

4G Core Network Training

BUPT – Beijing 2018-June

GOALS

- Show what Openairinterface offers in its 4G Core Network. What is actually the picture. How we run it.

Plan

- What is actually inside openair-cn
 - Core network entities released in github openair-cn repository.
 - HSS overview
 - MME Conformance
 - SPGW Conformance
 - Messages/interfaces
- Deployment used at Eurecom
 - Generic VM creation
 - Building NFs

3GPP Basic Picture of a 4G Core network.

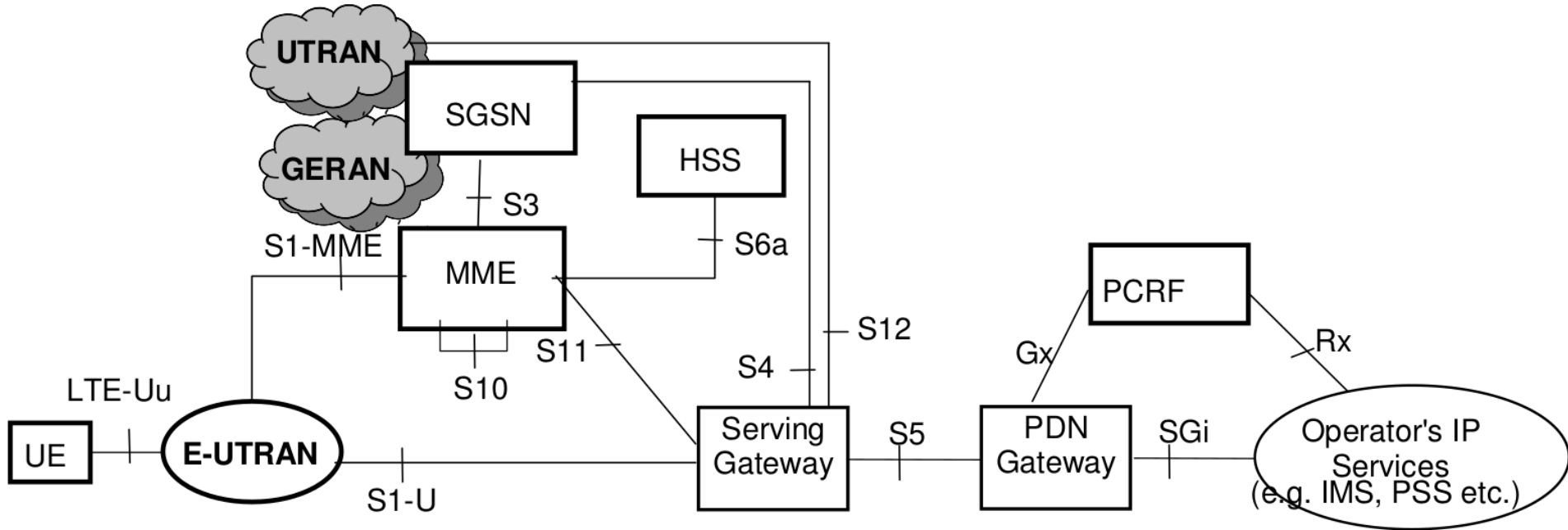
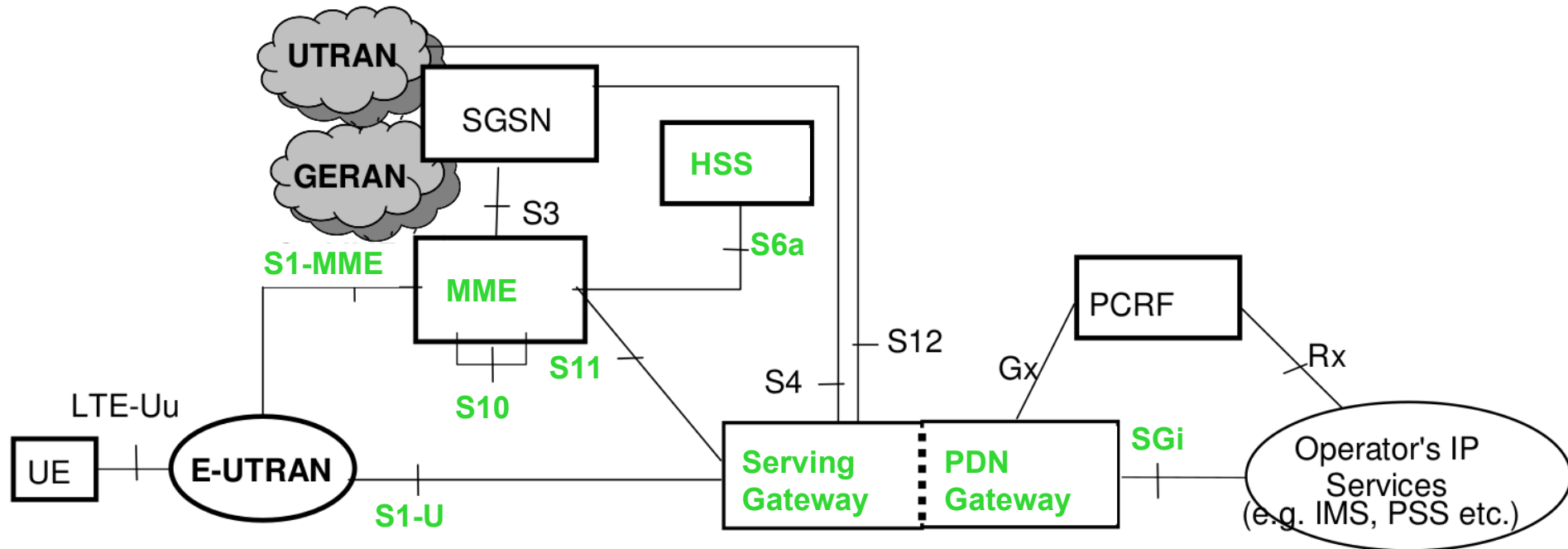


