Report

Hypatia - LEO satellite network simulation framework

Introduction

Hypatia is a simulation framework designed for satellite network research. This report summarizes the system setup, installation, and testing procedures based on the provided documentation.

System Requirements

To ensure proper functionality, the following system requirements must be met:

- **Python Version:** 3.7 or later
- **Operating System:** A recent Linux distribution (e.g., Ubuntu 18+)

In our project, we tried with this system set-up but it gave a lot of bugs, specially with Python 3.12.7 So we downgrade the version to 3.11, then to 3.8.20 then to 3.7. For the OS, we worked directly on WSL. We have found a very efficient way to install, build and run the dependencies which is: Google T4 Server provided by Google Colab Platform.

Installation Steps

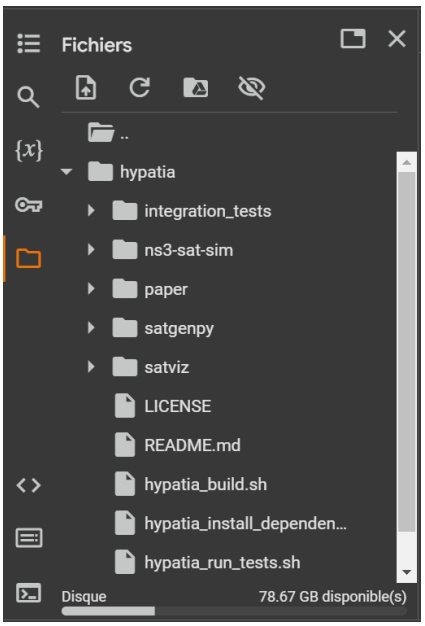
0. Copy Hypatia repo from github to colab:

Before start working on Google T4 Server, we need the hypatia repository deployed on github to be on colab, we lunch the following command to copy the remote repository to our local machine (T4 Server) :

git clone https://github.com/snkas/hypatia.git

```
[ ]  !git clone https://github.com/snkas/hypatia.git

Cloning into 'hypatia'...
remote: Enumerating objects: 1177, done.
remote: Counting objects: 100% (660/660), done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 1177 (delta 510), reused 481 (delta 481), pack-reused 517 (from 1)
Receiving objects: 100% (1177/1177), 33.47 MiB | 14.28 MiB/s, done.
Resolving deltas: 100% (743/743), done.
```

Now, we see that the hypatia repository is disponible on our colab workspace.



1. Install Dependencies

Before running Hypatia, dependencies need to be installed. This was done using the following command:

bash hypatia_install_dependencies.sh

```
[ ]  !cd hypatia && bash hypatia_install_dependencies.sh

     Preparing metadata (setup.py) ... done
   Building wheels for collected packages: networkload
     Building wheel for networkload (setup.py) ... done
     Created wheel for networkload: filename=networkload-1.3-py3-none-any.w
     Stored in directory: /tmp/pip-ephem-wheel-cache-16hq53gq/wheels/22/84/
   Successfully built networkload
   Installing collected packages: networkload
   Successfully installed networkload-1.3
   Reading package lists... Done
   Building dependency tree... Done
   Reading state information... Done
   gnuplot is already the newest version (5.4.2+dfsg2-2).
   0 upgraded, 0 newly installed, 0 to remove and 25 not upgraded.

   Hypatia dependencies have been installed.
```

This script will automatically download and install the necessary packages required for Hypatia to function correctly.

2. Build Modules

Hypatia consists of multiple modules that need to be built. The following command attempts to build all modules:

bash hypatia_build.sh

After running the command **`bash hypatia_build.sh`**, we encountered numerous unexpected bugs and errors. We resolved the issue by replacing the **`basic-simulation.cc`** file, as the original version contained coding errors that we fixed.

```
[ ]  !cp /content/basic-simulation.cc /content/hypatia/ns3-sat-sim/simulator/contrib/basic-sim/model/core/basic-simulation.cc
```

