Technical Note

Deployment of OMNET++

Lesly Maygua-Marcillo[1], Luis Urquiza-Aguiar[1], Martha Paredes-Paredes[1], Pablo Barbecho Bautista[2]

[1] Departamento de Electrónica, Telecomunicaciones y Redes de Información, Facultad de Eléctrica y Electrónica, Escuela Politécnica Nacional (EPN), C. Ladrón de Guevara E11-253, Quito PO.Box 17-01-2759, Ecuador; lesly.maygua@epn.edu.ec, cecilia.paredes@epn.edu.ec.
[2] Telematics Engineering Dept., Universitat Politécnica de Catalunya (UPC), Barcelona, Spain; pablo.barbecho@upc.edu.
* Correspondence: luis.urquiza@epn.edu.ec; Tel.:+593 2297-6300 ext.2311

Abstract: Nowadays, network simulators are frequently used to study services, applications and to solve problems in complex scenarios of communications. For this reason, the objective of this article is to introduce detailed information about installation of OMNET++ to new users. In this report, we show how to install OMNET++ simulator over a widely used distribution of Linux and how to integrate "Inet", "Inetmanet" and "Veins" frameworks with the simulator. We take care to explain this integration step-by-step. This tutorial includes: a virtual machine as additional material. Our objective is make it easy for a beginner research community in OMNET++ and to maintain an open environment of knowledge.

Keywords: Network simulator, OMNET++, veins, inet, inetmanet.

---

1. Introduction

Communications are present in everything, and each day studies of this topic are realized. A very useful tool to let studies of communications is a network simulator. With this tool we can build realistic scenarios, predicts the behavior of wired and wireless networks [1] [2]. Analyze parameters and protocols such as: QoS analysis [3], protocols analysis[8], etc.

Nowadays, there are many network simulators such as: NS-2, NS-3 [4], OPNET, GNS3, NetSim, among others [5] [7].

For this, reason we consider that is important the use of network simulators. And, in this occasion, we want to contribute to the development and easy start of the use of the simulator Next, we detailed important information for beginner research community in OMNET++.

2. Installation of OMNET++ simulator

With the purpose of reduce the size simulator it is necessary use as operative system Lubuntu, in this case we use Lubuntu 16.04.

2.1. Prerequisite Package for Lubuntu

First, it is advisable to update the Lubuntu with its latest components through the following instructions in a terminal:

```
sudo apt-get update

sudo apt-get upgrade
```

doi:10.20944/preprints201807.0302.v1

Now, the core of OMNET++ requires various packages that are needed to make different tasks, which are detailed below:

Minimal Requirements for Python

- `gcc, g++`.

Requirements for Network Simulation Cradle

- `flex, bison`.

Xml-based version of the config store

- `libxml2-dev`.

Doxygen and related inline documentation

- `doxygen, graphviz`.

Qtenv 3D visualization support

- `sudo add-apt-repository ppa:ubuntugis/ppa`
- `sudo apt-get update`.

Osgearth development

- `openscenegraph-plugin-osgearth libosgearth-dev`

Recommended to install parallel simulation support

- `openmpi-bin libopenmpi-dev`

Optional Pcap

- `libpcap-dev`

2.2. Download and Install OMNET++

Once the prerequisites have been installed, download OMNeT++ from http://omnetpp.org [7]. Copy the archive to the directory where you want to install it.Open a terminal, and extract the archive downloaded using the following command:

```
xvfz omnetpp-5.3-src.tgz
```

After that, a omnetpp-5.3 subdirectory with the OMNeT++ files will be created.

