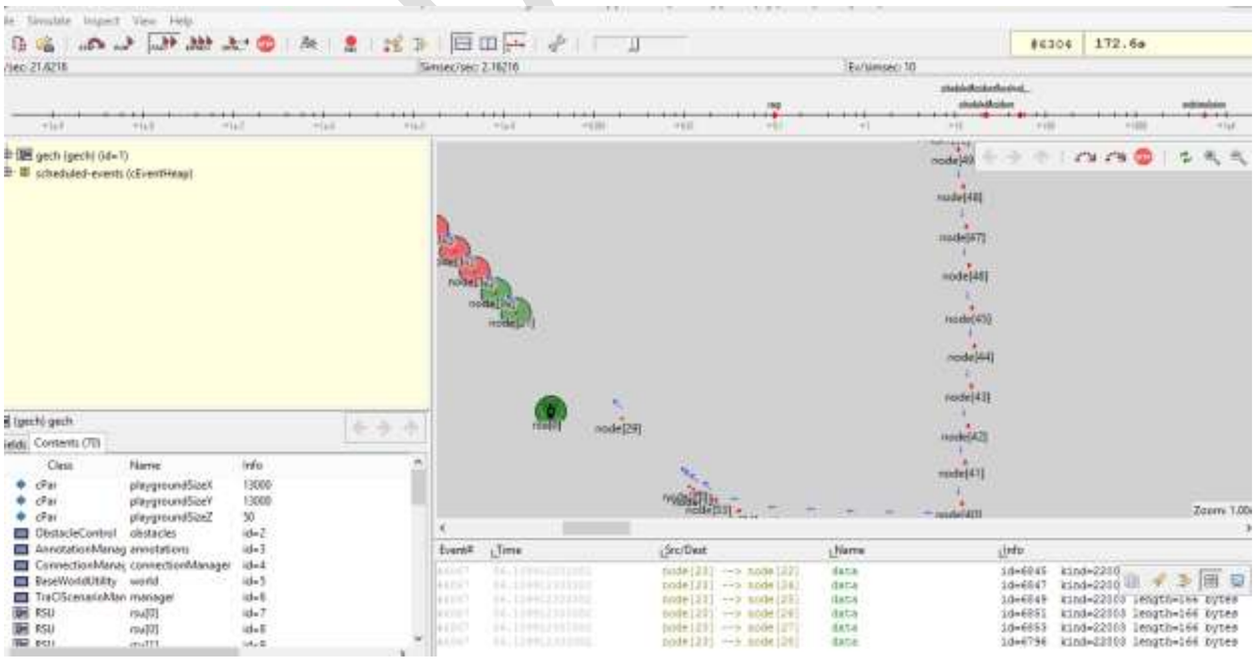
```
/c/omnetpp-5.0
Welcome to OMNeT++ 5.0!

/c/omnetpp-5.0$ /C/veins-veins-4.4/sumo-launchd.py -vv -c /C/sumo-0.25.0/bin/sumo-gui
Logging to c:\users\toshiba\appdata\local\temp/sumo-launchd.log
Listening on port 9999
```

- Then, you can simulate *the Veins demo scenario* in the *OMNeT++ IDE*, for both Windows and Linux users, right-click on ***veins/examples/veins/omnetpp.ini*** and choose ***Run As > 1 OMNeT++ simulation***



gech30.rou.xml
gech30.sumo.cfg
gech50.launchd
gech50.rou.alt.xml
gech50.rou.xml
gech50.sumo.cfg
memcheck
newscheme.anf
nodebug.anf
omnetpp.ini
RAP.anf
README
RSUExampleScenario.ned
run
images
out

If everything worked as intended this will give you a working simulation scenario using OMNeT++ and SUMO running in parallel it shows like the following figure



Class	Name	Info
cPar	playgroundSizeX	13000
cPar	playgroundSizeY	13000
cPar	playgroundSizeZ	50
ObstacleControl	obstacles	id=2
AnnotationManag	annotations	id=3
ConnectionManag	connectionManager	id=4
BaseWorldUtility	world	id=5
TraCIScenarioMan	manager	id=6
RSU	rsu[0]	id=7
RSU	rsu[0]	id=8

Event#	Time	Src/Dest	Name	Info
[illegible]	[illegible]	node[23] --> node[22]	data	id=6845 kind=2200
[illegible]	[illegible]	node[23] --> node[24]	data	id=6847 kind=2200
[illegible]	[illegible]	node[23] --> node[25]	data	id=6849 kind=22003 length=166 bytes
[illegible]	[illegible]	node[23] --> node[26]	data	id=6851 kind=22003 length=166 bytes
[illegible]	[illegible]	node[23] --> node[27]	data	id=6853 kind=22003 length=166 bytes
[illegible]	[illegible]	node[23] --> node[28]	data	id=6798 kind=22003 length=166 bytes

5. Example of simulation in VEINS

5.1 Importing networks and generation of routes in SUMO

SUMO offers the possibility to import real network topologies for simulation, which is an important advantage, since it is a very interesting possibility. The user can use real and concrete scenarios in order to study the behavior of a given IVC.

The SUMO simulator is able to import networks from several sources; however, it will use **OpenStreetMap**

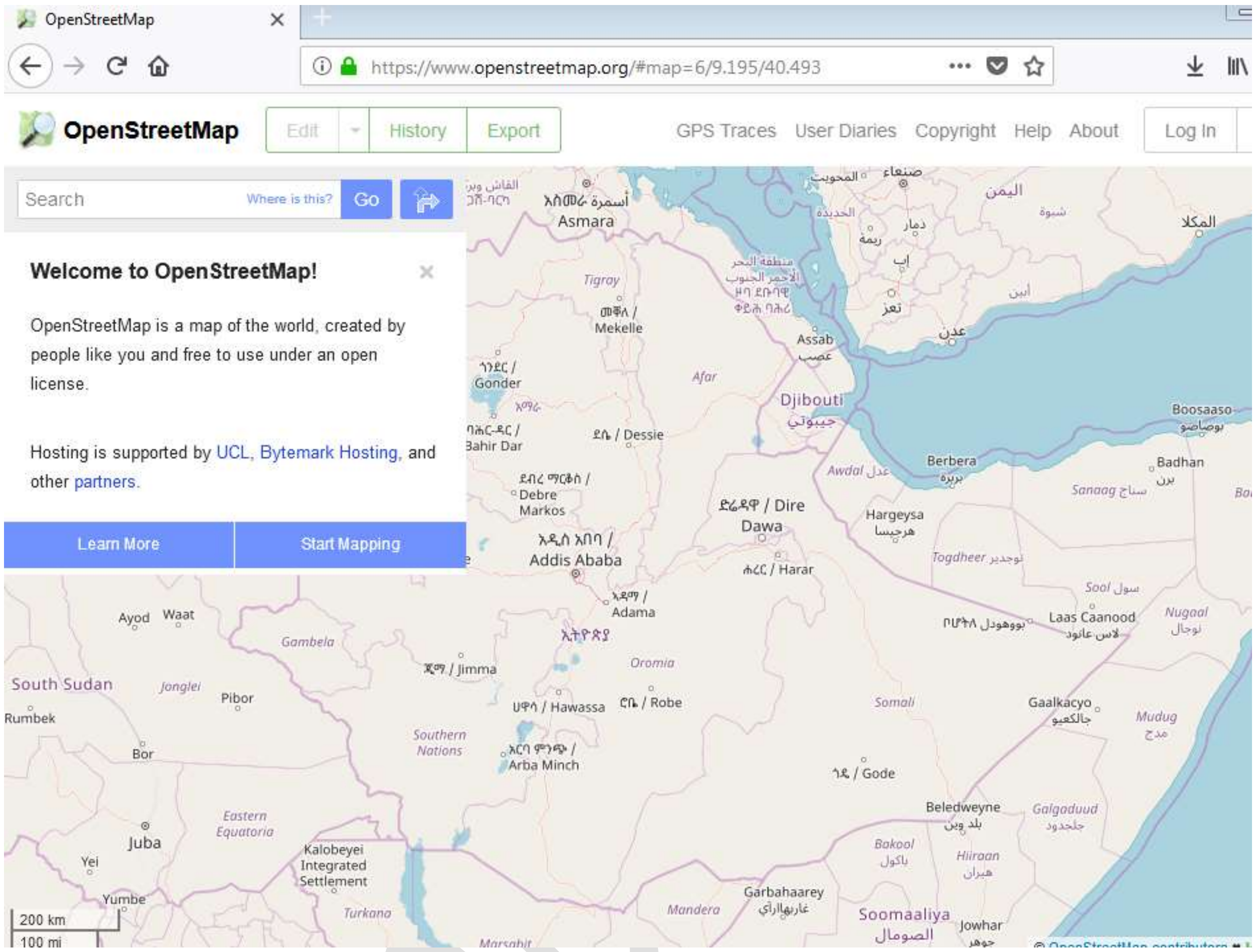
5.1.1 OpenStreetMap

The **OpenStreetMap** (OSM) project (www.openstreetmap.org) has collected an enormous amount of free spatial data and the database is growing every day. Many people want to use this data for their own GIS projects but have been hindered by the use of a non-standard data format in the OSM project.

