

ARNO Testbed Description

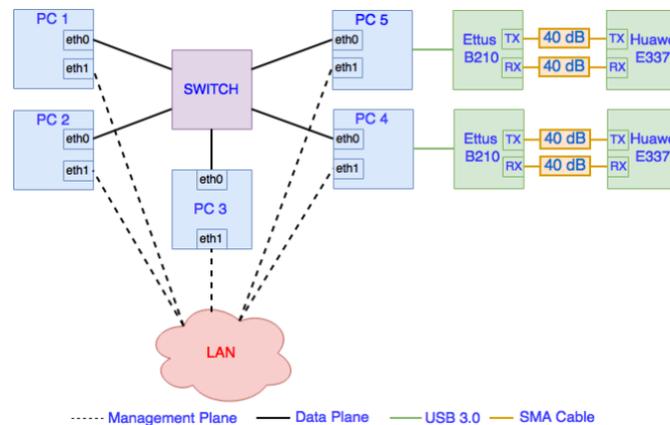


Figure 1 ARNO Testbed

Figure 1 shows a block diagram of the ARNO testbed. PC 1 is a mini-pc (Up-board First Generation) featuring an Intel Atom x5-Z8350 Quad Core Processor. It hosts Ubuntu 14.04 LTS with a 4.7 kernel. The kernel hosted in PC 1 is directly precompiled by OpenAirInterface (OAI) team for including the GPRS Tunnelling Protocol (GTP) kernel module.

PC 2 and PC 3 are Dell T410 PowerEdge desktop servers with Intel Xeon E5620 and hosting Ubuntu 14.04 with 3.19 low-latency kernel.

PC 4 is a Mini-ITX featuring an Intel i7 7700 Quad Core @ 4.0 GHz and hosting Ubuntu 14.04, 3.19 low-latency kernel. PC4 is connected to an Ettus B210 through an USB 3.0 link.

PC 5 is a desktop computer with an Intel i7 4790 @ 3.60 GHz and hosting Ubuntu 14.04 with 3.19 low-latency kernel. PC 5 is connected to an Ettus B210 through an USB 3.0 link.

The Ettus B210 USRPs are a fully integrated, single-board, Universal Software Radio Peripheral (USRP) platforms and act as radio front-end performing Digital to Analog and Analog to Digital Conversion (DAC/ADC), Digital Up and Down Conversion (DUC/DAC), low pass filtering, and amplification.

Two Huawei E3372 LTE dongles are utilized as User Equipments (UEs). They support LTE category 4 and frequency-division duplexing (FDD) communication in the following bands: 900 MHz, 1800 MHz, 2100 MHz and 2600 MHz. They support a maximum rate of 150Mb/s in downlink and 50Mb/s uplink with a signal bandwidth of 20MHz. The dongles are connected to the Ettus B210 USRPs through SMA cables with 40 dB of attenuation.

All the PCs composing ARNO, have a management plane interface and also a data plane interface. The management and data planes belong to two different networks. The former one (i.e., 10.30.x.x) belongs to the lab LAN to assure the continuous reachability of the machines. The latter one is handled by a Cisco Catalyst 2960G switch (indicated as SWITCH in Figure 1) and it is used exclusively as ARNO-5G testbed data plane for the communication between the LTE network entities. All the PCs are connected to the management plane through 100 Mb/s Ethernet links while they are all connected to the SWITCH by a 1 Gigabit Ethernet link.

The SWITCH is configured to have different subnets for the backhaul link and for the fronthaul link. Therefore the EPC and the CU interfaces belonging to the backhaul link are configured in the 10.10.20.x subnet while the fronthaul link and the interfaces of the CU and DU are configured in the 10.10.30.x subnet.

The utilised mobile network software is the OpenAirInterface (OAI) by Eurecom. The core network utilizes openair-cn, an implementation of the Evolved Packet Core (EPC) network. It implements the EPC 3GPP specifications, that means it contains the implementation of the following network elements: the Serving Gateway (S-GW), the PDN Gateway (PDN GW), the Mobile Management Entity (MME) and the Home Subscriber Server (HSS). For the RAN, openairinterface5g provides a standard-compliant implementation of Release 13 LTE and later for the evolved NodeB (eNB) and User Equipment (UE). It also provides an implementation of some eNB functional splits and therefore a possible deployment of Cloud/Virtualised RAN (C/V-RAN).

The functional splits implemented by the OAI platform are the IF5 and IF4.5 also known as Option 8 and Option 7-1 in the 3GPP terminology.

How to Access ARNO Testbed

EVER-WISHFUL testbed accept users from only one trusted central authority (iMinds) identity providers. Here, experiments can configure experiments interconnecting resources from multiple testbeds at the same time, reserve and access them via iMinds tools such as jFed. In order to use this tool, users need to provide a user certificate and a password to login. Experimenters can get an account and download their certificates from iMinds authority provider. Follow the manual to create an account and get the credential: Get an [Account](#). Once successfully logged in, experimenters can set up their experiments by choosing which types of resources and from which testbeds, configure those resources (operative system, software to be installed, network configurations, measurement options, etc), launch the experiments and access the resources, just by double clicking on their icons.