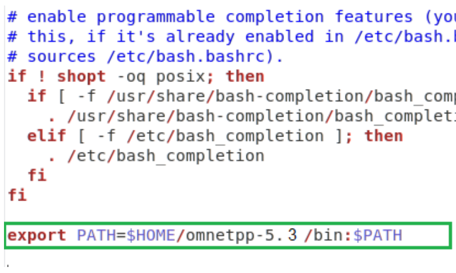
OMNeT++ needs its bin/ directory. To add bin/ to PATH permanently, it is advisable to set environment variables. To set the environment variables, you should edit **`.bashrc`** in your home directory. Use a text editor, for example: `gedit ~/.bashrc`

Add the following line at the end of the file. Save this change. Fig.1.

```
export PATH=$HOME/omnetpp-5.3/bin:$PATH
```

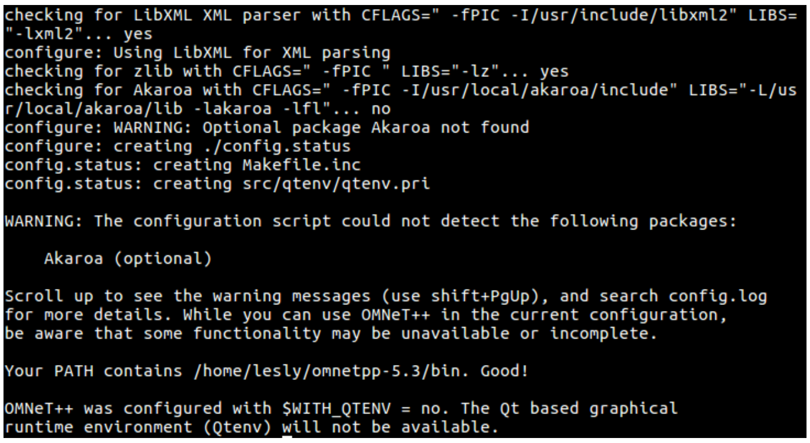
Now, is time to compile and build the default examples. You should be patient; the compilation takes several minutes. For this, run the command below:

```
./configure
```



```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH=$HOME/omnetpp-5.3/bin:$PATH
```

Figure 1. Set the environment variables



```
checking for LibXML XML parser with CFLAGS=" -fPIC -I/usr/include/libxml2" LIBS=
"-lxml2"... yes
configure: Using LibXML for XML parsing
checking for zlib with CFLAGS=" -fPIC " LIBS="-lz"... yes
checking for Akaroa with CFLAGS=" -fPIC -I/usr/local/akaroa/include" LIBS="-L/us
r/local/akaroa/lib -lakaroa -lfl"... no
configure: WARNING: Optional package Akaroa not found
configure: creating ./config.status
config.status: creating Makefile.inc
config.status: creating src/qtenv/qtenv.pri

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

    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 /home/lesly/omnetpp-5.3/bin. Good!

OMNeT++ was configured with $WITH_QTENV = no. The Qt based graphical
runtime environment (Qtenv) will not be available.
```

Figure 2. Configuring OMNET++.

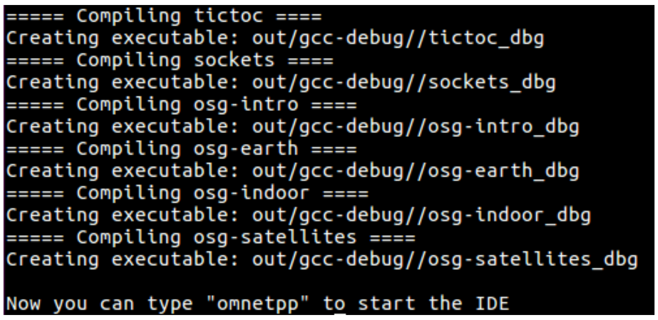
If the build is successful, the message "Good!" will appear. In the case of some problem is shown try the following command:

```
./configure WITH_TKENV=no WITH_QTENV=no
```

Now, when ./configure has finished, you can compile OMNeT++. Open a terminal and type:

```
make
```

A message of "type omnetpp to start the IDE" will appear, Fig.3.



