OAI 5G NR SA MODE DEMO OVER COLOSSEUM AND OAX CI

Presenter:
Shweta Shrivastava
s.shrivastava@northeastern.edu

Team:
Abhimanuy Gosain
agosain@coe.neu.edu
Mario Joa-Ng
m.joa-ng@northeastern.edu

OpenAirX-Labs, Northeastern University
Outline

• Overview of Colosseum
• OAI Standalone mode demo over Colosseum
  • Throughput Demo
  • Multiple UE connection
• CI Setup
• CI Demo
Colosseum: A Large-Scale Wireless Emulator Testbed

• A massive $20M wireless systems testbed developed by DARPA housed at Northeastern University
  • Funded by NSF

• 128 Computer servers (Intel Xeon), 128 programmable radios (USRP X310)

• Massive Channel Emulator (MCHEM)
  • 256 x 256 100 MHz RF channel emulation
  • Real-time emulation and complex RF scenarios

• Remotely accessible and used by the research community for experiments in
  • Spectrum Sharing
  • AI + Wireless
  • 5G (softwarization, slicing, security)
  • IoT
Running Experiments in Colosseum

- Every user experiment in Colosseum is run in containers
- Colosseum uses LXC containers
- Colosseum provides base container images that users can customize according to the requirements of their experiment
- Experimenters can install software/packages in the container and configuring those as needed*

- We also provide pre-configured containers for OAI 5G
  - OAI 5G CN container: oai-5gcn
  - OAI 5G RAN (gNB/NR UE) container: oai-5g-sa-ran

* For details on how to create custom Colosseum containers, please see:
  
  - https://www.youtube.com/watch?v=HmZITQ0xL1E&list=PLyPwVNte-Wvqovf58LWsfcvWLHQ-dGGQz&index=7
  - https://colosseumneu.freshdesk.com/a/solutions/articles/61000284967
OAI 5G NR SA Demo Setup - Colosseum

SRN 3

OAI NR UE HOST

LXC Container

OAI nrUE

Apps

mcchem

USRP X310

SRN 2

OAI GNB HOST

LXC Container

OAI GNB

MCHEM

(Massive Channel Emulator)

USRP X310

SRN 1

OAI 5G CN HOST

LXC Container

Docker network

IP range 192.168.70.128/26

MySQL

Nnf

N3

10.1.1.1

N4

OAI-UPF (OAI-SPGWU)

OAI-NRF

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)

N3

10.1.1.1

OAI-AMF

OAI-SMF

Nsnf

Namf

Nsmf

OAI-UPF (OAI-SPGWU)
OAI 5G Throughput Demo
OAI 5G Multiple UE Demo
OAX Continuous Integration

- Based on Eurecom’s CI framework
- Provides additional resources for OAI CI, particularly for testing SDR-based OAI NR UE
- Will be integrated with Eurecom Gitlab codebase and 5G MRs/merges will trigger OAX CI
- Currently uses Colosseum testbed as the test platform
- Extensible to include other PAWR platforms
  - Use of software LXC/Docker containers
  - Ansible toolchain automates container creation (credit: Zylinium Research)
  - Container migration enables Testing-at-a-scale (Colosseum, ARENA, POWDER, AERPAW)
OAX CI Setup

Execution Sequence
1. A new MR or code push to develop branch triggers Jenkins CI run on oai-Jenkins-mgmt.colosseum.net
2. Master job initiates code analysis/simulator build & runs
3. Worker job initiates Colosseum pipeline on server 2 (oai-ci-server-mgmt.colosseum.net)
4. Ansible toolchain builds OAI and creates LXC containers
5. Containers and config files are pushed to Colosseum
6. Ansible module initiates a Colosseum batch reservation
7. Colosseum resource manager assigns resources for the job
8. At start of Colosseum reservation, the LXC image and config files are pulled and SRNs are configured
9. Test run with OAI modules
10. After the test completes, log files are transferred to Colosseum NAS storage
11. Log files are analyzed for test success or failure using python scripts and xml definitions
12. Results of CI run are reported to Jenkins
13. Developers can view results on Jenkins portal

*Ansible and Airflow Toolchain Credit: Zylinium Research
CI: Status and Next Steps

- Completed (Steps 1 – 10 in the block diagram), including
  - Triggering of Jenkins jobs from Gitlab
  - Building OAI LXC container (Ansible)
  - Configuring test based on XML scenario file
  - Running the tests
  - Collecting logs (Airflow)

- In progress
  - Parsing of log files and generating HTML reports

- To be done
  - Add end-to-end SA mode test including OAI core network
  - Integration with official Eurecom Gitlab repo
Helpful Links

• Colosseum website:
  https://www.northeastern.edu/colosseum/

• Colosseum introductory webinar:
  https://colosseumneu.freshdesk.com/a/solutions/articles/61000284565

• Deploying OAI 5G Standalone Network:
  https://openairx-labs.northeastern.edu/deploying-oai-in-5g-standalone-mode/

• Deploying OAI 5G SA on Colosseum:
  https://openairx-labs.northeastern.edu/deploying-oai-5g-sa-mode-on-colosseum/
Thank You!