Institute for the Wireless Internet of Things
at Northeastern University

SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems

Leonardo Bonati

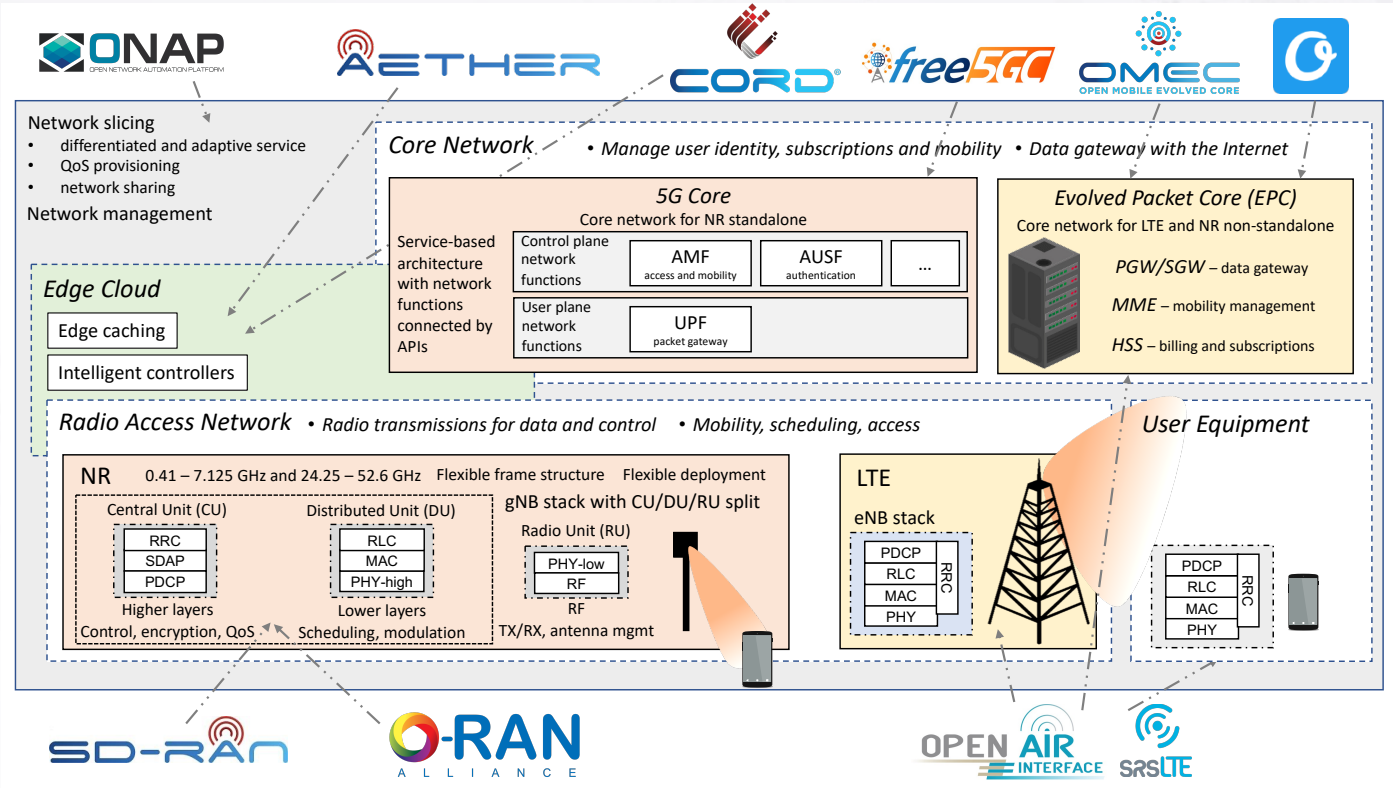
bonati.l@northeastern.edu

L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, *"SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems,"* in Proceedings of ACM MobiSys, June 2021

Repository: https://github.com/wineslab/colosseum-scope

Key Role of Softwarization and Virtualization in NextG Networks

- **Deploy** custom services on generic hardware
- **Program** network functionalities in **software**
- **Dynamically optimize** network performance



L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "*Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead,*" Computer Networks, vol. 182, December 2020

Need for:

- Open research tools to **prototype** NextG solutions in **controlled environments**
- **Large-scale datasets** to design/train AI/ML solutions

SCOPE: Toward Easier Reconfigurability

- Extends srsRAN with:
  - Network slicing functionalities
  - MAC-layer functionalities (e.g., custom scheduling policies)
  - PHY-layer functionalities (e.g., control of MCS)
- Automatic **data-collection** of RAN statistics
- Flexible **open APIs** to **control** the RAN in **real time**
- Portable implementation through LXC containers
- Enables to **prototype** custom control logic **at-scale**

Fully-integrated with Colosseum

Easier to run experiments*

- w/o SCOPE: Several hours to **setup** and **manually run** every single experiment
- w/ SCOPE: **Configure** container **once**, "seamlessly" **run different experiments** (at scale) from **single CLI**, easily modify configuration parameters/scenarios

Easier to collect data*

- w/o SCOPE: **manually** perform several hundreds of experiments and collect data → **may take a very long time**
- w/ SCOPE: **set up once** and **run multiple automatic jobs** on Colosseum in parallel, get data at the end → **easier and faster**

* Compared to traditional lab setup

SCOPE & Colosseum Example

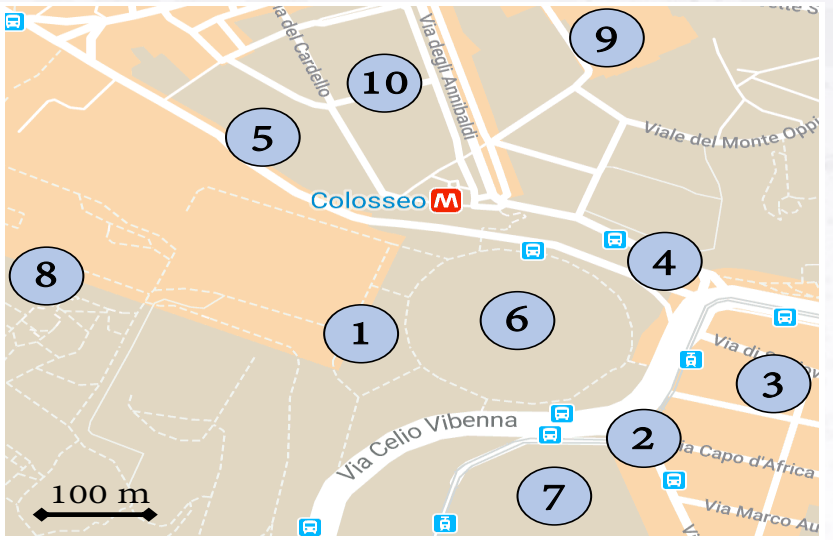
W/ Colosseum

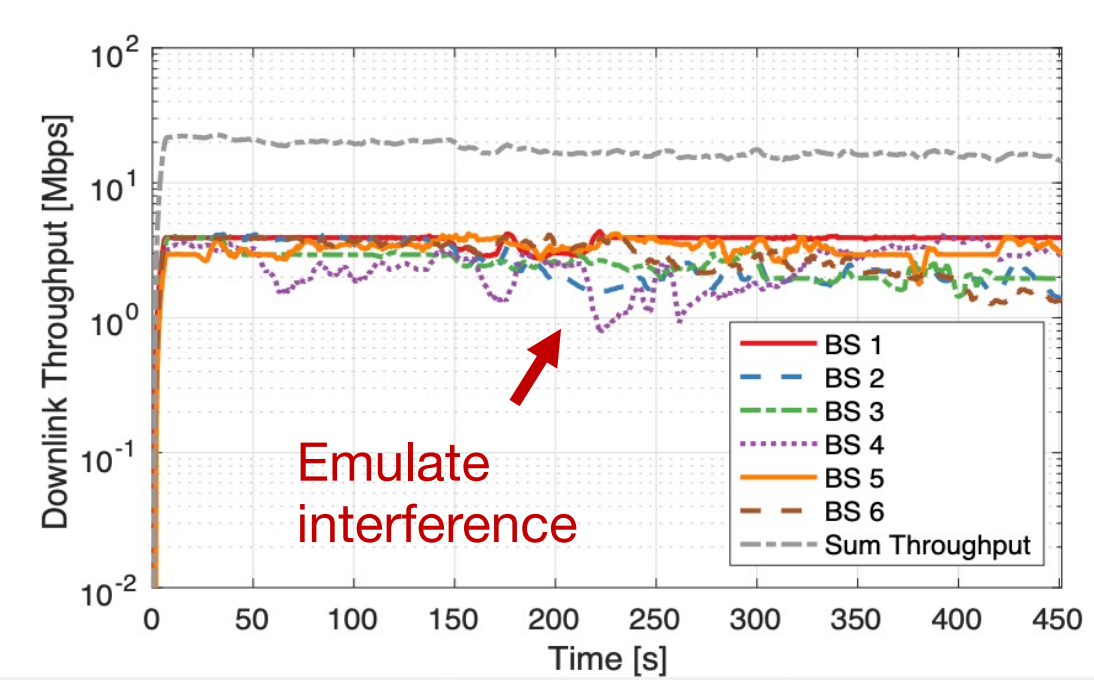
- Emulate Rome downtown scenario
- Deploy softwarized base stations & UEs
- Evaluate performance

