OpenAirInterface and C-RAN

Presenter : N. Nikaein
EURECOM, Mobile Communication Department
RAN Evolution to the Cloud Workshop
Paris 24/06/2013

This work is licensed under a CC attribution Share-Alike 3.0 Unported license.
Outline

- What is OAI
- HW/SW platforms
- Quick Demos
- Toward C-RAN
- Provides open-source (hardware and software) wireless technology platforms
  - Target innovation in air-interface technologies and networking through experimentation
  - System approach with high level of realism
  - SDR architecture and full GPP

- We rely on the help of
  - Publicly-funded research initiatives (ICT, ANR, CELTIC)
  - Direct contracts with industrial partners
  - Widespread collaboration with a network of partners using open-source development and tools
    - LINUX/RTAI based SW development for PCs
    - LEON3/GRLIB-based HW and SW development for FPGA targets
    - LINUX networking environment
  - Experimental Licenses from ARCEP (French Regulator) for medium-power outdoor network deployments
    - 1.9 GHz TDD, 5 MHz channel bandwidth
    - 2.6 GHz FDD (two channels), 20 MHz channel bandwidth
    - 800 MHz FDD (two channels) : 10 MHz channel bandwidth
    - 3.5 GHz TDD expected for future experimentation
OAI Platforms

Three main platforms

- **Simulation/Emulation**
  - Link-level
  - System-level

- **Soft Modem**
  - EXMIMO 2
  - EMOS

- **FPGA Modem**
  - EXMIMO I
  - SoC Arch.

- **Development and integration methodology**
  - Tight interaction between the system emulation and soft modem

- **Rich R&D environment**
  - Protocol behavior, integration and validation
  - Performance evaluation
  - System/application testing

- **C implementation under realtime Linux optimized for x86, could be ported to the ARM architecture**
ExpressMIMO2 software architecture

Application

ExpressMIMO2 (LEON)

Kernel Space

PCIexpress

Linux driver (openair_rf.ko)

User Space

Octave API

C API

Octave

Using real-time Linux extension (RTAI, Xenomai, RT-preempt)

Modem control and sync. (lte-softmodem)

Linux network driver nasmesh.ko)

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom

targets/ARCH/EXMIMO/DRIVER/eurecom
Software Architecture
In-lab System Validation Platform

Web Portal / Interface

- **Input:**
  - Description of application scenario
  - Initialization and configuration of all the blocks

- **Execution:**
  - PHY procedures, L2 protocols, traffic generator
  - PHY abstraction, channel model, and mobility model
  - Emulation medium: shared memory

- **Output:**
  - Execution logs
  - System/protocol operation
  - Key performance indicators: latency, jitter, throughput/goodput

OAEMU

**Application**

- Wireshark
- Log Gen.
- MSC & VCD
- Result Gen.

**Traffic Gen.**

- L3 Protocols
- OAI Network Interface
- L2 Protocols
- PHY Procedures
- PHY / PHY Abstraction

**Config. Gen.**

- DRB Config.
- Traffic Gen.

**Ch. Realization**

- Channel Descriptor
- Mobility Gen
- Channel Trace
- EMOS
EQUIPMENT AND SW
Prototype Equipment Timeline

**PLATON/RHODOS**
- 2003
- Cellular Systems
- Pure Software Radio
- WCDMA-TDD
- All-IPv4/v6

**CBMIMOI – V1**
- 2004
- AgileRF/Express MIMO
- CBMIMOI – V1
- 2005
- WCDMA-TDD
- All-IPv4/v6

**CBMIMOI – V2**
- 2006
- ExpressMIMO2
- CBMIMOI – V2
- 2007
- AgileRF/Express MIMO
- CBMIMOI – V2
- 2008
- AgileRF/Express MIMO
- CBMIMOI – V2
- 2009
- AgileRF/Express MIMO
- CBMIMOI – V2
- 2010
- Cellular, AdHoc and P2MP Topologies
- FPGA SoC (Virtex 5)+ Interface for partner Processing Engines
- Agile Tuning modules (0.2 – 7 GHz)
- Maximum Channel BW 20 MHz
- OFDM(A)/WCDMA
- 2011
- Cellular, AdHoc and P2MP Topologies
- FPGA SoC (Virtex 5)+ Interface for partner Processing Engines
- Agile Tuning modules (0.2 – 7 GHz)
- Maximum Channel BW 20 MHz
- OFDM(A)/WCDMA
- 2012
- Cellular, AdHoc and P2MP Topologies
- FPGA SoC (Virtex 5)+ Interface for partner Processing Engines
- Agile Tuning modules (0.2 – 7 GHz)
- Maximum Channel BW 20 MHz
- OFDM(A)/WCDMA
- 2013
- Cellular, AdHoc and P2MP Topologies
- FPGA SoC (Virtex 5)+ Interface for partner Processing Engines
- Agile Tuning modules (0.2 – 7 GHz)
- Maximum Channel BW 20 MHz
- OFDM(A)/WCDMA