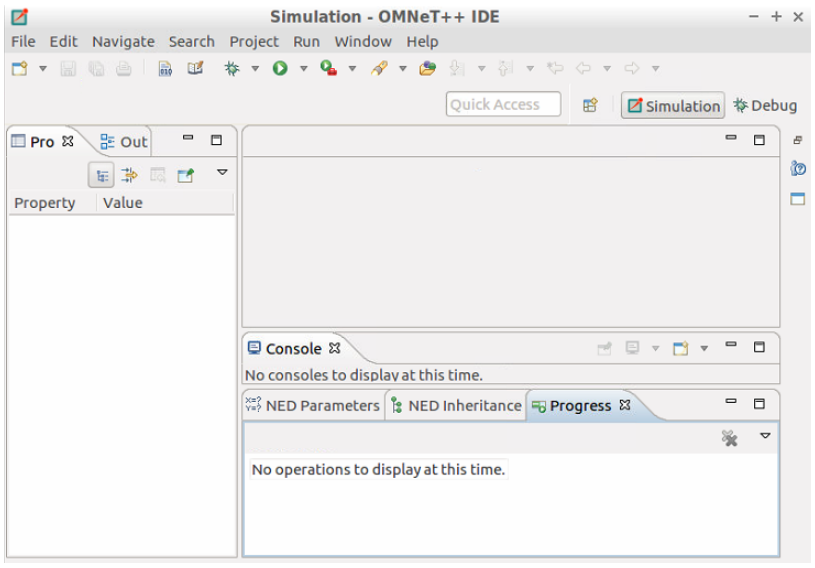
```
===== Compiling tictoc ====
Creating executable: out/gcc-debug//tictoc_dbg
===== Compiling sockets ====
Creating executable: out/gcc-debug//sockets_dbg
===== Compiling osg-intro ====
Creating executable: out/gcc-debug//osg-intro_dbg
===== Compiling osg-earth ====
Creating executable: out/gcc-debug//osg-earth_dbg
===== Compiling osg-indoor ====
Creating executable: out/gcc-debug//osg-indoor_dbg
===== Compiling osg-satellites ====
Creating executable: out/gcc-debug//osg-satellites_dbg

Now you can type "omnetpp" to start the IDE
```

Figure 3. Make compiling

2.3. Verifying the installation

To verify the installation. First, you can typing the following command in the terminal to try the OMNeT++ Simulation IDE Fig.3:

```
omnetpp
```



No consoles to display at this time.

No operations to display at this time.

Figure 4. Simulation IDE.

If you need to recompile the simulator components with different parameters (e.g. different optimization). You should change the top-level OMNeT++ directory, edit `configure.user`, typing:

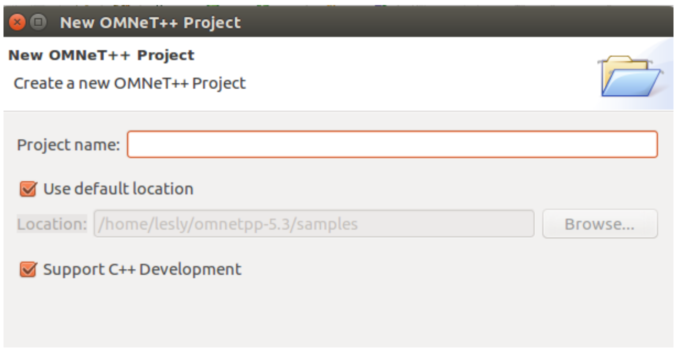
```
./configure

make cleanall

make
```

2.4. Create a project

∗ Go to **File** → **New** → **OMNeT++ Project**

Fill the settings with the information below Fig.5:

∗ Choose a project name -> Next
∗ Initial Content -> Choose Empty/project
∗ Select project type -> Choose OMNeT++ Simulation
∗ Select Finish



New OMNeT++ Project
Create a new OMNeT++ Project

Project name:

[✓] Use default location

Location: /home/lesly/omnetpp-5.3/samples   Browse...

[✓] Support C++ Development

Figure 5. New OMNET++ project.

doi:10.20944/preprints201807.0302.v1

2.4.1. Running the first script in OMNet++

You can execute the OMNeT++ projects by two ways. Through a terminal and with de simulator IDE.

Using the IDE, the steps for running a program file are:

∗ Go to Project_name → e.g. **File** → **Open file** select the project and double click on to open it.

