Part Ⅰ. Installation

(Tested in Ubuntu Linux 14.04 environment.)

1. OMNeT++

OMNeT is C++ based network simulator and it can be run on Eclipse. It is very helpful for researcher on network field, because it supports animation of packet flow on network and evaluation of network performance.

Assume that your home directory is "/home/foo".

Installation of OMNeT

a. Download OMNeT++ 4.6

https://omnetpp.org/component/jdownloads/download/32-release-older-versions/2290-omnet-4-6-source-ide-tgz

omnetpp-4.6-src.tgz file for Ubuntu

b. OMNeT++ Installation Manual and other documents

http://omnetpp.org/doc/omnetpp/InstallGuide.pdf
http://www.omnetpp.org/documentation

c. Prerequisite Installation of Libraries

refer to https://omnetpp.org/doc/omnetpp/InstallGuide.pdf chapter 5 Ubuntu

$ sudo apt-get install build-essential gcc g++ bison flex perl tcl-dev tk-dev blt libxml2-dev zlib1g-dev openjdk-6-jre doxygen graphviz openmpi-bin libopenmpi-dev libpcap-dev

Notice: if some packets cannot be installed, just pass or refer to the above document.

d. Decompress OMNeT++

$ mkdir ~/usr

$ mv omnetpp-4.6-src.tgz ~/usr

Note Navigate into the file where you have downloaded omnet++ and then perform this command

$ cd ~/usr

$ tar zxvf omnetpp-4.6-src.tgz

e. Register on environment variable

$ export PATH=/home/foo/usr/omnetpp-4.6/bin:$PATH *NOTE* Do this before 'make'

$ export TCL_LIBRARY=/usr/share/tcltk/tcl8.6

f. Configure and Install

$ cd omnetpp-4.6

$ ./configure

$ make

note It is not possible to 'make' without exporting the path to the command existing folder.

g. Download Java (Optional if your PC already has available Java)

http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

* Note: If your computer uses an Intel CPU, download jre for Linux x86, such as jre-8u45-linux-i586.tar.gz

h. Decompress Java and move (Optional if your PC already has available

Java)

$ tar zxvf jre-8u45-linux-i586.tar.gz

$ mv jre1.8.0_45 /home/foo/usr/omnetpp-4.6/ide/jre

i. Start in a terminal

$ omnetpp

2. Sumo

(http://sumo.dlr.de/wiki/Installing/Linux_Build)

a. Update apt-get command

$ sudo apt-get install autoconf

$ sudo apt-get install libproj-dev proj-bin proj-data libtool libgdal1-dev libxerces-c3-dev libfox-1.6-0 libfox-1.6-dev

b. Symbolic link

Check where a symbolic link /usr/lib/libgdal.so exists or not.

If not, make a symbolic link as follows:

$ sudo ln -s /usr/lib/libgdal1.7.0.so /usr/lib/libgdal.so

note please check the version of the libgdal.x file you have on your system

To find the version

$locate libgdal

Create a symbolic link to this particular file

c. Download Version: 0.22 (Statable Version) at SUMO official site

Official site : http://sumo-sim.org/wiki/Downloads

d. Decompress SUMO installation package

$ tar -xzvf sumo-src-0.22.0.tar.gz

$ sudo mv -v sumo-0.22.0 /usr/local/src (optional)

$ cd /usr/local/src/sumo-0.22.0

e. Configure

$ ./configure --with-python --enable-debug

or

$ ./configure --prefix=/home/usr/local/src/sumo-0.22.0 --enable-debug --with-python

note: "/home/usr/local/src/sumo-0.22.0" shall be your target installation folder, usually same to the current folder

note check the version of python on your system whether it is python3 or python and use the respective version when typing the $configure command

note If you get the error that the xerces/lib is missing, you will have to install the packages. The best way to do is by installing synaptic and marking the necessary packages for installation.

** libxerces-c-dev

** libxerces-c3.1

** libxerces-2-java

// if an error about fox-1.6 occur then you can use sudo apt-get install fox-1.6 then step 'E' will be completed success

PLEASE LOOK AT THE FOLLOWING LINK TO CORRECTLY INSTALL SUMO-0.22.0 CORRECT VERSION

http://sumo.dlr.de/wiki/Installing/Linux_Build_Libraries#Ubuntu_14.04_.22Trusty_Tahr.22_.28tested_with_SUMO_0.22.0.29

f. Install

$ make

$ sudo make install

*note: * if you have access permission on the installation folder, no need "sudo"

g. Call SUMO and Check

$ sumo-gui

Or

$ sumo

If you type sumo-gui, you run gui version of sumo. On the other hand, if typing sumo, command line mode is run. If not running, check path setting in your .bashrc file located at your default home directory.

3. Veins

Veins is a VANET project packet in OMNeT. It makes connection to SUMO via TraCI (Traffic Control Interface), which feeds vehicle mobility to Veins. The follows are a general procedure to run a standard veins. Usually, we may use project augmented Veins, like Veinsvadd shown in Step 4.

Import Veins into OMNeT

a. Download Veins, select a proper version by checking SUMO version.

http://veins.car2x.org/download/

b. Follow the step

http://veins.car2x.org/tutorial/

But if both Sumo and OMNeT is installed on your computer, then skip all the step except final step. The next is core contents of final step.

$ <VEINS_INST_DIR>/sumo-launchd.py –v –c sumo-gui

Then terminal will be blocked and it will wait the connect request of OMNeT.

c. Run the veins project

After execution of OMNeT++, find existing project ~~~ on File – import menu. And set the veins installation directory. If success, you can see mixim directory on your project window on Omnet++> Then click mouse right button placing your mouse on mixim/samples/veins/OMNeTpp.ini and click "Run As OMNeTpp". Then this veins project will be compiled and run.