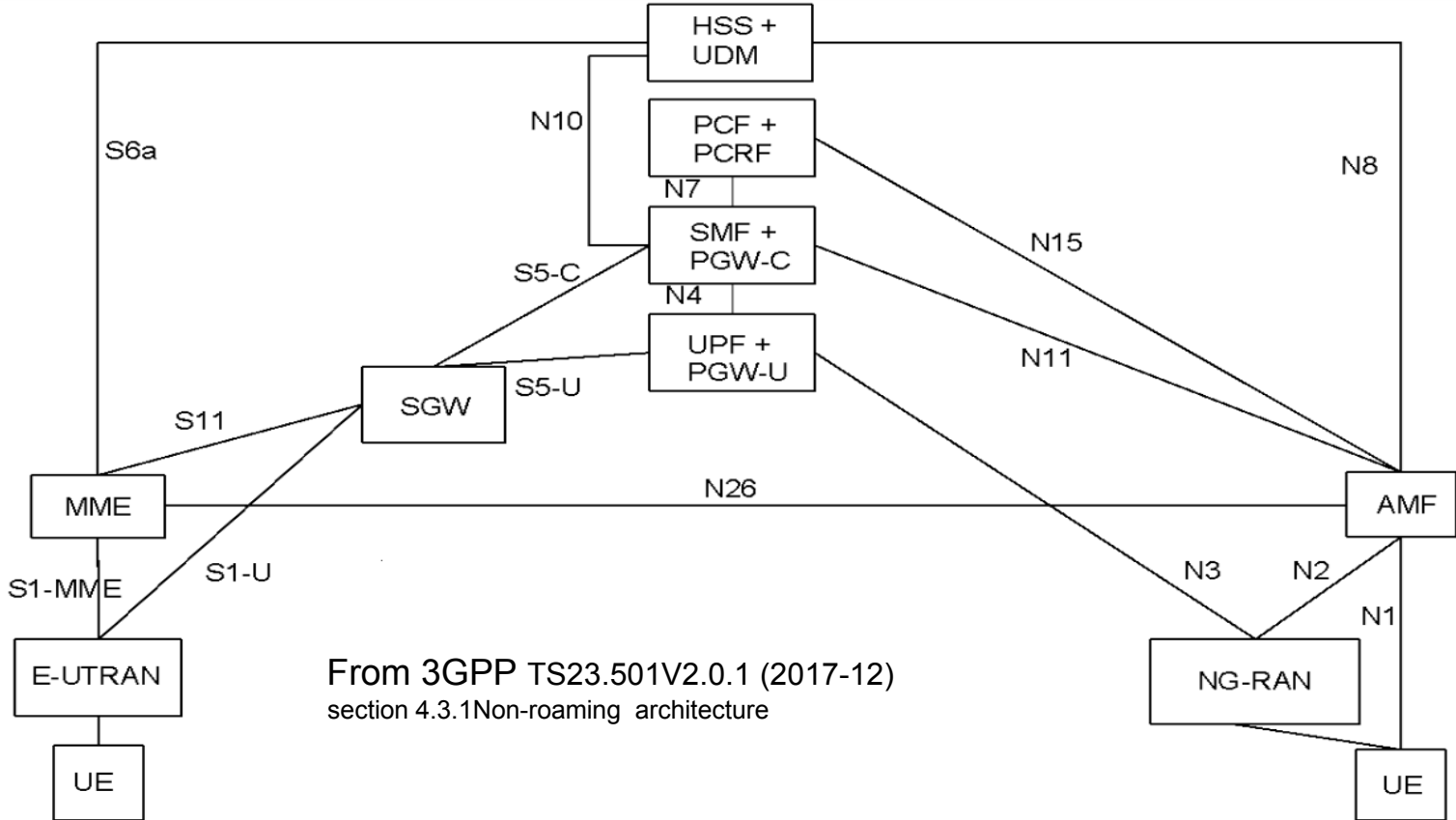
Figure 4.2.1-1: Non-roaming architecture for 3GPP accesses