W/o Colosseum

- Go **on-site** with equipment
- **Outcome may vary** based on hour of the day, weather conditions, etc.

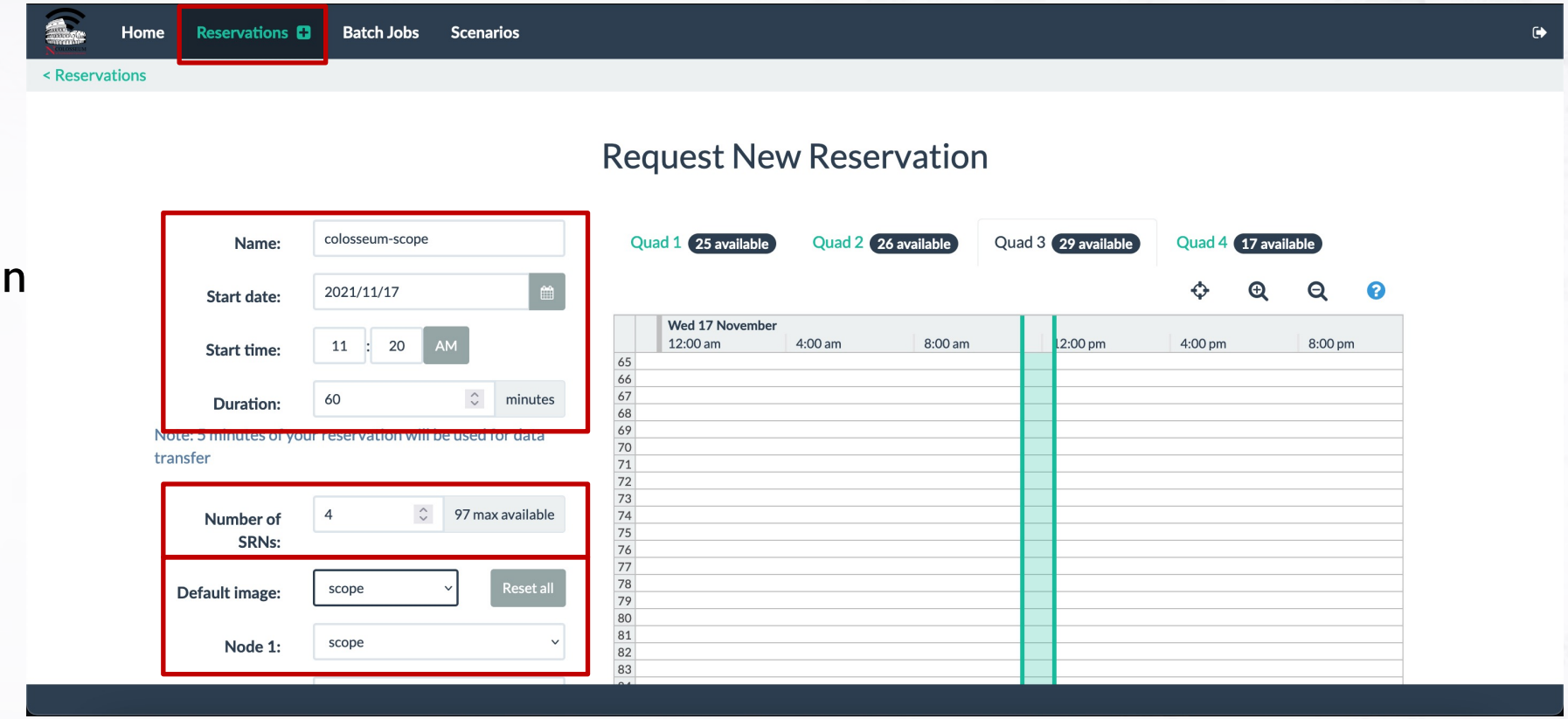
Rome BS locations from OpenCellid: https://opencellid.org