After 43 minutes of building, the script arrived finally to build most of the hypatia modules.

```
[ ]  !cd hypatia && bash hypatia_build.sh

[2892/2894] Compiling src/fd-net-device/helper/encode-decode.cc
[2893/2894] Compiling src/fd-net-device/helper/raw-sock-creator.cc
[2894/2894] Linking build/debug_all/src/fd-net-device/ns3.31-raw-sock-creator-debug
[2895/2897] Compiling src/tap-bridge/model/tap-encode-decode.cc
[2896/2897] Compiling src/tap-bridge/model/tap-creator.cc
[2897/2897] Linking build/debug_all/src/tap-bridge/ns3.31-tap-creator-debug
Waf: Leaving directory `/content/hypatia/ns3-sat-sim/simulator/build/debug_all'
Build commands will be stored in build/debug_all/compile_commands.json
'build' finished successfully (42m54.614s)

Modules built:
antenna                  aodv                     applications
basic-sim (no Python)    bridge                   buildings
config-store             core                     csma
csma-layout              dsdv                     dsr
energy                   fd-net-device            flow-monitor
internet                 internet-apps            lr-wpan
lte                      mesh                     mobility
mpi                      netanim                  network
nix-vector-routing       olsr                     point-to-point
point-to-point-layout    propagation              satellite (no Python)
satellite-network (no Python) sixlowpan               spectrum
stats                    tap-bridge               test (no Python)
topology-read            traffic-control          uan
virtual-net-device       wave                     wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite                    click                    openflow
visualizer

Nothing to build for satgenpy.
Nothing to build for satviz.
Nothing to build for paper.

Hypatia modules have been built.
```

3. Run Tests

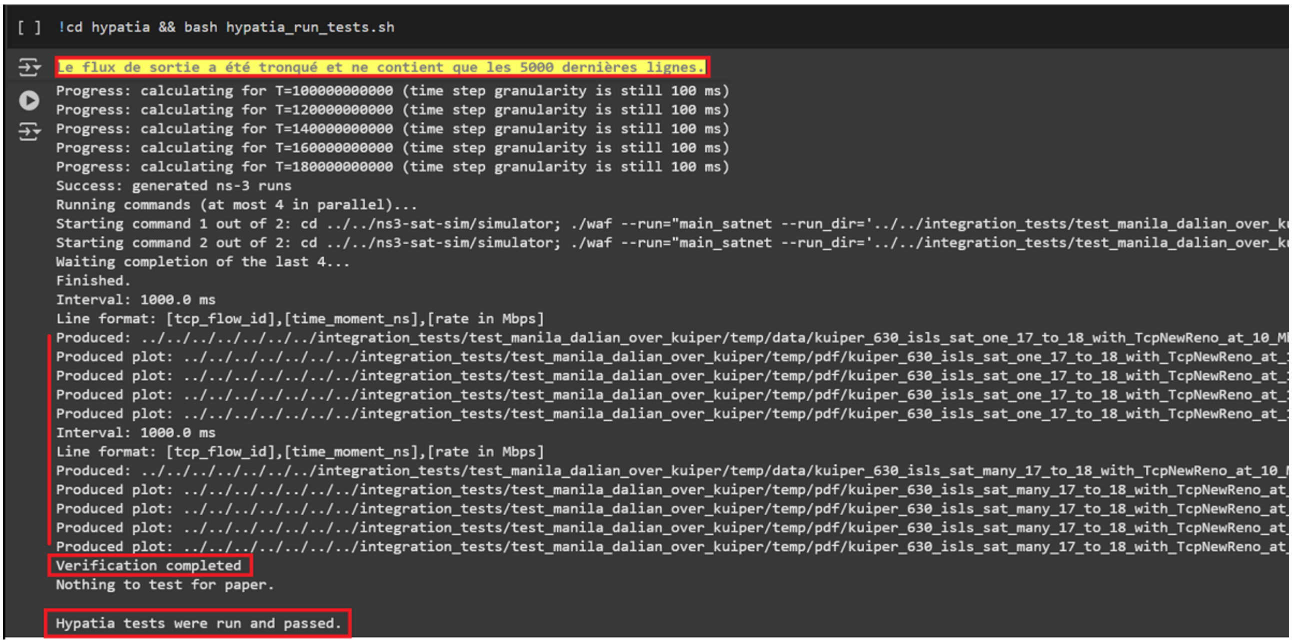
To verify that Hypatia is correctly installed and functioning, we run the test suite using:

bash hypatia_run_tests.sh

The first time we launched this command; it was blocked after one hour of running those tests so it didn't work for the first or second time because there were many bugs on the original files of hypatia. After extensive interpretation and analysis, we replaced all the necessary files - after fixing all bugs and errors - and executed the following commands:

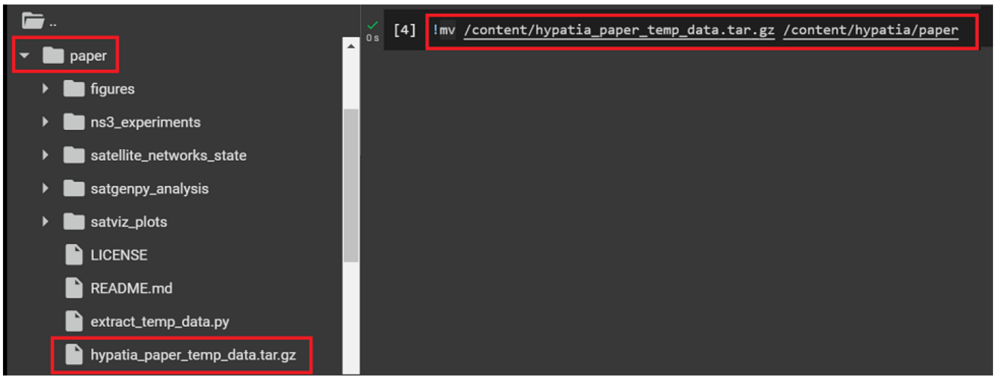
```
[ ]  !cp /content/top.html /content/hypatia/satviz/static_html/top.html

[ ]  !cp /content/visualize_constellation.py /content/hypatia/satviz/scripts/visualize_constellation.py

[ ]  !pip install dms2dec

[ ]  !apt-get install screen
```

After that, we launch the run command which ensures that all modules and dependencies are correctly set up and that the system operates as expected - it takes about 38 minutes to run all the tests and make all the plots.



Paper Reproduction

By following the given paper documentation. It's evident that some components of Hypatia require extensive computational time. In order to expedite the process, we download pre-generated data from the github repository then we upload it to colab.



To proceed, we ensure the following Python packages are installed: numpy, exputil and networkload

```
[5]  !pip install numpy
     !pip install git+https://github.com/snkas/exputilpy.git@v1.6
     !pip install git+https://github.com/snkas/networkload.git@v1.3

     Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (1.26.4)
     Collecting git+https://github.com/snkas/exputilpy.git@v1.6
       Cloning https://github.com/snkas/exputilpy.git (to revision v1.6) to /tmp/pip-req-build-xxc933v1
       Running command git clone --filter=blob:none --quiet https://github.com/snkas/exputilpy.git /tmp/pip-req-build-xxc933v1
       Running command git checkout -q 014750099016c725346d5a164e738c93c8a42224
       Resolved https://github.com/snkas/exputilpy.git to commit 014750099016c725346d5a164e738c93c8a42224
       Preparing metadata (setup.py) ... done
     Building wheels for collected packages: exputil
       Building wheel for exputil (setup.py) ... done
       Created wheel for exputil: filename=exputil-1.6-py3-none-any.whl size=7663 sha256=21580c40b98a76342d76220287b806b8f4593f
       Stored in directory: /tmp/pip-ephem-wheel-cache-whrpw3k6/wheels/d2/f5/59/a168001a09e7693b7837403fdc7e6b5927f889c7d540039
     Successfully built exputil
     Installing collected packages: exputil
     Successfully installed exputil-1.6
     Collecting git+https://github.com/snkas/networkload.git@v1.3
       Cloning https://github.com/snkas/networkload.git (to revision v1.3) to /tmp/pip-req-build-ks36cmza
       Running command git clone --filter=blob:none --quiet https://github.com/snkas/networkload.git /tmp/pip-req-build-ks36cmz
       Running command git checkout -q 3531c28466a8461cdd0cc1708426389305cd48f8
       Resolved https://github.com/snkas/networkload.git to commit 3531c28466a8461cdd0cc1708426389305cd48f8
       Preparing metadata (setup.py) ... done
     Building wheels for collected packages: networkload
       Building wheel for networkload (setup.py) ... done
       Created wheel for networkload: filename=networkload-1.3-py3-none-any.whl size=26075 sha256=1dcf3cd7b6bf6a0028e4f047838c6
       Stored in directory: /tmp/pip-ephem-wheel-cache-k9osfp9g/wheels/22/84/cb/2b60043c2d2dae925702da8cbfcf9a618393034ec568d09
     Successfully built networkload
     Installing collected packages: networkload
     Successfully installed networkload-1.3
```

