OpenAirInterface 5G Overview, Installation, Usage

Florian Kaltenberger
First North America OpenAirInterface Workshop, Bell Labs Murray Hill, 25.6.2019

Unleashing the potential of open-source in the 5G arena
Overview

- Overview and Ecosystem
- Features of current master & develop branch
- Ongoing feature branches
- Hardware targets
- Installation & Usage
- Debugging tools
- 5G-NR demo
What is OpenAirInterface?

- Open-source software-based implementation of 3GPP Technologies
  - Starting at LTE (Rel 8), including features from LTE-Advanced (Rel 10/11/12), LTE-Advanced-Pro (Rel 13/14), going on to 5G Rel (15/16/…)
  - Spanning the full protocol stack of 3GPP standard
    - E-UTRAN (eNB, gNB, UE)
    - EPC (MME, S+P-GW, HSS)
  - Realtime RF and scalable emulation platforms
  - Works with many SDR platforms (ExpressMIMO2, USRP, LimeSDR, …)

- Makes it is feasible to put a fully-compliant 4G eNodeB (and soon 5G gNB) and EPC in a commodity x86-based computer (or data center)

- Objectives
  - Building a community of individual developers, academics and major industrials embracing open-source for 5G
  - Become a strong voice and maybe a game-changer in the 3GPP world
    - Real impact from “the little guys” on 3GPP systems
Collaborative Web Tools

- **Main page:**
  - [https://www.openairinterface.org](https://www.openairinterface.org)

- **Code available from**
  - RAN (eNB, UE, gNB, nr-UE)
    - [https://gitlab.eurecom.fr/oai/openairinterface5g](https://gitlab.eurecom.fr/oai/openairinterface5g)
  - EPC and 5GC
    - [https://github.com/OPENAIRINTERFACE/openair-cn](https://github.com/OPENAIRINTERFACE/openair-cn)

- **Mailing lists**
  - [https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/MailingList](https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/MailingList)

- **Developer meetings (Eurecom & Webex)**
  - General: Tuesday Mornings 11:00 CET
  - NR: Friday, alternating between 11:00 CET (Europe/Asia) and 16:00 CET (Europe/Americas)

- **Forum in Chinese**
  - [http://bbs.opensource5g.org/forum.php](http://bbs.opensource5g.org/forum.php)

- **Other tools:**
  - [https://openairinterface.slack.com](https://openairinterface.slack.com)
  - [https://trello.com/oaidev](https://trello.com/oaidev)
The OpenAirInterface Software Alliance

- Launched in 2014 as an endowment fund (French “Fonds de Dotation”)
- Current strategic members (Orange, TCL, Nokia Bell Labs, Fujitsu, PAWR)
- Many associate members (Samsung, Interdigital, ng4t, Cisco, B-COM, INRIA, IMT, TNO, III, Rutgers WINLAB, U. Washington, IITH, BUPT, etc.)

Goals:
- Promote OpenAirInterface and its open-source licensing model
- Support the community of developers and users
OSA Boards

- **Strategic Board**
  - OSA strategic members
  - Webex meeting every 2 weeks
  - Overview of recent activity
  - Discussion on
    - strategic objectives
    - Legal matters
    - Links with other organizations (e.g. standardization)

- **Technical Board**
  - Members: OSA, Orange, Nokia, Fujitsu, TCL, EURECOM, BCOM, Fraunhofer IIS, OpenCells
  - Webex meeting every 2 weeks
  - Roadmap implementation
  - Architecture evolution
  - Documentation & Coding guidelines
  - Review merge requests
    - Code quality
    - Adherence to architecture
The OAI Licensing model

- FRAND License is based on Apache V2.0 but allows committing software with patent rights into OSA and still keep licensing rights -> Inline with 3GPP fair use licensing policy
- We work closely with ETSI on implications of open-source for licensing/certification
Releases and branches

- **Master branch:**
  - Contains major releases and critical fixes
  - Updated last January 2019 release 1.0.0 (current release 1.0.3)
  - Update target: every 6 months
  - Every feature in master must be tested in CI
  - Roadmap defined by technical committee

- **Develop branch:**
  - Contains new features, new tests, bugfixes
  - Updated every week through merge request (MR) process
  - Every MR must pass regression test CI

- **Feature branches:**
  - For cutting edge development (e.g. NB-IoT, 5G-NR, etc)
  - Regular MR with develop desirable (at least should be updated regularly with develop to stay in sync)
  - Might have their own tests

- **Bugfix branches**
Master branch update since last workshop

- **v1.0.0** -> January 2019.
  - This version implements the RAU/RRU architectural splits (FAPI, nFAPI and IF4.5)
  - Repository tree structure prepares future integrations of features such as LTE-M, NB-IoT, and 5G-NR.
  - Preliminary X2 support has been implemented.
  - S1-flex has been introduced.
  - New tools: config library, telnet server, ...
  - A lot of bugfixes and a proper automated Continuous Integration process validates contributions.