SCOPE Example

- Log into Colosseum website*
- Request a new reservation
- Specify name/date/ time/duration
- Set number of nodes



- Select the **scope** container image for the nodes (default credentials: **root**/**scope**)

Institute for the Wireless Internet of Things at Northeastern

* https://experiments.colosseum.net

SCOPE API at a High-level

Control RAN via **open APIs**:

- Control **network slicing** (e.g., associate users to slices, set slicing policies and modify at run-time)
- Control **MAC-layer parameters** (e.g., change slicing policies at run-time)

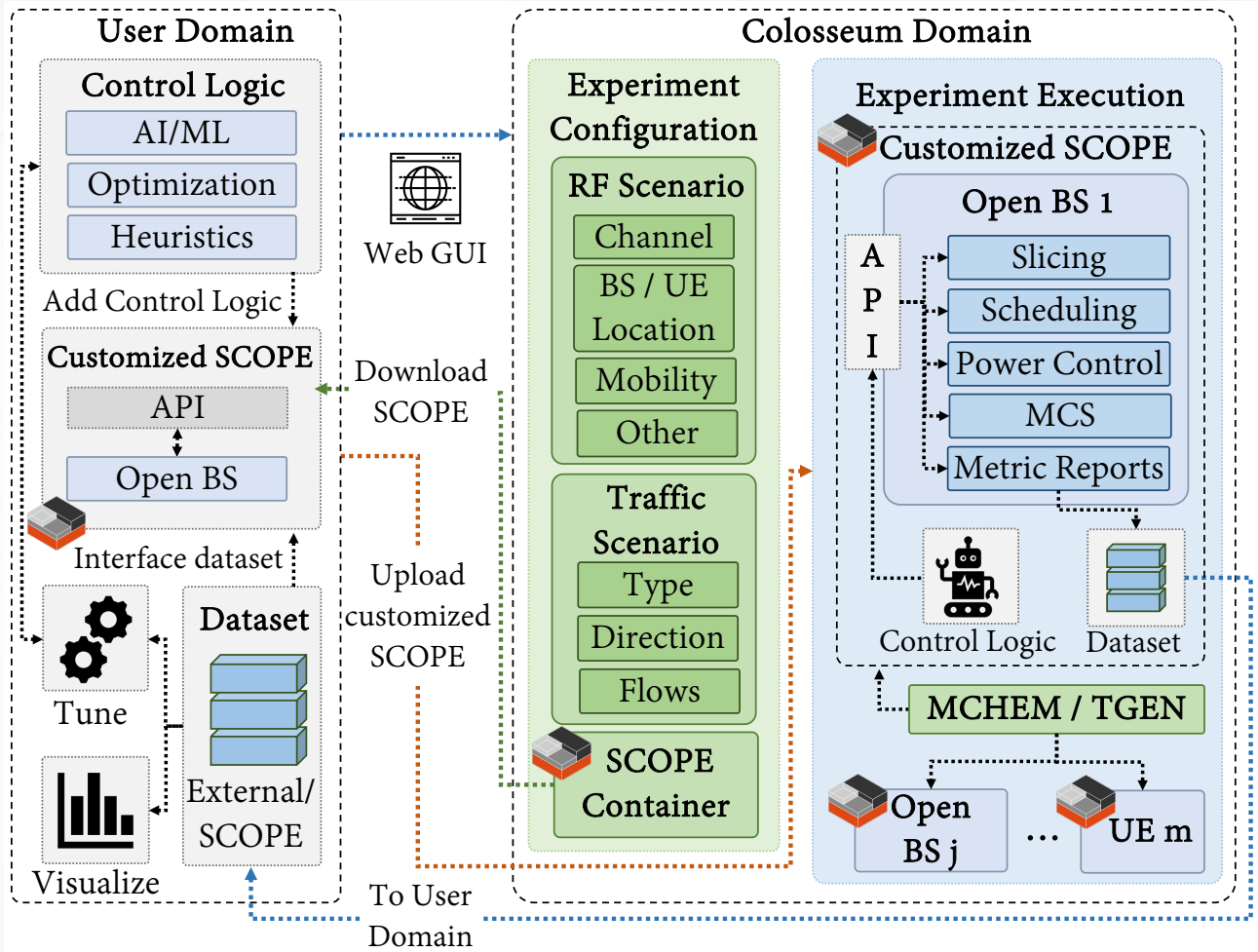
```
1.  import scope_api as sc, time

2.  while experiment_running:
3.    wnd_metrics = sc.read_metrics(time_window)

4.    for slice_id, slice_metrics in wnd_metrics.items():
5.      slice_users = slice_metrics['ue']
6.      slice_rbg = slice_metrics['rbg']
7.      sc.set_mcs(slice_users, mcs_level, 'dl')

8.      if slice_metrics['buffer'] > threshold:
9.        sc.set_slice(slice_id, 'proportionally', slice_rbg + 2)
10.     else:
11.       sc.set_slice(slice_id, 'round-robin', slice_rbg - 2)

12.   time.sleep(timeout)
```

