École d'été Rescom
la 5G et l'Internet des Objets
du 13 au 17 juin 2016 / Guidel plage

EURECOM
Sophia Antipolis

HUAWEI NOKIA cnrs TELECOM Bretagne INSTITUT Mines-Télécom

Cloud-native and programmable radio access networks

Navid Nikaein

Communication System, Eurecom

FLEx
FIRE LTE TESTBEDS FOR OPEN EXPERIMENTATION

Coherent

MCN
Mobile Cloud Networking

5G PPP
The 5G Infrastructure Public Private Partnership

European Commission

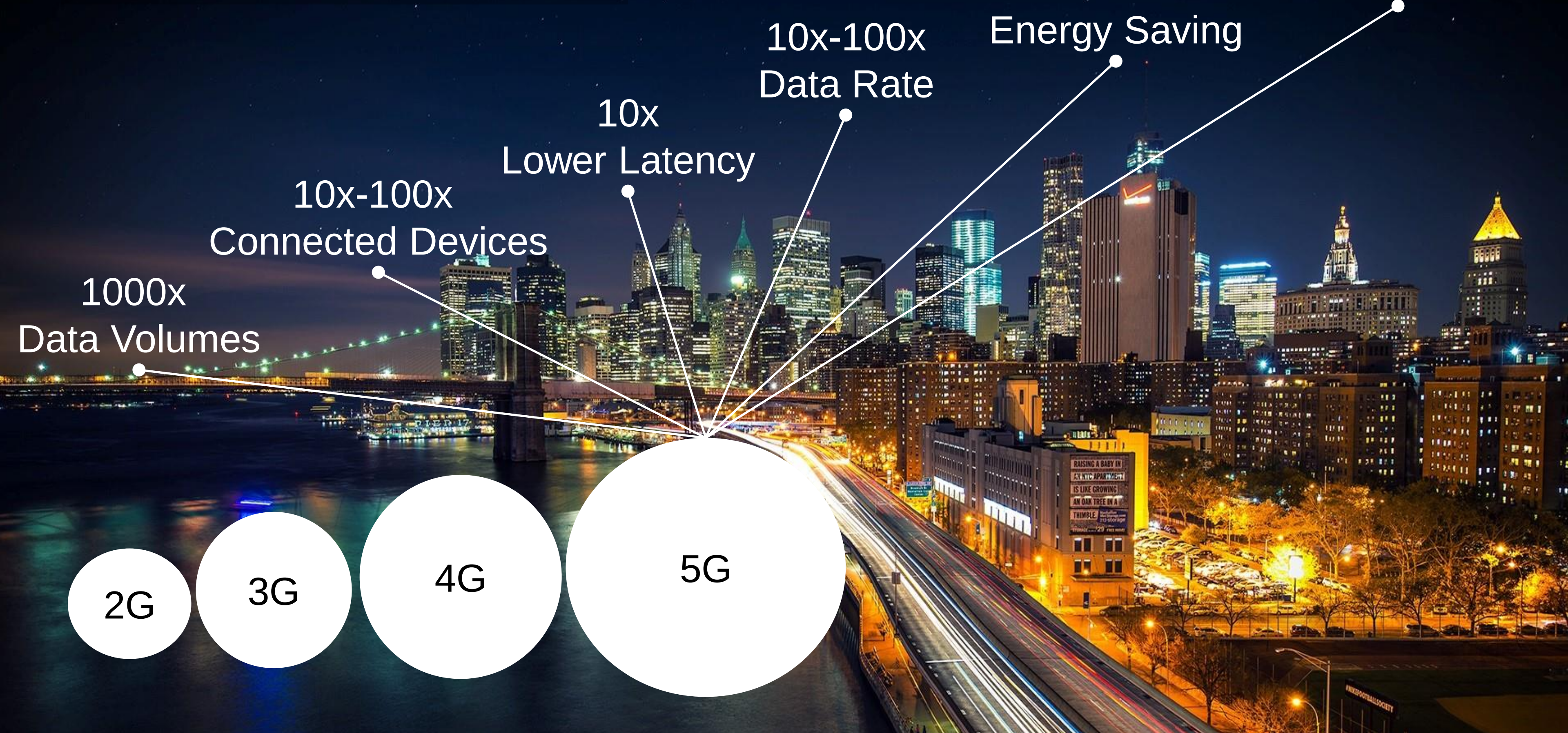
OPEN AIR INTERFACE

16th June, 2016

Economics of mobile are changing

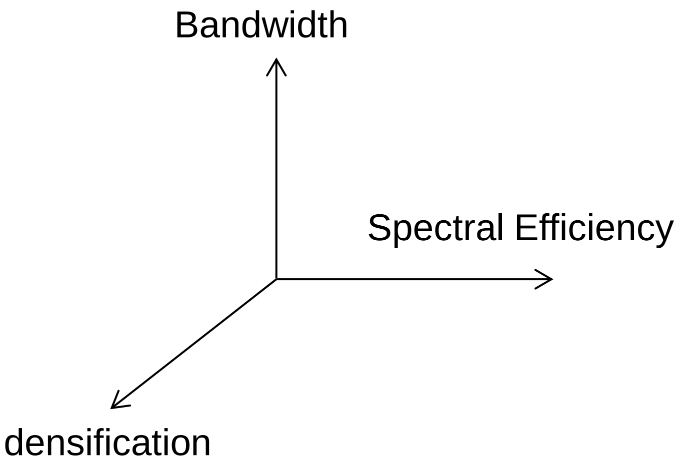
- **Softwarization and Commoditization**
  - Software implementation of network functions on top of GPP with no or little dependency on a dedicated hardware
    - Full GPP vs. accelerated vs. system-on-chip
  - Programmable RF
- **Virtualization and Cloudification**
  - Execution of network functions on top of virtualized computing, storage, and networking resources controlled by a cloud OS.
  - Share I/O resources among multiple guests
- **Emergence of rich ecosystem and opensource for telecom**
  - NFV, SDN and MEC
  - Open APIs and standardized I/F

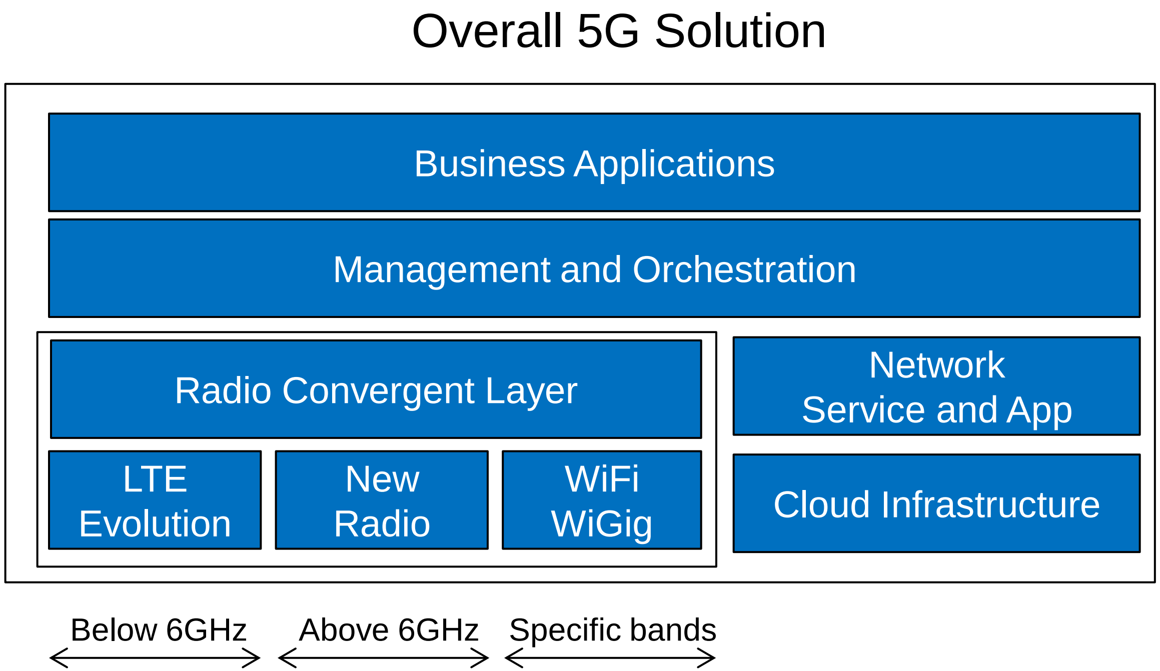
5G Target KPIs



Not all of these Requirements need to be satisfied simultaneously

5G will be a paradigm shift

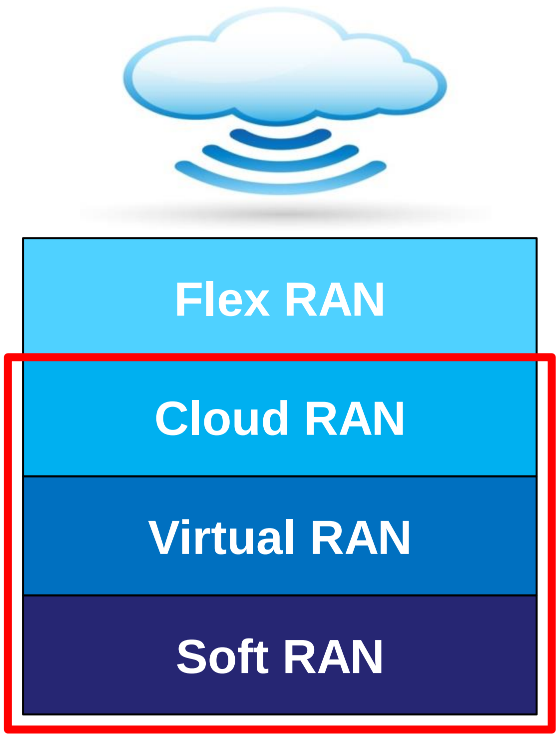




- **5G is not just a new radio/spectrum, but also a new architecture and business helper**