- **v1.0.1** -> February 2019: Bug fix for the UE L1 simulator.
- **v1.0.2** -> February 2019: Full OAI support for 3.13.1 UHD
- **v1.0.3** -> June 2019: Bug fix for LimeSuite v19.04.0 API
Major develop branch updates since last workshop

- **New Features**
  - LTE-M,
  - X2 interface and handover,
  - CU/DU split (F1 interface),
  - CDRX,
  - eMBMS,
  - multi RRU support

- **Continuous Integration**
  - LTE-M,
  - F1 interface,
  - CDRX,
  - OAI UE (standard & “noS1”)

- **Other improvements**
  - Build system:
    - One single build including simulators and “noS1” mode (now a runtime option --noS1)
    - TUN interface now used by default (for UE, eNB-noS1, and UE-noS1)
      - To use legacy kernel module, Can be disabled using the --nokrnmod 0
  - more code cleanup
  - cppcheck
Roadmap for master/develop branch

- **v1.1.0: summer 2019**
  - Current develop branch
  - D2D?
  - CI testing of X2 handover
  - Improve CI for LTE-M
  - Improve stability of F1 interface

- **v1.2.0: winter 2019/2020**
  - 5G-NR
  - NB-IoT
  - Dual connectivity
  - LTE-M enhancements (CDRX and extended coverage)
  - Multi-RRU enhancements (reciprocity)
OpenAirInterface eNB features (PHY)

- The Physical layer implements 3GPP 36.211, 36.212, 36.213 and provides the following features:
  - LTE release 8.6 compliant, and implements a subset of release 10;
  - FDD and TDD configurations 1 (experimental) and 3;
  - Bandwidth: 5, 10, and 20 MHz;
  - Transmission modes: 1, 2 (stable), 3, 4, 5, 6, 7 (experimental);
  - Max number of antennas: 2
  - CQI/PMI reporting: aperiodic, feedback mode 3-0 and 3-1;
  - PRACH preamble format 0
  - All downlink (DL) channels are supported: PSS, SSS, PBCH, PCFICH, PHICH, PDCCH, PDSCH, PMCH;
  - All uplink (UL) channels are supported: PRACH, PUSCH, PUCCH (format 1/1a/1b), SRS, DRS;
  - HARQ support (UL and DL);
  - Highly optimized base band processing (including turbo decoder).
- Expected throughputs DL
  - 5 MHz, 25 PRBS/MCS 28 = 16-17 Mbit/s (measured with COTS UE Cat 3/4)
  - 10 MHz, 50 PRBS/MCS 28 = 34-35 Mbit/s (measured with COTS UE Cat 3/4)
  - 20 MHz, 100 PRBS/MCS 28 = ~70 Mbit/s (measured with COTS UE Cat 3/4)
- Expected throughputs UL
  - 5 MHz, 20 PRBs / MCS 20 = 9 Mbit/s (measured with COTS UE Cat 3/4)
  - 10 MHz, 45 PRBs / MCS 20 = 17 Mbit/s (measured with COTS UE Cat 3/4)
  - 20 MHz, 96 PRBs / MCS 20 = ~35 Mbit/s (measured with COTS UE Cat 3/4)
OpenAirInterface eNB features (MAC)

- The MAC layer implements a subset of the 3GPP 36-321 release v8.6 in support of BCH, DLSCH, RACH, and ULSCH channels.

- The eNB MAC implementation includes:
  - RRC interface for CCCH, DCCH, and DTCH
  - Proportional fair scheduler (round robin scheduler soon)
  - DCI generation
  - HARQ Support
  - RA procedures and RNTI management
  - RLC interface (AM, UM)
  - UL power control
  - Link adaptation
OpenAirInterface eNB features (PDCP)

- The current PDCP is header compliant with 3GPP 36-323 Rel 10.1.0 and implement the following functions:
  - User and control data transfer
  - Sequence number management
  - RB association with PDCP entity
  - PDCP entity association with one or two RLC entities
  - Integrity check and encryption using the AES and Snow3G algorithms
OpenAirInterface eNB features (RLC)

