

Ns-3 Installation

1. Prerequisites:

All In One Ubuntu libraries (dependencies) Installation command (in Ubuntu Desktop) [in terminal]

```
apt-get install gcc g++ python python-dev mercurial python-setuptools git synaptic  
qt4-dev-tools libqt4-dev cmake libc6-dev libc6-dev-i386 g++-multilib gdb valgrind gsl-bin  
libgsl2 libgsl-dev flex bison libfl-dev tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-  
dev libgtk2.0-0 libgtk2.0-dev vtun lxc uncrustify doxygen graphviz imagemagick texlive  
texlive-extra-utils texlive-latex-extra texlive-font-utils texlive-lang-portuguese dvipng  
python-sphinx dia python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev  
ipython libboost-signals-dev libboost-filesystem-dev openmpi-bin openmpi-common  
openmpi-doc libopenmpi-dev
```

2. Downloading ns3 code (3.20).

Go to <https://www.nsnam.org/> and download any version, e.g., ns 3.20.

3. Copy the downloaded ns3 zip file to home folder and keep in your destination folder [e.g. ns3]. Then go to that folder in terminal [by typing cd home/ns3] Unzip the zip file

```
tar xjf ns-allinone-3.20.tar.bz2
```

4. Build ns3

```
cd ns-allinone-3.20  
./build.py --enable-examples --enable-tests
```

5. Test or verify

```
cd ns-3.20  
./test.py  
./test.py -c core [to check only core libraries]  
./waf -d debug --enable-examples --enable-tests configure [to debug and configure all the  
examples and test files]
```

6. Run the 1st script

```
cd examples  
cd tutorial  
cp first.cc ../../scratch/
```

```
cd ../../
./waf --run scratch/first
```

Run the 2nd script

```
cd examples
cd tutorial
cp second.cc ../../scratch/
cd ../../
./waf --run scratch/second
```

7. installation of netanim in ns3 and testing on first file

a. Check synaptic that the libraries are being installed correctly or if we install anything that doesn't create any problem

```
sudo synaptic
The files we need:
(i)    qt4-qmake
(ii)   libqt4-dev
(iii)  libxml2-dev
```

b. Install netanim

```
cd Desktop
cd ns-allinone-3.20
cd netanim-3.105
qmake NetAnim.pro      [as root]
make
```

c. Verify

```
./NetAnim
```

8. Check whether everything runs fine.

```
cd ..
cd ns-3.20
cd scratch
gedit first.cc
```

- (i) Include **netanim** header file

```
#include "ns3/netanim-module.h"
```
- (ii) Add some code before simulator run statement (Simulator::Run ());

```
AnimationInterface anim ("anim1.xml");
anim.SetConstantPosition(nodes.Get(0), 1.0, 2.0);
anim.SetConstantPosition(nodes.Get(1), 2.0, 3.0);
```
- (iii) Go to netanim-3.105 directory

```
cd ..
```

```
cd ..
cd netanim-3.105
./NetAnim
Open
/home/bunu/Desktop/ns-allinone-3.20/ns-3.20/netanim1.xml
```

9. Install Openflow, Brite and Click

(i) Openflow

- a. Obtain the OFSID code. An ns-3 specific OFSID branch is provided to ensure operation with ns-3. Use mercurial to download this branch and waf to build the library:

```
$ hg clone http://code.nsnam.org/jpelkey3/openflow
$ cd openflow
```

- b. From the “openflow” directory, run:

```
$ ./waf configure
$ ./waf build
```

(ii) Click

- a. Building Click

The first step is to fetch (`git clone git://github.com/kohler/click.git`) and build Click. At the top of your Click source directory:

```
$: ./configure --enable-userlevel --disable-linuxmodule --enable-nsclick --enable-wifi
$: make
```

The `-enable-wifi` flag may be skipped if you don't intend on using Click with Wifi. *
Note: You don't need to do a 'make install'.

Hint: If you have click installed one directory above ns-3 (such as in the ns-3-allinone directory), and the name of the directory is 'click' (or a symbolic link to the directory is named 'click'), then the `-with-nsclick` specifier is not necessary; the ns-3 build system will successfully find the directory.

If it says 'enabled' beside 'NS-3 Click Integration Support', then you're good to go.
Note: If running modular ns-3, the minimum set of modules required to run all ns-3-click examples is wifi, csma and config-store.

(iii) Brite

a. Building BRITE Integration

The first step is to download and build the ns-3 specific BRITE repository:

```
$ hg clone http://code.nsnam.org/BRITE
$ cd BRITE
$ make
```

This will build BRITE and create a library, libbrite.so, within the Brite directory.

- (iv)** To link to an ns-3 build with this OpenFlow switch module, Click Integration support and Brite support run the following from the ns-3-dev (or whatever you have named your distribution):

```
$: /usr/bin/python2.7 ./waf configure --enable-examples --enable-tests --with-
openflow= path/to/openflow --with-nsclick= path/to/click --with-brite= path/to/Brite
```

```
$ ./waf build
```

```
Under ---- Summary you should see:
"NS-3 OpenFlow Integration    : enabled"
"NS-3 Click Integration      : enabled"
"NS-3 Brite Integration       : enabled"
```

indicating the libraries have been linked to ns-3. Run:

Examples

For an example demonstrating its use in a simple learning controller/switch, run:

```
$: ./waf --run openflow-switch
To see it in detailed logging, run:
$: ./waf --run "openflow-switch -v"
$: ./waf --run nsclick-simple-lan
$: ./waf --run 'brite-generic-example'
$ ./waf --run brite-generic-example --vis
$ mpirun -np 2 ./waf --run brite-MPI-example
```

Validation

This model has one test suite which can be run as follows:

```
$ ./test.py --suite=openflow
```