High-level example of SCOPE APIs.

- Control **PHY-layer parameters** (e.g., power control and adaptation, MCS selection at run-time)
- **Data collection** (e.g., collect and query performance metrics at run-time)

Prototyping At-Scale Example

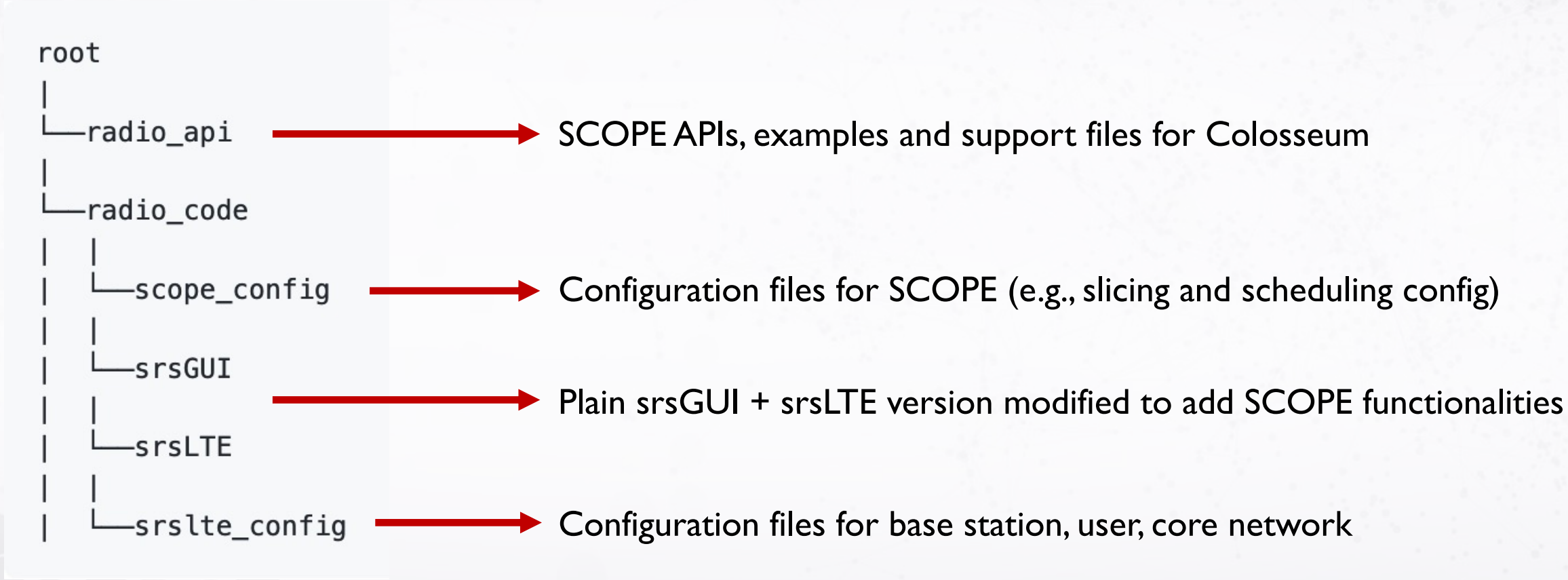
- Extend the provided "plain" SCOPE container with **custom control logic**
- **Select** RF and traffic **scenarios** to run
- Interface with SCOPE APIs to **optimize** network performance **at run-time**
- Save metrics and statistics for **offline parsing/dataset creation**