4G Core network entities released in openair-cn repo.



Implemented

Perspective: 4G Core Network inter-working with 5G Core Network



From 3GPP TS23.501V2.0.1 (2017-12)
section 4.3.1 Non-roaming architecture

HSS

- Contributed back by Sprint (USA), started from OAI HSS rel 10.
 - Also contributed tiny/skeleton TDF (Traffic detection), CTF (Charging), PCEF, PCRF.
- Use Cassandra database (1 Node or fully distributed cluster of data centers)
- Interfaces
 - s6a(MME)/s6d(SGSN), s6c (GMSC), s6t (Service Capabilities Exposure Function SCEF) implemented.
 - Fully written in C++11

HSS

- External libraries
 - **FreeDiameter** (modified for overlapping AVPs between dictionaries).
 - **C-ares** (an asynchronous resolver library. It is intended for applications which need to perform DNS queries without blocking, or need to perform multiple DNS queries in parallel).
 - **cpp-driver** (Cassandra db driver).
 - **pistache** (HTTP and REST framework for C++ written in pure-C++11).
 - **rapid json** (A fast JSON parser/generator for C++ with both SAX/DOM style API).
 - **spdlog** (Very fast, header only, C++ logging library).

MME

- Tech details:
 - Written in C.
 - Main Libraries
 - Asn1c.
 - Openssl.
 - NwGTPv2-c (BSD 2-clause) (GTPv2-c library written in C).
 - freeDiameter (not the same as HSS Rel14 actually).
 - bstring (better string library).
 - liblfd (lock free library: queues).

MME

- MME got a boost in its Mobility management features thanks to the contribution of Blackned (Dincer Beken).
 - Implementation of X2HO (also contributed by B-COM), S1HO, S10HO, Paging in MME.
 - Multiple PDN almost finished (will be released very soon).
 - Dedicated bearers should follow.
 - Tested with NG4T Ran emulator, Blackned HSS, third party S/P-GW.

OAI MME Conformance interfaces

interfaces	1	S1-MME interface	supported	Still rel 10 should be upgraded to rel 14 soon
	2	S11 interface	supported	Rel 12 (Blackned)
	3	S6a interface	supported	Still rel 10 should be upgraded to rel 14 soon (Sprint contribution on HSS)
	4	S10 interface	supported	Rel 12 (Blackned)

OAI MME conformance functions 1/3

1	NAS signalling	23.401 V10.13.0 4.4.2	supported	
2	NAS signalling security	23.401 V10.13.0 4.4.2	supported	
3	Inter CN node signalling for mobility between 3GPP access networks (terminating S3)	23.401 V10.13.0 4.4.2	not supported	Intra LTE HO only
4	UE Reachability in ECM-IDLE state (including control, execution of paging retransmission and optionally Paging Policy Differentiation)	23.401 V10.13.0 4.4.2	supported	We are working on this
5	Tracking Area list management	3GPP TS 23.401 V10.13.0 4.4.2	supported	

