

ns-3 Tutorial (Part II) Installation

IWING Team, Kasetsart University

Supported platforms

- ▶ **ns-3 is primarily developed on GNU/Linux platforms.**
- ▶ **Current supported platforms are listed as follow**
 - ▶ Linux x86 gcc 4.4, 4.3, 4.2, 4.1, and, 3.4.6.
 - ▶ Linux x86_64 gcc 4.4.0, 4.3.2, 4.2.3, 4.2.1, 4.1.3, 3.4.6
 - ▶ MacOS X ppc and x86 (gcc 4.0.x and 4.2.x)
 - ▶ Cygwin gcc 3.4.4 (debug only), gcc 4.3.2 (debug and optimized)

Installation instructions

- ▶ Install prerequisite packages
- ▶ Download ns3 codes
- ▶ Build ns3
- ▶ Validate ns3

Prerequisite packages for Ubuntu

- ▶ *Minimal requirements for Python:* gcc g++ python python-dev
- ▶ *Debugging and GNU Scientific Library (GSL) support:* gdb valgrind gsl-bin libgsl0-dev libgsl0ldbl
- ▶ *Network Simulation Cradle (nsc):* flex bison
- ▶ *Reading pcap packet traces:* tcpdump
- ▶ *Database support for statistics framework:* sqlite sqlite3 libsqlite3-dev
- ▶ *Xml-based version of the config store:* libxml2 libxml2-dev
- ▶ *A GTK-based configuration system:* libgtk2.0-0 libgtk2.0-dev
- ▶ *Experimental with virtual machines and ns-3:* vtun lxc

Prerequisite packages for Ubuntu

- ▶ *Doxygen and related inline documentation:* doxygen graphviz imagemagick texlive texlive-latex-extra texlive-generic-extra texlive-generic-recommended
- ▶ *Ns3 manual and tutorial:* texinfo dia texlive texlive-latex-extra texlive-extra-utils texlive-generic-recommended texi2html
- ▶ *Support for Gustavo Carneiro's ns-3-pyviz visualizer:* python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev
- ▶ *Support ns-3 development repositories:* mercurial bazaar

All In One Ubuntu Installation command

```
sudo apt-get install gcc g++ python python-dev gdb \  
valgrind gsl-bin libgsl0-dev libgsl0ldbl flex bison tcpdump \  
sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev \  
libgtk2.0-0 libgtk2.0-dev vtun lxc \  
doxygen graphviz imagemagick texlive \  
texlive-latex-extra texlive-generic-extra \  
texlive-generic-recommended texinfo dia \  
texlive-extra-utils texi2html python-pygraphviz \  
python-kiwi python-pygoocanvas libgoocanvas-dev \  
mercurial bzip2
```

Pre-Install

- ▶ www.nsnam.org
 - ▶ Homepage > Developers > Tools > Wiki
 - ▶ Installation > 2.1.1 Ubuntu/Debian
- ▶ `sudo apt-get install gcc g++ python python-dev`
- ▶ `sudo apt-get install build-essential`
- ▶ `sudo apt-get install mercurial`
- ▶ `sudo apt-get install bzip2`
- ▶ `sudo apt-get install gdb valgrind`
- ▶ `sudo apt-get install gsl-bin libgsl0-dev libgsl0ldbl`
- ▶ `sudo apt-get install flex bison libfl-dev`
- ~~▶ `sudo apt-get install g++-3.4 gcc-3.4`~~
- ▶ `sudo apt-get install sqlite sqlite3 libsqlite3-dev`
- ▶ `sudo apt-get install libxml2 libxml2-dev`
- ▶ `sudo apt-get install libgtk2.0-0 libgtk2.0-dev`
- ▶ `sudo apt-get install vtun lxc`
- ▶ `sudo apt-get install uncrustify`
- ▶ `sudo apt-get install doxygen graphviz imagemagick`
- ▶ `sudo apt-get install texlive texlive-extra-utils texlive-latex-extra`
- ▶ `sudo apt-get install python-sphinx dia`
- ▶ `sudo apt-get install python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev`
- ▶ `sudo apt-get install libboost-signals-dev libboost-filesystem-dev`
- ~~▶ `sudo apt-get install openmpi*`~~

Downloading ns3 code (ns 3.18)

▶ By Mercurial

```
cd  
mkdir repos  
cd repos  
hg clone http://code.nsnam.org/ns-3-allinone  
cd ns-3-allinone  
./download.py -n ns-3.18
```