3. Adding INET Framework

INET is an open-source model library to simulate wired, wireless and mobile networks in OMNeT++. Contains models for several Internet protocols.

To install this framework there are two ways: automatic and manual installation [11].

- **Automatic Installation**

Recent versions of the OMNeT++ can download and install INET for you. It is:

– Open the OMNeT++ IDE (omnetpp)

– A prompt will ask if you want to install INET.

– Keep the boxes checked and proceed.

If you skipped this step:

– Go to Help -> Install Simulation Models.Fig.6.

– A dialog will appear with the available simulation models, select INET and follow the prompts.

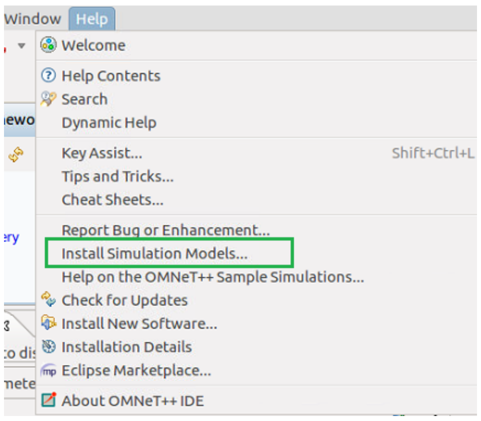


Figure 6. Help - Simulation models.

The IDE will download in both ways, unzip, and automatically build INET.

- **Manual Installation**

– Download the INET packages in https://inet.omnetpp.org/Download.html

– Unpack it into the directory of your preference using: `xvfz inet-<version>.tgz`

– Start the OMNeT++ IDE, and import the project: File -> Import -> Existing Projects to the Workspace. A project named INET will be appear.

– Build with Project -> Build, or hit `Ctrl+B`.

You should be able to launch simulations.

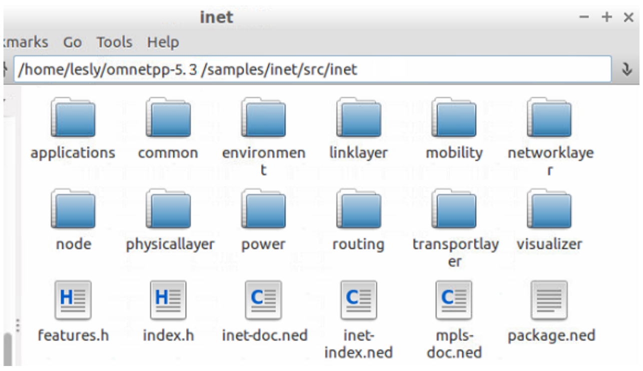


Figure 7. Inet framework installed

4. Adding INETMANET Framework

INETMANET is an extension of INET Framework but this adds features and protocols for mobile ad-hoc networks [11]. The steps to install this framework are very similar to INET and these are show bellow:

- Download Inetmanet package
- Unpack into the directory of your preference by using: `xvfz inetmanet-<version>.tgz`
- Start the OMNeT++ IDE, and import the project: File -> Import -> Existing Projects to the Workspace.
- Finally, build with Project -> Build.

5. Adding VEINS Framework and SUMO

This framework of OMNeT++ includes a suite of models to make vehicular network simulations as realistic as possible. The GUI and IDE of SUMO and OMNeT++ can be used for quickly setting up and interactively simulations.

Below are shown the steps to the installation [12].

- First, check that the OMNET++ version previously installed is compatible with the selected Veins release at https://veins.car2x.org/download/.
- Then, download Veins using https://veins.car2x.org/download/
- Unpack it in your selected directory.
- Import the project into your OMNeT++ IDE workspace. Click in File -> Import -> General: Existing Projects to the Workspace and selecting the directory to unpacked the module framework Fig.8.
- Build the newly imported project. Go to: Project -> Build All in OMNeT++ IDE.

With this steps you are ready to run your first IVC evaluations, but to ease debugging, the next step will ensure that SUMO works good.

