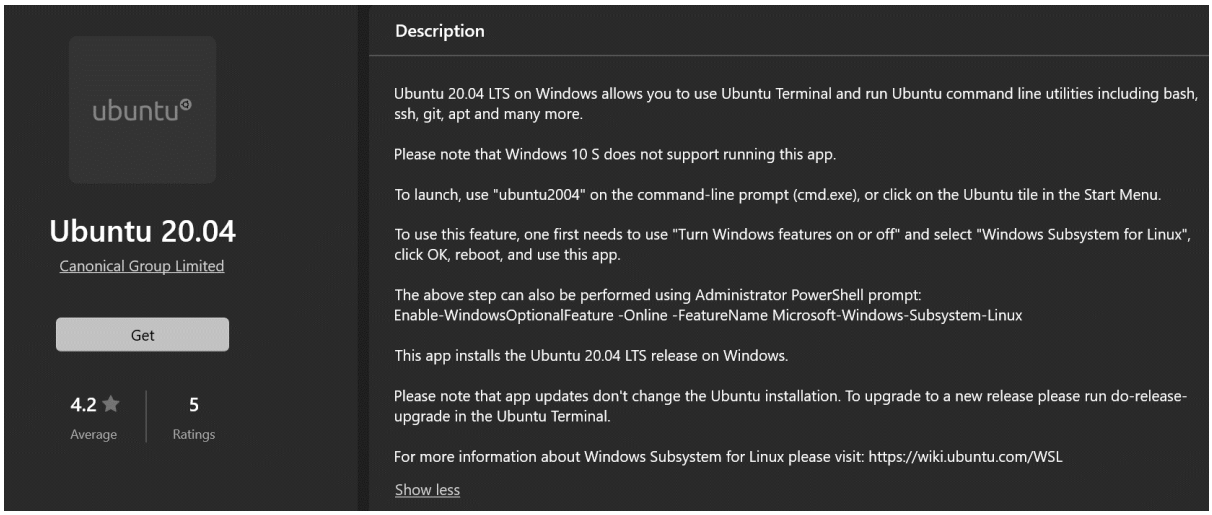
Installation of ns-3 and SUMO

Tested steps:

1. Ubuntu was downloaded from the microsoft store and installed according to steps included there:



Or using the powershell:

“wsl --install”

You can find these steps here: https://docs.microsoft.com/en-us/windows/wsl/install#set-up-your-linux-user-info

GUI for wsl download and setup: https://medium.com/@japheth.yates/the-complete-wsl2-gui-setup-2582828f4577 or here https://www.youtube.com/watch?v=tmdGaXv30ug :

Type these into Linux shell:

“sudo apt update && sudo apt -y upgrade”

“sudo apt install build-essential”

“sudo apt install net-tools”

“sudo apt install xrdp -y && sudo systemctl enable xrdp”

“sudo apt install -y tasksel”

“sudo tasksel install xubuntu-desktop” (if the gui progress bar doesn`t appear, retry this step)

“sudo apt install gtk2-engines”

2. Display server:

https://sourceforge.net/projects/vcxsrv/

Download from this site: next, next, done.

In case wsl 1 is already installed on your machine i tis best to upgrade it to wsl 2 with this command:

“wsl --set-version Ubuntu-20.04 2”

Or sometimes you need to also take these steps:

"dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart"

And download the linux kernel update package.

Here is how to upgrade: https://docs.microsoft.com/en-us/windows/wsl/install-manual

Run these commands every time you want to start the wsl 2:

1. XLaunch – run (leave everything as is and check the disable access control)
2. run powershell (as an administrator)
3. turn off the firewall
4. run these commands in powershell:
   "wsl"
   "cat /etc/resolv.conf" – copy the ip for the nameserver and insert this id here with a "0" in the end like this:

   "export DISPLAY=172.31.16.1:0"

   "startxfce4"

SUMO installation according to this video: https://www.youtube.com/watch?v=tmdGaXv30ug :

1. Type these commands in Linux shell, possibly in the home directory:
   "sudo apt update"
   "sudo apt-get install cmake python g++ libxerces-c-dev libfox-1.6-dev libgdal-dev libproj-dev libgl2ps-dev swig"
   "sudo apt install git"
   "git clone --recursive https://github.com/eclipse/sumo"
   "export SUMO_HOME="$PWD/sumo""
   "mkdir sumo/build/cmake-build && cd sumo/build/cmake-build"
   "cmake ../.."
   "make -j8" the number after j determines the number of cores used tu process simulations

NS3 installation from this video: https://www.youtube.com/watch?v=cPpJ_mJLkzo :

1. run wsl
2. download ns-3 from here: www.nsnam.org
3. Type these commands in the Linux shell:
   "sudo apt update"

“sudo apt install build-essential autoconf automake libxmu-dev python3-pygraphviz cvs mercurial bzr git cmake p7zip-full python3-matplotlib python-tk python3-dev qt5-qmake qt5-default gnuplot-x11 wireshark”

4. From the downloads directory in wsl move the downloaded ns3-allinone to home directory(the directory is optional)
5. Extract here for ns3-allinone súbor
6. In ns3-allinone directory type this command:
   “./build.py –enable-examples –enable-tests”

NS3 + SUMO:

1. run sumo-gui
2. Generate the desired map and vehicles with: select area a generate scenario.
3. The generated scenario should be present here: “~/sumo/tools/“ its name is a timestamp e.g. “2021-11-23-10-50-30”
4. This command generates a xml file in this directory: “~/sumo/tools/2021-11-23-10-50-30“:
   “sumo -c osm.sumocfg --fcd-output name_of_file.xml”
5. This command generates a tcl file in this directory: “~/sumo/tools/“:
   “python traceExporter.py -i 2021-11-23-10-50-30/ name_of_file.xml --ns2mobility-output=2021-11-23-10-50-30/ name_of_file _2.tcl”
6. The tcl file can now be used as a mobility file in our ns3 simulation which is written in C++.
7. Place your C++ source code here: “/ns-allinone-3.34/ns-3.34/scratch/”
8. Then move to this directory: “/ns-allinone-3.34/ns-3.34/” run the command:
   “./waf --run scratch/name.cc”
9. In case of debugging run:
   “./waf --run scratch/ name --command-template="gdb %s "”, then type run and backtrace