- The RLC layer implements a full specification of the 3GPP 36-322 release v9.3
- RLC TM (mainly used for BCCH and CCCH)
  - Neither segment nor concatenate RLC SDUs
  - Do not include a RLC header in the RLC PDU
  - Delivery of received RLC PDUs to upper layers
- RLC UM (mainly used for DTCH)
  - Segment or concatenate RLC SDUs according to the TB size selected by MAC
  - Include a RLC header in the RLC PDU
  - Duplication detection
  - PDU reordering and reassembly
- RLC AM, compatible with 9.3
  - Segmentation, re-segmentation, concatenation, and reassembly
  - Padding
  - Data transfer to the user
  - RLC PDU retransmission in support of error control and correction
  - Generation of data/control PDUs
OpenAirInterface eNB features (RRC)

- Based on 3GPP 36.331 v14.3.0.
  - System Information broadcast (SIB 1, 2, 3, and 13)
  - RRC connection establishment
  - RRC connection reconfiguration (addition and removal of radio bearers, connection release)
  - RRC connection release
  - RRC connection re-establishment
  - Inter-frequency measurement collection and reporting
  - eMBMS for multicast and broadcast
  - X2 Handover
  - Paging
  - RRC inactivity timer
The X2AP layer is based on 3GPP 36.423 v14.6.0 and implements the following functions:

- X2 Setup Request
- X2 Setup Response
- X2 Setup Failure
- Handover Request
- Handover Request Acknowledge
- UE Context Release
- X2 timers (t_reloc_prep, tx2_reloc_overall)
- Handover Cancel
eNB Functional Splits

- IF4.5/IF5: similar to IEEE P1914.1
- FAPI/nFAPI (IF2): specified by small cell forum, implementation (open-nFAPI) by CISCO
- IF1: same as F1 in 3GPP Rel 15
eNB Functional Split Architecture (under integration)
Buggy/Missing/Incomplete Features in develop

- **Essential**
  - TM3/4/8/9/10 : incomplete
    - TM3 PHY (Eurecom) and MAC (Fujitsu) available – to be integrated
  - PUCCH format 2 : incomplete
  - Measurement gap handling : missing
  - Carrier Aggregation : incomplete

- **Needs improvement**
  - RLC AM mode
Very recent and upcoming eNB features

- DRX/eDRX handling
- Multi RRU handling and synchronization
- Rel 12 Dual Connectivity
- Rel 13 LTE-M
- Rel 14 NB-IoT
- Rel 14 D2D/Sidelink/ProSe
  - see also presentation during workshop
- Rel 14 feMBMS
- Rel 15 5G-NR
Support for eMTC in OAI

- **Contributors**
  - Orange Cairo, Eurecom

- **Release 13 eMTC integrated in develop**

- **Tested with commercial LTE-M Modules**
  - Nimbelink/Pycom Modules (Sequans Cat-M chipset)
  - Nimbelink Modules (QCOM chipset)
  - Nordic Semiconductor (nRF91 prototype)

- **Integration in OAI CI (Nimbelink, Pycom)**

- **Currently limited (testing) to CEMode A (CE Levels 0,1)**

(c) Eurecom 2018
Current Status

- **Configuration:** lots of new eMTC-related parameters to be fed to L1/L2 stack
  - **RRC**
    - Handling of Rel-13 information elements for eMTC
    - eMTC System Information handling
    - Extra bits in MIB
    - SIB1/SI: Quite Different from legacy LTE (repetitions, frequency-hopping, no DCI)
  - **PRACH handling**
    - Support for up to 4 CE levels: different number of repetitions per level => signal combining across repetitions for each level
    - New thread for eMTC PRACH (different parametrization in LTE Cell for eMTC)
  - **MPDCCH**
    - Support for EPDCCH allocation (only in MPDCCH configuration for now)
    - 3 new DCI formats
    - New procedures (search space, etc.)
    - No Repetitions yet
    - Limited to 4+2 PRB configuration
  - **PDSCH**
    - No Repetitions
  - **PUSCH**
    - No Repetitions yet
  - **PUCCH**
    - No Repetitions yet (removal of slot-frequency hopping)
  - **MAC**
    - RA procedures for eMTC
    - Basic scheduler for testing
Overview of current implementation

- Very basic scheduler
  - 1 downlink subframe every 2 frames with fixed mcs
  - 1 uplink subframe every 2 frames with fixed mcs

- Need to test repetition mechanisms for coverage enhancement
  - Will implement basic UE procedures to enhance dlsim/ulsim to test repetitions in TX/RX

- New elements in configuration file
  - targets/PROJECTSGENERIC-LTE-ENB/CONF/enb.band13.tm1.50PRB.emtc.conf
LTE-M roadmap