AdHoc/Mesh and Cellular Topologies
- FPGA-SoC (Virtex 2)
- 2x2 OFDM(A) @ 2 GHz, 5MHz channels
- Cellular (towards LTE)
ExpressMIMO2 Spartan6 PCIe board

- 4 LIME microsystems RF ASICs (LTE small-cell eNB/UE compliant) – TDD/FDD (0 dBm output, 5 dB noise-figure)
- PCIe-based acquisition (1-way initially, 4-way possible)
- LEON3 embedded system for controlling x86 SDR
Software Roadmap

Wireless3G4Free


OpenAirInterface (WIDENS/CHORIST)

OpenAir4G

OpenAir.11p

OpenAir4GEPC

TD-SCDMA SDR
IPv6 interconnect
No longer supported

AdHoc/Mesh and Cellular Topologies
In-house MIMO-OFDMA TDD waveform (WiMAX 2004 like)
Distributed Signal Processing and Mesh-Topology functions (L2.5 relaying)

LTE compliant waveform
Mesh extensions from WIDENS/CHORIST
3GPP-LTE protocol stack (openair2+openair3)
OpenAir4G software
Access Stratum + EPCLite

- Targeting
  - Experimentation with commercial UE (1-2 years time frame)
  - Reconfigurable LTE/LTE-A through APIs (3 years time frame)

GNU GPL License
QUICK DEMOS
## OAI Wireshark Trace Control Plane

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.05896</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>88</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>6</td>
<td>0.07519</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>88</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>7</td>
<td>0.08315</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>93</td>
<td>SystemInformation [ SIB2 SIB3 ]</td>
</tr>
<tr>
<td>8</td>
<td>0.09269</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>64</td>
<td>RACH Preamble chosen for UE 0 (RAPID=0, attempt=0)</td>
</tr>
<tr>
<td>9</td>
<td>0.09322</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>88</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>10</td>
<td>0.09426</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>71</td>
<td>RAR (RA-RNTI=1, SF=0) (RAPID=56: TA=1, UL-Grant=848, Temp C-RNTI=51364)</td>
</tr>
<tr>
<td>11</td>
<td>0.10658</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC UL_CCCH</td>
<td>74</td>
<td>RRCConnectionRequest</td>
</tr>
<tr>
<td>12</td>
<td>0.10801</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>80</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>13</td>
<td>0.11070</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_CCCH</td>
<td>113</td>
<td>RRCConnectionSetup</td>
</tr>
<tr>
<td>14</td>
<td>0.12820</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>80</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>15</td>
<td>0.13621</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>82</td>
<td>UL-SCH: (SF=2) UEId=0 (Padding) (Short BSR) (1:remainder)</td>
</tr>
<tr>
<td>16</td>
<td>0.14537</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>82</td>
<td>UL-SCH: (SF=2) UEId=0 (Power Headroom Report) (1:6 bytes) (Padding:remainder)</td>
</tr>
<tr>
<td>17</td>
<td>0.14693</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>80</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>18</td>
<td>0.15019</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>75</td>
<td>DL-SCH: (SF=6) UEId=0 (1.2 bytes) (Padding:remainder)</td>
</tr>
<tr>
<td>19</td>
<td>0.15133</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_DCCH</td>
<td>75</td>
<td>SecurityModeCommand [8-bytes]</td>
</tr>
<tr>
<td>20</td>
<td>0.15749</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>93</td>
<td>SystemInformation [ SIB2 SIB3 ]</td>
</tr>
<tr>
<td>21</td>
<td>0.16427</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>82</td>
<td>UL-SCH: (SF=2) UEId=0 (1.2 bytes) (Padding:remainder)</td>
</tr>
<tr>
<td>22</td>
<td>0.16807</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>80</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>23</td>
<td>0.17639</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC UL_DCCH</td>
<td>82</td>
<td>SecurityModeComplete [7-bytes]</td>
</tr>
<tr>
<td>24</td>
<td>0.18852</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>80</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>25</td>
<td>0.18381</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_DCCH</td>
<td>77</td>
<td>UECapabilityEnquiry [8-bytes]</td>
</tr>
<tr>
<td>26</td>
<td>0.19338</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>82</td>
<td>UL-SCH: (SF=2) UEId=0 (Padding) (Short BSR) (1:remainder)</td>
</tr>
<tr>
<td>27</td>
<td>0.19667</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>MAC-LTE</td>
<td>82</td>
<td>UL-SCH: (SF=3) UEId=0 (1:14 bytes) (Padding:remainder)</td>
</tr>
<tr>
<td>28</td>
<td>0.20876</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_DCCH</td>
<td>139</td>
<td>RRCConnectionReconfiguration [67-bytes]</td>
</tr>
<tr>
<td>30</td>
<td>0.22173</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC UL_DCCH</td>
<td>82</td>
<td>RRCConnectionReconfigurationComplete [7-bytes]</td>
</tr>
<tr>
<td>31</td>
<td>0.22758</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>88</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>32</td>
<td>0.23967</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>88</td>
<td>SystemInformationBlockType1</td>
</tr>
<tr>
<td>33</td>
<td>0.24427</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>93</td>
<td>SystemInformation [ SIB2 SIB3 ]</td>
</tr>
<tr>
<td>34</td>
<td>0.25206</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>LTE RRC DL_SCH</td>
<td>80</td>
<td>SystemInformationBlockType1</td>
</tr>
</tbody>
</table>
OAI Wireshark Trace
Data Plane