OAI MME conformance functions 2/3

6	Mapping from UE location (e.g. TAI) to time zone, and signalling a UE time zone change associated with mobility	23.401 V10.13.0 4.4.2	not supported	
7	PDN GW and Serving GW selection	23.401 V10.13.0 4.4.2	supported	spgw selection & neighboring MME selection via WRR
8	MME selection for handovers with MME change	23.401 V10.13.0 4.4.2	supported	S1 (inter (S10) and intra MME S1AP handover), X2 HO supported
9	SGSN selection for handovers to 2G or 3G 3GPP access networks	23.401 V10.13.0 4.4.2	not supported	
10	Roaming (S6a towards home HSS)	23.401 V10.13.0 4.4.2	not supported	
11	Authentication	23.401 V10.13.0 4.4.2	supported	
12	Authorization	23.401 V10.13.0 4.4.2	supported	

OAI MME conformance functions 3/3

13	Bearer management functions including dedicated bearer establishment	3GPP TS 23.401 V10.13.0 4.4.2	not supported	
14	Lawful Interception of signalling traffic	3GPP TS 23.401 V10.13.0 4.4.2	not supported	
15	Warning message transfer function (including selection of appropriate eNodeB)	3GPP TS 23.401 V10.13.0 4.4.2	not supported	
16	UE Reachability procedures	3GPP TS 23.401 V10.13.0 4.4.2	not supported	UE Reachability Notification Request procedure, UE Activity Notification procedure
17	Support Relaying function (RN Attach/Detach)	3GPP TS 23.401 V10.13.0 4.4.2	not supported	

SPGW

- Tech details:
 - Written in C.
 - Main Libraries
 - NwGTPv2-c.
 - Libfluid or libgtpnl.
 - libevent.
 - bstring (better string library).
 - liblfd (lock free library: queues).
 - Use OVS or GTP in kernel only.
 - Strong dependency on kernel 4.9 actually

OAI SGW conformance interfaces

1	GTP-based S5/S8 interface	not supported	missing split between SGW and PGW
2	S1-U interface	supported	
3	S11 interface	supported	

OAI SGW conformance functions 1/2

1	the local Mobility Anchor point for inter-eNodeB handover (except when user data is transported using the Control Plane ClOT EPS Optimisation)	3GPP TS 23.401 V10.13.0 4.4.3.2	supported	X2HO supported
2	sending of one or more "end marker" to the source eNodeB, source SGSN or source RNC immediately after the Serving GW switches the path during inter-eNodeB and inter-RAT handover, especially to assist the reordering function in eNodeB	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	
3	Mobility anchoring for inter-3GPP mobility (terminating S4 and relaying the traffic between 2G/3G system and PDN GW)	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	
4	ECM-IDLE mode downlink packet buffering and initiation of network triggered service request procedure and optionally Paging Policy Differentiation	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	buffering is not supported for paging in idle mode
5	Lawful Interception	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	

OAI SGW conformance functions 2/2

6	Packet routing and forwarding	3GPP TS 23.401 V10.13.0 4.4.3.2	supported	Nat is no more supported inside SPGW if OVS used.
7	Transport level packet marking in the uplink and the downlink, e.g. setting the DiffServ Code Point, based on the QCI, and optionally the ARP priority level, of the associated EPS bearer	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	could be supported quickly
8	Accounting for inter-operator charging. For GTP-based S5/S8, the Serving GW generates accounting data per UE and bearer	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	
9	Interfacing OFCS according to charging principles and through reference points specified in TS 32.240	3GPP TS 23.401 V10.13.0 4.4.3.2	not supported	

GTPv2 Messages implemented

PATH MANAGEMENT MESSAGES

Messages	Status (S10/S11)	Comments
Echo Request/Response	Not implemented	Required (not with OAI eNB)
Version not supported Indication	Not implemented	

GTPv2 Messages **implemented** 1/4

TUNNEL MANAGEMENT MESSAGES

Messages	Interface	Comments
Create Session Request/Response	S11	
Create Bearer Request/Response	S11	Not used now (dedicated bearers)
Modify Bearer Request/Response	S11	
Delete Session Request/Response	S11	
Delete Bearer Request/Response	S11	TODO
Downlink Data Notification/Acknowledge	S11	network triggered service request procedure