The mapping from OSM data to other formats is not an exact science. OSM rules on how to map certain features are often not well defined and there is no mandatory quality control. This openness allows a lot of flexibility and is part of the reason why **OSM** has been able to collect so much data in such a short time frame, but it makes using the data more difficult.

5.1.2 Get a map from OpenStreetMap

Open your web browser and go to the web site: https://www.openstreetmap.org and search a city that you want, for example Ethiopia, Addis Abeba

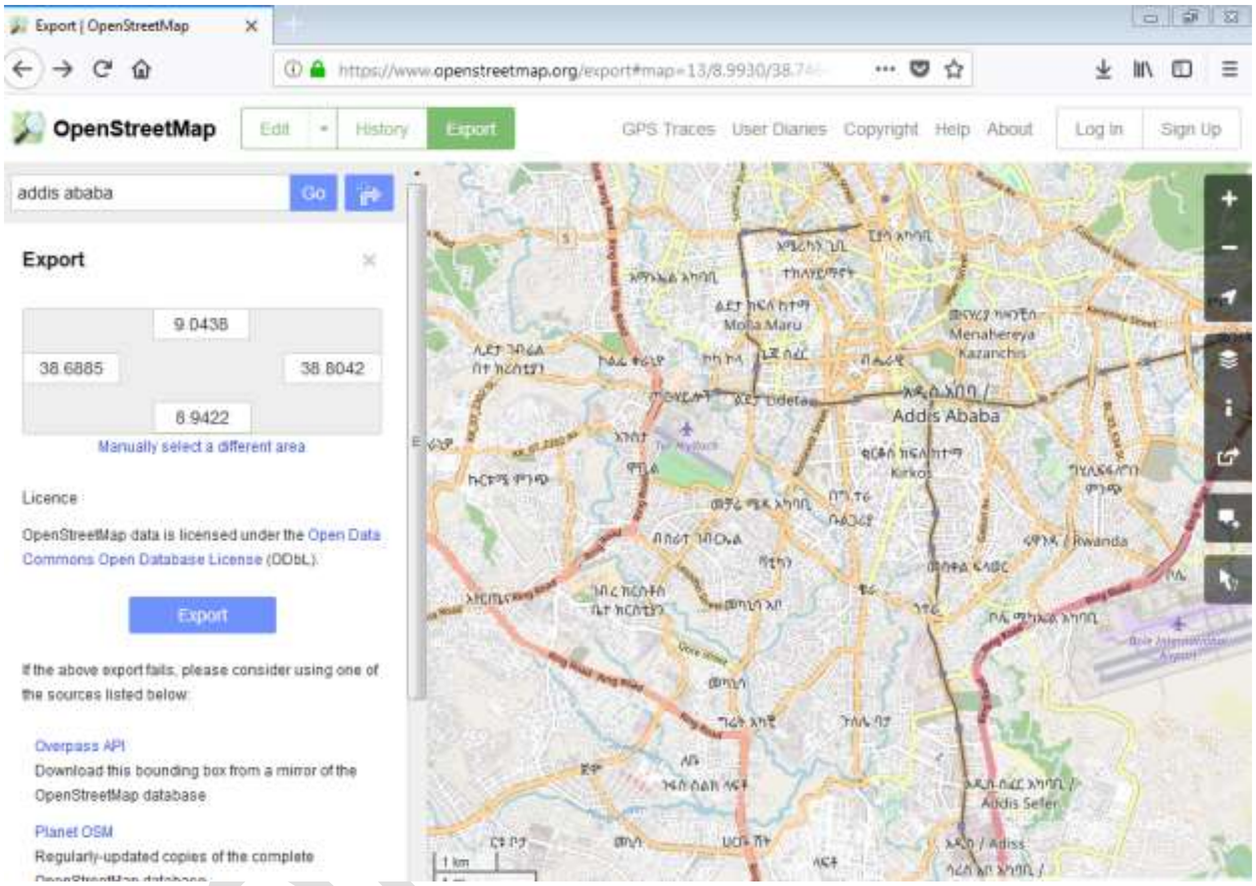


Welcome to OpenStreetMap!

OpenStreetMap is a map of the world, created by people like you and free to use under an open license.

Hosting is supported by UCL, Bytemark Hosting, and other partners.

Then, click on ***Export*** button which is located in top and after that click on ***manually select a different area.*** Choose the area which you want to export and click on ***Export*** button. A window appears asked if you want to open or save the map, you must click on ***save*** for download the file ***map.osm***.



addis ababa

Export

9.0438

38.6885 38.8042

8.9422

Manually select a different area

Licence

OpenStreetMap data is licensed under the Open Data Commons Open Database License (ODbL).

Export

If the above export fails, please consider using one of the sources listed below:

Overpass API
Download this bounding box from a mirror of the OpenStreetMap database

Planet OSM
Regularly-updated copies of the complete OpenStreetMap database

5.1.3 Preparation the map for use in SUMO

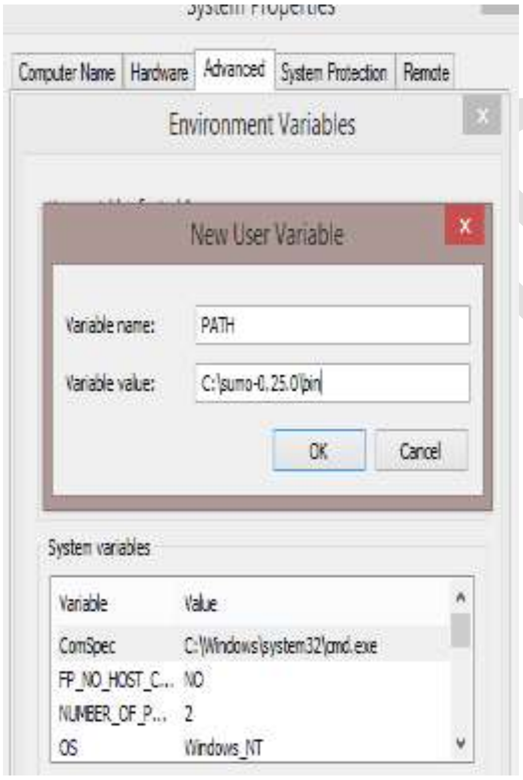
Copy the file *map.osm which you downloaded in previous section* in ***sumo/bin*** directory, after open a terminal and execute the following command in order to convert the map into a road network that is understood by SUMO.

- Here open terminal
- Navigate to sumo/bin: by typing **cd /sumo/bin**
- Write the command below :**netconvert --osm-files map.osm -o ab.net.**xml here **ab** is the name you given for the file.