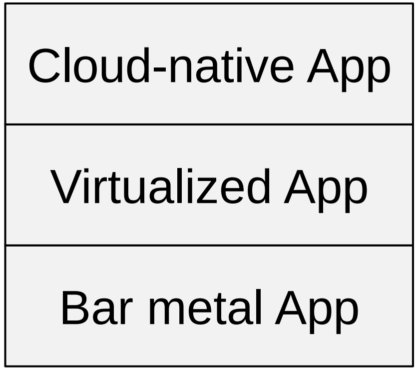
Tutorial – Part I

- Technology
- Challenges
- Results
- Conclusion



Cloud Computing

- **Cloud Computing disrupts IT consumption and operations**
  - on-demand, self-service, elastic, pay-as-you-go, metered service
  - Additional advantage: automated management, remote access, multi-tenancy, rapid deployment and service provisioning, load-balancing
- **New business models based on sharing**
  - Public, private, local, remote, community, and hybrid clouds
- **Promising business potentials (CAPEX/OPEX)**
  - Start small and grow on-demand



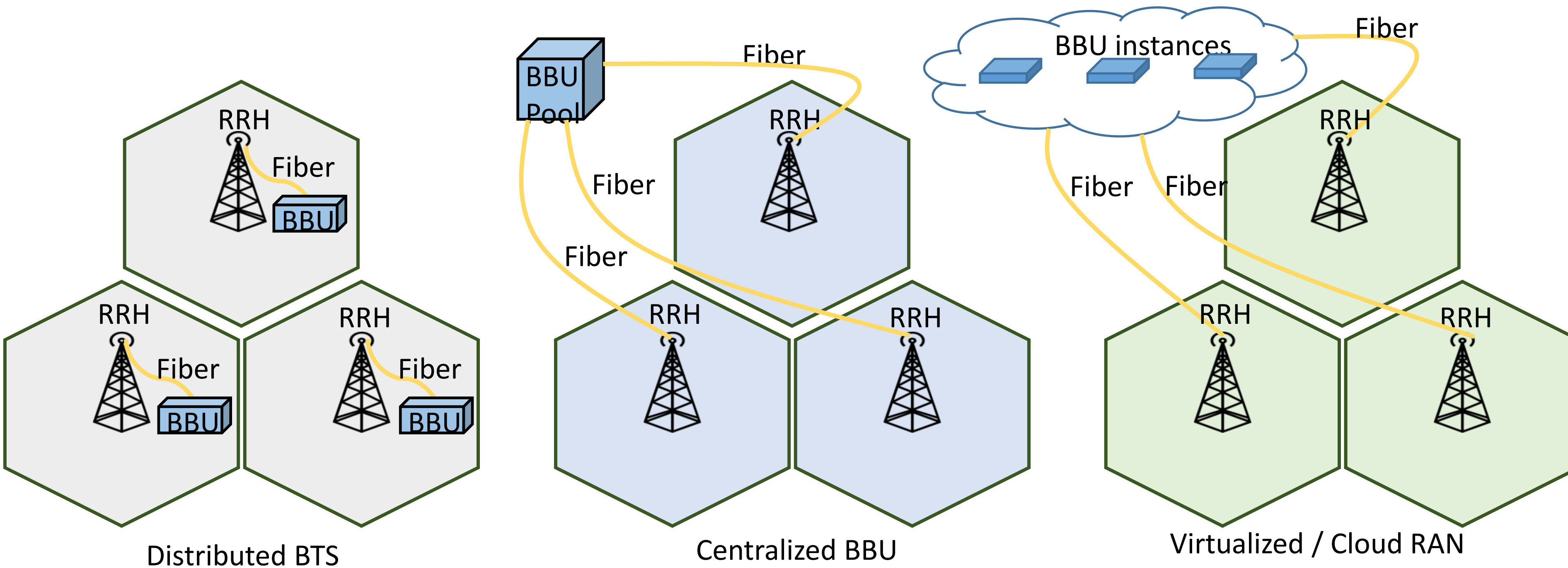
GP-Cloud computing vs C-RAN applications

	GP-Cloud Computing	Cloud RAN
Data rate	Mbps, bursty,	Gbps, stream
Latency / Jitter	Tens of ms	< 1, jitter in ns
Lifetime of data	Long	Extremely short
Number of clients	Millions	Thousands – Millons
Scalability	High	Low
Reliability	Redundancy, load balancing	Redundancy, Offloading / load balancing
Placement	Depends on the cost and performance	Specific areas
Time scale (operation, recovery)	Non-realtime	Realtime

Cloud RAN Primer

- **Main idea:**
  - Decouple the base station processing from the radio unit
  - Perform the processing at the high performance cloud infrastructure
  - Transport the data through a high speed medium
- **Components**
  - Remote radio head (RRH): lightweight (passive) radio element with power amplifier and antennas
  - Base band Unit (BBU): a centralized pool of virtualized base station covering a large set of cells (10 – 100)
  - Fronthaul (FH): data distribution channel between the BBU pool and RRHs

Cloudification of RAN

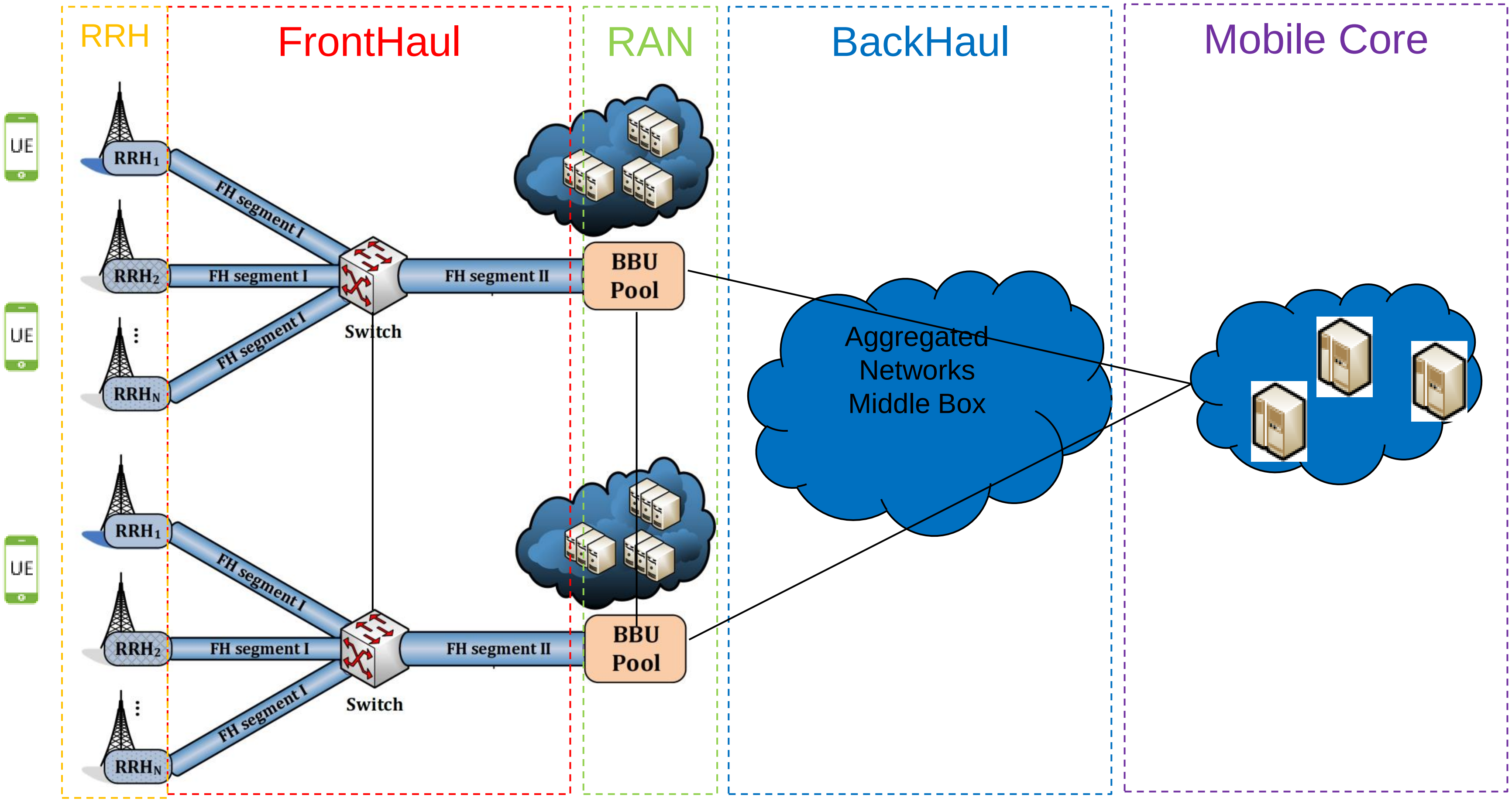


EURECOM

Comparison
Traditional BS, Distributed BS, and C-RAN

Architecture	Radio and BaseBand	Advantages	Drawbacks
Traditional BS	Co-located at the cell site In-BS processing	-	High power consumption Underutilized resources
Distributed BS	Split of BBU from RRH Group of RRH	Lower power consumptions Better placement of RRH	Underutilized resources
C-RAN	Split of BBU from RRH. Network of RRHs. Collocated BBUs	Even lower power consumption Better placement of BBU and RRH Lower the number of BBU Simpler network densification Rapid network deployment	Fronthaul Capacity requirement (non commodity)