GTPv2 Messages **implemented** 2/4

TUNNEL MANAGEMENT MESSAGES

Messages	Interface	Comments
Delete Indirect Data Forwarding Tunnel Request/ Response	S11	S1-based handover
Modify Bearer Command and Failure Indication	S11	HSS Initiated Subscribed QoS Modification procedure
Update Bearer Request/Response	S11	Many procedures
Delete Bearer Command and Failure Indication	S11	eNodeB requested bearer release or MME-Initiated Dedicated Bearer Deactivation procedure.
Create Indirect Data Forwarding Tunnel Request/Response	S11	Handover procedures or TAU/RAU procedure with Serving GW change and data forwarding

GTPv2 Messages **implemented** 3/4

TUNNEL MANAGEMENT MESSAGES

Messages	Interface	Comments
Release Access Bearers Request/Response	S11	S1 release procedure and eNodeB initiated Connection Suspend procedure
Stop Paging Indication	S11	network triggered service request procedure. TODO.
Modify Access Bearers Request/Response	S11	If both the SGW and the MME support the MABR feature
Remote UE Report Notification/Acknowledge		Proximity Services

GTPv2 Messages implemented 4/4

MOBILITY MANAGEMENT MESSAGES

Messages	Interface	Comments
Forward Relocation Request/Response/Complete Notification/Complete Acknowledge	S10	S1-based handover relocation procedure
Context Request/Response/Acknowledge	S10	TAU/RAU procedure
Forward Access Context Notification/Acknowledge	S10	sent from the Old MME to the New MME over the S10 interface to forward the RNC/eNodeB contexts to the target system.
Relocation Cancel Request/Response	S10	S1 Based handover Cancel procedure
Configuration Transfer Tunnel	S10	transfer information from an eNodeB to another eNodeB in unacknowledged mode

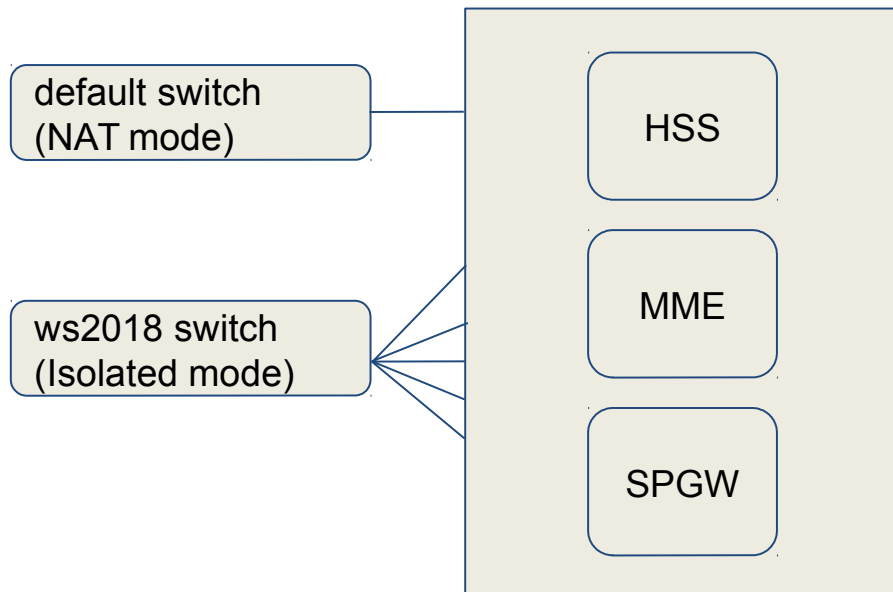
GTPv2 Messages **implemented** 3/4

TUNNEL MANAGEMENT MESSAGES

Messages	Interface	Comments
Release Access Bearers Request/Response	S11	S1 release procedure and eNodeB initiated Connection Suspend procedure
Stop Paging Indication	S11	network triggered service request procedure. TODO.
Modify Access Bearers Request/Response	S11	If both the SGW and the MME support the MABR feature
Remote UE Report Notification/Acknowledge		Proximity Services

END OF FEATURES

CN generic deployment - libvirt



Ubuntu cloud image xenial (16.04)
or ubuntu image built from ISO
32 GB disk
2 cores
4 GB RAM
Customized kernel 4.9.x (1..108)
(SPGW/GTP constraints)
8 physical network interfaces

Note on required disk size:
Kernel 4.9 compilation and install (15GB) needs at
least 21GB disk.
Installation of HSS, MME, SPGW need 1GB disk
space.