Institute for the Wireless Internet of Things at Northeastern

SCOPE Quick Tour

SCOPE structure:



Repository: https://github.com/wineslab/colosseum-scope

SCOPE Quick Tour: radio_api

Main SCOPE API scripts:

- **scope_start.py**: quick start script for running SCOPE on Colosseum testbed
  - **python3 scope_start.py --config-file radio_interactive.conf**
  - Parse configuration files
  - Configure/start cellular applications
  - Using the quick start script outside Colosseum might require minor adjustments
- **scope_api.py**: APIs to interact with/reconfigure the base station
- **constants.py**: constant parameters used by the other scripts
- **support_functions.py**: additional support functions

SCOPE Quick Tour: radio_api

Exemplary scripts (to be run at the base station):

- **heuristic.py**: read performance metrics from dataset and implement arbitrary heuristic policies, in this case:
  - read run-time performance metrics
  - modify slicing and scheduling policies based on metrics

- **slice_heuristic.py**: periodically modify number of PRBs allocated to the network slices

SCOPE Quick Tour: radio_api

Configuration files:

- **radio_interactive.conf**: exemplary configuration file to use with the **scope_start.py** script
- **radio.conf**: dummy configuration file replaced by Colosseum when running batch jobs
- **heuristic.conf**: exemplary configuration file to use with **heuristic.py** script
- **slice_heuristic.conf**: Exemplary configuration file to use with **slice_heuristic.py** script

+ Additional support files for Colosseum

Main API Configuration Parameters

General configuration:

- **capture-pkts**: enable packet capture/dumps on .pcap files
- **config-file**: JSON-formatted configuration file where to read these parameters from
- **iperf**: generate traffic through iPerf3

MAC- PHY-layer configuration:

- **global-scheduling-policy**: set MAC-layer scheduling policy for all slices (choose among round-robin, waterfilling and proportionally fair)
- **force-dl-modulation**/**force-ul-modulation**: Force downlink/uplink modulation of base station/users

Main API Config Parameters, cont'd

Network slicing:

- **custom-ue-slice**: use users-slice associations passed through the slice-users parameter
- **network-slicing**: enable network slicing
- **slice-allocation**: configure slicing policies
- **slice-scheduling-policy**: select scheduling policy for each slice of the network

SCOPE Quick Tour: scope_config

Configuration files periodically reloaded at run time by the "*Scopified*" srsRAN:

- **scope_cfg.txt**: global configuration file to enable/disable SCOPE functionalities
- **config**: directory populated at run time with user-related parameters
- **metrics/csv**: CSV files on user performance are automatically logged in this directory at run time