We install gnuplot by the following command:

sudo apt-get install gnuplot

Then, we extract temp data from the hypatia_paper_temp_data.tar.gz

```
[7]  !cd hypatia/paper/ && python extract_temp_data.py

     > Deleting..... figures/a_b/tcp_cwnd/pdf
     > Creating..... figures/a_b/tcp_cwnd/pdf
     > Extracting... figures/a_b/tcp_cwnd/pdf
     > Deleting..... figures/a_b/tcp_isls_vs_gs_relays/pdf
     > Creating..... figures/a_b/tcp_isls_vs_gs_relays/pdf
     > Extracting... figures/a_b/tcp_isls_vs_gs_relays/pdf
     > Deleting..... figures/a_b/tcp_mayhem/pdf
     > Creating..... figures/a_b/tcp_mayhem/pdf

     > Creating..... ns3_experiments/traffic_matrix/pdf
     > Extracting... ns3_experiments/traffic_matrix/pdf
     > Deleting..... ns3_experiments/traffic_matrix_load/runs
     > Creating..... ns3_experiments/traffic_matrix_load/runs
     > Extracting... ns3_experiments/traffic_matrix_load/runs
     > Deleting..... ns3_experiments/traffic_matrix_load/data
     > Creating..... ns3_experiments/traffic_matrix_load/data
     > Extracting... ns3_experiments/traffic_matrix_load/data
     > Deleting..... ns3_experiments/traffic_matrix_load/pdf
     > Creating..... ns3_experiments/traffic_matrix_load/pdf
     > Extracting... ns3_experiments/traffic_matrix_load/pdf

     Removing temporary temp_data/ directory which was used for extraction

     Finished.
```

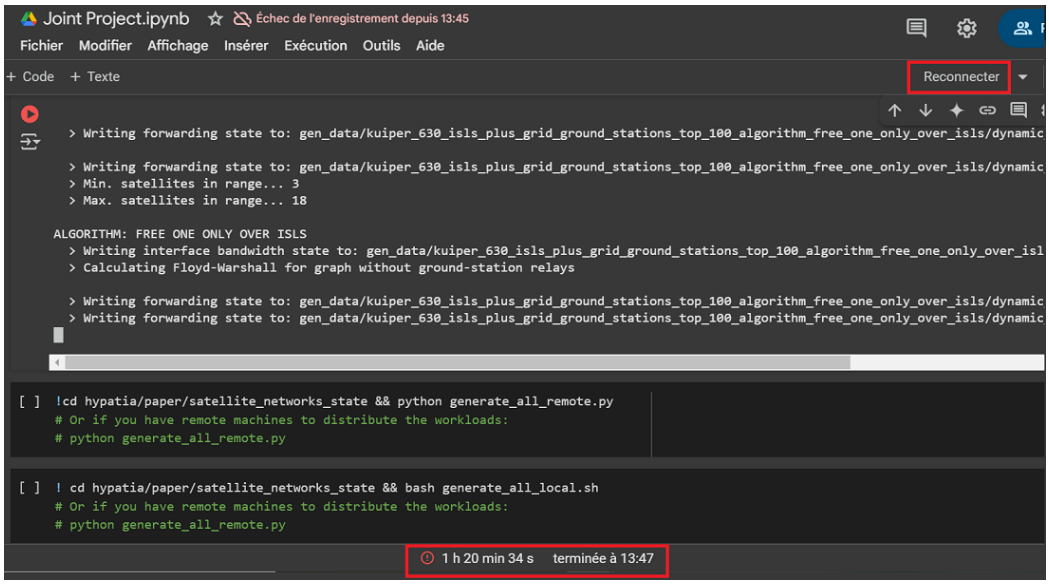
Step 1: generating LEO satellite network dynamic state over time

