OAI CI Framework @ OpenAirX-Labs

Presenter:
Shweta Shrivastava
s.shrivastava@northeastern.edu

PIs:
Abhimanyu Gosain, Tommaso Melodia
agosain@coe.neu.edu, t.melodia@northeastern.edu

June 24, 2021
OpenAirX-Labs (OAX)

• A lab ecosystem incubated within Institute for Wireless Internet of Things @ Northeastern University
• Part of the Platforms for Advanced Wireless Research (PAWR) program
• Funded by NSF and the U.S. Department of Defense's Office of the Under Secretary of Defense for Research and Engineering [DOD OUSD (R&E)]
• Supported by OSA board as the official North American Designate Affiliate Development Partner
Memberships and Partners

OAX is an active participant in the following open-source ecosystems

OAX has several industry and academic partners

Consortia
- Qualcomm
- Interdigital
- Facebook

Industry
- Zylinium
- Radisys
- Omniprohys
- Xilinx
- National Instruments

Academia
- Eurecom
- NC State University
- Mississipi State University
- Rutgers University
OpenAirX-Labs Charter

• North American home for development, testing and integration of OpenAirInterface 5G SA software stack.

• Provide a benchmark, end-to-end 5G Standalone reference architecture to promote innovation

• Central Point of Contact for testing and evaluation status across member lab partners.
  • Provide development and test status for SA features

• Continuous Integration (CI) system to accelerate the development and testing of the software stack.
  • Promote stable, compliant and performant stack
**Test Platforms**

**Colosseum**
- Colosseum testbed
  - Large-scale wireless emulator
  - 128 computational nodes
    - Intel Xeon E5-2600 CPUs
  - 128 USRP X310 SDRs
  - Massive channel emulator
  - Octoclocks

**Arena**
- Arena testbed
  - Open access wireless testing platform
  - 12 computational servers
    - Intel Xeon E2186G CPUs
  - 24 SDRs (16 USRP N210 + 8 USRP X310)
  - 64 antenna grid.
  - 4 NI OctoClocks

---

**Diagram**
- SRN 1
  - Ubuntu Container
  - OAI CN(docker)
  - OAI gNB
  - SDR (Ettus USRP X310)
- SRN 2
  - Ubuntu Container
  - OAI UE
  - SDR (Ettus USRP X310)
- Server 1
  - Ubuntu Container
  - OAI CN(docker)
  - OAI gNB
  - SDR 1
- Server 2
  - Ubuntu Container
  - OAI UE
  - USRP X310/ N210
- Antenna x 2
- Antenna x 2
Test Platforms

AERPAW Testbench @ NCSU

Server 1
- OAI CN (Docker)
- OAI gNB
- USRP X310

Server 2
- OAI CN (Docker)
- OAI gNB
- USRP X310

SDR 1
- Propsim Channel Emulator

SDR 2
- 10G Eth connection

USB connection

Queue UE

AERPAW Testbench @ MSU

Server 1
- Intel i9-9900
- OAI CN (docker)
- OAI gNB
- USRP B205mini-i
- Antenna x 2

Server 2
- NUC Intel 10710U
- OAI UE
- USRP B205mini-i
- Antenna x 2

PAWR-AERPAW Team at NCSU:
Prof. Rudra Dutta, Prof. Ismail Guvenc, Prof. Mihail L. Sichitiu, Ozgur Ozdemir, Udita Bhattacharjee

PAWR-AERPAW Team at MSU:
Prof. Vuk Marojevic, Keith Powell
OAX Continuous Integration

- Mirrors Eurecom’s CI framework
- 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 Framework

**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. At start of Colosseum reservation, the LXC image and config files are pulled and SRNs are configured.
8. After the test completes, log files are transferred to Colosseum NAS storage.
9. Log files are analyzed for test success or failure using Python scripts and xml definitions.
10. Results of CI run are reported to Jenkins.
11. Developers can view results on Jenkins portal.

*Ansible and Airflow Toolchain Credit: Zylinium Research*
Thank You!

Please see https://openairx-labs.northeastern.edu/ for more details!