Remark : you may get command line - sumo simulator is showing "SUMO_HOME variable not set

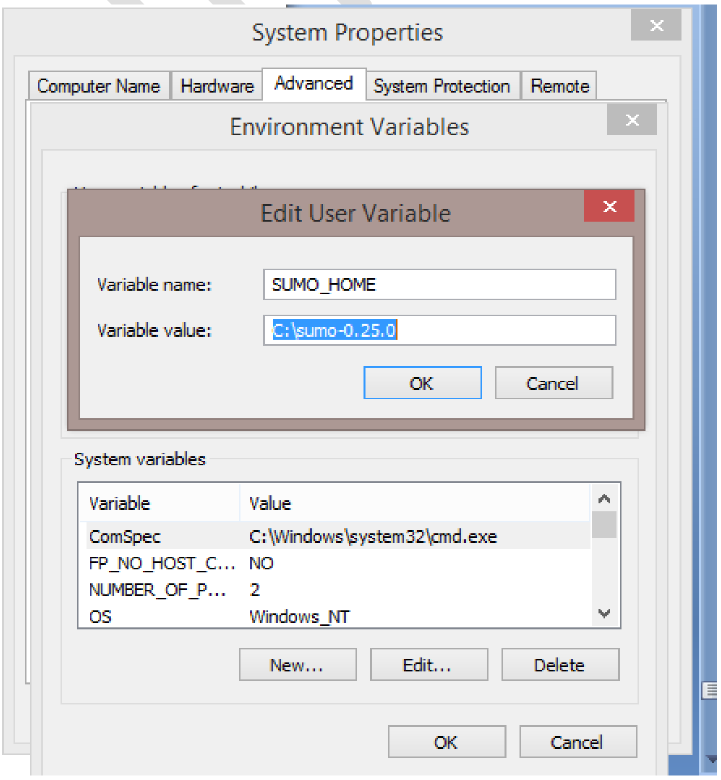
So you must set the environmental variable as follows

- Goto environmental variable
- Under user variable click on **new** button and fill like the following



Variable name: PATH
Variable value: C:\sumo-0.25.0\bin

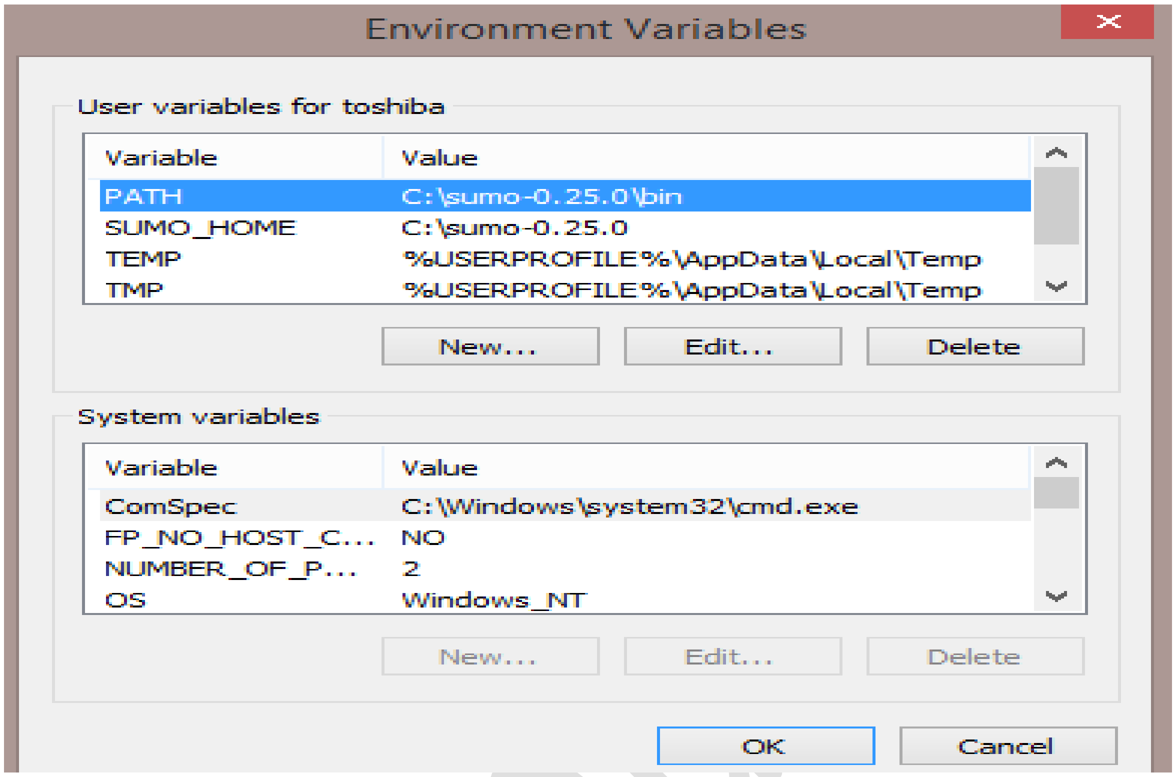
Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT



Variable name: SUMO_HOME
Variable value: C:\sumo-0.25.0

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

Then you will get the following



Variable	Value
PATH	C:\sumo-0.25.0\bin
SUMO_HOME	C:\sumo-0.25.0
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2
OS	Windows_NT

- Now, it is necessary to create the ***typemap.xml*** file in side **sumo/bin** folder

Then, open your web browser and go to the following web site:

- https://github.com/bluemix/SUMO/blob/master/typemap.xml and copy its contents into the typemap.xml file which you created before and save it.
- Then, go to the terminal and execute the next command in order to show the map correctly in SUMO**. cd C:/sumo/bin** directory
  polyconvert --net-file ab.net.xml --osm-files map.osm --type-file typemap.xml -o ab.poly.xml

5.1.4 Generation of routes in SUMO

- ✓ Having defined the network topology, it only remains to generate the so-called traffic demand, that is, the description of the routes that follow the vehicles.
- ✓ There are several methods to generate traffic demand in SUMO:
- ✓ Using Route definitions.
- ✓ Using definitions travel.
- ✓ Using definitions of flows (similar to above but uniting vehicles with similar travel in groups).
- ✓ Using definitions of flows at intersections rotation rate (the link target is not specified, and instead the probability of making turns at intersections shown).
- ✓ Using random routes.

In this case it will use random routes. There is a Python script developed with the aim of producing random routes, his name is ***randomTrips.py***. Currently, it is the most recommended method to achieve this functionality. However, note that the results are not always entirely realistic.
So, open a terminal, go to the **sumo/bin** folder, and type the next command.

python /sumo/sumo-0.25.0/tools/trip/randomTrips.py --net-file ab.net.xml --route-file ab.rou.xml --begin 0 --end 100 –length

5.1.5 Prepare files before simulating

In this step you must copy the files that you have generated in sections above from **sumo** folder to **veins** folder

Copy ab.net.xml ,ab.poly.xml and ab.rou.xml to veins-veins-4.4/examples/veins/

After that, you have to edit the configuration files of VEINS. So, go to the path */veins-veins-4.4/examples/veins/*. Then, open with ***notepad*** the files ***erlangen.launchd.xml*** and ***erlangen.sumo.cfg*** and write the name of files which you copied before i.e. ***ab.net.xml, ab.rou.xml, ab.poly.xml,***

Then use the example above to simulate the network

6. NS-2 Installation Manual

In this tutorial, we will see how to install NS2 2.34 in ubuntu 14.04 linux operating system. It is not a good idea to use direct commands in the terminal like

// These commands ok for 12.04 but not recommended for UBUNTU 14.04

sudo apt-get install ns2

sudo apt-get install nam

This results "Segmentation Fault and core dumped".

Remove both ns2 and nam using:

sudo apt-get remove ns2

sudo apt-get remove nam

Just follow the below step by step instructions to install successfully.

To install allinone version of NS2 :

STEP 1: Does we need to do anything before starting the installation? YES

Install all necessary dependencies using below commands one after another.

sudo apt-get install tcl8.5-dev tk8.5-dev
sudo apt-get install build-essential autoconf automake
sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev

STEP 2:

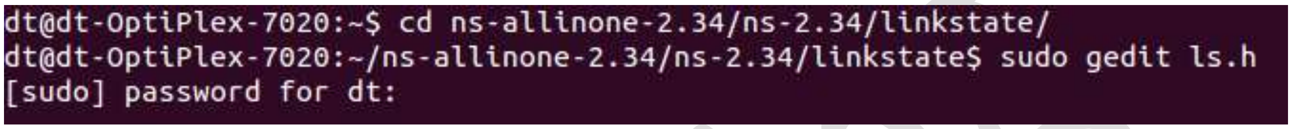
1. Download the NS2 Package from this link.
   https://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.34/ns-allinone-2.34.tar.gz/download

2. Copy the downloaded file to your /Home folder in ubuntu 14.04.
3. Right click on the file and select "Extract here" option. (You can also do this using command line).