- Integration of IoT applications / MEC.
- Add the CDRX support
- Improve Scheduler
  - Design of joint LTE/LTE-M scheduling framework
    - How to share resources efficiently
    - Need UE L2 stub for testing scalability (minimal Cat-M1 functionality in OAI UE)
      - handle more than 1 UE
      - handle more than 1 HARQ process
- Coverage extension options
  - Repetitions for physical channels
OAI D2D Objectives

- Interfaces for ProSe applications running on top of OAI at UE
- Integration of Rel 14 Sidelink procedures (L1/L2)
- Extensions to support UE-Network relaying scenarios

**Testing**
- ProSe application from Perspecta Labs (not public)
- Public D2D application available for individual testing of PC5 features: multicast traffic, discovery, 1-to-1 connection establishment and Unicast traffic, Relay traffic.
- Small field deployment with OAI-based UEs and Infrastructure
  - Off-network and relay coverage scenarios
Recent progress and code availability

Recent progress
- Robustified operation of sidelink L1/L2 procedures in RF mode
  - Multiple bug fixes allowing much more stable performance
- Integrated support for UE-to-network relay functionality
  - Network level (IP driver) extensions and relay supporting configuration
  - Extensions in OAI RAN and the interfaces with the RF USRP B210 devices to support UL/DL and SL operations concurrently
  - Extensions at the UE NAS layer and the Core Network to integrate relay functionality signaling
    - Work in progress

Code availability
- OAI RAN Code available at LTE-sidelink branch of public OAI-RAN repository
  - To be merged with develop branch soon
- D2D testing application publicly available
  - https://gitlab.eurecom.fr/tien-thinh.nguyen/d2d-l3-stub
DRX/eDRX handling

- **Lead:**
  - Louis-Adrien Dufrene (Orange)

- **Principle:**
  - In RRC Connected state, the UE only monitors PDCCH during specific time windows
  - Some are periodic, other depends on current data exchange
  - Allow energy savings for the UE

- **Current status**
  - Implemented for LTE FDD, tested in CI
  - With the Idle DRX and the RRC inactivity timer (that allows the eNB to release UE RRC connection), OAI provides several energy saving features

- **Next steps**
  - September 2019: Provide some optimizations to the current CDRX implementation, support of more configuration possibilities (timers durations)...
  - End of S2-2019: Add the CDRX support for LTE-M (eMTC) devices
  - In the long run: add PSM and eDRX support, but these features also impact the EPC
Dual Connectivity

- **Contributors**
  - Eurecom, UPC

- **Goal:**
  - Rel12 dual connectivity with split radio bearer
  - Work will be reused for EUTRAN – NR dual connectivity

- **Status**
  - Just started X2 control-plane procedures
Multi RRU support

- **Contributors**
  - Eurecom, Orange

- **Goal:**
  - Support multiple RRU's connected to a single RAU over IF 4.5 to form a phase coherent distributed antenna array
  - Needs synchronization at 3 levels: Time, Frequency, Phase

- **Current status**
  - Time synchronization implemented & merged into develop
  - Frequency synchronization: need to hack USRP B200 mini as they are not suited for phase synchronous operation

- **Next steps**
  - Phase synchronization through reciprocity calibration
  - Requires channel measurements between RRU's
Remote Radio Unit

- Supports SISO 20 MHz
- **Total cost:** ~750$
  - UPBoard (100$)
  - USRPB200-mini (500$ in quantities)
  - PA/LNA/Switch (100$)
  - PoE+ module (50$)

40MHz reference driving LO of AD9364 RFIC transceiver
Eurecom C-RAN Deployment

Band 38 (TDD, 2.6 GHz)

20Gb optical Ethernet
1Gbit Ethernet
10MHz clock reference

level -3
RRU

EURECOM
Data Center

vEPC
MEC
RCC/RAU

Agg. switch

Comm. Sys
RF Lab

level -4

Shannon
Meeting Room

FDU
dist switch

Fourier
Meeting Room

FDU
dist switch
NB-IoT

- Contributors
  - Eurecom, Bcom, NTUST, Nokia

- Branch
  - Develop-nb-iot
NB-IoT

- **Status last workshop**
  - PHY ready & demonstrated with fixed scheduler up to Msg4

- **Current status**
  - MAC scheduler (UL/DL) and FAPI procedures ready
  - Successful reception of Msg 5 (RRConnectionSetupComplete) at eNB
  - Tests with core network (both Nokia Bell Labs LTE box and OpenAir-CN) ongoing

- **Roadmap**
  - Complete end-to-end validation
  - non-IP CIoT services (requires core support)
  - Integration of multi-carrier PUSCH
  - Integration into develop
**Contributors**
- Javier Morgade (Vicomtech)