▶ By Tarball

```
cd  
mkdir tarballs  
cd tarballs  
wget http://www.nsnam.org/releases/ns-allinone-3.18.tar.bz2  
tar xjf ns-allinone-3.18.tar.bz2
```


http://code.nsnam.org/

Repositories list

<u>Name</u>	<u>Description</u>	<u>Contact</u>
ns-3-allinone	ns-3 allinone scripts	<ns-developers@isi.edu>
ns-3-bib	unknown	unknown
ns-3-dev	ns-3 development tree	<ns-developers@isi.edu>
ns-3-dev-ref-traces	reference traces for ns-3-dev regression	<ns-developers@isi.edu>
ns-3.10	ns-3.10 release	<ns-developers@isi.edu>
ns-3.4	ns-3.4 release	<ns-developers@isi.edu>
ns-3.4-ref-traces	reference traces for ns-3.4	<ns-developers@isi.edu>
ns-3.5	ns-3.5 release branch	<ns-developers@isi.edu>
ns-3.5-ref-traces	unknown	unknown
ns-3.5.1	unknown	unknown
ns-3.6	ns-3.6 release	<ns-developers@isi.edu>
ns-3.6-ref-traces	reference traces for ns-3.6	<ns-developers@isi.edu>
ns-3.7	ns-3.7 release	<ns-developers@isi.edu>
ns-3.7-ref-traces	reference traces for ns-3.7 release	<ns-developers@isi.edu>
ns-3.8	ns-3.8 release	<ns-developers@isi.edu>
ns-3.8-ref-traces	reference traces for ns-3.8 release	<ns-developers@isi.edu>
ns-3.9	ns-3.9 release	<ns-developers@isi.edu>

Archived release:
<http://www.nsnam.org/releases/>

OLDER RELEASES



All ns-3 releases are archived: the short links below provide quick access to the source code, releases notes, and documentation of every past release:

[ns-3.15](#) (August 2012)

[ns-3.14](#) (June 2012)

[ns-3.16](#) (December 2012)

[ns-3.12](#) (August 2011)

[ns-3.11](#) (May 2011)

[ns-3.13](#) (December 2011)

[ns-3.9](#) (August 2010)

[ns-3.8](#) (May 2010)

[ns-3.10](#) (January 2011)

[ns-3.6](#) (October 2009)

[ns-3.5](#) (July 2009)

[ns-3.7](#) (January 2010)

[ns-3.3](#) (December 2008)

[ns-3.2](#) (September 2008)

[ns-3.4](#) (April 2009)

[ns-3.1](#) (July 2008)

The full directory of releases is available [here](#).



Building ns-3 with ns-3-allinone

- ▶ For the first run, using *build.py*

```
cd repos
./build.py --enable-examples --enable-tests
```

```
ns3@ns3-vbox:~/repos/ns-3-allinone$ ./build.py --enable-examples --enable-tests
# Build NSC
Entering directory `nsc'
=> python scons.py
scons: Reading SConscript files ...
Checking target architecture...(cached) x86
scons: done reading SConscript files.
```

```
[1707/1707] pfile: build/debug/src/template/libns3-template.pc
Waf: Leaving directory `/home/ns3/repos/ns-3-allinone/ns-3-dev/build'
'build' finished successfully (12m18.864s)
```

Modules built:

```
aodv          applications          bridge
```

```
visualizer    wifi                  wimax
```

```
Leaving directory `./ns-3-dev'
```

```
ns3@ns3-vbox:~/repos/ns-3-allinone$
```

Validating ns3 by test.py

- ▶ cd ns-3.18
- ▶ ./test.py

```
ns3@ns3-vbox:~/repos/ns-allinone-3.10/ns-3.10$ ./test.py
Waf: Entering directory `/home/ns3/repos/ns-allinone-3.10/ns-3.10/build'
Waf: Leaving directory `/home/ns3/repos/ns-allinone-3.10/ns-3.10/build'
'build' finished successfully (3.582s)
PASS: TestSuite rocketfuel-topology-reader
PASS: TestSuite lte-propagation-loss-model
PASS: TestSuite lte-bearer
PASS: TestSuite lte-device
PASS: TestSuite lte-phy
```

```
PASS: Example src/contrib/flow-monitor/examples/wifi-olsr-flowmon.py
PASS: Example examples/routing/simple-routing-ping6.py
PASS: Example examples/tutorial/first.py
PASS: Example examples/wireless/wifi-ap.py
PASS: Example examples/wireless/mixed-wireless.py
167 of 167 tests passed (167 passed, 0 skipped, 0 failed, 0 crashed)
ns3@ns3-vbox:~/repos/ns-allinone-3.10/ns-3.10$
```

Running: Hello Simulator

```
./waf --run hello-simulator
```

```
ns3@ns3-vbox:~/repos/ns-allinone-3.10/ns-3.10$ ./waf --run hello-simulator
Waf: Entering directory `/home/ns3/repos/ns-allinone-3.10/ns-3.10/build'
Waf: Leaving directory `/home/ns3/repos/ns-allinone-3.10/ns-3.10/build'
'build' finished successfully (1.465s)
Hello Simulator
ns3@ns3-vbox:~/repos/ns-allinone-3.10/ns-3.10$ █
```