CN deployment - Networking plan

NW ref	VM Device	Subnet	Connection to host
Management	ens3	DHCP, NAT192.168.122.0/24	“default” switch in NAT mode
S11	ens4	192.168.17.0/24	“ws2018” switch in isolated mode
<i>spare</i>	<i>ens5</i>	<i>192.168.27.0/24 (unused)</i>	“ws2018” switch in isolated mode
S1-MME	ens6	192.168.37.0/24	“ws2018” switch in isolated mode
S1-U	ens7	192.168.47.0/24	“ws2018” switch in isolated mode
S5	<i>ens8</i>	<i>192.168.57.0/24 (unused)</i>	“ws2018” switch in isolated mode
S6a	ens9	192.168.67.0/24	“ws2018” switch in isolated mode
SGi	ens10	192.168.77.0/24	“ws2018” switch in isolated mode

VM networking - default network

file default.xml content:

```
<network>
  <name>default</name>
  <uuid>c605dfba-fe70-497f-87f3-f1b90e58e752</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535'/>
    </nat>
  </forward>
  <bridge name='virbr0' stp='on' delay='0'/>
  <mac address='52:54:00:5b:cc:7b'/>
  <dns>
    <forwarder addr='192.168.106.12'/>
    <forwarder addr='192.168.106.10'/>
  </dns>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254'/>
    </dhcp>
  </ip>
</network>
```

```
ubuntu@ubuntu# sudo virsh net-create default.xml
```

```
ubuntu@ubuntu# sudo virsh net-autostart default
```

```
ubuntu@ubuntu# sudo virsh net-define ws2018.xml # make it persistent
```

VM networking - 4G network

file ws2018.xml content:

```
<network>
  <name>ws2018</name>
  <uuid>94a99b8d-5737-4357-8e79-b1b21e9e56f2</uuid>
  <bridge name='virbr2' stp='on' delay='0'/>
  <mac address='52:54:00:bd:e0:83'/>
  <domain name='ws2018'/>
  <ip address='192.168.100.1' netmask='255.255.255.0'>
  </ip>
</network>
```

```
ubuntu@ubuntu# sudo virsh net-create ws2018.xml
```

```
ubuntu@ubuntu# sudo virsh net-autostart ws2018
```

```
ubuntu@ubuntu# sudo virsh net-define ws2018.xml # make it persistent
```

VM networking

```
ubuntu@ubuntu# virsh net-list --all
```

Name	State	Autostart	Persistent
default	active	yes	yes
ws2018	active	yes	yes

```
ubuntu@ubuntu# ifconfig
```

```
...
```

```
virbr0Link encap:Ethernet HWaddr 00:00:00:00:00:00  
  inet addr:192.168.122.1 Bcast:192.168.122.255 Mask:255.255.255.0  
  UP BROADCAST MULTICAST MTU:1500 Metric:1
```

```
...
```


```
virbr2Link encap:Ethernet HWaddr 00:00:00:00:00:00  
  inet addr:192.168.100.1 Bcast:192.168.100.255 Mask:255.255.255.0  
  UP BROADCAST MULTICAST MTU:1500 Metric:1
```

```
...
```


VM Creation

```
ubuntu@ubuntu# uvt-kvm create ws2018 release=xenial --memory 4096 --cpu 2 --disk 32 --ssh-public-key-  
file ~/.ssh/id_rsa.pub --template $PWD/templates/ws2018.xml --meta-data $PWD/meta_data/ws2018.cfg  
--password eurecom
```

```
<domain type='kvm'>  
  <memory unit='KiB'>25165824</memory>  
  <currentMemory unit='KiB'>25165824</currentMemory>  
  <vcpu placement='static'>2</vcpu>  
  <os>  
    <type arch='x86_64' machine='pc-i440fx-xenial'>hvm</type>  
    <boot dev='hd'>  
  </os>  
  <features>  
    <acpi/>  
    <apic/>  
    <pae/>  
  </features>  
  <cpu mode='host-model'>  
    <model fallback='allow'>  
    <topology sockets='1' cores='2' threads='1'>  
  </cpu>  
  <devices>  
    <interface type='bridge'>  
      <source network='default' bridge='virbr0'>  
      <model type='virtio'>  
    </interface>  
    <interface type='network'>  
      <source network='ws2018' bridge='virbr2'>  
      <model type='virtio'>  
    </interface>  
    <interface type='network'>  
      <source network='ws2018' bridge='virbr2'>
```



```
network-interfaces: |  
  auto ens3  
  iface ens3 inet dhcp  
  auto ens4  
  iface ens4 inet static  
    address 192.168.17.221  
    netmask 255.255.255.0  
  ...  
  auto ens10  
  iface ens10 inet static  
    address 192.168.77.221  
    netmask 255.255.255.0
```



```
apt:  
  preserve_sources_list: true
```

```
primary:  
  - arches: [amd64, default]
```

```
security:  
  uri: http://security.ubuntu.com/ubuntu
```