**Current status in develop**
- **UE PHY:**
  - LTE MBMS-dedicated cell (feMBMS) procedures subset for LTE release 14
  - LTE non-MBSFN subframe (feMBMS) Carrier Aquistion Subframe-CAS procedures (PSS/SSS/PBCH/PDSCH)
  - LTE MBSFN MBSFN subframe channel (feMBMS): PMCH (CS@1.25KHz) (channel estimation for 25MHz bandwidth) (experimental)
- **UE MAC:**
  - MBMS-dedicated cell (feMBMS) RRC interface for BCCH
  - MBMS-dedicated cell (feMBMS) RRC interface for MCCH, MTCH
- **UE RLC:**
  - MBMS-dedicated cell (feMBMS) SI-MBMS/SIB1-MBMS management

**Nexts steps (Q3/Q4):**
- Above procedures on eNB side
- MCE (Multicast Control Entity) --> MCE_APP colocated with eNB
- M2 Interface / M2AP (eNB) --> MBMS Control Interface in between eNB and MCE
- M3 Interface / M3AP (eNB) --> MBMS Control Interface in between MCE and MME (M3AP support will be added for OAI-CN)
- M1 Interface / MBMS-GW support for OAI-CN
5G New Radio

- **Contributors**
  - Eurecom, Nokia BL, Orange, Fraunhofer, IISc, TCL, NTUST

- **Goals**
  - Phase 1 (Summer 2019): “noS1” 5G-NR only (with pre-configured gNB and UE, no core network)
  - Phase 2 (Autumn 2019): non-standalone (E-UTRA – NR dual connectivity with 4G core)
  - Phase 3 (2020): standalone (with 5G core)
Development phases

Non-standalone (ENDC)

EPC

- S1-U
- S1-MME
- S1-U

RAN

- eNB
- gNB
- gNB-CU
- gNB-DU
- gNB-DU

AMF/UPF

- NG-C/U

Standalone

RAN

- gNB
- gNB-CU
- gNB-DU
- gNB-DU

Phase 1

Phase 2

Phase 3
Status of OpenAirInterface 5G-NR software

- **PHY (gNB & UE)**
  - Highly efficient 3GPP compliant LDPC encoder and decoder (BG1 and BG2 supported)
  - Highly efficient 3GPP compliant polar encoder and decoder
  - Encoder and decoder for short blocks
  - NR-PSS and NR-SSS (validated with R&S FSW)
  - NR-PBCH (validated with R&S FSW)
    - Supports up to 8 SSB
    - Flexible periodicity
  - NR-PDCCH (validated with OAI UE)
    - common search space configured by MIB
    - user-specific search space configured by RRC
    - DCI formats: 00, 10
  - NR-PDSCH (validated with OAI UE)
    - Single symbol DMRS, dmrs-TypeA-Position Pos2, DMRS configuration type 1
    - PDSCH mapping type A
  - NR-PUSCH (validated in simulation only)
  - NR-PUCCH (validated in simulation only)
    - Format 0 (ACK/NACK)
  - NR-PRACH (partially validated in simulation)
    - Formats 0,1,2,3, A1-A3, B1-B3

- **PHY (UE specific)**
  - Initial sync and selection of strongest beam
  - Time tracking based on PDCCCH DMRS
  - Frequency offset estimation
  - First version of dual stream receiver for PDSCH
Status of OpenAirInterface 5G-NR software

- **Higher layers (gNB specific)**
  - Import of 38.331 RRC messages using asn1c (new version)
  - Application to read configuration file and program gNB RRC
  - Generation of MIB *(validated with R&S FSW)*
  - RRC -> MAC configuration
  - MAC -> PHY configuration using FAPI P5 interface
  - MAC dummy scheduler using FAPI P7 interface
  - “noS1” mode: re-use 4G RLC, PDCP, and IP interface to inject user-plane traffic

- **Higher layers (UE specific)**
  - Initial sync and detection of MIB
  - MAC -> PHY Configuration of PHY via UE FAPI P5 interface
  - Basic MAC to control PHY via UE FAPI P7 interface
  - “noS1” mode: re-use 4G RLC, PDCP, and IP interface to receive user-plane traffic *(under validation)*

- **Software architecture updates**
  - new rfsimulator:
    - allows to run gNB and UE without any hardware
    - Good for debugging
  - new threading architecture for UE
    - Also required to making code more thread-safe
Notes on L1/L2 Interface

- **OpenAirInterface eNB uses FAPI for L1/L2 interface**
  - Specified in SCF082.09.05 (Small Cell Forum, 2017-05-18) up to Rel 13
  - Available in open-source from CISCO (https://github.com/cisco/open-nFAPI)