Typical cloud RAN Deployment



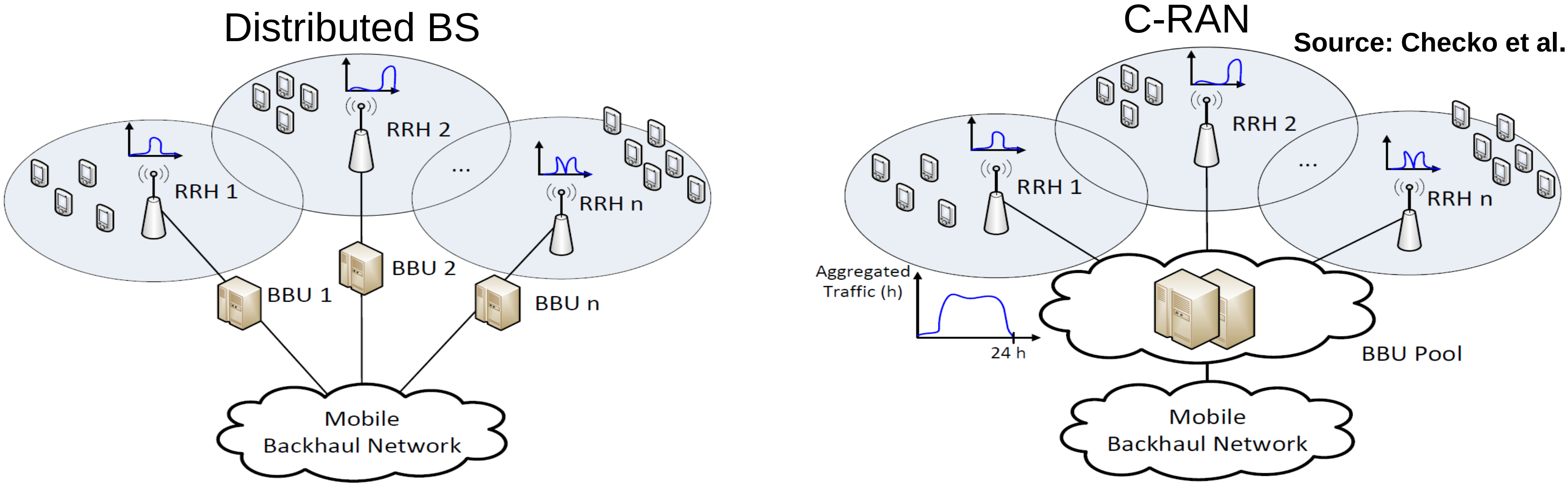
EURECOM

Benefit of a Cloudified RAN

- **Cooperation**
  - Coordinated signal processing
  - Joint scheduling
  - Interference management through channel feedbacks
- **Interconnections**
  - Maximize statistical multiplexing gain
  - Load balancing
- **Clustering**
  - RRH aggregation and assignment to BBU pools
  - Reduce the number of BBUS to save energy

Cloudified RAN Benefit

Adapt to spatio-temporal traffic fluctuation



Source: Checko et al.

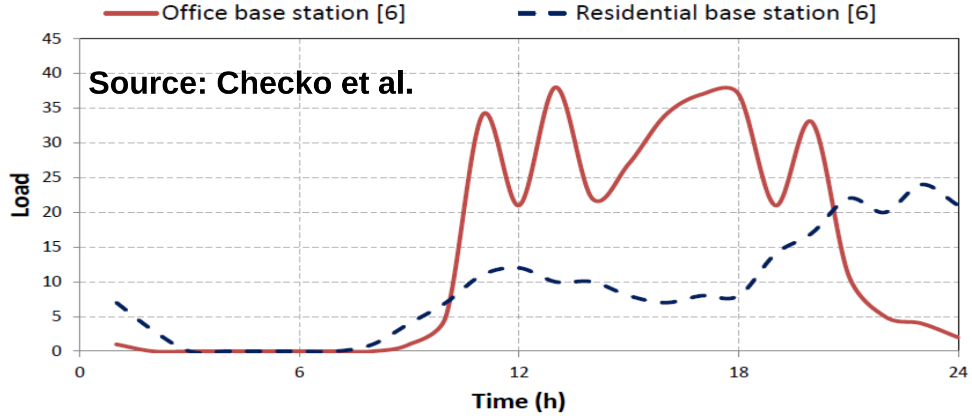
- **Statistical Multiplexing Gain**
- **Scalability**
  - Elasticity (scale up/down)
  - Workload sharing (scale in/out)

EURECOM

Cloudified RAN Benefit

Exploit workload variations through the statistical multiplexing gain among multiple RAN

- BS are often dimension for the peak traffic load!
- Peak traffic load →10x off-the-peak hours
- Exception: load-aware BS

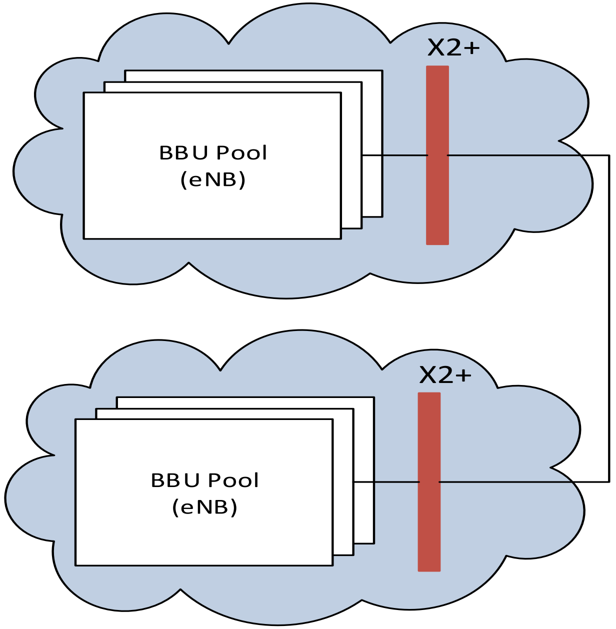


- Observation: Centralized BBUs' processing < Σ of BSs' processing
- Statistical multiplexing gain = $\frac{\Sigma \text{ of BSs' processing}}{\text{Centralized BBUs' processing}}$
- Gain: depends on traffic pattern, BBU to RRH mapping, BBU load balancing

Cloudified RAN Benefit

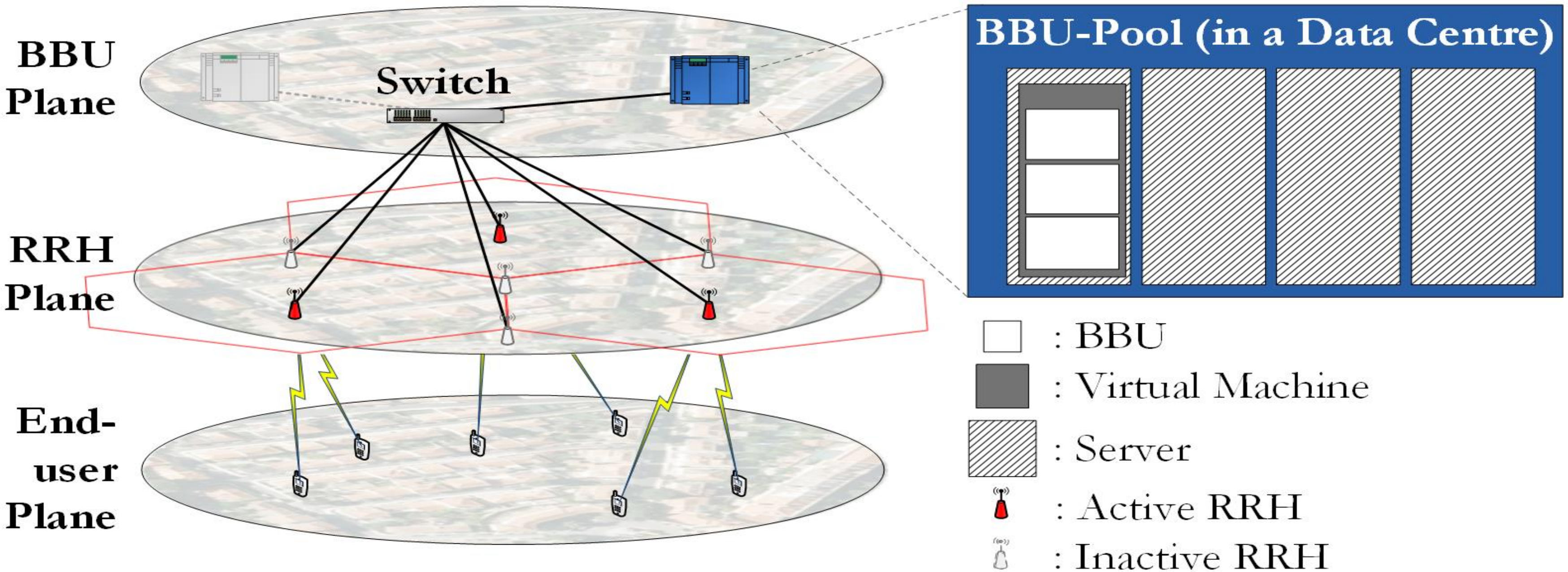
Improve of spectral efficiency (throughput, latency)