etc.

VM Creation - other steps (not detailed)

- Retrieve its IP address on management interface (ens3)
- Create a user
- Set encrypted password to user (login 'eurecom', passwd 'eurecom')
- Set user as sudoer
- Copy authorized keys to user .ssh dir

Get openair-cn code

- Login as the sudoer

```
ubuntu@ubuntu# ssh eurecom@192.168.122.170
```

```
# Install kernel for SPGW
```

```
sudo apt install libncurses5-dev libncursesw5-dev bc binutils gcc libssl-dev make autoconf
```

```
eurecom@ws2018:~$ cd /usr/src; sudo wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.9.108.tar.xz
```

```
eurecom@ws2018:/usr/src$ sudo tar xf linux-4.9.108.tar.xz && cd linux-4.9.108/
```

```
eurecom@ws2018:/usr/src/linux-4.9.108$ sudo make olddefconfig && sudo make -j`nproc`
```

```
eurecom@ws2018:/usr/src/linux-4.9.108$ sudo make modules_install && sudo make install
```

```
eurecom@ws2018:~$ git clone https://github.com/OPENAIRINTERFACE/openair-cn.git
```

```
eurecom@ws2018:~$ cd openair-cn; git checkout develop
```

```
eurecom@ws2018:~/openair-cn$ cd scripts
```

Build OAI HSS, MME, SPGW

```
eurecom@ws2018:~/openair-cn/scripts$ ./build_hss_rel14 -i && ./build_hss_rel14 --clean
```

```
eurecom@ws2018:~/openair-cn/scripts$ ./build_mme -i && ./build_mme --clean
```

```
eurecom@ws2018:~/openair-cn/scripts$ ./build_spgw -i && ./build_spgw --clean
```

Stop here

Configure HSS - verify cassandra

Verify that Cassandra is installed and running

```
#nodetool status
```

The output should look something like:

```
eurecom@ws2018:~/openair-cn/scripts$ nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address    Load          Tokens  Owns (effective)  Host ID                               Rack
UN 127.0.0.1  51.64 KB     256     100.0%           41399cd4-c832-4d9a-9fbc-895a8301638e  rack1
```

Configure HSS - configure cassandra

Stop Cassandra and cleanup the log files before modifying the configuration

```
sudo service cassandra stop
```

```
sudo rm -rf /var/lib/cassandra/data/system/*
```

```
sudo rm -rf /var/lib/cassandra/commitlog/*
```

```
sudo rm -rf /var/lib/cassandra/data/system_traces/*
```

```
sudo rm -rf /var/lib/cassandra/saved_caches/*
```

Configure HSS - Update cassandra configuration

(c) Update the Cassandra configuration **if needed** (default configuration set 1 cassandra node listening on **localhost**)

Update `/etc/cassandra/cassandra.yaml` as indicated below. The `<Cassandra_Server_IP>` address should be the IP address of the Cassandra server that the HSS will use to connect to Cassandra. The `"..."` below indicate configuration lines between values that need to be modified.

```
...
cluster_name: "HSS Cluster"
...
seed_provider:
- class_name: org.apache.cassandra.locator.SimpleSeedProvider
- seeds: "<Cassandra_Server_IP>"
...
listen_address: <Cassandra_Server_IP>
...
rpc_address: <Cassandra_Server_IP>
...
endpoint_snitch: GossipingPropertyFileSnitch
```

or

```
endpoint_snitch: SimpleSnitch
```