- **This interface was updated by OAI to 5G for gNB and UE**

- **SCF about to release 5G NR FAPI**
  - Pre-release (v0.0.16) under review
  - OAI committed to adapt implementation

- **Agreement signed between OSA and SCF**
  - OSA members have access to this specification
Hardware Requirements

- **SDR platform**
  - ExpressMIMO2 (discontinued)
  - USRP B200, X300, N300 (recommended)
  - Blade RF
  - LMS-SDR
  - Skylark Iris
  - Syrtem

- **Host PC**
  - A powerful x86 PC (recommended)
    - Intel Core i5, i7, i9
    - Intel Xeon
    - Intel Atom
    - >4 cores, >= 3GHz, SSE 4, AVX
  - Low-cost x86 PC (for RRH)
    - Up board (up2), Euclid board

- **Antennas, Duplexers, etc**
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>USRP B210</th>
<th>USRP X310</th>
<th>USRP N310</th>
<th>Blade RF 2</th>
<th>LMS SDR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data acquisition</strong></td>
<td>USB3</td>
<td>Gbit EtherNet, PCIexpress</td>
<td>Gbit Ethernet</td>
<td>USB3</td>
<td>USB3</td>
</tr>
<tr>
<td><strong>MIMO and bandwidth capabilities</strong></td>
<td>2x1 MIMO 20MHz or 2x2 MIMO 10MHz</td>
<td>2x2 MIMO, 120MHz</td>
<td>4x4 MIMO 100MHz</td>
<td>2x2 MIMO 20MHz</td>
<td>2x2 MIMO 20MHz</td>
</tr>
<tr>
<td><strong>RF chip</strong></td>
<td>AD9361</td>
<td>n/a**</td>
<td>AD9371 (x2)</td>
<td>AD9361</td>
<td>LMS7002M</td>
</tr>
<tr>
<td><strong>Frequency range</strong></td>
<td>70MHz – 6GHz (depends on daughterbrd)</td>
<td>DC-6GHz</td>
<td>10 MHz – 6GHz</td>
<td>47MHz to 6GHz</td>
<td>300 MHz – 3.8GHz</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>€1,130</td>
<td>~€5,000</td>
<td>~€10,000</td>
<td>$480 - $720</td>
<td>$299</td>
</tr>
<tr>
<td><strong>Duplexing</strong></td>
<td>FDD* or TDD*</td>
<td>FDD* or TDD*</td>
<td>FDD* or TDD</td>
<td>FDD*</td>
<td>FDD* or TDD*</td>
</tr>
<tr>
<td><strong>Output power</strong></td>
<td>10dBm</td>
<td>n/a**</td>
<td>12-18dBm</td>
<td>8dBm</td>
<td>10dBm</td>
</tr>
<tr>
<td><strong>Noise figure</strong></td>
<td>&lt;8dB</td>
<td>n/a**</td>
<td>5.5-7.5dB</td>
<td>?</td>
<td>&lt;7dB</td>
</tr>
<tr>
<td><strong>EVM</strong>*</td>
<td>Very good</td>
<td>Excellent</td>
<td>Very good</td>
<td>???</td>
<td>Average</td>
</tr>
<tr>
<td><strong>Open source</strong></td>
<td>FPGA/driver</td>
<td>FPGA/driver</td>
<td>FPGA/Driver</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td><strong>Compatibility</strong></td>
<td>4G/5G (40MHz with ¾ sampling)</td>
<td>4G/5G (80MHz with ¾ sampling)</td>
<td>5G up to 100MHz</td>
<td>4G</td>
<td>4G</td>
</tr>
</tbody>
</table>

*needs external RF elements  
** depends on daughterboard  
*** subjective to the author 😊
Other experimental targets

- **Epiq Sidekiq**
  - Based on AD 9361
  - Mini PCIe or M.2 form factor

- **CPRI - PClexpress**
  - IT Avero
  - Based on Xilinx eval board

- **CPRI gateway**
  - Bell Labs
  - Based on Xilinx or Intel platform

- **Skylark Iris platform**
  - Based on Lime platform
  - Scalable for massive MIMO

- **SYRTEM UED platform**
  - Based on Xilinx ZC706 eval board + AD9371 daughterboard
  - 2 full duplex channels with up to 122.88 MHz sampling
  - Not 100% open source
OAI eNB + OAI UE