- **Centralization of BBU pool in C-RAN facilitates the inter BBU cooperation**
  - **Joint scheduling**
    - minimize inter cell interference (e.g. eICIC)
  - **Joint and coordinated signal processing**
    - utilize interference paths constructively (e.g. CoMP, MU-MIMO)
  - **Shared Context**
    - reduce control plane signaling delay (e.g. handover, co-scheduling via X2+)



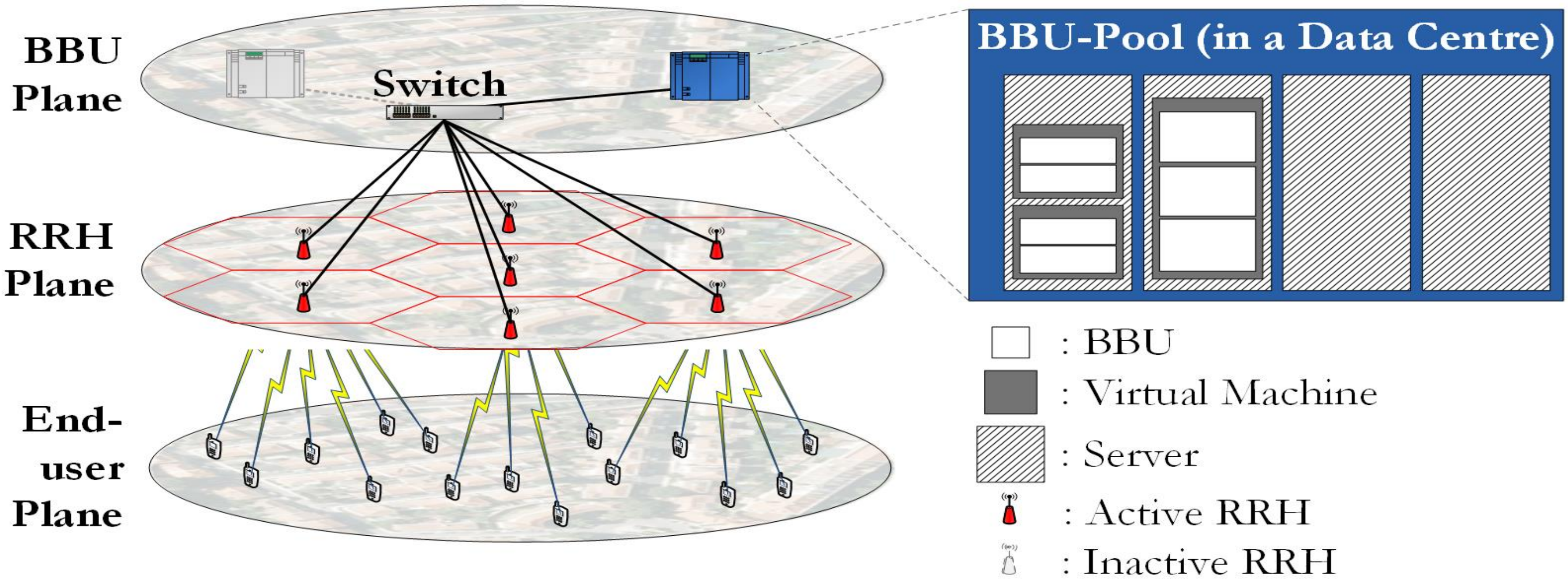
Cloud-RAN *Example*

- **With few users, 3 RRH-BBU pairs cover the service area and provide the requested capacity.**



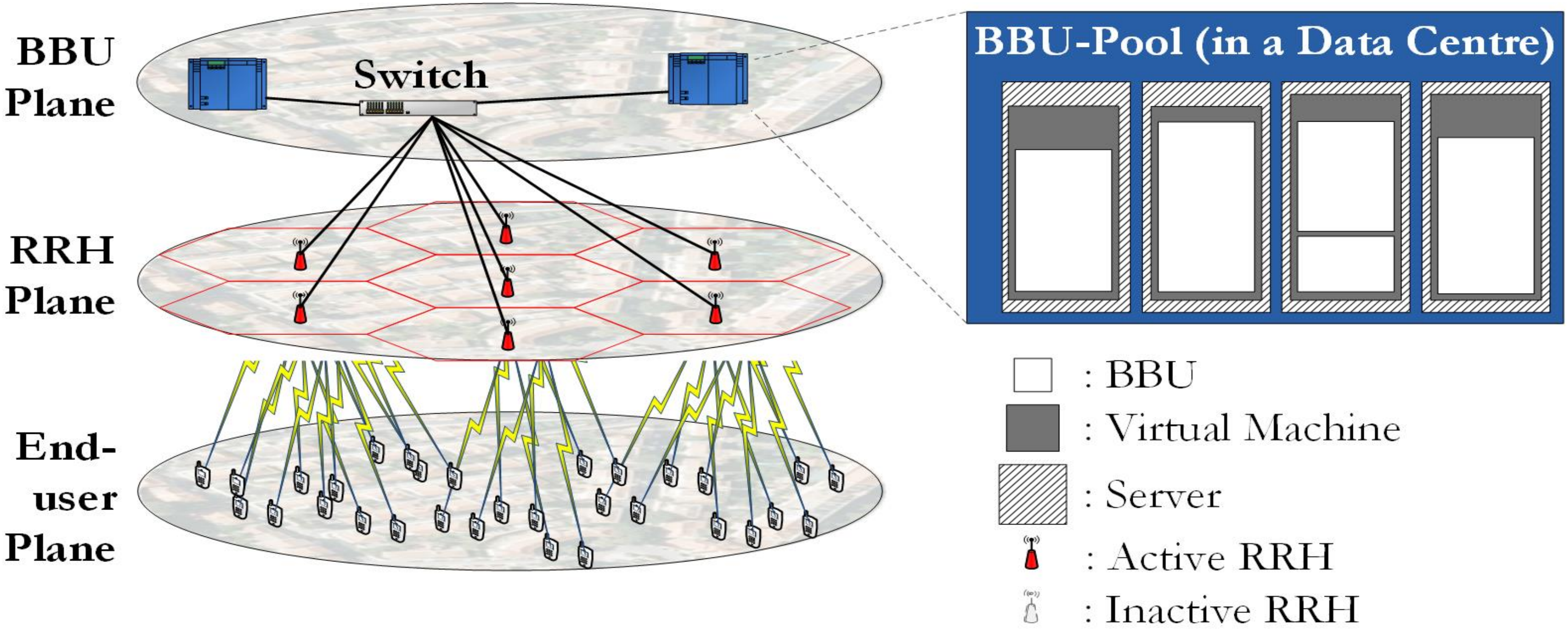
Cloud-RAN *Example*

- **With more users, extra RRHs are activated and BBUs instantiated, to provide the requested capacity.**



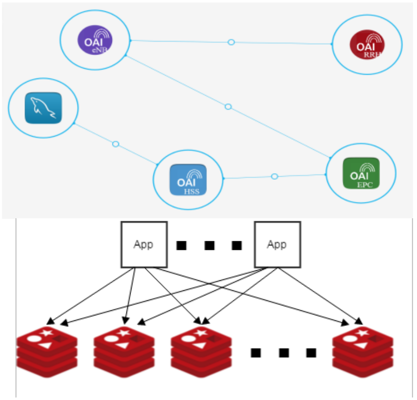
Cloud-RAN *Example*

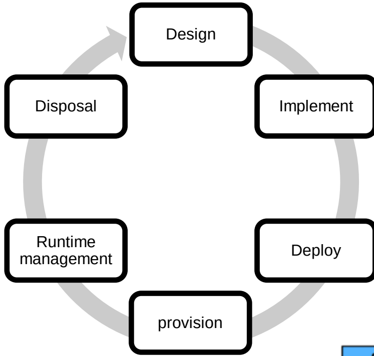
- **With more users, extra RRHs are activated and BBUs instantiated, to provide the requested capacity.**

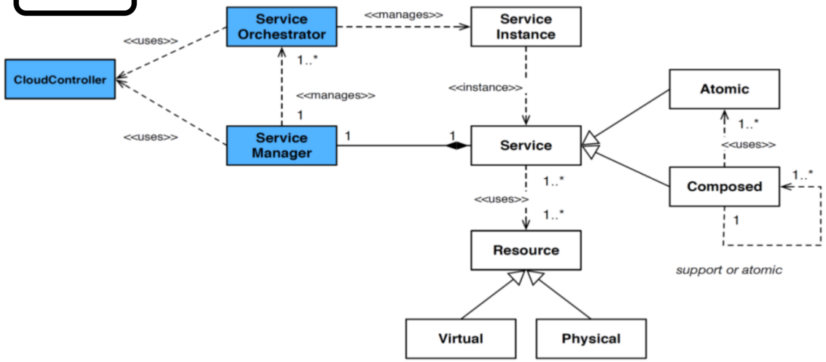


Cloud-Native RAN

- **Mircoservice Architecture along with NFV**
  - Flexible Functional split
  - Move form monolitic to a composed and metered service
  - Stateless, composable, reusable
- **Scalability**
  - Scale in and out, pay-as-you-go
- **Reliability**
  - Redundancy and stateless
- **Multitenancy**
  - Share the resources
  - (spectrum, radio, and infrastructure)
- **Placement**
  - Optimize the cost and performance
  - Supported Hardware, in particular for RAN
- **Realtime edge services**
  - Direct access to the radio information