In order to generate the satellite network state. We need the *satgenpy* or *satgen* Python module, which are located at the root of Hypatia. For each satellite network defined in here (Kuiper-630, Starlink-550, Telesat-1015), it generates for a ranging set of scenarios the following static state:

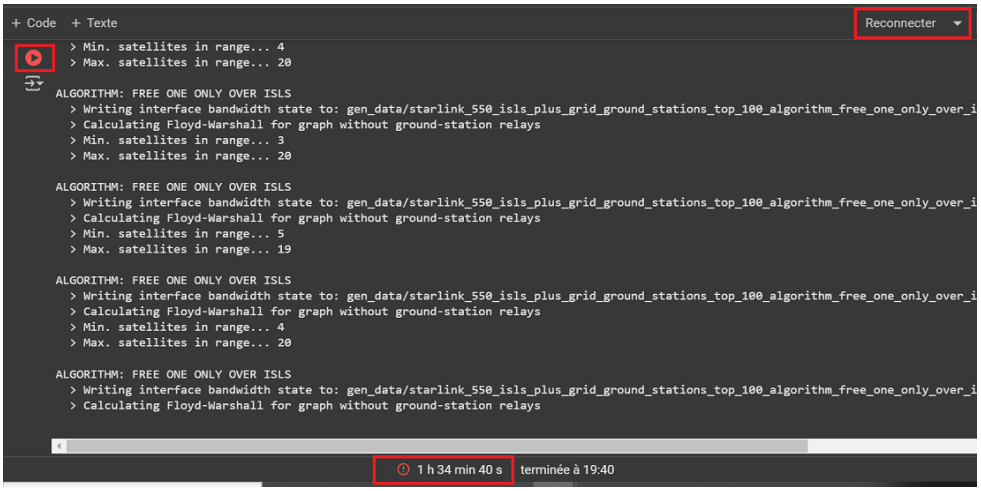
- List of satellites which are encoded using TLEs (*tles.txt*)
- List of ISLs (*isls.txt*)

- List of ground stations (*ground_stations.txt*)
- Description of the maximum GSL and ISL length in meters (*description.txt*)
- Number of GSL interfaces each node (satellite and ground station) has, and their total bandwidth sum (*gsl_interfaces_info.txt*)

Before launching the command: **bash generate_all_local.sh,** we checked that all hypatia dependencies were successfully installed (including *satgenpy* modules which are essential for generating satellite network states).



After more than 80 minutes, it disconnects from the Google T4 server. We retry this again and again but the same problem persists and this time after more than one hour and a half.



It must generate :

gen_data

|-- 25x25_algorithm_free_one_only_over_isls

|--kuiper_630_isls_none_ground_stations_paris_moscow_grid_algorithm_free_one_only_gs_relays

|-- kuiper_630_isls_plus_grid_ground_stations_top_100_algorithm_free_one_only_over_isls

|-- starlink_550_isls_plus_grid_ground_stations_top_100_algorithm_free_one_only_over_isls

|-- telesat_1015_isls_plus_grid_ground_stations_top_100_algorithm_free_one_only_over_isls

