ANTLAB
ADVANCED NETWORK TECHNOLOGIES LAB

POLITECNICO
MILANO 1863

O-RAN Projects for MRN

Eugenio Moro, MRN 2024

Projects overview

- Goal -> Implement an O-RAN xApp and test it on real hardware
- What you'll learn
  - Software Defined Radio
  - Software defined gNB
  - O-RAN, concepts and architecture
  - Protocol design
  - 5G software development
  - RAN data collection/control through xApps
  - Manage over-the-air radio experiments and hardware equipment
- Prerequisites
  - Basic coding skills in C/C++ or Python
  - Basic terminal knowledge
- 4 Milestones – You can stop and cash-in partial points
- You'll be 1/3 through your thesis (at the final milestone)

ANTLAB

Milestones

	ETA	Points	Goal	You'll learn how to…
M1	1 Week	1	Reproduce xApp setup with gNBemu	• Manage docker containers • Basic git
M2	2 Weeks	2	Implement and test E2SM protocol with your xApp and gNBemu	• Define protocols • Extend gNBemu to test protocols • Write a basic xApp
M3	2 Weeks	3	Test with real gNB and rfsimulator	• Work with a software stack • Modify OAI's codebase
M4	???	Option 2	Improve your xApp Experiments with SDRs	• Manage experiments w/ hardware

Misc

- Teams of 3
- Milestones must be sequentially cleared
- At milestone completion, upload the deliverables on webeep and send me an email
- A milestone is cleared if pass conditions are met. If not, you can retry
- You can start the next (or 1st) milestone at any time
- You will get the points of the last milestone completed before exam verbalization
- You'll receive a form where you can define your team and choose your project

Option 1 and Option 2

- Option 1 – full written exam
  - All milestones from 1 to 3
  - Your final project mark will be the sum of all the milestones you complete
- Option 2 – reduced written exam
  - Option 2 requires you to extend the xApp you have at the end of M3
  - You have time until the end of M3 to decide if you want to continue with Option 2 or stop at Option 1
  - If, after you complete M3, you decide to go for it, then drop me an email and we will meet to discuss together how to continue
  - There will be no milestones for Option 2, but we will define some passing conditions and your final project evaluation will be based on how many of those are met

Milestone 1

- Reproduce the demo seen during class
- Suggested steps:
  1. Deploy the xApp development environment (following the instructions in ***xApp development basics***)
  2. Test the environment by running the base xApps
- Deliverables:
  1. 60s video where you start both components and demonstrate that they properly work
- Duration: 1 week
- Points: 1

Milestone 2

- Implement and test the E2 SM of your xApp
- Suggested steps:
  1. Choose a project
  2. Analyze the data collection and control requirements of the xApp
  3. Extend the provided base protobuf SM
  4. Extend the xApp and the gNB emulator (exchanging random data is fine)
  5. Test it using the setup of Milestone 1
- Deliverables:
  1. 2 pages (max) written report motivating your protocol design choices and describing your protobuf definition
  2. 60s video demo
  3. Code
- Duration: 2 weeks
- Points: 2

Milestone 3

- Test the xApp with a real gNB and simulated RF channel
- Suggested steps:
  1. Move the new message handler functions and protocols in the gNB's codebase
  2. Setup a base 5G deployment: OAI gNB – simulated RF channel – OAI UE
  3. Finalize the xApp, if needed
  4. Test the xApp in reasonable scenarios
- Deliverables
  1. 2 pages (max) written report describing how you connected the handler functions with the gNB code, the test scenario and your conclusions + run instructions
  2. 60s video demo
  3. Code
- Duration: 3 weeks
- Points: 3

Option 2

- Come to the lab to play with real equipment
- Deploy and test with over-the-air transmissions and real UEs
- Enrich you xApp and conduct thesis-worth experimental research

If you really liked it, you could decide to make a thesis out of it:

- You can keep your team (thesis can be co-authored)
- You can manage the effort and work on your thesis even if it is not your last semester

Project 1

- Implement an xApp that collects PHY/MAC metrics:
  - Per-UE RSRP
  - Per-UE BER (uplink and downlink)
  - Per-UE MCS (uplink and downlink)
  - Cell load (i.e. allocated PRBs)
- All values timestamped and saved in a CSV file
- 500ms data collection loop
- Pass conditions:
  - M2: gNB emu sends random data, xApp is complete, test with several UEs
  - M3: OAI gNB sends actual data, test with 1 softUE

Project 2

- Implement an xApp that monitors each UE's BER and forces a low MCS if it is below a threshold
- Normal operation if BER is higher than threshold in next monitoring interval
- The target MCS and threshold BER are defined as parameters in the xApp
- Monitoring interval of 500ms
- Limited to downlink
- Pass conditions:
  - M2: gNB emu sends BER (either higher or lower than thr.), xApp reacts accordingly, test with multiple UEs
  - M3: OAI gNB sends actual BER, received MCS is applied, test with 1 softUE

Project 3

- Implement an xApp that limits the resources that can be allocated to specific UEs
- More specifically, the xApp sets a maximum number of allocable PRBs to any UE of choice
- UEs are chosen interactively in the xApp based on the RNTI
- The gNB scheduler remains otherwise untouched
- Resource limit defined as parameter in xApp
- No monitoring
- Limited to downlink
- Pass conditions:
  - M2: xApp send random but sensible data, gNB emu stores data in memory
  - M3: test with 1 UE, allocate 100%, 50%, 10% PRBs from xApp

ADVANCED
NETWORK
TECHNOLOGIES
LAB