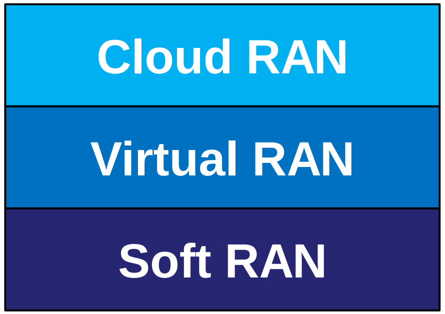


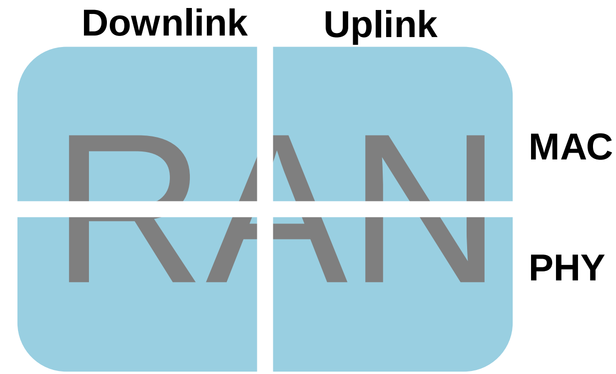




EURECOM

Cloud-RAN Challenges





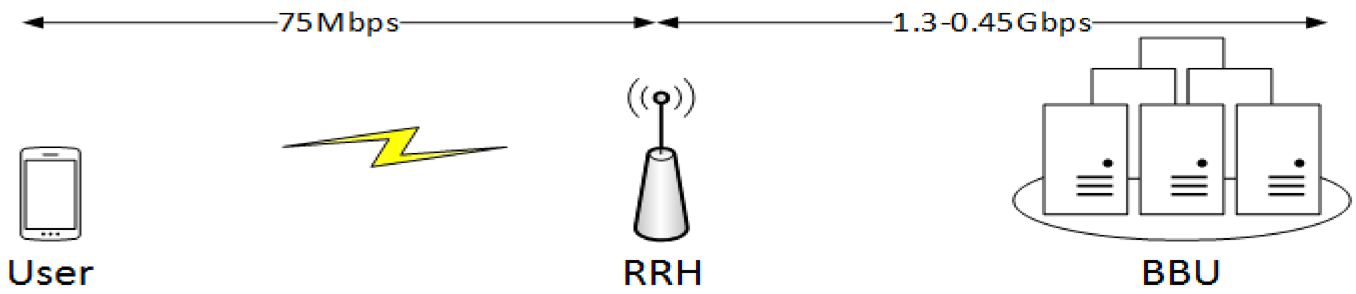
- **Capacity, latency, and jitter requirements for fronthaul**
- **BBU processing budget and protocol deadlines**
- **Realtime, virtualization environment and BBU performance**
- **Active RRH and Flexible Functions Split**
- **E2E Service modelling and template definition**
- **NFV Service manager and orchestrator**

Capacity, latency, and jitter requirements for fronthaul

- **Transport Network between RRH and BBU**
  - Dark fiber
  - WDM/OTN: Wavelength-division multiplexing (WDM)/Optical Transport Network (OTN)
  - Unified Fixed and Mobile access (microwave)
  - Carrier Ethernet
- **Protocols**
  - Common Public Radio Interface (CPRI)
  - Open Base Station Architecture Initiative (OBSAI)
  - Open Radio equipment Interface (ETSI-ORI)
- **Key requirements**
  - Supported Topology (star, ring, mesh), reliability, distance, multiplexing, capacity, scalability

Capacity, latency, and jitter requirements for fronthaul

- **Latency required by the HARQ RRT deadline**
  - 250 us maximum one-way latency adopted by NGMN, limiting the length of BBU-RRH within 20-40 Km (given that the speed of light in fiber is approximately 200×10^6 m/s)
- **Jitter required by advanced CoMP schemes**
  - <65 ns(MIMO, 36.104) timing accuracy in collaboration between base stations, which is the tightest constraint.
  - Frequency error < 50 ppb (macro BS)
  - BER < 10e-12
- **20MHz channel BW, SISO, 75 Mbps for users**
  - 2.6Gbps on Fronthaul without compression (0,87Gbps with 1/3)



EURECOM
Sophia Antipolis

Capacity, latency, and jitter requirements for fronthaul

$$C = 2 \cdot N_{Antenna} \cdot M_{Sector} \cdot F_{Sampling} \cdot W_{I/Q} \cdot C_{carriers} \cdot O_{coding+proto} \cdot K_{comp}$$

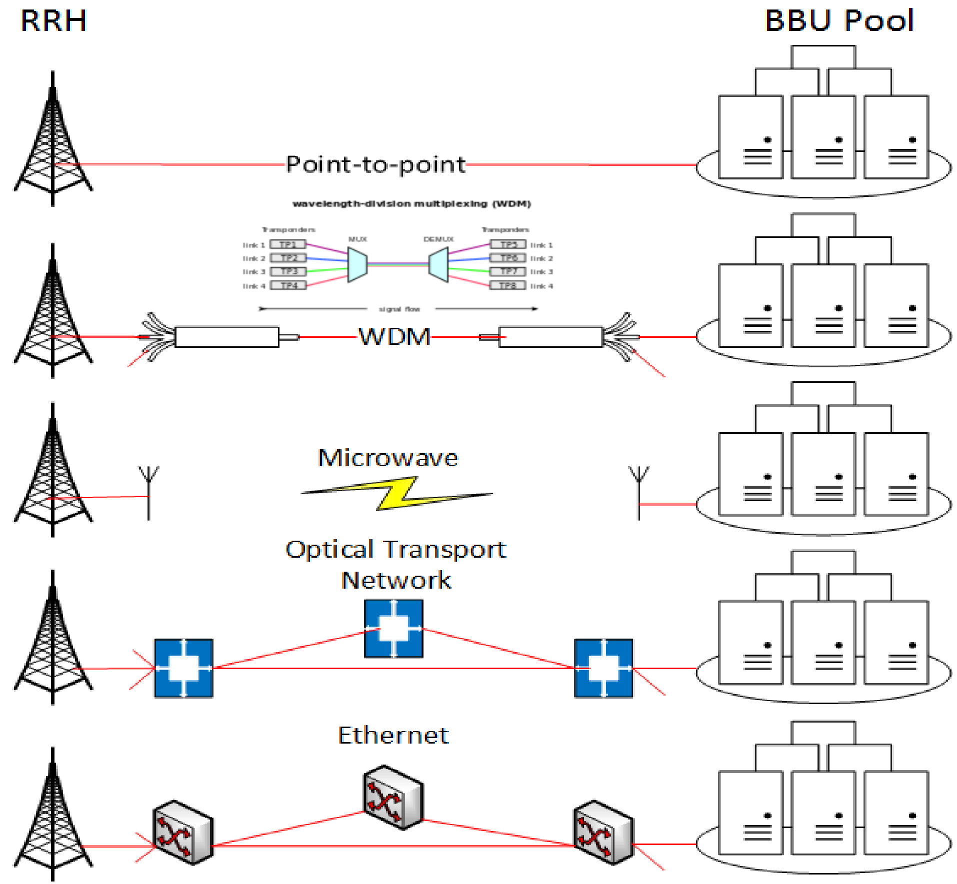
Bandwidth	$N_{antenna}$	M_{sector}	$F_{sampling}$	$W_{I/Q}$	$O_{coding+proto}$	$C_{Carriers}$	K	Data Rate
1.4MHz	1x1	1	1.92	32	1.33	1	1	163Mb/s
5MHz	1x1	1	7.68	32	1.33	1	1	650Mb/s
5MHz	2x2	1	7.68	32	1.33	1	1	1.3Mb/s
10MHz	4x4	1	15.36	32	1.33	1	1/2	2.6Gb/s
20MHz	1x1	1	30.72	32	1.33	1	1	2.6Gb/s
20MHz	4x4	1	30.72	32	1.33	1	1/3	3.4Gb/s
20MHz	4x4	1	30.72	32	1.33	1	1	10.4Gb/s

- **Costs**
  - Tens of BS over long distance → 100 Gbps
- **Savings**
  - Equipment's
  - Energy