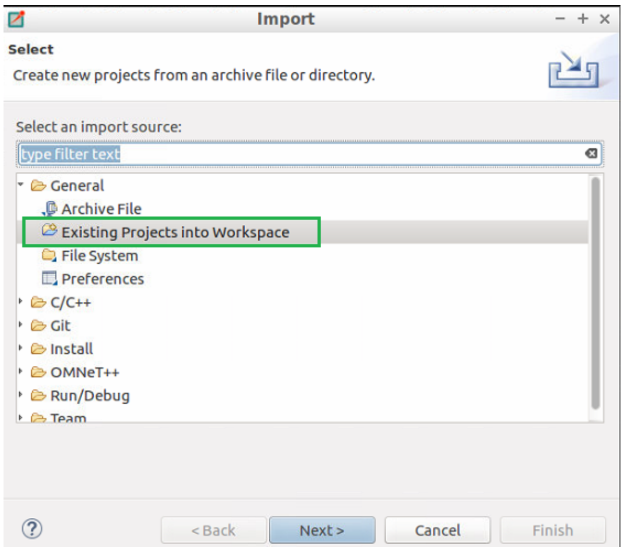


Figure 8. Installing Veins-Existing projects into workspace.

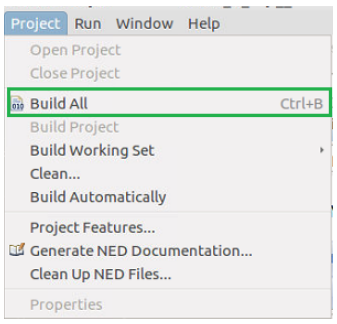


Figure 9. Building Veins.

5.1. SUMO installation

Before installing latest SUMO version, we strongly recommend to check OMNET++ and VEINS compatibility at http://veins.car2x.org/download/ [6]. As prerequisites we have to install the following packages [13][14]:

```
sudo apt-get install libgdal-dev libxerces-c-dev libproj-dev libfox-1.6-dev
```

Once the prerequisites have been installed, download SUMO from https://sourceforge.net/projects/sumo/files/sumo/ [15]. Copy the archive .tar.gz to the directory where you want to install it. SUMO also needs its bin/ directory. To add bin/ to PATH permanently, it is advisable to set environment variable. To set the environment variable, you should edit `.bashrc` in your home directory. Use a text editor, for example: `gedit ~/.bashrc`

Add the following line at the end of the file. Save this change.

```
export SUMO_HOME=$HOME/omnetpp-5.3/bin
```

Then, SUMO can be installed following the next steps in our simulator at SUMO folder [13][14]. You should be patient; the compilation takes several minutes. For this, run the commands below:

```
./configure
make
```

```
make install
```

Next, we can verify the correct installation of this component through : `sumo-gui`. When typing this command you should see a window similar to the shown in Fig.10

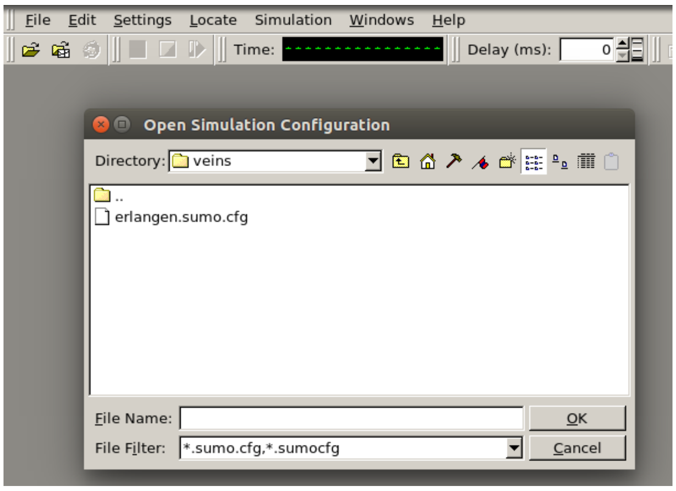

File Edit Settings Locate Simulation Windows Help
Time:
Delay (ms): 0
Open Simulation Configuration
Directory: veins
..
erlangen.sumo.cfg
File Name:
File Filter: *.sumo.cfg,*.sumocfg
OK
Cancel

Figure 10. Opening Sumo.

