Table of Contents

- Introduction and Our Goal
- Basic Testbed Setup
- The SimulaMet EPC VNF
- Managing Builds
- Live Demo
Setting Up a 4G/5G Testbed

**Hardware:**
- User Equipment (modems, smartphones, etc.)
- Programmable sim cards
- Software-Defined Radio boards

**For the rest (eNodeBs, EPC):**
- OpenAirInterface Open Source software
- Running on regular Linux PCs
- But: difficult to install and maintain!
Our Goal: An OpenAirInterface VNF

- Main purpose: testbed setups for research and development
- OAI EPC as VNF
  - Easy to use, EPC should (hopefully) work “out of the box”
  - Build of OAI software inside VMs, according to specified Git repositories and commits
    ⇨ get exactly the desired installation (e.g. using Mosaic5G branches)
- NSs using the VNF and possibly other VNFs
  - Example 1: add Mobile Edge Computing services to EPC
  - Example 2: use FlexRAN for network slicing
  - ...
Basic Testbed Setup

OpenSource MANO

Juju containers
- MME
- HSS
- SPGW-C
- SPGW-U

UEs eNodeBs

The Internet

PDN

SimulaMet EPC NS

OpenStack Cluster

S1-C
S1-U
SGi
S11
S6a
SXab
What is needed for the VNF?

- Base VDU image
- The VNF itself
- Juju Charms to configure the components
- Management of the build process
Base VDU Image

• VDU image goals:
  - Full-featured base VDU image, including development and debug tools
  - Different versions of Ubuntu LTS (Xenial, Bionic, Focal)
  - Up-to-date (i.e. all updates installed)

• Packer scripting:
  - Fully automatic installation using Packer
  - Preseeding (Ubuntu < 20.04); Subiquity (Ubuntu ≥ 20.04)
    • Additional PPA, keyboard layout setup, EFI boot, etc.
    • All updates installed
    • => Fresh, state-of-the art installation (avoids issues with “old” installations)
  - Details: https://github.com/simula/nornet-vmimage-builder-scripts
The SimulaMet EPC VNF

HSS: Home Subscriber Server
MME: Mobile Management Entity
SPGW-C: Control Plane of the Packet Data Network Gateway
SPGW-U: User Plane of the Packet Data Network Gateway
VNF Parameters Example

- # ====== HSS ===============================================
  hss_git_repository: 'https://github.com/simula/openairinterface-openair-hss.git'
  hss_git_commit: 'dreibh/cassandra-build-fix-22oct2020'
  hss_S6a_address: '172.16.6.129'
  network_realm: 'simula.nornet'
  network_k: '449C4B91AEACD0ACE182CF3A5A72BFA1'
  network_op: '1006020F0A478BF6B699F15C062E42B3'
  network_imsi_first: '242881234500000'
  network_msisdn_first: '24288880000000'
  network_users: '1024'

- # ====== MME ===============================================
  mme_git_repository: 'https://github.com/simula/mosaic5g-openair-cn.git'
  mme_git_commit: 'dreibh/mosaic5g'
  mme_S1C_ipv4_interface: '192.168.247.102/24'
  mme_S1C_ipv4_gateway: '0.0.0.0'
  mme_S1C_ipv6_interface: ''
  mme_S1C_ipv6_gateway: ''
  mme_S11_ipv4_interface: '172.16.1.102/24'
  mme_S6a_address: '172.16.6.2'
  network_mcc: '242'
  network_mnc: '88'
  ...
Managing VNFD/NSD Builds

- Multiple manual steps to generate and deploy VNFs and NSs
  - Strictly verify all YAML files with yamllint (very useful, to avoid problems and surprises!)
  - Copy Charm files to VNFDs and build Charms (charm build …)
    - Including our library “VDUHelper”
  - Verify descriptor(s) and generate VNFD package(s)
    (validate_descriptor.py, generate_descriptor_pkg.sh)
  - Verify descriptor(s) and generate NSD package(s)
- Initial approach: write a Makefile
- Better approach:
  - Git for source management ⇒ information about all relevant source files
  - Let CMake write Makefiles and take care of dependencies!
Live Demo
Sources

- Get the sources here: https://github.com/simula/5gvinni-oai-ns
  - Open Source, GPL-licensed
  - README: how to set up a testbed
  - vmimage-builder-scripts/ (submodule): VDU preseeded image build script
  - juju/: The Juju Charms used by the VNF
  - SimulaMet-OAI-EPC_vnfd/: VNF descriptor
  - SimulaMet-OAI-EPC_nsd/: NS descriptor for simple example