Capacity, latency, and jitter requirements for fronthaul

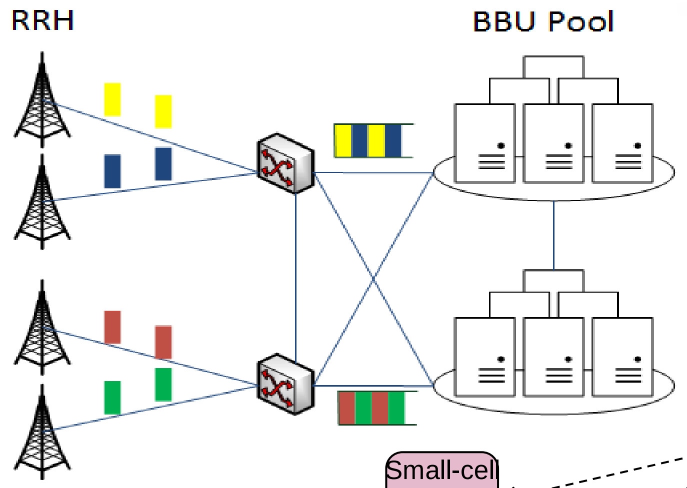
Medium	Bit rate	Distance	Remark
Fiber	100Gbps	~20Km	OTN: expensive
Copper	10Gpbs	100m	Low cost, SYNC
Wireless	1Gbps	2-15Km	LoS, high latency

- **Synchronization**
  - Frequency of transmission
  - Handover, coding
- **Solution**
  - GPS
  - PHY layer clock, SyncEth
  - Packet-based sync (IEEE 1588v2)

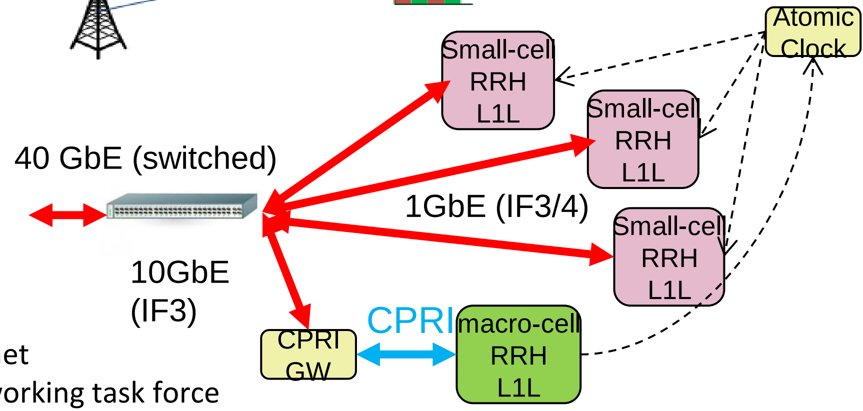


EURECOM

Capacity, latency, and jitter requirements for fronthaul



- **Asynchronous Ethernet**
  - Reduce the fronthaul capacity
  - I/Q transport over Ethernet
  - Some DSP in RRH to reduce transport speed/cost (split)
    - Decoupling of user-processing and cell-processing (iFFT/FFT)
- **Advantages**
  - Cost saving (reuse, commodity hardware)
  - Switching (packet-based)
  - Multiplexing / load balancing
  - Flexible topology (mesh)
- **Challenges**
  - Distributed computation
  - Cheap synchronization ((GPS, 1588v2)
  - Real-time I/Q over Eth links (copper, low-cost fiber)

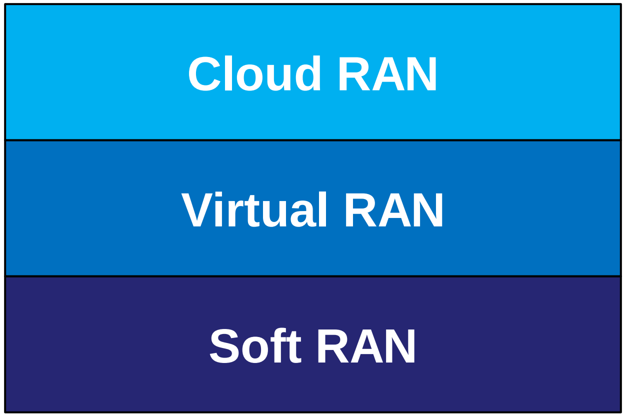


- **Hot topics**
  - IEEE 1904.3 - encapsulation and mapping of IQ data over Ethernet
  - IEEE 802.1 – CPRI fronthaul discussion with Time Sensitive Networking task force
  - CPRI → CPRI2?
  - 3GPP - proposal on a study item on variable rate multi-point to multi-point packet-based fronthaul interface supporting load balancing

EURECOM

Soft RAN
BBU processing budget

- **4G Feasible on General Purpose Processors (x86)**
- **An eNB is approximately 1-2 x86 cores on Gen 3 Xeon silicon**
  - Perhaps more power efficient solutions from TI, Freescale or Qualcomm
  - But: lose commodity software environment and common HW platform to high-layer protocols and cloud



Soft RAN
BBU processing budget for peak rate

eNB Rx stats (1subframe)

- OFDM demod : 109.695927 us
- ULSCH demod: 198.603526 us
- ULSCH Decoding : 624.602407 us

➔ 931 us (<1 core)

eNB Tx stats (1 subframe)

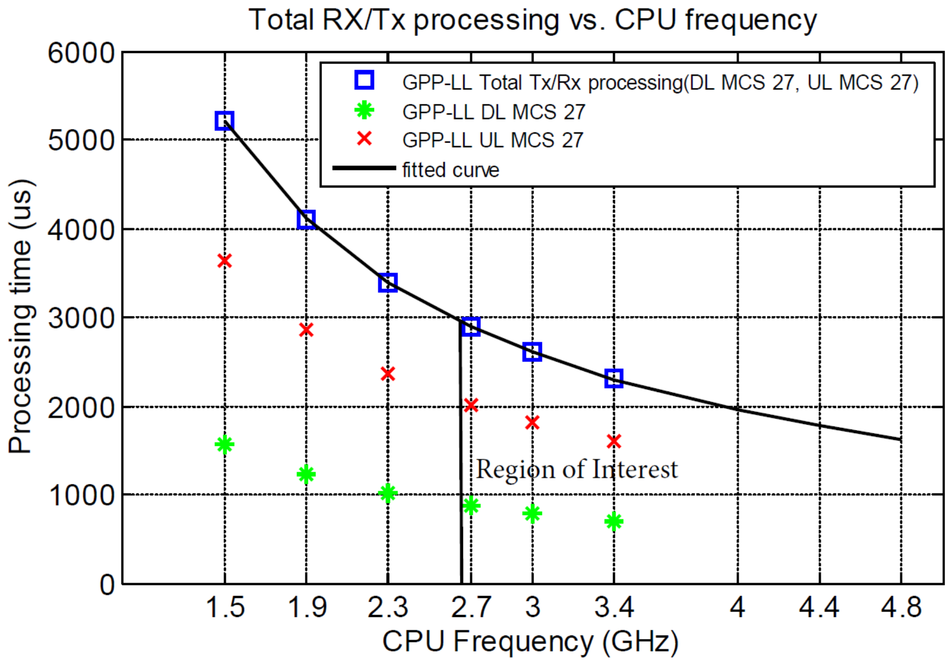
- OFDM mod : 108.308182 us
- DLSCH mod : 176.487999 us
- DLSCH scrambling : 123.744984 us
- DLSCH encoding : 323.395231 us

➔ 730 us (< 1core)

- **Efficient base band unit is challenging**
- **With AVX2 (256-bit SIMD), turbo decoding and FFT processing will be exactly twice as fast**
  - <1 core per eNB
  - .4 core per eNB without TC ← can this be exploited efficiently with HW acceleration? (Solution adopted in China Mobile CRAN project, offload of TC on Altera FPGA)
- **Configuration**
  - gcc 4.7.3, x86-64 (3 GHz Xeon E5-2690),
  - 20 MHz bandwidth (UL mcs16 – 16QAM, DL mcs 27 – 64QAM, transmission mode 1 - SISO)
  - 1000 frames, AWGN channel

Soft-RAN
Processing Budget for Peak Rate

- **Processing time reduces with the increase of CPU Freq.**
- **min CPU Freq is 2.7GHz**
  - HARQ deadline
- **$T_{subframe}$ = α/ x,**
  - α =8000
  - x is the CPU freq GHZ



- **Note: FDD LTE HARQ requires a round trip time (RTT) of 8ms**
  - $Tx+RX \leq Tharq/2-(acquisation+transport+offset) \approx 3ms$
  - ~2ms RX and 1ms TX (can't be fully parallelized)

Soft RAN Considerations

- **Key Consideration to meet the deadlines (SF, protocol)**
  - Real-time OS (linux with deadline scheduler) and optimized BIOS
    - Problem: OS scheduler latency (kernel is not pre-emptible)
  - Real-time data acquisition to PC
  - SIMD optimized integer DSP (SSE4, AVX2)
  - Parallelism (SMP)
  - x86-64
    - more efficient for Turbo decoding because of the number of available registers is doubled
- **Remove bottlenecks with**
  - hardware accelerators or hybrid CPUs
    - Turbo decoders (easily offloaded to FPGA-based accelerators), FFT, PDCP (de)enryption
  - GPUs or Xeon PHY-type devices
    - Perhaps interesting for Turbo-decoders and encoders than FFT
    - Main issue in both FPGA/GPU offloading
      - High-speed low-latency bus between CPU memory and external processing units