STEP 3:

Now go to ns-allinone-2.34/ns-2.34/linkstate sub folder.

double click on "ls.h" file to open.



```
dt@dt-OptiPlex-7020:~$ cd ns-allinone-2.34/ns-2.34/linkstate/
dt@dt-OptiPlex-7020:~/ns-allinone-2.34/ns-2.34/linkstate$ sudo gedit ls.h
[sudo] password for dt:
```

go to line number 137 and change the below line

from

void eraseAll() { erase(baseMap::begin(), baseMap::end()); }

to

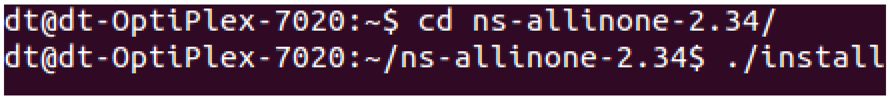
void eraseAll() { this->erase(baseMap::begin(), baseMap::end()); }

STEP 4:

Open the Terminal by pressing "ALT+CNTL+T" keys combination. And move to ns-allinne-2.34 folder from home through terminal

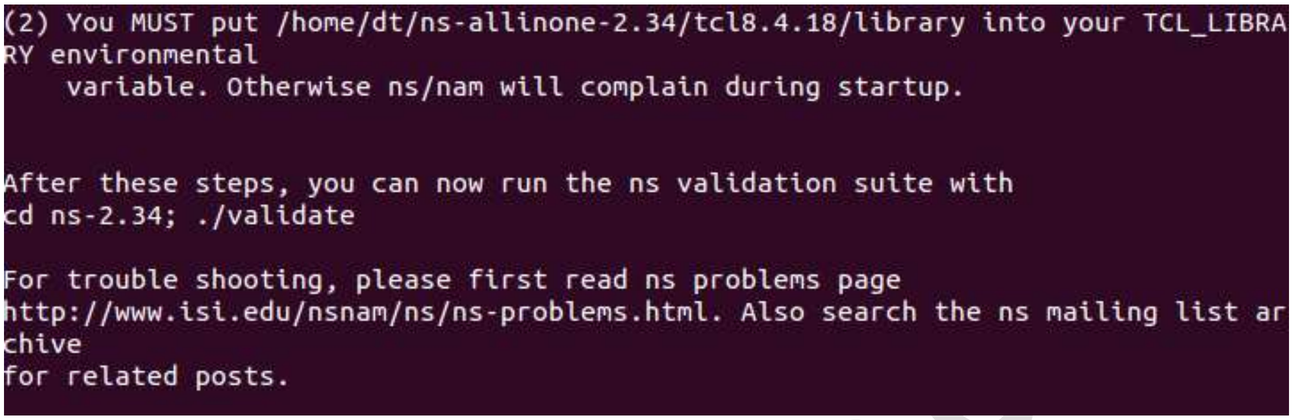
```
dt@dt-PC:~$ cd ns-allinone-2.34/
dt@dt-PC:~/ns-allinone-2.34$
```

Now type ./install on terminal



```
dt@dt-OptiPlex-7020:~$ cd ns-allinone-2.34/
dt@dt-OptiPlex-7020:~/ns-allinone-2.34$ ./install
```

If NS2 is installed successfully you will get:


```
(2) You MUST put /home/dt/ns-allinone-2.34/tcl8.4.18/library into your TCL_LIBRA
RY environmental
    variable. Otherwise ns/nam will complain during startup.


After these steps, you can now run the ns validation suite with
cd ns-2.34; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list ar
chive
for related posts.
```

If it is Not installed successfully you will get the following error:

1.

ld: libotcl.so: hidden symbol `__stack_chk_fail_local' isn't defined

ld: final link failed: Bad value

*make: *** [libotcl.so] Error 1*

otcl-1.13 make failed! Exiting ...

Solution:

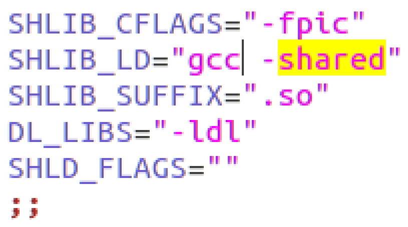
In *otcl-1.13/configure,* line number 6304

change :

SHLIB_LD="ld -shared"

TO :

SHLIB_LD="gcc -shared"


```
SHLIB_CFLAGS="-fpic"
SHLIB_LD="gcc -shared"
SHLIB_SUFFIX=".so"
DL_LIBS="-ldl"
SHLD_FLAGS=""
;;
```

2.

tools/ranvar.cc: In member function ‘virtual double GammaRandomVariable::value()’:
tools/ranvar.cc:219:70: error: cannot call constructor ‘GammaRandomVariable::
tools/ranvar.cc:219:70: error: for a function-style cast, remove the redundant
‘::GammaRandomVariable’ [-fpermissive]
*make: *** [tools/ranvar.o] Error 1*

Solution:
In ns-2.34/tools/ranvar.cc, line 219
Change:
return GammaRandomVariable:: GammaRandomVariable(1.0 + alpha_, beta_).value() * pow (u, 1.0 / alpha_);

TO:
return GammaRandomVariable(1.0 + alpha_, beta_).value() * pow (u, 1.0 / alpha_);

3.

In file included from mac/mac-802_11Ext.cc:66:0:
mac/mac-802_11Ext.h: In member function ‘u_int32_t PHY_MIBExt::getHdrLen11()’:
mac/mac-802_11Ext.h:175:19: error: expected primary-expression before ‘struct’
mac/mac-802_11Ext.h:175:41: error: ‘dh_body’ was not declared in this scope
mac/mac-802_11Ext.h:175:51: error: ‘offsetof’ was not declared in this scope
mac/mac-802_11Ext.h:177:3: warning: control reaches end of non-void function [-Wreturn-type]
*make: *** [mac/mac-802_11Ext.o] Error 1*
Ns make failed!

Solution:
In mac/mac-802_Ext.h, line 65 add the following header:

#include<cstddef>

4.

mobile/nakagami.cc: In member function 'virtual double Nakagami::Pr(PacketStamp, PacketStamp*, WirelessPhy*)':*
mobile/nakagami.cc:183:73: error: cannot call constructor 'ErlangRandomVariable::
mobile/nakagami.cc:183:73: error: for a function-style cast, remove the redundant '::ErlangRandomVariable' [-fpermissive]
mobile/nakagami.cc:185:67: error: cannot call constructor 'GammaRandomVariable::
mobile/nakagami.cc:185:67: error: for a function-style cast, remove the redundant '::GammaRandomVariable' [-fpermissive]
*make: *** [mobile/nakagami.o] Error 1*

Solution:

In ns-2.34/mobile/nakagami.cc,

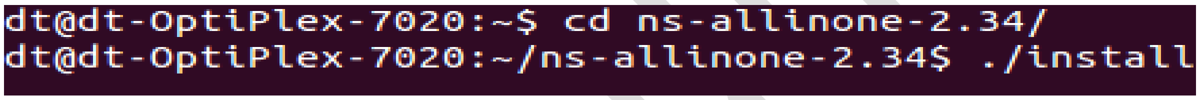
Replace:

```
if (int_m == m) {
        resultPower = ErlangRandomVariable::ErlangRandomVariable(Pr/m, int_m).value();
      } else {
        resultPower = GammaRandomVariable::GammaRandomVariable(m, Pr/m).value();
      }
      return resultPower;
  }
```

with:

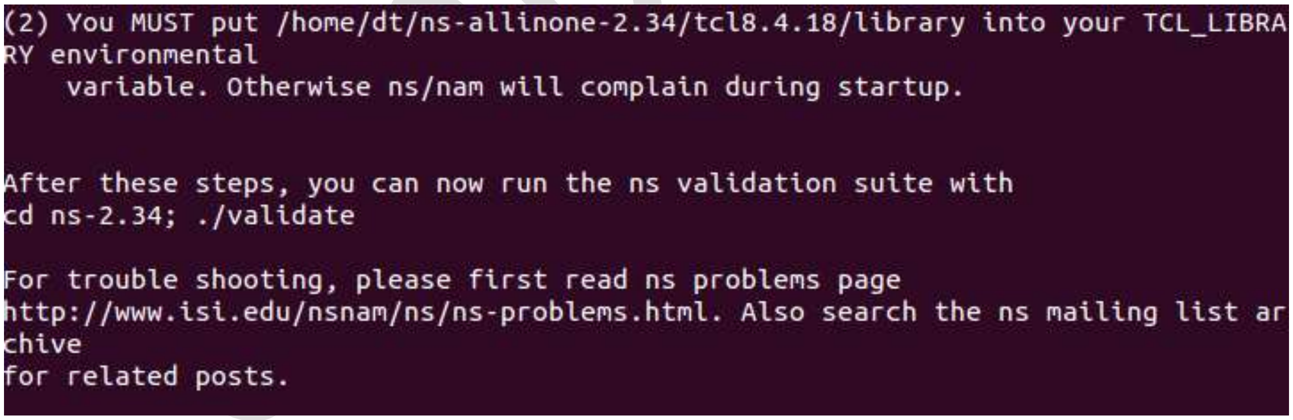
```
 if (int_m == m) {
           resultPower = ErlangRandomVariable(Pr/m, int_m).value();
         } else {
            resultPower = GammaRandomVariable(m, Pr/m).value();
         }
        return resultPower;
   }
```