INSTALLATION
Software Requirements

- **Operating system**
  - Ubuntu >= 18.04
    - works for both openairinterface5g and openair-cn
    - For real-time operation, a low-latency kernel is recommended
    - For P/S-GW, gtp kernel module needs to be patched
    - See details on Wiki
  - CentOS Linux release 7.4.1708 (Core)
    - Better real-time performance than Ubuntu low-latency

- **Get code from our gitlab server**
  - RAN (eNB+UE): [https://gitlab.eurecom.fr/oai/openairinterface5g](https://gitlab.eurecom.fr/oai/openairinterface5g)
    - Branch develop latest features (recommended)
    - Several feature branches for cutting-edge developments
OpenAirInterface5G directories

- **cmake_targets**
  - New directory for building all the targets
  - Contains “mother” build_oai script

- **targets**
  - Hardware specific code (drivers, tools, etc)
  - lte-softmodem,

- **openair1**
  - Basic DSP routines for implementing subset of LTE specifications under x86 (36.211, 36.212, 36.213 3GPP specifications)
  - Channel simulation, sounding and PHY abstraction software,

- **openair2**
  - MAC/RLC/PDCP/RRC

- **openair3**
  - Contains interfaces S1-C, S1-U (GTP, SCTP, S1AP) and NAS UE

- **common/utils**
  - Utilities such as the T tracer or the ITTI
Compiling OpenAirInterface5G

- **Top-level build script** `./build_oai` located in
  - `cd openairinterface5g/cmake_targets`

- **Recent simplifications**
  - No more separate executables for “noS1”, basic simulator, etc
  - → there is only one lte_softmodem and lte-uesoftmodem!

- **Compilation options**
  - `-I` installs additional required software
  - `-w <hw_target>` select HW target
  - `--eNB` compiles the lte-softmodem
  - `--UE` compiles the lte-uesoftmodem and UE NAS parts
  - `--lte-simulators` compiles the unitary simulators
  - `-h` help

- **This creates executables in openairinterface5g/targets/bin**
  - `Liboai_device.so` symbolic link to library of current hardware (oai_usrpdevif.so, rfsimulator.so, tcp_bridge_oai.so, …)
  - `Liboai_eth_transpro.so` Ethernet transport library (IF4.5 or IF5 split)
  - `Libparams_libconfig.so` Library for parameter handling
  - `Libcoding.so` Library for channel coding
  - Nasmesh.ko and rb_tool: kernel driver and ioctl tool for noS1
  - `ue_ip.ko`, usim, nvram, conf2uedata: UE NAS driver (if TUN interface is not used)
  - `dlsim`, `ulsim`, ….: unitary simulators
Simulators

- ulsim/dlsim: unitary simulator for PHY
- Basic simulator / rfsimulator
- L1 simulator: uses IF4.5 interface
- L2 FAPI simulator: uses nFAPI interface
Basic simulator

<table>
<thead>
<tr>
<th>eNB</th>
<th>Device driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UE</th>
<th>Device driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td></td>
</tr>
</tbody>
</table>

Socket

rfsimulator

L2 simulator

<table>
<thead>
<tr>
<th>eNB</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
</tbody>
</table>

nFAPI

UE

L1 simulator

<table>
<thead>
<tr>
<th>eNB</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td>(w/o OFDM)</td>
</tr>
</tbody>
</table>

Channel (freq dom)

Socket

Multi-user channel model

Device driver

Socket

IF 4.5

Basic simulator

<table>
<thead>
<tr>
<th>eNB</th>
<th>Device driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UE</th>
<th>Device driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td></td>
</tr>
</tbody>
</table>

Socket

Multi-user channel model

Device driver

Socket

IF 4.5

L2 simulator

<table>
<thead>
<tr>
<th>eNB</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
</tbody>
</table>

nFAPI

UE

L1 simulator

<table>
<thead>
<tr>
<th>eNB</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDCP</td>
<td></td>
</tr>
<tr>
<td>RLC</td>
<td></td>
</tr>
<tr>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td>PHY</td>
<td>(w/o OFDM)</td>
</tr>
</tbody>
</table>

Channel (freq dom)

Socket

Multi-user channel model

Device driver

Socket

IF 4.5
Unitary Simulators

- **Simulates one physical channel (TX and RX) using (standardized) channel models**
  - Good for debugging and performance analysis (Monte-Carlo simulations)

- **Build using**
  - `./build_oai phy_simulators`

- **In develop**
  - `ulsim`
  - `dlsim`

- **In develop-nr**
  - `Polartest, Idpctest, smallblocktest`
  - `Nr_pbchsim, nr_prachsim`
  - `nr_dlschsim, nr_ulschsim, nr_pucchsim`
  - `nr_dlsim, nr_ulsim`
How to connect COTS phone to OAI eNB