To run an example in sumo we can follow:

First, go to the directory where the example is located. Type the following command:

```
sumo-gui -c example.sumo.cfg
```

Immediately, the example will be shown on the screen Fig.11.

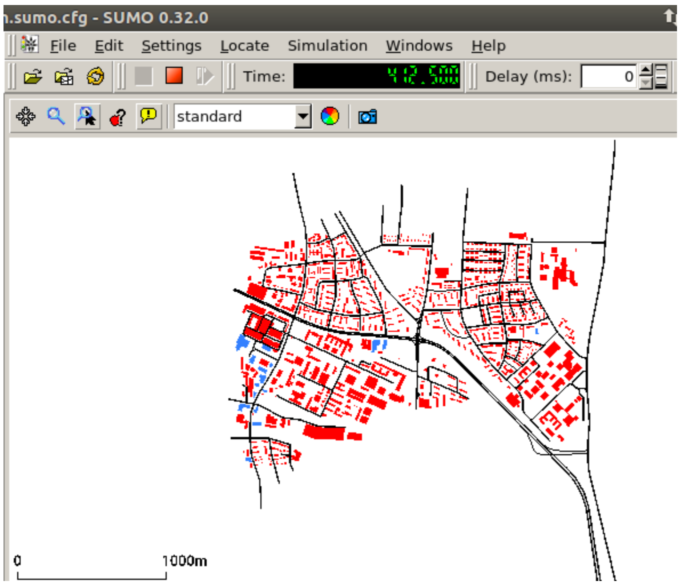


Figure 11. Running example in SUMO

5.2. SUMO Map generation

We can use our own map for vehicular network simulations in Veins, using SUMO tools. In order to simplify this procedure, first we have to export the desired map from https://www.openstreetmap.org [16]. The file will has the extension .osm.

For the next step, we have to copy the generated file .osm into the sumo-0.32.0/bin directory. Using SUMO tools, we convert the map from .osm to .net file. Type the following command:

```
netconvert -osm-files map.osm -o mymap.net.xml
```

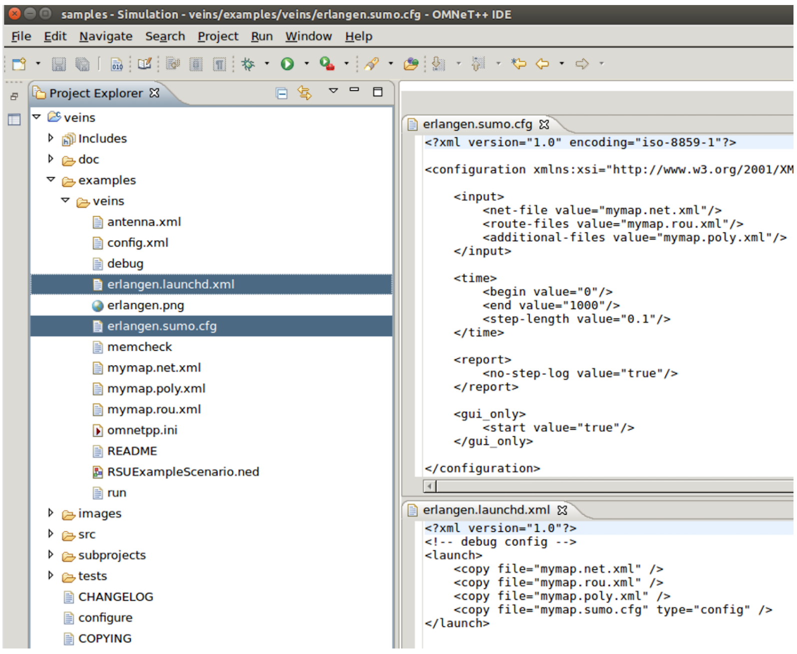
To generate routes for vehicles with default configurations, execute the following python script:

```
python randomTrips.py -n mymap.net.xml -r mymap.rou.xml
```

