Cooperative Mobile Systems [CMS 2022w]

Installation

Mario Franke

Goal

This handout will guide you in the process of getting everything set up to follow the labs.

You can skip most of this installation procedure by downloading a ready-made virtual machine image called "Instant Veins 5.2-i1".[1] You may then continue with installing R.

As an alternative, we provide this guide to install all required tools either natively on your personal computer, or as a virtual machine (e.g. Oracle VM VirtualBox) on your own. This helps you to get familiar with the tools and with the used operating system.

To address any technical difficulties while you set up your system refer to the Q&As omnetpp[2] tag in StackOverflow or have a look to the OMNeT++ installation guide[3]

Attention

We do not make any warranties about the completeness, reliability, and accuracy of the information in this document. Any action you take upon the information in this document is strictly at your own risk and we will not be liable for any losses and damages.
If you are setting up your own system, make sure to use English as the system language for better support from our side.

[1] http://veins.car2x.org/download/
[2] https://stackoverflow.com/search?q=omnetpp
[3] https://omnetpp.org/doc/omnetpp/InstallGuide.pdf

1 Preparations

For best support and compatibility, it is necessary to use a Linux-based operating system, e.g., Debian. You can set up the system either natively on your personal computer, or as a virtual machine, e.g, VMware or VirtualBox. To get better performance from your virtual machine activate hardware acceleration (VT-x, or AMD-V) in the BIOS of your computer. Furthermore, it is recommended to install the support tools of the used virtualization software (i.e., *VMware Tools* for VMware, *Guest Additions* for VirtualBox).

System requirements:

- Operating system: Linux (e.g., Debian 11 or Ubuntu 20.04 LTS.)
- OMNeT++ 5.7 (Discrete Event Simulator)[4]
- SUMO 1.8 (Traffic simulator platform)[5]
- Veins 5.2 (Framework for vehicular network simulations)[6]

When downloading the required tools make sure you get the correct version as source files mentioned in the list above. Set up your machine and install the required software: For Debian and derivatives, the following packages are needed. Other systems will likely ship with similarly named packages.

Debian and derivatives

```
sudo apt update
sudo apt install git-core gitk git-gui tig clang bison flex tcl-dev tk-dev ↘
    openjdk-17-jre qtbase5-dev libopenscenegraph-dev xvfb libwebkit2gtk-4.0-dev ↘
    doxygen graphviz python3-numpy python3-pandas python3-matplotlib ↘
    libxerces-c-dev libproj-dev libgdal-dev libfox-1.6-dev libavformat-dev ↘
    libavcodec-dev libswscale-dev python-dev cmake xterm clang-format uncrustify ↘
    valgrind gdb lldb nemiver
```

2 Installing OMNeT++

1. Extract the OMNeT++ source files to a directory of your choice, e.g., /home/username/software/omnetpp-5.7. We'll call this directory `<your-dir>` in the remainder of this document.
2. Navigate into the extracted OMNeT++ directory: `cd <your-dir>/omnetpp-5.7`
3. Add OMNeT++/`bin` directory to your path by executing:
   `<your-dir>/omnetpp-5.7$ source setenv`
   Additionally the following line to the bottom of ~/`.profile` file:
   `export PATH=$PATH:/home/<your-dir>/omnetpp-5.7/bin`
4. To apply the changes you made in the ~/`.profile` file, please logout and login again
5. Configure the makefiles for OMNeT++: `<your-dir>/omnetpp-5.7$ ./configure`
6. Compile the code: `<your-dir>/omnetpp-5.7$ make`
   Note: You can use `make -j <number-of-cores>` or `make -j $(nproc)` to compile the software with multiple cores. For speed ups use the same option in the subsequent steps requiring compilation.

[4] https://omnetpp.org/omnetpp
[5] https://sourceforge.net/projects/sumo/files/sumo/
[6] http://veins.car2x.org/download/

Note: This process can take a long time. As OMNeT++ is not needed to install SUMO, feel free to, e.g., minimize this window and continue with the next step.