Frame 34332: 225 bytes on wire (1800 bits), 225 bytes captured (1800 bits)
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
User Datagram Protocol, Src Port: 33443 (33443), Dst Port: distinct (9999)
MAC-LTE DL-SCH: (SF=6) UEId=0 (3:remainder) [158-bytes]

MAC PDU Header (3:remainder) [1 subheaders]
Sub-header (tclid=3, length is remainder)
RLC-LTE (DRB:3) [DL] [UM] DRB:3 SN=61 [158-bytes]

PDCP-LTE (SN=61)(156 bytes data)

Internet Protocol Version 4, Src: 10.0.1.1 (10.0.1.1), Dst: 10.0.1.2 (10.0.1.2)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x39bf [correct]
Identifier (BE): 4131 (0x1023)
Identifier (LE): 8976 (0x2310)
Sequence number (BE): 7 (0x0007)
navid@oai:~$ ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_req=1 ttl=64 time=30.7 ms
64 bytes from 10.0.1.2: icmp_req=2 ttl=64 time=22.5 ms
64 bytes from 10.0.1.2: icmp_req=3 ttl=64 time=23.8 ms
64 bytes from 10.0.1.2: icmp_req=4 ttl=64 time=21.5 ms
64 bytes from 10.0.1.2: icmp_req=5 ttl=64 time=13.7 ms
64 bytes from 10.0.1.2: icmp_req=6 ttl=64 time=51.9 ms
64 bytes from 10.0.1.2: icmp_req=7 ttl=64 time=30.5 ms
64 bytes from 10.0.1.2: icmp_req=8 ttl=64 time=38.2 ms
64 bytes from 10.0.1.2: icmp_req=9 ttl=64 time=36.0 ms
64 bytes from 10.0.1.2: icmp_req=10 ttl=64 time=21.7 ms
^C
--- 10.0.1.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 13.732/29.098/51.949/10.378 ms

TOWARD C-RAN
One approach for DSP is to use SIMD engines on general-purpose processors GPP, CPUs (e.g. x86) for front-end and channel decoding

- This was started (first?) in the SpectrumWare project in the late 1990s at MIT for GSM soft-base stations and resulted in a company called VANU inc. (still exists!)
- A couple of years later (1999) we developed Wireless3G4Free.com (precursor to OpenAirInterface.org) which did the same for TD-SCDMA
- OpenAirInterface (2007) provided OFDMA => OpenAir4G (2011)

Key elements

- Real-time extensions to Linux OS
- Real-time data acquisition to PC
- SIMD optimized integer DSP (64-bit MMX → 128-bit SSE2/3/4 → 256-bit AVX2)
- x86-64 : more efficient than legacy x86
Performance Bottleneck