In order to get a realistic simulation we can include buildings in the map with the following command:

```
polyconvert -n mymap.net.xml -osm-files map.osm -o mymap.poly.xml
```

We have generated three files .net .rou and .poly all as xml files. We have to move these files to the examples folder in veins directory and modify the files .launch and .cfg with the names of the new files "mymap" Fig.12.



```
<?xml version="1.0" encoding="iso-8859-1"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XM

    <input>
        <net-file value="mymap.net.xml"/>
        <route-files value="mymap.rou.xml"/>
        <additional-files value="mymap.poly.xml"/>
    </input>

    <time>
        <begin value="0"/>
        <end value="1000"/>
        <step-length value="0.1"/>
    </time>

    <report>
        <no-step-log value="true"/>
    </report>

    <gui_only>
        <start value="true"/>
    </gui_only>

</configuration>
```

```
<?xml version="1.0"?>
<!-- debug config -->
<launch>
    <copy file="mymap.net.xml" />
    <copy file="mymap.rou.xml" />
    <copy file="mymap.poly.xml" />
    <copy file="mymap.sumo.cfg" type="config" />
</launch>
```

Figure 12. Map configuration for Veins

6. Additional Material

This tutorial includes a virtual machine with OMNET ++ simulator. This machine include the **inet**, **inetmanet**, **veins** and **sumo** frameworks. Below, we show a table (Table 1) with the used software versions in this tutorial with the objective to avoid installation problems.

Table 1. My caption

Software	Version
OMNeT++	5.3
INET	3.4.0
VEINS	4.7
SUMO	0.32.0

Here concludes our tutorial. In the next section is added the additional material to perform test.

Bibliography

1. Ravi, Kishore and Narasimha Sarma, Simulation Analysis of Multi-Dimensional WSNs using NS-3, TENCON, 2017.
2. Weiwei, Liu and Xichen, Wang and Wenli, Zhang ; Lin,Yang and Chao, Peng, Coordinative simulation with SUMO and NS3 for Vehicular Ad Hoc Networks, APCC, 2016.
3. Fahimeh, Arab and Mohsen, Karimi and Seyed, Mostafa, Analysis of QoS parameters for video traffic in homeplug AV standard using NS-3, Smart Grids Conference, 2017.
4. G. Carneiro, "Ns-3: Network simulator 3," in *UTM Lab Meeting April*, vol. 20, 2010.
5. S. Siraj, A. Gupta, and R. Badgujar,"Network simulation tools survey," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, no. 4, pp. 199–206, 2012.
6. Veins Changelogs, 2018. [Online]. Available: https://veins.car2x.org/download/
7. OMNET++. (2017) Main page. [Online]. Available: https://omnetpp.org/
8. Nosiba, Alfadil and Hamid, Ali, Performance Evaluation of TCP Congestion Control Mechanisms Using NS-2, SGCAC , 2016.
9. Documentation INET Framework. (año). [Online]. Available: https://inet.omnetpp.org/Introduction.html
10. Andras Varga, Install Guide Omnetpp, v.5.3, 2016. [Online]. Available: https://omnetpp.org/doc/omnetpp/InstallGuide.pdf
11. Installing INET, 2017. [Online]. Available: https://inet.omnetpp.org/Installation.html
12. Installing Veins, 2017. [Online]. Available: https://veins.car2x.org/tutorial/#step3
13. Install Sumo, 2017. [Online]. Available: http://veins.car2x.org/tutorial/#step1
14. Installing sumo on Linux, (2017) [Online]. Available: http://sumo.dlr.de/wiki/Installing#Linux
15. Sumo versions, 2017. [Online]. Available:https://sourceforge.net/projects/sumo/files/sumo/
16. OpenStreetMaps, 2018. Main page. [Online]. Available:https://www.openstreetmap.org