Realtime, virtualization environment and BBU performance RTOS issues

- **Low-latency radio applications for PHY (e.g. 802.11x,LTE) should run under an RTOS**
  - Meet strict hard deadline to maintain the frame/subframe and protocol timing
  - efficient/elastic computational resources (e.g. CPU, memory, network)
- **Example OS**
  - eCos/MutexH for generic GNU environment
  - RTAI for x86
  - VXWorks ($$$)
- **Example: RTAI / RT-PREEMPT kernel can achieve worst-case latencies below 30μs on a loaded-PC. More than good enough for LTE, but not for 802.11x because of MAC timing.**
- **Should make use of POSIX multithreading for SMP**
  - Rich open-source tool chains for such environments (Linux, BSD, etc.)
  - Simple to simulate on GNU-based systems for validation in user-space
  - Allow each radio instance to use multiple threads on common HW

Realtime, virtualization environment and BBU performance
Issues with standard Linux Kernels

- **Scheduler latency**
  - Kernel is not pre-emptible
  - Overhead in disabling/enabling interrupts
- **Mainstream kernel solutions, the RT-Preempt patch and out-of-the-box Linux kernel (>3.14) converts Linux into a fully preemptible kernel**
  - Kernel preemption (RT-PREEMPT) – mainstream until 2.6.32 (patches afterwards)
  - Latency reduction (soft-RT kernels) with DEADLINE_SCHED
    - ☞ Version >3.14
- **Patches / dual-OS solution**
  - ADEOS + RTAI/Xenomai

EURECOM Sophia Antipolis

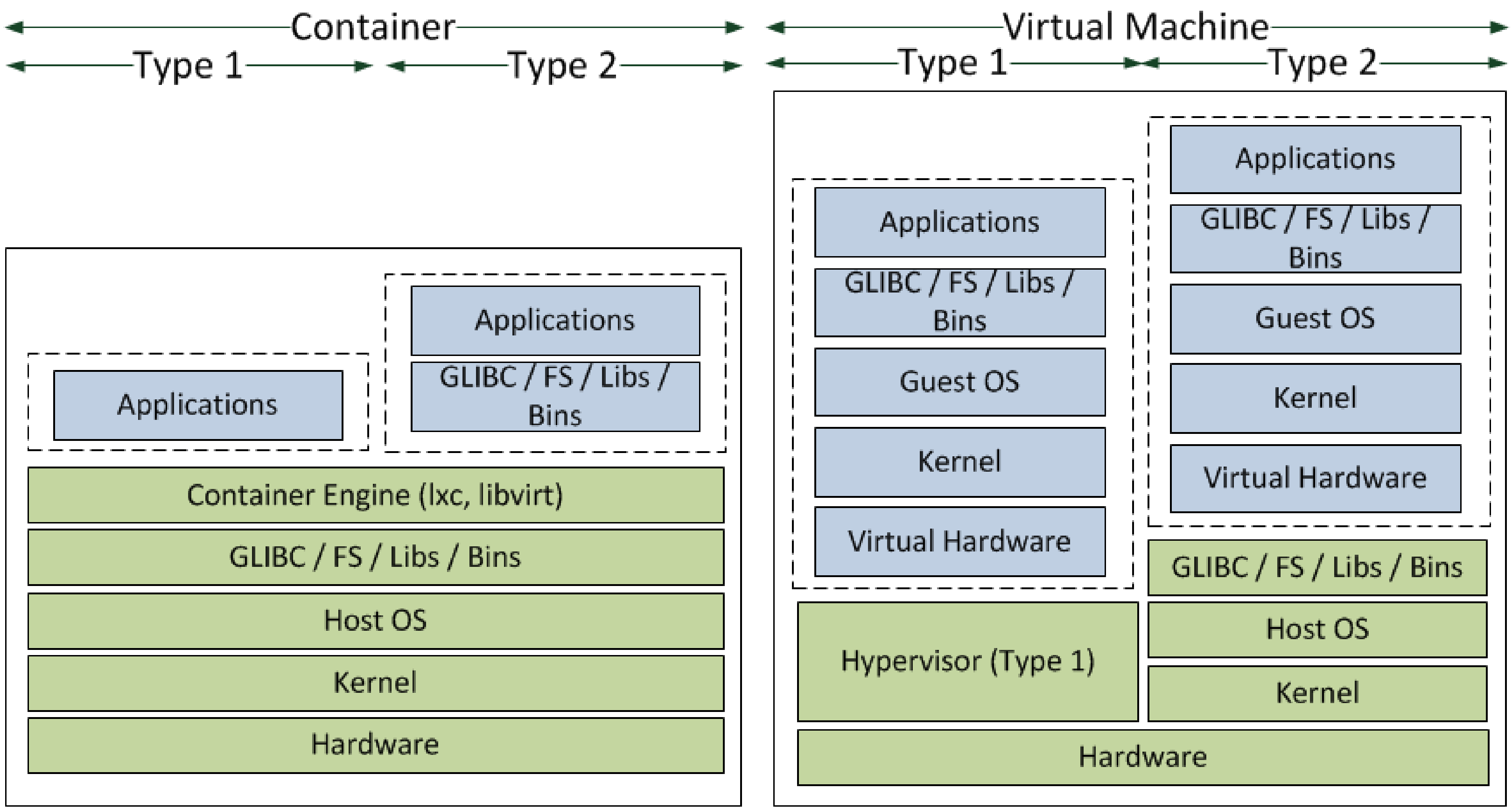
Realtime, Virtualization environment and BBU performance

- **Virtual Machine (VM) – e.g., KVM:**
  - A complete OS is deployed as a guest
  - Virtualisation layer that emulates physical resources
  - Hypervisor that manages requests for CPU, memory, hard disk, network and other hardware resources
- **Virtualisation Environment (VE) – e.g., LXC and Docker:**
  - No hardware emulation nor hypervisor and guest OS (containers).
  - Use and share the OS and potentially device drivers of the host
  - OS scheduler manages the request for physical resources

Realtime, Virtualization environment and BBU performance

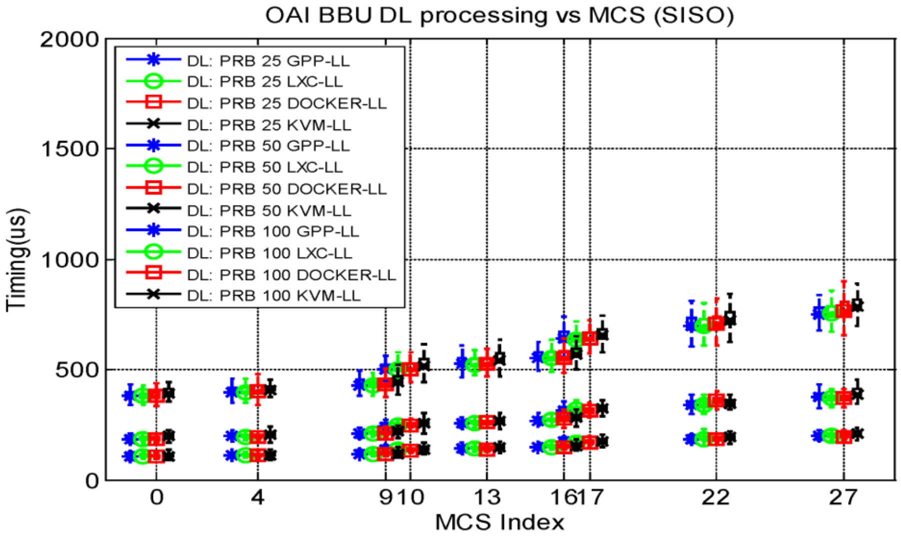
- **General Purpose Platform (GPP)**
  - dedicated machine.
- **Kernel-based Virtual Machine (KVM)**
  - Linux virtualisation infrastructure that turns it into a hypervisor.
- **Linux Container (LXC)**
  - operating-system-level capability for running isolated Linux Virtual Environments (VE) on a single control host.
- **Docker**
  - LXC-based portable container engine that encapsulates an application with all its dependencies.
- **Options:**
  - low latency kernel, prioritization of processes.

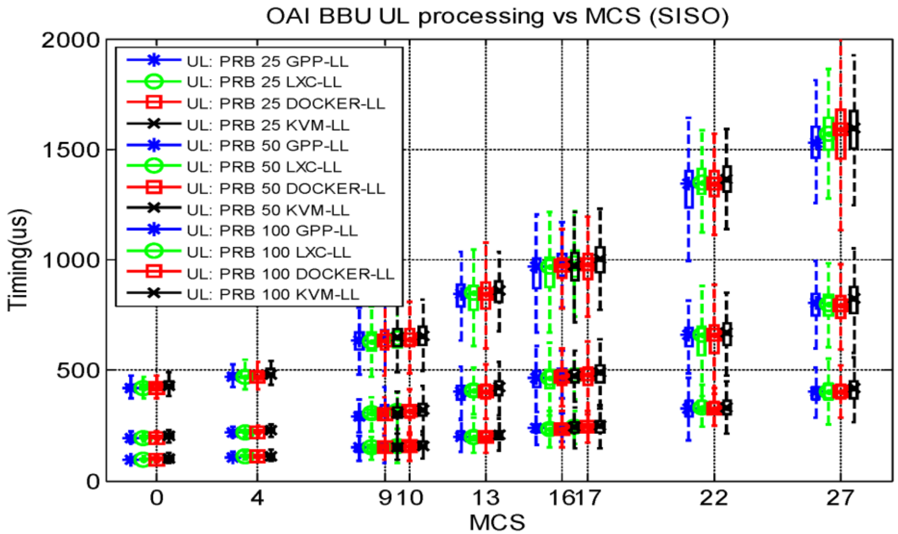
Realtime, Virtualization environment and BBU performance