- **Additional requirements**
  - Core network (e.g., OAI EPC, see later in training)
  - Sim card with corresponding parameters

- **Compile eNB**
  - `./build_oai --w USRP --eNB`

- **Configure eNB**
  - `targets/PROJECTS/GENERIC-LTE-EPC/CONF/`
  - Select the config file that is most appropriate for your configuration (Band and Hardware)
  - Check
    - MCC, MNC, TAC (need to match EPC)
    - `downlink_frequency`, bandwidth, etc
    - IP addresses of S1-MME and S1-U interfaces

- **Run eNB**
  - `sudo ./lte-softmodem -O <file.conf>`
How to connect OAI UE to OAI eNB

- **Compile UE**
  - ./build_oai -w USRP -UE
  - Initialize NAS (except when using TUN interface)
    - “init_nas_s1 UE” or “init_nas_noS1”

- **Run UE**
  - sudo ./lte-softmodem -U -C <freq> -r [25|50|100] -ue-scan-carrier -ue-txgain xx -ue-rxgain yy (-d) (--noS1 --nokrnmod 0)
How to setup RCC and RRU

- **RCC**
  - Check RCC parameters in RCC config file
    - IF name, local and remote IP addresses and ports
    - local_rf = "no"
    - tr_preference = "udp_if4p5"
  - ./lte-softmodem -O <file.conf>

- **RRU**
  - Check RRU parameters in RRU config file
    - IF name, local and remote IP addresses and ports
    - local_rf = "yes"
    - tr_preference = "udp_if4p5"
  - ./lte-softmodem -O <file.conf>
**Troubleshooting**

- **eNB not connection to MME / RRH**
  - Check IP addresses in config files
  - Check MCC, MNC matching

- **I get a lot of UUUs and LLLs**
  - Check the performance setting of CPU (C-states, CPU frequency)
  - Check USB3 connection (some cables are bad)

- **Phone does not connect**
  - Analyze S1AP messages in wireshark
  - Check keys in SIM card and HSS
  - ...

- **Throughput is very low**
  - Check radio conditions: duplexer, antennas, interference
DEBUGGING TOOLS
The T tracer

- The T tracer is a framework to debug and monitor the eNB softmodem.
- Combines logging, timing analysis, signal visualization, MAC PDU analysis (with wireshark)
- It is made of two main parts:
  - an events collector integrated to the real-time processing,
  - a separate set of programs to receive, record, display, replay and analyze the events sent by the collector.
- Can work locally or over network
The T tracer: usage of GUI

- **eNB is compiled by default with –T-tracer option:**
  - Otherwise disable with –disable-T-tracer

- **Compile eNB GUI:**
  - cd openairinterface5g/common/utils/T
  - make

- **Run lte-softmodem normally**
  - sudo ./lte-softmodem –O <…> --T_stdout 0

- **Run T tracer GUI**
  - ./enb –d ../T_messages

- **Other features**
  - Recording & replay
  - VCD file generation (for gtkwave)
eNB GUI

- HARQ ACK
- HARQ NAK
- New DCI
- Retr. DCI
Telnet server

- Telnet server can be used to show and change parameters at runtime
  - Log level and verbosity
  - Threads and their priority
  - Some PHY parameters (e.g. turbo iterations)

- Easily extendable

- Usage
  - ./build_oai -w USRP -eNB -build-telnetsrv
  - sudo ./lte-softmodem -O <...> --telnetsrv
  - Telnet 127.0.0.1 9090
  - Use online help
Use case I: classical 3GPP network

- OAI EPC
- Commercial/3rd party EPC
- OAI eNB
- Commercial/3rd party eNB
- OAI UE
- COTS UE
Use case II: simplified network

- Non-3GPP setup (no-S1 mode):
  - OAI eNB <-- OAI UE
Use case III: cloud-RAN

Main target of EURECOM deployment
Epiq Sidekiq

- **Based on AD 9361 chipset**
  - 70MHz - 6GHz with up to 50MHz bandwidth per channel

- **Sidekiq™ - MiniPCIe**
  - MiniPCIe card form factor (30mm x 51mm x 5mm)
  - 2 independent RF channels (2xRx or Tx+Rx)
  - PCIe Gen1.1 x1 (2.5 Gbps) interface to host + USB 2.0 interface

- **Sidekiq™ - M.2**
  - M.2 T3042-D3-B card form factor (30mm x 42mm x 4mm)
  - Up to 2x2 MIMO
  - PCIe Gen2 x1 (5 Gbps) interface to host + USB 2.0 interface

- **Under beta-testing**