

Wns3

# Pip Packaging and Educational

## Use of ns-3

Gabriel Ferreira

Universidade de Brasília



**UnB**

# Current state of ns-3 distribution

Wns3

- ns-3 is typically distributed as a source tarball
  - Requires additional tools (e.g. compiler, cmake, ninja, tar, etc)
  - Requires some setup (e.g. setting up PATH environment variables)
  - You need to be aware that there is a sequence of steps to build the software
    - Not the case for many students, since introductory classes switched from C to Python
- Some package managers package ns-3 as a C++ library
  - Debian, Spack, Homebrew
  - Limited reach and does not ship with everything needed to develop



**UnB**

# Python bindings and pip packaging

UnB

- Cppyy-based python bindings can be used to provide Python APIs from the C++ ones
  - We just need to package a pre-built ns-3 package with it...
- Pip packaging comes to the rescue
  - One build, many Linux distributions (up to a certain GLibc version)
  - Windows and Mac still in the works due to Cppyy limitations
  - 'pip install ns3' and you are good to go
  - And visualizer works too!



UnB

# Pip installation: Ubuntu 22.04 on WSL

# Wns3

```
gabriel@DESKTOP-2SF520H:~$ pip install ns3
Defaulting to user installation because normal site-packages is not writeable
Collecting ns3
  Downloading ns3-3.38.post272-cp310-cp310-manylinux_2_28_x86_64.whl (68.0 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.0/68.0 MB 4.4 MB/s eta 0:00:00
Collecting cppy==2.4.2
  Downloading cppy-2.4.2.tar.gz (26 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Collecting CPyCppyy==1.12.12
  Downloading CPyCppyy-1.12.12.tar.gz (206 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 206.6/206.6 KB 19.6 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting cppy-cling==6.27.1
  Using cached cppy_cling-6.27.1-py2.py3-none-manylinux2014_x86_64.whl (32.4 MB)
Collecting cppy-backend==1.14.10
  Using cached cppy_backend-1.14.10-py2.py3-none-linux_x86_64.whl
Building wheels for collected packages: cppy, CPyCppyy
  Building wheel for cppy (pyproject.toml) ... done
  Created wheel for cppy: filename=cpypy-2.4.2-py3-none-any.whl size=15351488 sha256=f17f36e7d8d510a85946be6d62ab95582555ed86bf5f3621720df282c3ee48b
  Stored in directory: /home/gabriel/.cache/pip/wheels/73/66/49/38da85078450bb1a5b63405aec0935de2e2991717394ac2f64
  Building wheel for CPyCppyy (pyproject.toml) ... done
  Created wheel for CPyCppyy: filename=CPyCppyy-1.12.12-cp310-cp310-linux_x86_64.whl size=4173446 sha256=a18a745abaa3c288fd6825844e4713d5675e10fc5edb3818b06dbeabbea8f71b
  Stored in directory: /home/gabriel/.cache/pip/wheels/5d/7a/78/ca93def377eef72601bc832691be3fca85d43b4bbedd23c91a
Successfully built cppy CPyCppyy
Installing collected packages: cppy-cling, cppy-backend, CPyCppyy, cppy, ns3
  WARNING: The scripts cling-config, cppy-generator, genreflex and rootcling are installed in
  '/home/gabriel/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-
  warn-script-location.
Successfully installed CPyCppyy-1.12.12 cppy-2.4.2 cppy-backend-1.14.10 cppy-cling-6.27.1 ns
3-3.38.post272
```

```
gabriel@DESKTOP-2SF520H:~$ python3
Python 3.10.6 (main, May 29 2023, 11:10:38) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from ns import ns
>>> ns.LogComponentEnable("DefaultSimulatorImpl", ns.LOG_LEVEL_ALL)
>>> ns.Simulator.Stop(ns.Seconds(10))
DefaultSimulatorImpl:DefaultSimulatorImpl(0x7f3c628d78c0)
DefaultSimulatorImpl:SetScheduler(0x7f3c628d78c0, ns3::MapScheduler[])
DefaultSimulatorImpl:Stop(0x7f3c628d78c0, 1000000000)
DefaultSimulatorImpl:Schedule(0x7f3c628d78c0, 1000000000, 0x7f3c62833740)
>>> ns.Simulator.Run()
DefaultSimulatorImpl:Run(0x7f3c628d78c0)
handle 1000000000
DefaultSimulatorImpl:Stop(0x7f3c628d78c0)
>>> ns.Simulator.Destroy()
DefaultSimulatorImpl:Destroy(0x7f3c628d78c0)
DefaultSimulatorImpl:DoDispose(0x7f3c628d78c0)
DefaultSimulatorImpl::~DefaultSimulatorImpl(0x7f3c628d78c0)
>>>
```



# UnB

# Pip installation: Jupyter notebooks

Wns3

- We could already use the Python bindings via Jupyter but it required a workaround and a local build

```
import sys

# You must run jupyter from the ns-3-dev directory,
# or adjust these paths to the absolute directory path of ns-3-dev
sys.path.append("./ns-3-dev/build/bindings/python")
sys.path.append("./ns-3-dev/build/lib")
```

```
from ns import ns
```

+ Code + Text Copy to Drive

```
!pip install ns3
from ns import ns
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-who
Collecting ns3
  Downloading ns3-3.38.post272-cp310-cp310-manylinux_2_28_x86_64.whl (68.0 MB)
  68.0/68.0 MB 7.5 MB/s eta 0:00:00
Collecting cppy==2.4.2 (from ns3)
  Downloading cppy-2.4.2.tar.gz (26 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
```



UnB

# Educational use of ns-3

Wns3

- ns-3 plus Jupyter allow for rich computer network demonstrations
- Examples are hosted on GitHub, but it cannot render animations
  - [https://github.com/Gabrielcarvfer/ns3\\_for\\_education](https://github.com/Gabrielcarvfer/ns3_for_education)
- Notebooks can be viewed locally or via a visualizer
  - e.g. Google Colab
    - <https://colab.research.google.com/github/>



UnB

Demonstration

Wns3



**UnB**