Step 2: build ns-3 simulator

Here we install all ns-3 dependencies (inherited from basic-sim ns-3 module) using these commands:

sudo apt-get update

sudo apt-get -y install openmpi-bin openmpi-common openmpi-doc libopenmpi-dev lcov gnuplot

pip install numpy statsmodels

pip install git+https://github.com/snkas/exputilpy.git@v1.6

git submodule update --init --recursive

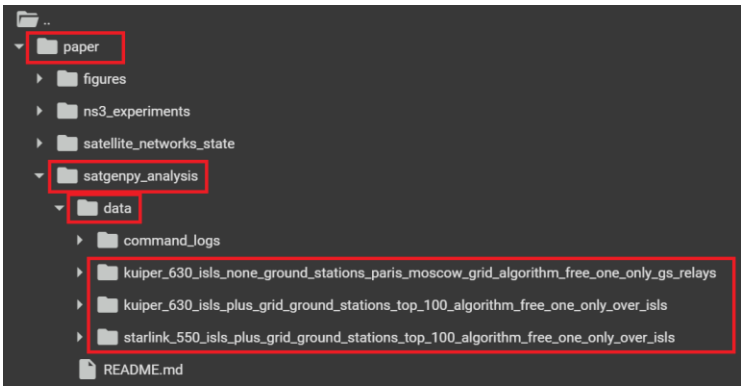
Then, we launch the optimized build using: **bash build.sh --optimized,** we obtain:

```
!cd hypatia/ns3-sat-sim && bash build.sh --optimized
[2897/2897] Linking build/debug_all/src/tap-bridge/ns3.31-tap-creator-debug
Waf: Leaving directory `/content/hypatia/ns3-sat-sim/simulator/build/debug_al…
Build commands will be stored in build/debug_all/compile_commands.json
'build' finished successfully (42m54.614s)

Modules built:
antenna                  aodv                     applications
basic-sim (no Python)    bridge                   buildings
config-store             core                     csma
csma-layout              dsdv                     dsr
energy                   fd-net-device            flow-monitor
internet                 internet-apps            lr-wpan
lte                      mesh                     mobility
mpi                      netanim                  network
nix-vector-routing       olsr                     point-to-point
point-to-point-layout    propagation              satellite (no Python)
satellite-network (no Python) sixlowpan               spectrum
stats                    tap-bridge               test (no Python)
topology-read            traffic-control          uan
virtual-net-device       wave                     wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite                    click                    openflow
visualizer

Nothing to build for satgenpy.
Nothing to build for satviz.
Nothing to build for paper.

Hypatia modules have been built.
```

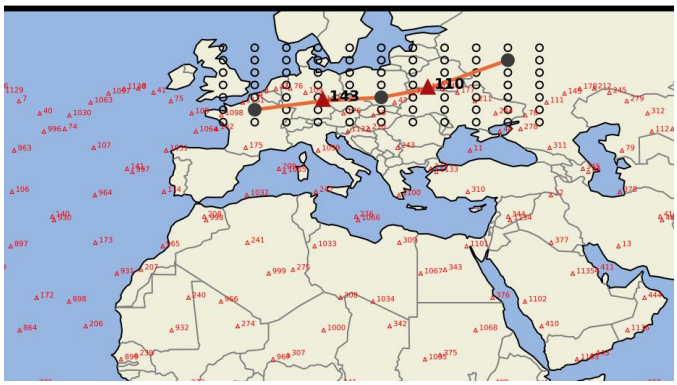
Step 3: performing analysis using satgenpy

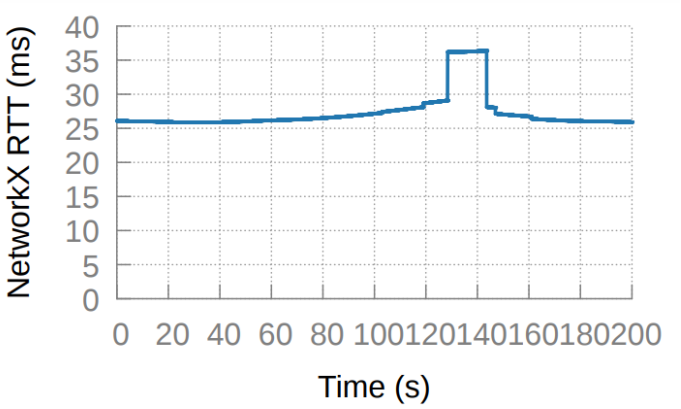
Till now we have the generated satellite network state data over time in satellite_networks_state, here we can launch an analysis of the constellations, for both as a whole, as well as for a few particular pairs. So before perform the full analysis, we shall check that all *satgenpy* dependencies were installed then we launch the command: **python perform_full_analysis.py**