Now move to ns-allinone-2.34 directory write the command ./install



```
dt@dt-OptiPlex-7020:~$ cd ns-allinone-2.34/
dt@dt-OptiPlex-7020:~/ns-allinone-2.34$ ./install
```

hit enter and wait for some time till it shows path information:



```
(2) You MUST put /home/dt/ns-allinone-2.34/tcl8.4.18/library into your TCL_LIBRA
RY environmental
    variable. Otherwise ns/nam will complain during startup.


After these steps, you can now run the ns validation suite with
cd ns-2.34; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list ar
chive
for related posts.
```

That's done now and you are installed NS2.

STEP 5:

Now it's time to set the path information. In the terminal use sudo gedit .bashrc and hit enter. It will ask for password to enter (Its not visible).

```
dt@dt-PC:~$ sudo gedit .bashrc
[sudo] password for dt:
```

Go to the last line of the newly opened file (bashrc), copy and paste these 3 lines. Make sure that you changed **dt** with your username on ubuntu.

PATH=$PATH:/home/dt/ns-allinone-2.34/bin:/home/dt/ns-allinone-2.34/tcl8.4.10/unix:/home/dt/ns-allinone-2.34/tk8.4.10/unix

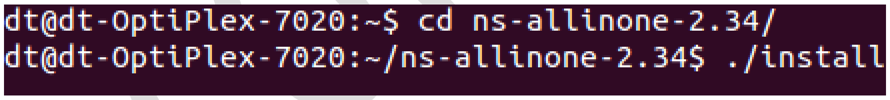
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/dt/ns-allinone-2.34/otcl-1.13:/home/dt/ns-allinone-2.34/lib

TCL_LIBRARY=$TCL_LIBRARY:/home/dt/ns-allinone-2.34/tcl8.4.10/library

Save the document and close. Reload the .bashrc using the following command on the terminal.

source ~/.bashrc

Now rerun **./install** command to include changes we have made:

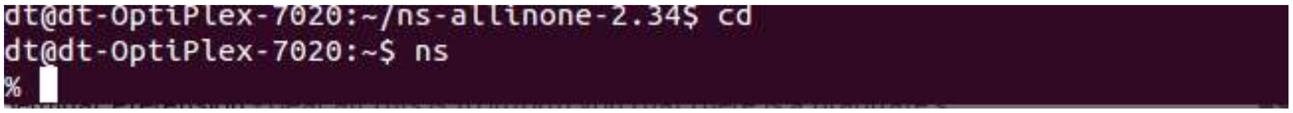


```
dt@dt-OptiPlex-7020:~$ cd ns-allinone-2.34/
dt@dt-OptiPlex-7020:~/ns-allinone-2.34$ ./install
```

Note: Make sure the version of tcl, otcl, tk match with the version you specified in bashrc if you are not using ns-allinone-2.34.

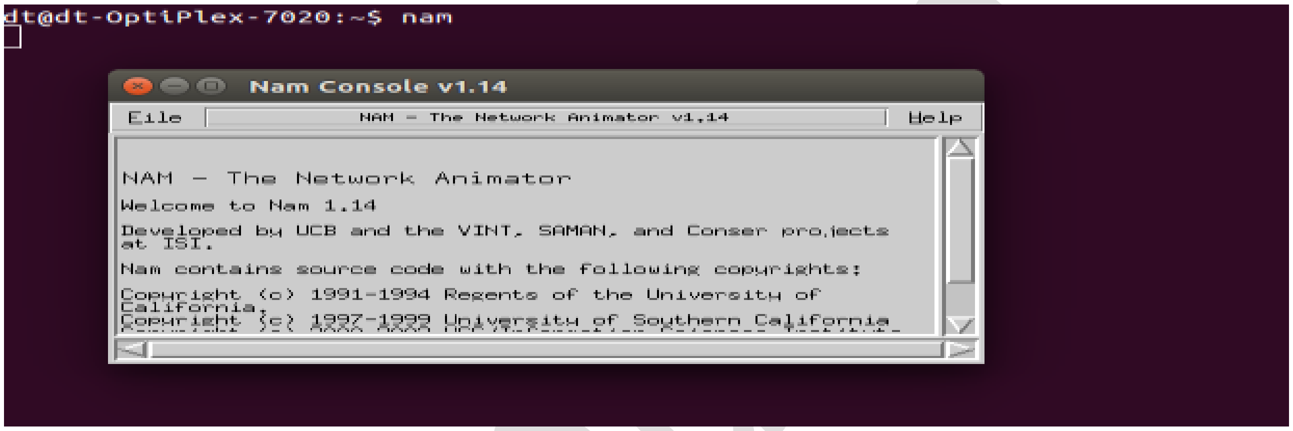
STEP 6:

Its done! open the terminal and type "ns" hit enter. You will get a % sign, it indicates the successful installation.



```
dt@dt-OptiPlex-7020:~/ns-allinone-2.34$ cd
dt@dt-OptiPlex-7020:~$ ns
%
```

To test NAM (NETWORK animator) write the command: **nam** and press inter on the terminal:



```
dt@dt-OptiPlex-7020:~$ nam
```

```
NAM - The Network Animator
Welcome to Nam 1.14
Developed by UCB and the VINT, SAMAN, and Conser projects
at ISI.
Nam contains source code with the following copyrights:
Copyright (c) 1991-1994 Regents of the University of
California.
Copyright (c) 1997-1999 University of Southern California
```

Now all requirements are fulfilled and you can proceed with other activities like running built in protocols on ns2 and cloning existing protocol to make changes.

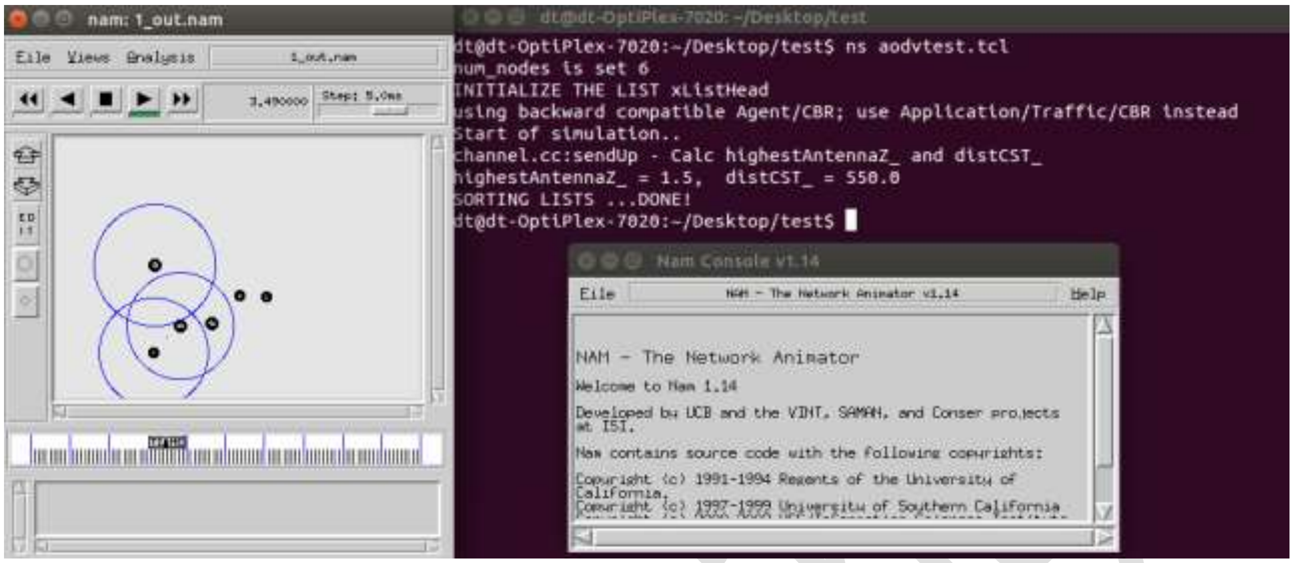
TCL sample script is located in:

souvikthegreat11.blogspot.com/2016/05/aodv-tcl-script.html

copy the tcl script from the above link and save it as **aodvtest.tcl.**

Move to directory of file and write the following command:

- **ns aodvtest.tcl**

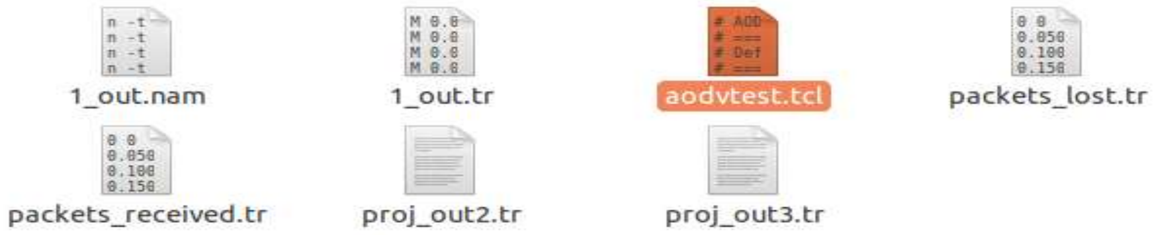


```
dt@dt-OptiPlex-7020:~/Desktop/test$ ns aodvtest.tcl
num_nodes is set 6
INITIALIZE THE LIST xListHead
using backward compatible Agent/CBR; use Application/Traffic/CBR instead
Start of simulation..
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
dt@dt-OptiPlex-7020:~/Desktop/test$
```

```
NAM - The Network Animator
Welcome to Nam 1.14
Developed by UCB and the VINT, SAMAN, and Conser projects
at ISI.
Nam contains source code with the following copyrights:
Copyright (c) 1991-1994 Regents of the University of
California.
Copyright (c) 1997-1999 University of Southern California
```

If the tcl script run without an error ns will generate the following trace file (with .tr extension) in your working directory (directory which holds tcl script).

Trace files are further analyzed by awk script to get status of nodes in your network as well as traffic condition. Some parameters which can be computed from trace file are delay, PDR, energy, bandwidth, etc.



6.1 Cloning protocols

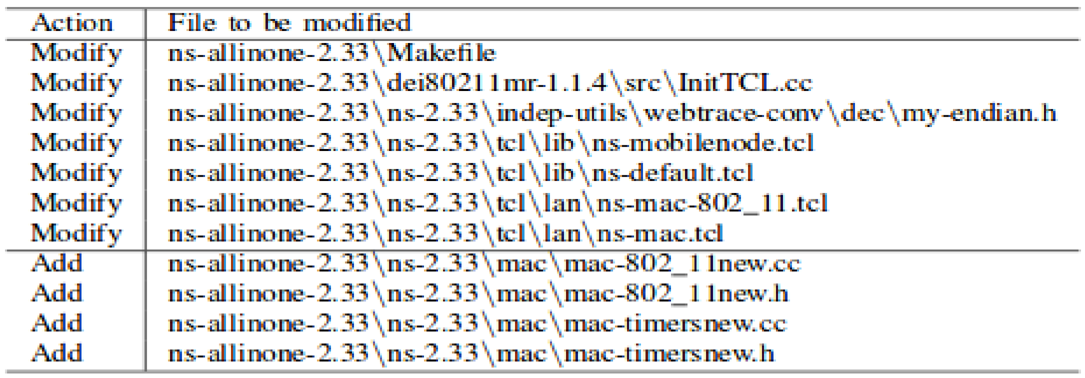
6.1.1 Cloning MAC layer protocols

Cloning a protocol mean creating identical copy of the protocol. When we brought this concept to ns2 there should be some consideration like: considering links to different files in ns2, file names and paths.

AS a sample we list the steps to clone MAC802_11 in the following section. The same steps will be followed with slight modification to clone other MAC layer protocols such as SMAC and TDMA in ns2.

Steps to clone MAC layer protocol in ns2

1) Check source code to be modified or added, maybe a backup is necessary for future diff.



Action	File to be modified
Modify	ns-allinone-2.33\Makefile
Modify	ns-allinone-2.33\dei80211mr-1.1.4\src\InitTCL.cc
Modify	ns-allinone-2.33\ns-2.33\indep-utils\webtrace-conv\dec\my-endian.h
Modify	ns-allinone-2.33\ns-2.33\tcl\lib\ns-mobilenode.tcl
Modify	ns-allinone-2.33\ns-2.33\tcl\lib\ns-default.tcl
Modify	ns-allinone-2.33\ns-2.33\tcl\lan\ns-mac-802_11.tcl
Modify	ns-allinone-2.33\ns-2.33\tcl\lan\ns-mac.tcl
Add	ns-allinone-2.33\ns-2.33\mac\mac-802_11new.cc
Add	ns-allinone-2.33\ns-2.33\mac\mac-802_11new.h
Add	ns-allinone-2.33\ns-2.33\mac\mac-timersnew.cc
Add	ns-allinone-2.33\ns-2.33\mac\mac-timersnew.h

2) Modify Makefile by

a) adding ”mac-80211new.o” next to existing ”mac-80211.o” in OBJCC section.

b) adding ”mac-timersnew.o” next to existing ”mac-timers.o” in OBJCCsection.

3) Modify InitTCL.c by adding following portion next to existing codes, from about line 70.

```
---BEGIN---
Mac/802_11new/Multirate set useShortPreamble_ false
Mac/802_11new/Multirate set gSyncInterval_ 0
Mac/802_11new/Multirate set bSyncInterval_ 0
Mac/802_11new/Multirate set CWMin_ 32
```

```
Mac/802_11new/Multirate set CWMax_ 1024
Mac/802_11new/Multirate set VerboseCounters_ 0
---END---
```

4) Modify my-endian.h by marking off first line and last line, this is to fix the compile error.

5) Modify ns-mobilenode.tcl by adding following 2 portions next to existing codes, from about line 500 and 700.

```
---BEGIN---
set god_ [God instance]
if {$mactype == "Mac/802_11new"} {
$mac nodes [$god_ num_nodes]
}
---END---
---BEGIN---
if {$mactype == "Mac/802_11new"} {
$self instvar mac_
set ns_ [Simulator instance]
set beacon_period [$ns_ delay_parse $beacon_period]
set cfp_duration [$ns_ delay_parse $cfp_duration]
$mac_(0) cfp $beacon_period $cfp_duration
}
---END---
```

6) Modify ns-default.tcl by adding following portion next to existing codes, from about line 700.

```
---BEGIN---
# Mac/802_11new
Mac/802_11new set CWMin_
Mac/802_11new set CWMax_ 1023
Mac/802_11new set SlotTime_ 0.000020
Mac/802_11new set SIFS_ 0.000010
Mac/802_11new set PreambleLength_ 144
Mac/802_11new set PLCPHeaderLength_48
Mac/802_11new set PLCPDataRate_ 1.0e6
```

Mac/802_11new set RTSThreshold_ 0

Mac/802_11new set ShortRetryLimit_7

Mac/802_11new set LongRetryLimit_4

Mac/802_11new set bugFix_timer_ true

Mac/802_11new set BeaconInterval_ 0.1

Mac/802_11new set ScanType_ PASSIVE

Mac/802_11new set ProbeDelay_ 0.0001

Mac/802_11new set MaxChannelTime_ 0.011

Mac/802_11new set MinChannelTime_ 0.005

Mac/802_11new set ChannelTime_ 0.12

---END---

7) Modify ns-mac-802

11.tcl by adding following portion next to existing codes, from about line 50.

---BEGIN---

Mac/802_11new set debug_ false Mac/802_11new instproc init {} {

eval $self next

set ns [Simulator instance]

$ns create-eventtrace Event $self

}

---END---

8) Modify ns-mac.tcl by adding following portion next to existing codes, from about line 65.