- Turbo decoder is the primary (by far) bottleneck in an HSPA/LTE receiver
  - OAI adopts a scalable SIMD max-logmap
  - Integer x86 SIMD arithmetic
  - Care must also be taken when it comes to the interleaving and data reforming operations in a turbo decoder. These can also benefit from SIMD

- OAI Radix-4 DFTs provide essentially the same performance as FFTW
  - Approx 300 cycles for 64-point DFT on a x86-64 core i5/i7/Xeon machine (gcc generated)
  - For larger DFTs, both are slightly less than the closed IPP libraries from Intel (based on SPIRAL code generator)

- PDCP de/encryption and ROHC
  - Not yet implemented
OAI Modem Performance (eNB)

- gcc 4.7.3, x86-64 (1.8 GHz Xeon E5-2650L), 20 MHz bandwidth (mcs 19 – 16QAM, transmission mode 1 - SISO)

eNB RX function statistics (per 1ms subframe)

<table>
<thead>
<tr>
<th>Function</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFDM_demod time</td>
<td>:202.992302 us (100 trials)</td>
</tr>
<tr>
<td>ULSCH demodulation time</td>
<td>:347.516264 us (100 trials)</td>
</tr>
<tr>
<td>ULSCH Decoding time</td>
<td>:1271.786873 us (100 trials, max 3032.161541)</td>
</tr>
<tr>
<td>sub-block interleaving</td>
<td>18.597523 us (700 trials)</td>
</tr>
<tr>
<td>demultiplexing</td>
<td>213.908853 us (100 trials)</td>
</tr>
<tr>
<td>rate-matching</td>
<td>19.591607 us (700 trials)</td>
</tr>
<tr>
<td>turbo_decoder(5632 bits)</td>
<td>103.893272 us (187021 cycles, 700 trials)</td>
</tr>
<tr>
<td>init</td>
<td>16.228172 us (cycles/iter 14606.423571, 700 trials)</td>
</tr>
<tr>
<td>alpha</td>
<td>5.040021 us (cycles/iter 18145.403571, 2800 trials)</td>
</tr>
<tr>
<td>beta</td>
<td>5.173861 us (cycles/iter 18627.263571,2800 trials)</td>
</tr>
<tr>
<td>gamma</td>
<td>1.609997 us (cycles/iter 5796.413571,2800 trials)</td>
</tr>
<tr>
<td>ext</td>
<td>3.851109 us (cycles/iter 13865.007857,2800 trials)</td>
</tr>
<tr>
<td>intl1</td>
<td>3.933486 us (cycles/iter 7080.792857,1400 trials)</td>
</tr>
<tr>
<td>intl2+HD+CRC</td>
<td>7.454629 us (cycles/iter 13419.312857,700 trials)</td>
</tr>
</tbody>
</table>
OAI Modem Performance (eNB)

Summary (processing for 1ms subframe)
- RX : 1820 µs (< 2 cores)
- TX : 330 µs (1/3 core)

On 3 GHz machine, < 2 cores for 20 MHz eNB

On future AVX2 (256-bit SIMD), turbo decoding and FFT processing will be exactly twice as fast
- 1 core per eNB

---

eNB TX function statistics (per 1ms subframe)

<table>
<thead>
<tr>
<th>Function</th>
<th>Time (us)</th>
<th>Trials</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFDM_mod time</td>
<td>176.144838</td>
<td>100</td>
</tr>
<tr>
<td>DLSCH modulation time</td>
<td>55.319101</td>
<td>100</td>
</tr>
<tr>
<td>DLSCH scrambling time</td>
<td>22.194255</td>
<td>100</td>
</tr>
<tr>
<td>DLSCH encoding time</td>
<td>79.016000</td>
<td>100</td>
</tr>
<tr>
<td>DLSCH turbo encoding time</td>
<td>27.376582</td>
<td>200</td>
</tr>
<tr>
<td>DLSCH rate-matching time</td>
<td>19.505191</td>
<td>200</td>
</tr>
<tr>
<td>DLSCH sub-block interleaving time</td>
<td>20.249680</td>
<td>200</td>
</tr>
</tbody>
</table>
Real-time Soft-MODEMS