3 Installing SUMO

1. Extract the SUMO source files to a directory of your choice: `<your-dir>`
2. Navigate into the extracted SUMO directory: `cd <your-dir>/sumo-1.8`
3. Add the SUMO directory to the SUMO_HOME variable by executing:
   `export SUMO_HOME="$PWD"`
4. Create and navigate to folders for makefiles:
   `<your-dir>/sumo-1.8$ mkdir build/cmake-build && cd build/cmake-build`
5. Generate native makefiles: `<your-dir>/sumo-1.8/build/cmake-build$ cmake ../..`
6. Compile the code: `<your-dir>/sumo-1.8/build/cmake-build$ make`
   Note: You can use `make -j <number-of-cores>` or `make -j $(nproc)` to compile the software with multiple cores.
7. Add SUMO/`bin` directory to your path by adding the following line to the bottom of ~/`.profile` file (requires relogin):
   `export PATH=$PATH:/home/<your-dir>/sumo-1.8/bin`

Note: This process can take a long time. As SUMO is not needed right away, feel free to, e.g., minimize this window and continue with the next step.

4 Get Veins from GitHub

1. Navigate to a directory of your choice: `<your-dir>`
2. Clone the Veins repository from GitHub:
   `<your-dir>$ git clone https://github.com/sommer/veins.git`
3. Navigate into the new directory: `cd <your-dir>/veins`
4. Checkout version Veins 5.2: `<your-dir>/veins$ git checkout tags/veins-5.2`

5 Installing and Running Veins (using the command line)

In order to continue with this step, the installation of OMNeT++ needs to be done.

1. Navigate into the Veins directory: `cd <your-dir>/veins`
2. Configure the makefiles for Veins: `<your-dir>/veins$ ./configure`
3. Compile the code: `<your-dir>/veins$ make`
   Note: You can use `make -j <number-of-cores>` or `make -j $(nproc)` to compile the software with multiple cores.
4. In a terminal, start the daemon[7] by executing: `<your-dir>/veins$ bin/veins_launchd -vv`

[7] http://veins.car2x.org/documentation/sumo-launchd/

5. In another terminal, navigate into the directory containing the Veins tutorial simulation:
   `cd <your-dir>/veins/examples/veins`
6. Run the tutorial simulation `<your-dir>/veins/examples/veins$ ./run`

6 Running Veins (using the IDE)

1. To import Veins into OMNeT++, start the OMNeT++ IDE (as in the previous step) and open the workbench. Then, from the menu bar select *File* → *Import* → *General* → *Existing Projects into Workspace* select the Veins directory: `<your-dir>/veins/`.
2. In a terminal, start the daemon by executing:
   `<your-dir>/veins$ bin/veins_launchd -vv`
3. In the OMNeT++ IDE, inside the Veins project find the `examples/veins/` directory, right-click on `omnetpp.ini` file and select *Run As* → *OMNeT++ Simulation*.

7 Installing R

In the lab, we'll evaluate simulations by recording and analyzing data. We'll use R in the labs. R is a free software environment for statistical computing and graphics. For Debian and derivatives, the following packages are needed. Other systems will likely ship with similarly packages:

Debian and derivatives

```
sudo apt update
sudo apt install r-recommended
sudo apt install libssl-dev
```

In order to prepare the R environment for the lab, the following statements need to be executed in R to install the required R packages:

Running R

```
R
```

Installing R packages

```
install.packages('tidyverse',dependencies = TRUE, ↘
    repos=c('http://cran.uni-muenster.de/'))
install.packages('sqldf',dependencies = TRUE, ↘
    repos=c('http://cran.uni-muenster.de/'))
q()
```

The setup might ask you whether a personal library shall be used. In this case select `yes`. You can use any type of text-editor to write your R scripts, however the RStudio[8] IDE is a good starting point.

[8] https://www.rstudio.com/products/rstudio/download/

Contact

- Course Website: https://nsm.inf.tu-dresden.de/teaching/courses/cms/2022w/
- OPAL: https://bildungsportal.sachsen.de/opal/auth/repository/catalog/26766245902