---BEGIN---

# IEEE 802.11new MAC settings if [TclObject is-class Mac/802_11new]

{

Mac/802_11new set delay_ 64us

Mac/802_11new set ifs_ 16us

Mac/802_11new set slotTime_ 16us

Mac/802_11new set cwmin_ 16

Mac/802_11new set cwmax_ 1024

Mac/802_11new set rtxLimit_ 16

Mac/802_11new set bssId_ -1

Mac/802_11new set sifs_ 8us

Mac/802_11new set pifs_ 12us

Mac/802_11new set difs_ 16us

Mac/802_11new set rtxAckLimit_ 1

Mac/802_11new set rtxRtsLimit_ 3

Mac/802_11new set basicRate_ 1Mb

Mac/802_11new set dataRate_ 1Mb

}

---END---

9) mac-80211new.cc

a) copy this file from ”mac-80211.cc”

b) substitute ”Mac80211” with ”Mac80211new”, there are 83 instances. (match case, only full matched word)

c) substitute ”Mac80211Class” with ”Mac80211newClass”, there are 2 instances. (match case, only full matched word)

d) substitute ”Mac/80211” with ”Mac/80211new”, there are 5 instances. (match case, only full matched word)

e) substitute ”mac-80211” with ”mac-80211new” in line 34.

f) substitute ”mac-timers.h” with ”mac-timersnew.h” in line 53.

g) substitute ”mac-80211.h” with ”mac-80211new.h” in line 54.

h) substitute ”classmac80211” with ”classmac80211new” in line 149.

i) substitute ”MAC80211” with ”MAC80211new” in line 1963.

10) mac-80211new.h

a) copy this file from ”mac-80211.h”

b) *substitute ”mac-80211.h” with ”mac-80211new.h” in line 34 and 37.

c) substitute ”nsmac80211.h” with ”nsmac80211new.h” in line 40, 41 and 602.

d) substitute ”mac-timers.h” with ”mac-timersnew.h” in line 47.

e) substitute ”Mac80211” with ”Mac80211new”, there are 4 instances. (match case, only full matched word)

f) from line 356 to line 365, postfix the name of each timer class with ”NEW”, for example, substitute ”DeferTimer” with ”DeferTimerNEW”

g) from line 554 to line 562, postfix the name of each timer class with ”NEW”, for example, substitute ”IFTimer” with ”IFTimerNEW”

11) mac-timersnew.cc

a) copy this file from ”mac-timers.cc”

b) substitute ”mac-timers.h” with ”mac-timersnew.h” in line 49.

c) substitute ”mac-80211.h” with ”mac-80211new.h” in line 50.

d) postfix the name of each timer class with ”NEW”, for example, substitute ”MacTimer” with ”MacTimerNEW” in line 79, 95, 114, 134, 157, 169, 183, 201, 215, 230, 245, 262, 277, 289, 315 and 349.

12) mac-timersnew.h

a) copy this file from ”mac-timers.h”

b) substitute ”_mac_timers_h_” with ”_mac_timersnew_h_” in line 36, 37 and 140.

c) substitute ”Mac80211” with ”Mac80211new”, there are 11 instances. (match case, only full matched word)

d) substitute ”MacTimer” with ”MacTimerNEW”, there are 18 instances. (match case, only full matched word)

e) postfix the name of each timer class with ”NEW”, for example, substitute ”BackoffTimer” with ”BackoffTimerNEW”

in line 73/75, 87/89, 95/97, 103/105, 112/114, 119/121, 126/128 and 133/135.

13) Remake load by entering ”make” command under ”ns-allinone-2.33\ns-2.33\”.

6.1.2 Cloning Routing Protocol

How to clone a protocol in ns2.35(AODV)

Step 1: Copy the file of aodv in the different folder name as taodv.

Step 2: Rename all the file names and inside document.

step 3: Replace aodv to taodv and AODV to TAODV.

Step 4: Go to #ns-allinone-2.35\ns-2.35\common\packet.h

```
typedef unsigned int packet_t;

static const packet_t PT_TCP = 0;
static const packet_t PT_UDP = 1;
static const packet_t PT_CBR = 2;
......
......
......
......
        // insert new packet types here
static const packet_t PT_TAODV = 73;  //newly added packet
static packet_t       PT_NTYPE = 74; // This MUST be the LAST one

class p_info {
public:
   p_info()
   {
      initName();
   }
   const char* name(packet_t p) const {
      if ( p <= p_info::nPkt_ ) return name_[p];
      return 0;
   }
........
........
........

static packetClass classify(packet_t type) {
        if (type == PT_DSR ||
           type == PT_MESSAGE ||
           type == PT_TORA ||
           type == PT_PUMA ||
           type == PT_AODV ||
           type == PT_TAODV ||
           type == PT_MDART)
           return ROUTING;
...........
..........
........
........

name_[PT_DCCP]="DCCP";
        name_[PT_DCCP_REQ]="DCCP_Request";
        name_[PT_DCCP_RESP]="DCCP_Response";
        name_[PT_DCCP_ACK]="DCCP_Ack";
        name_[PT_DCCP_DATA]="DCCP_Data";
        name_[PT_DCCP_DATAACK]="DCCP_DataAck";
        name_[PT_DCCP_CLOSE]="DCCP_Close";
        name_[PT_DCCP_CLOSEREQ]="DCCP_CloseReq";
        name_[PT_DCCP_RESET]="DCCP_Reset";
        name_[PT_TAODV]= "TAODV";
        name_[PT_NTYPE]= "undefined";


}
```

Step 5: #ns-allinone-2.35\ns-2.35\trace\cmu-trace.h

```
class CMUTrace : public Trace {
```

```
public:
   CMUTrace(const char *s, char t);
   void    recv(Packet *p, Handler *h);
   void    recv(Packet *p, const char* why);

   static void addPacketTracer(PacketTracer *pt);

..........
.........
..........
..........

void    format_imep(Packet *p, int offset);
      void    format_aodv(Packet *p, int offset);
      void    format_taodv(Packet *p, int offset);
   void    format_aomdv(Packet *p, int offset);
   void    format_mdart(Packet *p, int offset);
}
```

Step 6: #ns-allinone-2.35\ns-2.35\trace\cmu-trace.cc

```
#include <taodv/taodv_packet.h>

//Newly added
void
CMUTrace::format_taodv(Packet *p, int offset)
   {
     struct hdr_taodv *ah = HDR_TAODV(p);
     struct hdr_taodv_request *rq = HDR_TAODV_REQUEST(p);
     struct hdr_taodv_reply *rp = HDR_TAODV_REPLY(p);
```

```
switch(ah->ah_type) {
case TAODVTYPE_RREQ:

          if (pt_->tagged()) {
              sprintf(pt_->buffer() + offset,
                      "- taodv:t %x - taodv:h %d - taodv:b %d -taodv:d %d "
                      "- taodv:ds %d - taodv:s %d - taodv:ss %d "
                      "- taodv:c REQUEST ",rq->rq_type,
                       rq->rq_hop_count,
                       rq->rq_bcast_id,
            rq->rq_dst,
            rq->rq_dst_seqno,
            rq->rq_src,
            rq->rq_src_seqno);
      } else if (newtrace_) {
        sprintf(pt_->buffer() + offset,"-P  taodv -Pt 0x%x -Ph %d -Pb %d -Pd %d -Pds %d -
Ps %d -Pss %d -Pc REQUEST ",
        rq->rq_type,
        rq->rq_hop_count,
        rq->rq_bcast_id,
        rq->rq_dst,
        rq->rq_dst_seqno,
        rq->rq_src,
        rq->rq_src_seqno);


          } else {

              sprintf(pt_->buffer() + offset,
                  "[0x%x %d %d [%d %d] [%d %d]] (RREQ)",
        rq->rq_type,
        rq->rq_hop_count,
        rq->rq_bcast_id,
        rq->rq_dst,
        rq->rq_dst_seqno,
        rq->rq_src,
        rq->rq_src_seqno);
          }
           break;

    case TAODVTYPE_RREP:
    case TAODVTYPE_HELLO:
    case TAODVTYPE_RERR:

          if (pt_->tagged()) {
              sprintf(pt_->buffer() + offset,
                      "- taodv:t %x - taodv:h %d - taodv:d %d -tadov:ds %d "
                      "- taodv:l %f - taodv:c %s ",
                      rp->rp_type,
                      rp->rp_hop_count,
                      rp->rp_dst,
                      rp->rp_dst_seqno,
                      rp->rp_lifetime,
                      rp->rp_type == TAODVTYPE_RREP ? "REPLY" :
                      (rp->rp_type == TAODVTYPE_RERR ? "ERROR" :
                       "HELLO"));
          } else if (newtrace_) {

                  sprintf(pt_->buffer() + offset,
                    "-P  taodv -Pt 0x%x -Ph %d -Pd %d -Pds %d -Pl %f -Pc %s ",
                       rp->rp_type,
                        rp->rp_hop_count,
                       rp->rp_dst,
                        rp->rp_dst_seqno,
          rp->rp_lifetime,
          rp->rp_type == TAODVTYPE_RREP ?"REPLY" :
                        (rp->rp_type == TAODVTYPE_RERR ?"ERROR" :
                        "HELLO"));
                       } else {

                sprintf(pt_->buffer() + offset,"[0x%x %d [%d %d] %f] (%s)",
          rp->rp_type,
          rp->rp_hop_count,
          rp->rp_dst,
          rp->rp_dst_seqno,
          rp->rp_lifetime,
          rp->rp_type == TAODVTYPE_RREP ? "RREP":
                       (rp->rp_type == TAODVTYPE_RERR ?"ERROR" :
                         "HELLO"));
         }
         break;

    default:
      #ifdef WIN32
      fprintf(stderr,"CMUTrace::format_ taodv: invalid TAODV packet type\n");
      #else
         fprintf(stderr,"%s: invalid TAODV packet type\n",__FUNCTION__);
    #endif
         abort();
    }
}
...... ......
```

```
 //Newly added
 void CMUTrace::format(Packet* p, const char *why)
  {
         ......
        ......
        default:
        ......
        ......
               case PT_AODV:
            format_aodv(p, offset);
                      break;
               case PT_TAODV:                //Newly added
                      format_taodv(p, offset);
                      break;
             break;
        ......
        ......
 }
```