- Map OAI DSP to HW using RTAI or RT-PREEMPT
  - RT-PREEMPT has benefit that it is compatible/transparent with standard Linux
    - Same application can be executed (with RT faults) on regular kernel
  - Map modem processing based on time-scales to multiple threads
    - Allows efficient use of available CPU cores
    - Allows for prioritization of processing
Real-time SoftModems

Slot $2n-1$  
RX$_{DEC}$ $(2n-2)$  
RX$_{LLR}$ $(2(n-1))$  
TX $(2n+2)$, RX $(2n-1)$

Slot $2n$  
RX$_{DEC}$ $(2n-1)$  
RX$_{LLR}$ $(2n-1)$  
RX $(2n)$

Slot $2n+1$  
RX$_{DEC}$ $(2n)$  
RX$_{LLR}$ $(2n)$  
TX $(2n+4)$, RX $(2n+1)$

Slot $2(n+1)$  
RX$_{LLR}$ $(2n+2)$

Slot $2n+3$  
RX$_{LLR}$ $(2n+2)$  
RX $(2n+2)$
From ExpressMIMO2 to RRH+CPRI Integration

2.5 Gbit/s ➞ 1eNB 20MHz 1TX/RX SISO ≡ 1Core
4eNB 5MHz
15 Gbit/s ➞ 1eNB 20MHz 2TX/RX MIMO Tri-sector

Lab Configuration

BBUs
/Proc
Memory
Root Port

Fronthaul
OF CPRI (~40Gb Eth)
Parallel use of ExpressMIMO2

various configurations are possible

<table>
<thead>
<tr>
<th>Channel bandwidth [MHz]</th>
<th>1.4</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of resource blocks (N_{bb})</td>
<td>6</td>
<td>15</td>
<td>25</td>
<td>50</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td>Number of occupied subcarriers</td>
<td>72</td>
<td>180</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
</tr>
<tr>
<td>IDFT(Tx)/DFT(Rx) size</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>1336</td>
<td>2048</td>
</tr>
<tr>
<td>Sample rate [MHz]</td>
<td>1.92</td>
<td>3.84</td>
<td>7.68</td>
<td>15.36</td>
<td>23.04</td>
<td>30.72</td>
</tr>
<tr>
<td>Samples per slot</td>
<td>960</td>
<td>1920</td>
<td>3840</td>
<td>7680</td>
<td>15360</td>
<td>15360</td>
</tr>
</tbody>
</table>

16x 2.5Gbit/s = 40 Gbit/s peak

16-way PCIe Backplane

16-way PCIe Gen 2
(80 Gbit/s peak)

High-End Intel Xeon Computing Engine
3GHz Dual-Proc AVX2
20 Parallel Cores

OpenAir4G
RT-Linux
MODEMs
#CPU vs. #RRH: load balancing

- **For a RRH**
  - $MCS_i < MCS_{\text{max}}$

- **For a BBU**
  - $\overline{MCS} \ll N \times MCS_{\text{max}}$

- **Num CPU scales with**
  - $\text{num CPU} \approx \overline{MCS}$
EPCaaS Integration

Redundancy
- +1
- Multi Eth IF / M(V)NO

Largest XEON server
16 10-core processors on common motherboard (160 AVX2 cores!)
Contacts Information

- **URL:**
  - www.openairinterface.org
  - https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface

- **Partnership and collaboration:**
  - openair_admin@eurecom.fr

- **Technical Support:**
  - openair_tech@eurecom.fr
Collaborative Web Tools

- **OpenAirInterface SVN Repositories**
  - All development is available through [www.openairinterface.org](http://www.openairinterface.org)’s SVN repository (openair4G) containing
    - OPENAIR0 (open-source real-time HW/SW)
    - OPENAIR1 (open-source real-time and offline SW)
    - OPENAIR2 (open-source real-time and offline SW)
    - OPENAIR3 (open-source Linux SW suite for cellular and MESH networks)
    - TARGETS: different top-level target designs (emulator, RTAI, etc.)
  - Partners can access and contribute to our development
    - [https://svn.eurecom.fr/openair4G](https://svn.eurecom.fr/openair4G) (RO access)

- **OpenAirInterface TWIKI**
  - A TWIKI site for quick access by partners to our development via a collaborative HOW-TO

- **Forum and bugzilla**
  - external support services (not currently used effectively)

- **Mailing list**
  - openair4G-devel@eurecom.fr
EURECOM MEMBERS