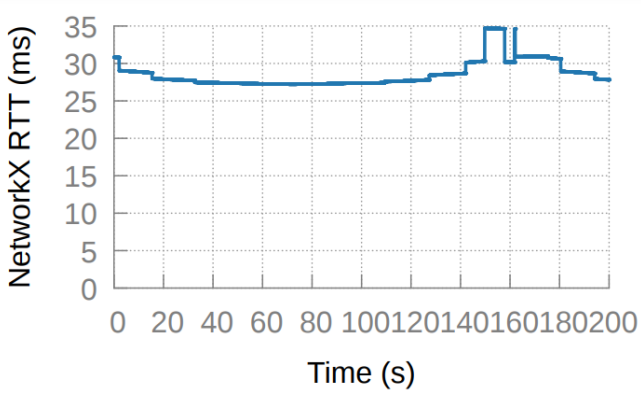
The analysis for each constellation is now in data/<satellite network name>

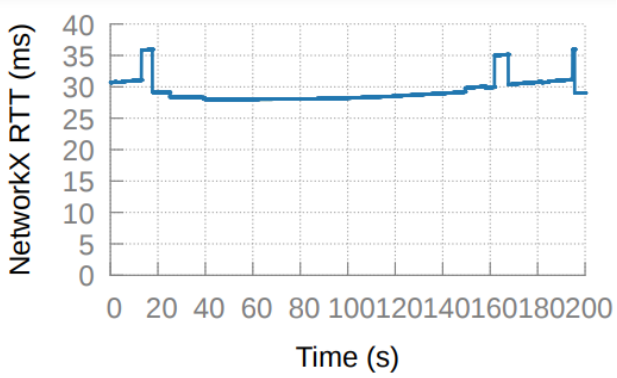


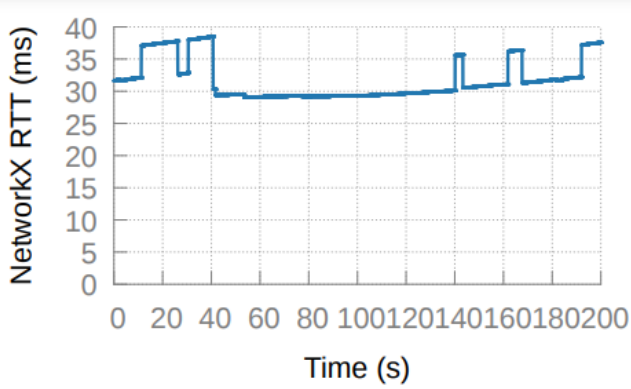
Here some screenshots from the PDF file generated for kuiper satellite:




NetworkX RTT (ms)
0
5
10
15
20
25
30
35
40
0 20 40 60 80 100120140160180200
Time (s)


NetworkX RTT (ms)
0
5
10
15
20
25
30
35
0 20 40 60 80 100120140160180200
Time (s)


NetworkX RTT (ms)
0
5
10
15
20
25
30
35
40
0 20 40 60 80 100120140160180200
Time (s)


NetworkX RTT (ms)
0
5
10
15
20
25
30
35
40
0 20 40 60 80 100120140160180200
Time (s)

Step 4: running ns-3 experiments

Working on it

Conclusion

Setting up Hypatia requires a Linux environment with Python 3.7+, installation of dependencies, module compilation, and testing. The documentation provides a structured approach to ensure a smooth setup and usage experience. Users are encouraged to refer to the tutorial located in the paper/README.md file for in-depth guidance on using Hypatia effectively.