SCOPE Quick Tour: scope_config, cont'd

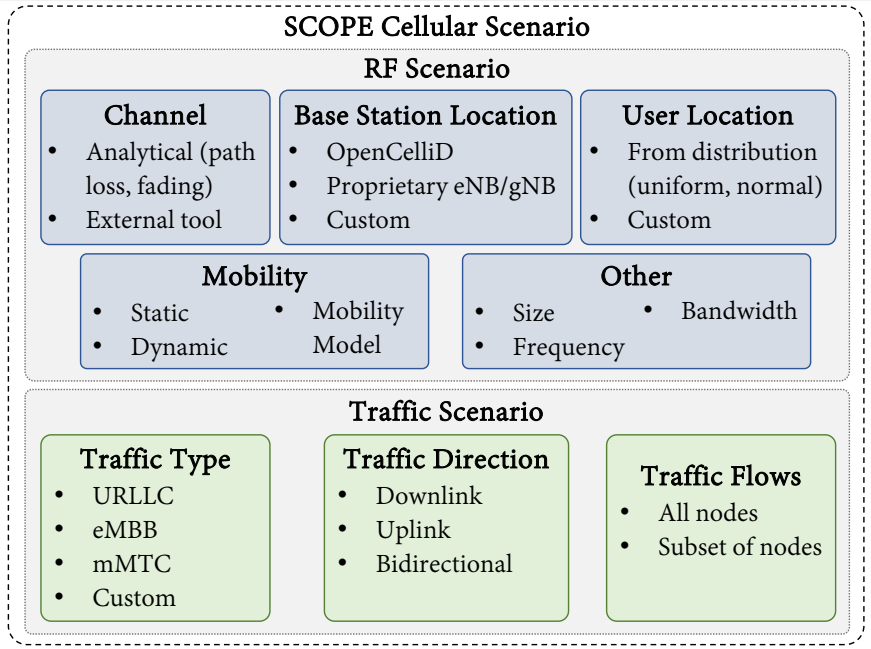
Configuration files:

- **slicing**: contains slicing- and user- related configuration files
  - **slice_allocation_mask_tenant_*.txt**: RBG allocation mask for tenant
  - **slice_scheduling_policy.txt**: specifies the scheduling policy to for each network slice
  - **ue_imsi_slice.txt**: slice-users associations
  - **ue_imsi_modulation_dl.txt**/**ue_imsi_modulation_ul.txt**: configuration file to force downlink/uplink modulation for specific users

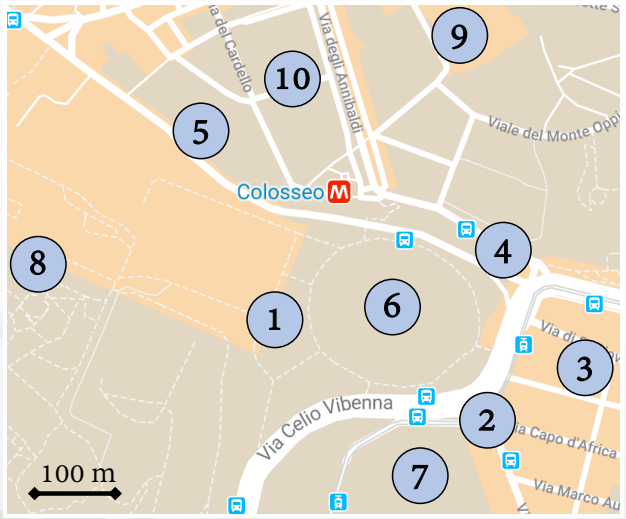
SCOPE Cellular Scenarios

Allows to specify:

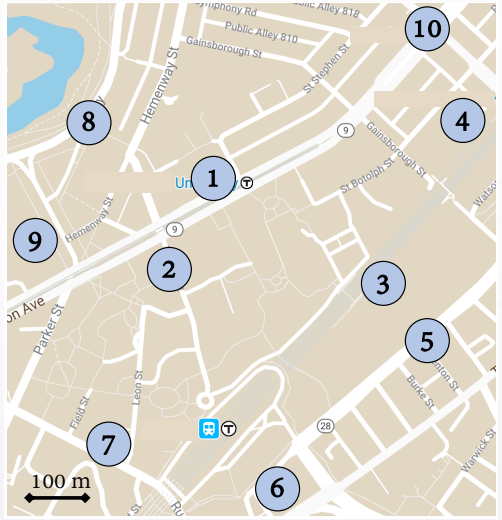
- **Wireless channel effects**, e.g., path loss, position/distance of BSs/UEs, mobility/speed, etc.
- **Traffic flows and types** among nodes