Step 7: #ns-allinone-2.35\ns-2.35\tcl\lib\ns-packet.tcl

```
set protolist {
# Common:
   Common
   Flags
   IP     # IP
.......
.......
.......
# Mobility, Ad-Hoc Networks, Sensor Nets:
   AODV     # routing protocol for ad-hoc networks
   TAODV     # routing protocol for ad-hoc networks
.........
........
.........

}
```

Step 8: #ns-allinone-2.35\ns-2.35\tcl\lib\ns-lib.tcl

```
Simulator instproc create-wireless-node args {
        ......
       ......
       switch -exact $routingAgent_ {
           ......
           ......
           AODV {
                  set ragent [$self create-aodv-agent $node]
           }
          TAODV {
          set ragent [$self create-taodv-agent $node]
       }
               ......
               ......
           }
           ......
           ......
     }
     ......
     ......
     # Newly added
```

```
  Simulator instproc create-taodv-agent { node } {
        #  Create TAODV routing agent
        set ragent [new Agent/TAODV [$node node-addr]]
        $self at 0.0 "$ragent start"     ;# start BEACON/HELLO messages
        $node set ragent_ $ragent
        return $ragent
  }
```

Step 9: #ns-allinone-2.35\ns-2.35\queue\priqueue.cc

```
//Newly added
  void
  PriQueue::recv(Packet *p, Handler *h)
  {
            ......
            ......
            case PT_AODV:
            case PT_TAODV:              //Newly added
            ......
            ......
  }
```

Step 10: #ns-allinone-2.35\ns-2.35\Makefile

```
OBJ_CC = \
        ......
        ......
        aodv/aodv_logs.o aodv/aodv.o \
        aodv/aodv_rtable.o aodv/aodv_rqueue.o \
        taodv/taodv_logs.o taodv/taodv.o \
        taodv/taodv_rtable.o taodv/taodv_rqueue.o \
        ......
        ......
        $(OBJ_STL)
```

*** Make sure you are making the same changes in makefile.vc and Makefile.in otherwise after ./configure
it will not take into account and object file will not generate.

Step 11: #ns-allinone-2.35\ns-2.35\tcl\lib\ns-agent.tcl

```
    Agent/AODV instproc init args {

          $self next $args
    }

    Agent/AODV set sport_   0
    Agent/AODV set dport_   0

    # Newly added
    Agent/TAODV instproc init args {

          $self next $args
    }

    Agent/TAODV set sport_   0
    Agent/TAODV set dport_   0
```

Step 12: #ns-allinone-2.35\ns-2.35\tcl\lib\ns-mobilenode.tcl

```
    Node/MobileNode instproc add-target { agent port } {
        ......
        ......
```

```
            # Special processing for AODV
            set aodvonly [string first "AODV" [$agent info class]]
            if {$aodvonly != -1 } {
                    $agent if-queue [$self set ifq_(0)]   ;# ifq between LL
 and MAC
            }
            # Newly added
            # Special processing for TAODV
            set taodvonly [string first "TAODV" [$agent info class]]
            if {$taodvonly != -1 } {
       $agent if-queue [$self set ifq_(0)]   ;# ifq between LL and MAC
            }
            ......
            ......
     }
```

Step 13: #ns-allinone-2.35\ns-2.35\queue\rtqueue.cc

Do not make any changes just go through the Packet Queue used by AODV.

Step 14: #ns-allinone-2.35\ns-2.35\routing\rtable.h

```
   class Neighbor {
       friend class AODV;
   friend class TAODV;    //Newly added
       friend class rt_entry;
       ......
       ......
       }

       class rt_entry {
```

```
    friend class rttable;
    friend class AODV;
  friend class TAODV;    //Newly added
  friend class TAODVLocalRepairTimer; //modified
  .....
  .....
  .....
  }
```

Step 15: #ns-allinone-2.35\ns-2.35\wpan\p802_15_4nam.cc

```
  packet_t nam_pktName2Type(const char *name)
  {
    //not all types included

    return (strcmp(packet_info.name(PT_TCP),name) == 0)?PT_TCP:
    ......
    ......
    ......
    (strcmp(packet_info.name(PT_AODV),name) == 0)?PT_AODV:
   (strcmp(packet_info.name(PT_TAODV),name) == 0)?PT_TAODV:

   }
```

Step 16: #ns-allinone-2.35\dei80211mr-1.1.4\src\InitTCL.cc

```
PacketHeaderManager set tab_(PacketHeader/SR) 1\n\
PacketHeaderManager set tab_(PacketHeader/AODV) 1\n\
PacketHeaderManager set tab_(PacketHeader/TAODV) 1\n\
```

Step 17: Rename every aodv file with taodv name inside taodv folder

Step 18: Open each and every file and rename aodv to taodv and AODV to TAODV

Step 19: Rename every timer class in taodv.h and taodv.cc

E.g. In taodv.h

```
class TAODVBroadcastTimer : public Handler {
public:
   TAODVBroadcastTimer(TAODV* a) : agent(a) {}
      void    handle(Event*);
private:
      TAODV    *agent;
      Event   intr;
};

.......
......
.....
....
.....
.....
class TAODV: public Tap,public Agent {



....
.....
friend class TAODVBroadcastTimer;
......
......

TAODVBroadcastTimer  btimer;
.....
......

}
```

*note rename every timer otherwise it will give an error

E.G.

In taodv.cc

```
void TAODVHelloTimer::handle(Event* p) {


   agent->sendHello();
   double interval = MinHelloInterval +
   ((MaxHelloInterval - MinHelloInterval) * Random::uniform());
   assert(interval >= 0);
   Scheduler::instance().schedule(this, &intr, interval);


}
```

*note rename every timer otherwise it will give an error

Step 20: Edit taodv_rtable.h

```
class taodv_rt_entry {
      friend class taodv_rtable;
      friend class TAODV;
      friend class LocalRepairTimer;
      friend class TAODVLocalRepairTimer;
 ......
```

......

......

}

Step 21: Recompilation:

Step 1: We should recompiled ``packet.cc`` as the ``packet.h" is modified.

this can be done by : ``touch common/packet.cc"

Step 2: ./configure (if this fails go to step 22)

Step 3: make clean

Step 4: make

Step 5: make install

Step 22: If ./configure fails then run ./install

```
$cd ns-allinone-2.35
$./install
$cd ns-allinone-2.35/ns-2.35
$sudo make install
```

You are now done with complete cloning of aodv routing protocol!

3 Generating TCL script using NSG2

To generate TCL script you can use ***NSg2***

- *Download NSg2.jar file*
- *Copy it into your home directory*
- *Open terminal*
- *Write* ***java –jar nsg2.jar***

You can use the following link for tutorial

https://www.youtube.com/watch?v=WcqBpNcq__M