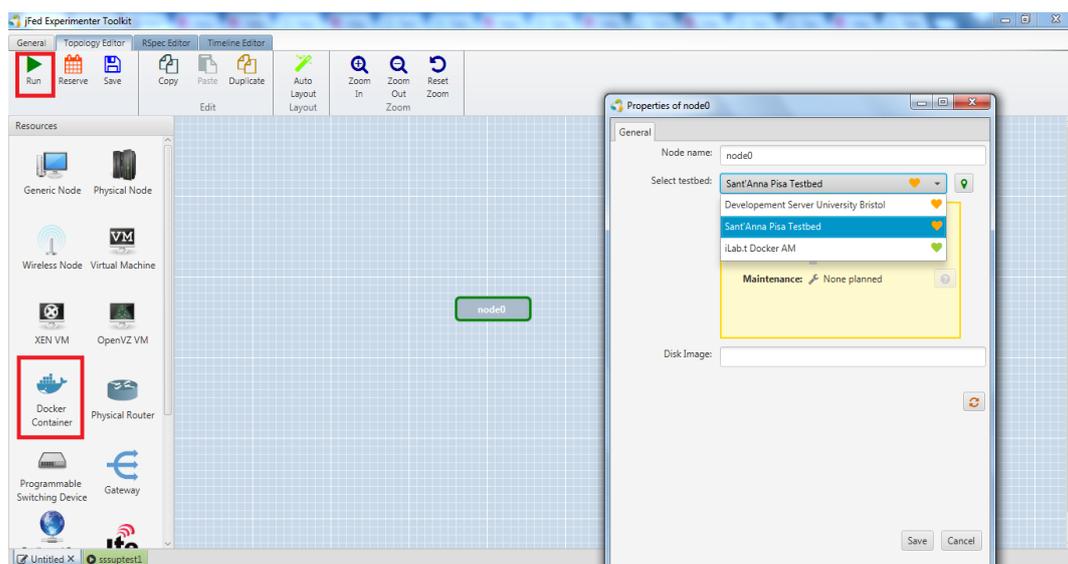


Figure 2 Experimental setup in jFed Experimenter Toolkit

Figure 2 shows Docker Container icon (red rectangle), by double clicking on it, a popup window opens where experimenter can select EVER-WISHFUL testbed, namely “Sant’Anna Pisa testbed”. After that, experimenter can provide their slice name by clicking on run (other red rectangle). This process creates a docker container in EVER-WISHFUL testbed. By double clicking on “node0”, experimenters can enter into each OAI component of EVER-WISHFUL testbed through ssh based on the specific container Rspec (i.e., Dockerfile). Note that a sample Dockerfile will be provided by EVER-WISHFUL team to access a OAI component (e.g., eNB). EVER-WISHFUL testbed provides different ways to access the testbed based on the experimenter knowledge on LTE and Radio Access Network (RAN).

Figure 3 shows an experimenter with basic OAI knowledge can perform experiment who has jfed credentials. The experiment scenario is described as follows.

The controller is installed in a container (as described above) within a Docker and is also reachable via ssh. C1, C2,..., Ci are generic containers deployed automatically once the experiment starts. In C1, C2,..., Ci are also automatically installed the Wishful framework and OpenAirInterface5G emulation platform. For the experimenters point of view C1, C2,..., Ci are like black boxes. The experimenter using the controller can configure and run the LTE Network nodes (EPCs, eNBs, RRUs, RCCs and Ues).

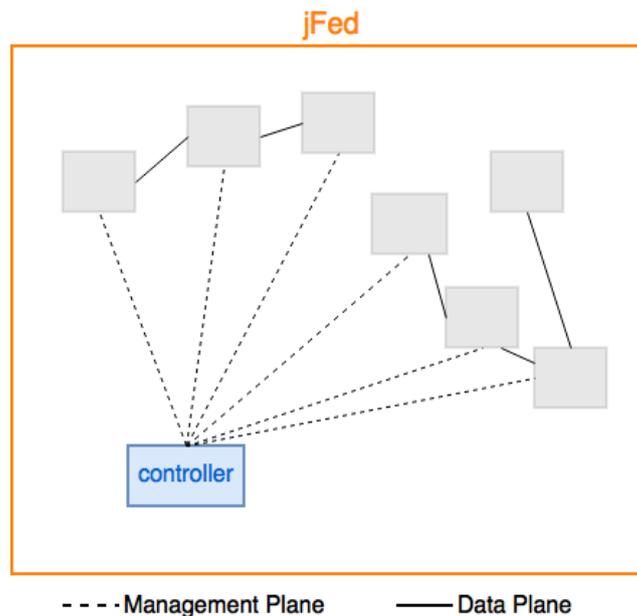


Figure 3 Accessing ARNO through jFed

Experiment Example

To run the above described experiment, the experimenter could follow the steps described below:

1. Write its own controller_experiment.py;
2. Connect to the Aggregate Manager machine through jfed or ssh;
3. Copy the controller_experiment.py into the Aggregate Manager;
4. Run controller_experiment.py