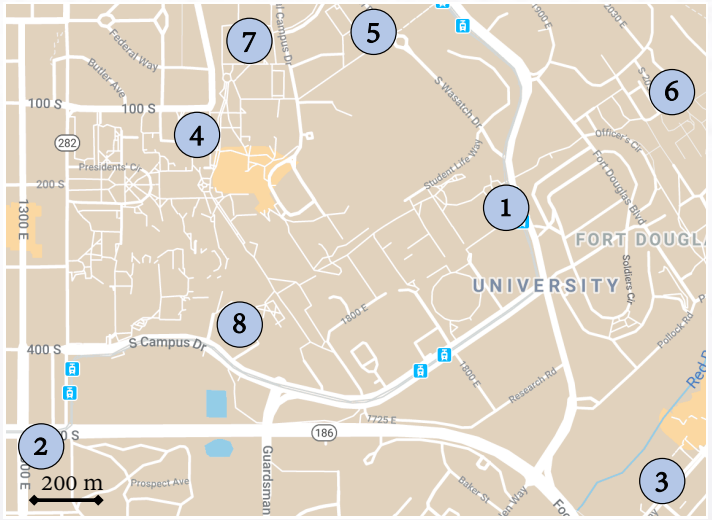
Example of designed large-scale scenarios (blue circles = BS):



Rome, Italy



Boston, MA

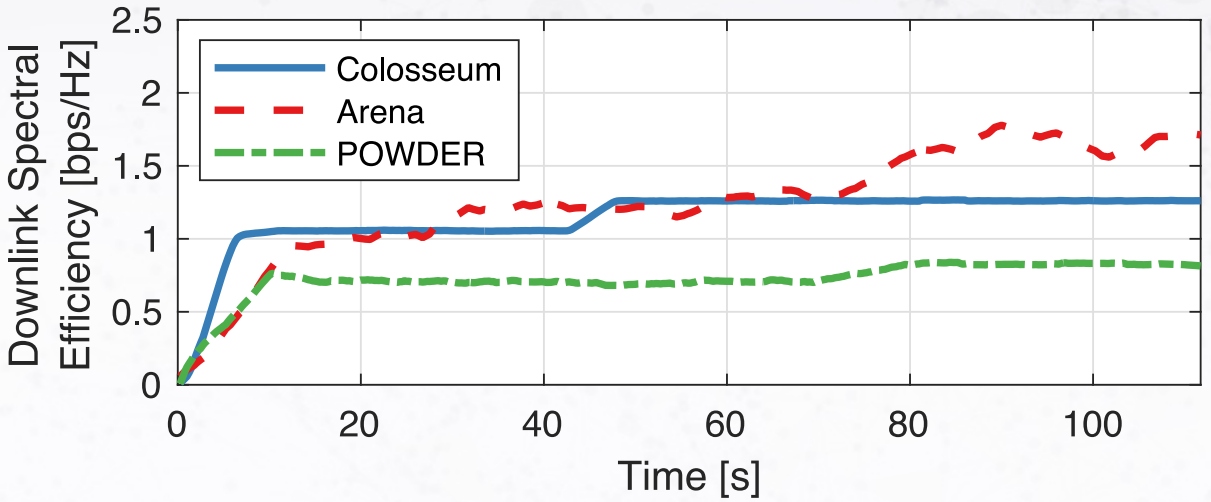


Salt Lake City, UT (POWDER)

Prototype At-scale, Test in the Wild

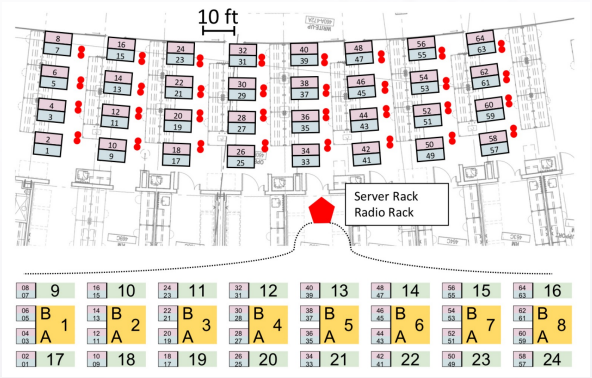
SCOPE can be **ported to different testbeds**

- **Prototype** on Colosseum
- **Validate in real environment**
- **Test** large-scale capabilities on **city-scale platforms**



Test at-a-scale on emulated scenarios

Validate in real wireless environment



Test large-scale capabilities

Institute for the Wireless Internet of Things
at Northeastern University

Thank You!

Leonardo Bonati

bonati.l@northeastern.edu

L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, *"SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems,"* in Proceedings of ACM MobiSys, June 2021

Repository: https://github.com/wineslab/colosseum-scope