Virtual-RAN
Processing Budget for Peak Rate

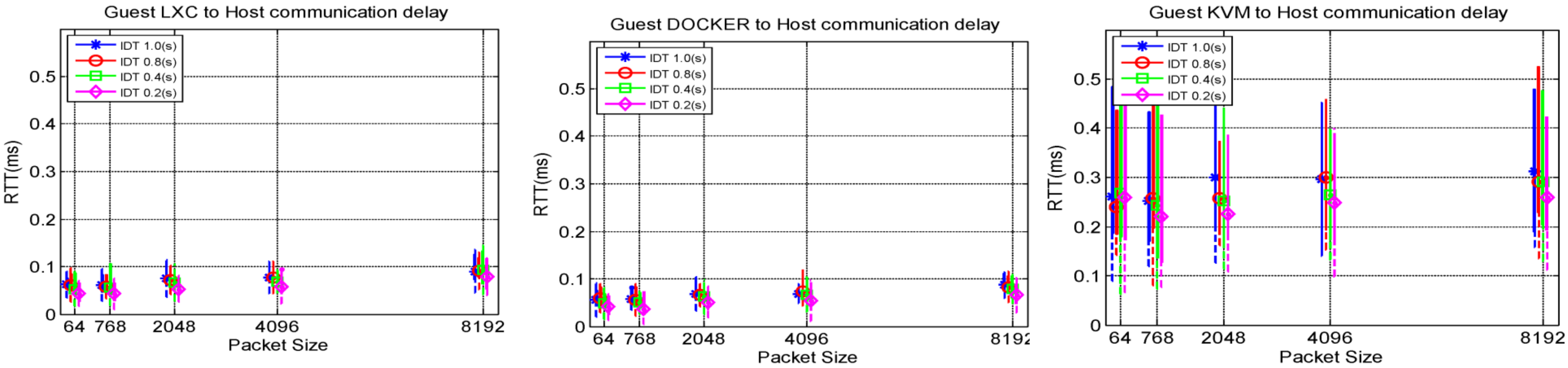
- **DL and UL BBU processing load for various MCS, PRB, and virtualization flavor**
  - Comparable BBU Processing time





Virtual-RAN Additional Consideration

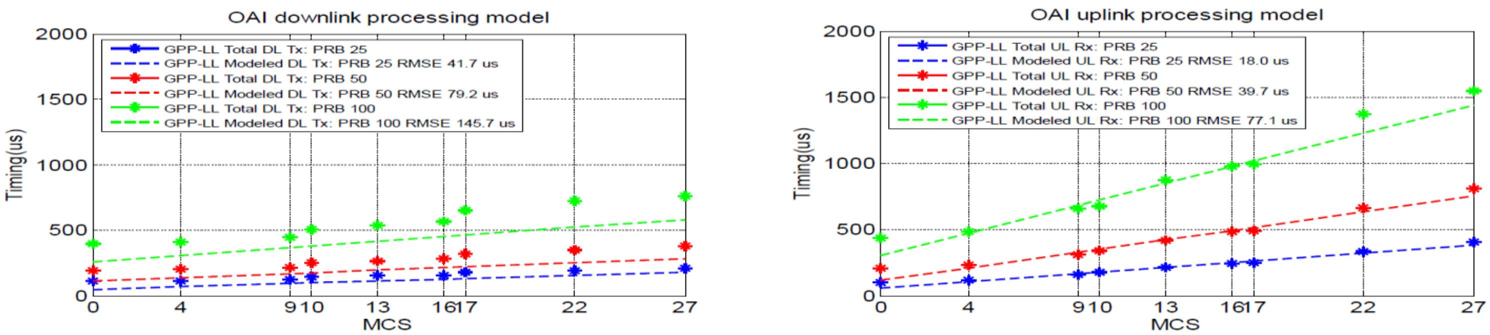
- **I/O access delay**
  - RF, ETH, and HW accelerator
  - RF Passthrough vs Hardware virtualization (and sharing)
  - Delay and jitter requirement on the fronthaul network
- **Limitation of the guest-only network data rate**



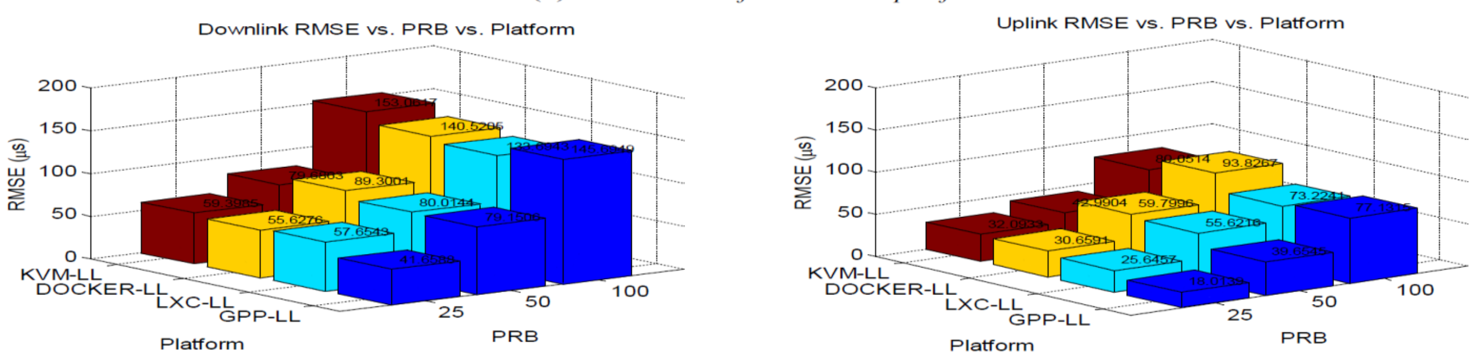
EURECOM Sophia Antipolis

Virtual-RAN
Modelling Processing Budget

- $T_{subframe} = c[x] + p[w] + u_s(x,y) + u_r(x), where$
- $u_s(x,y) = a(x).y + b(x), and\ x\ \in PRB, y\ \in MCS,\ w\ \in VE$



(a) *Fitted curves for GPP-LL platform*



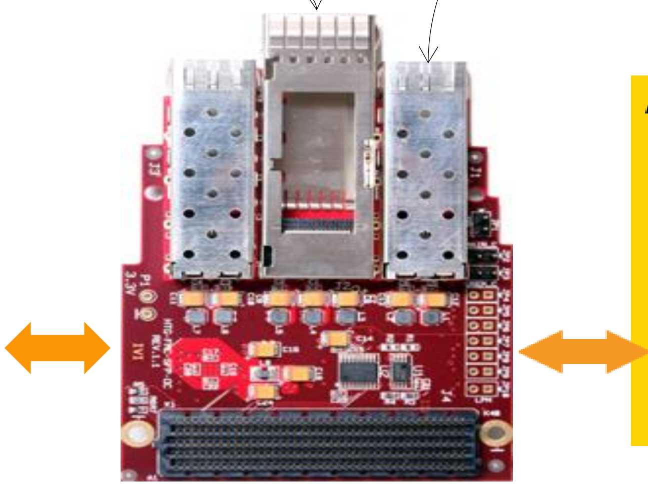
EURECOM

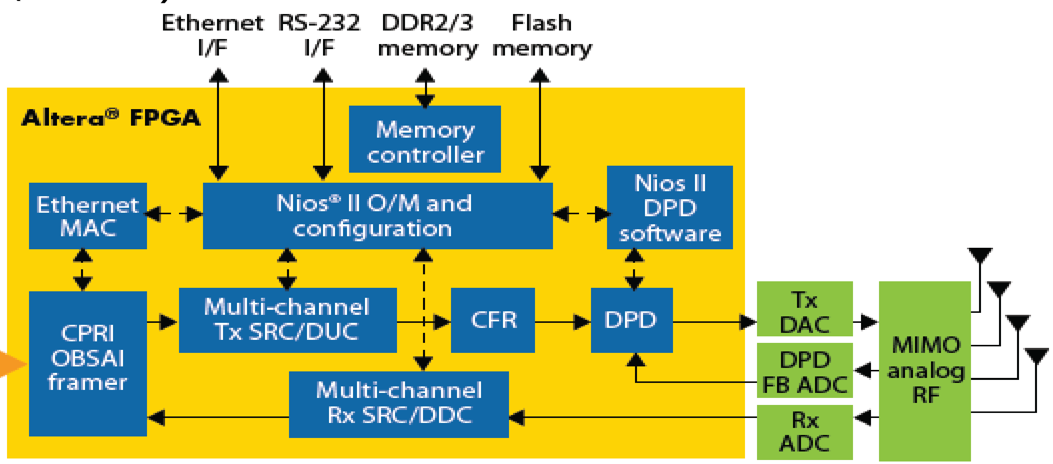
Cloud RAN
CPRI-based RRH

- **CPRI is**
  - A *synchronous protocol* for high-speed transport of I/Q baseband signals between BBU and RRH
    - Uses Gigabit ethernet-like (10,40,later 100) physical links based on 122.88 MHz clock and optical transport (for 40,100)
    - Line rates up to 9.8 Gbit/s (20 MHz antenna port ≈ 1.2 Gbit/s bi-directional)
    - All RRH are driven by common clock from BBU => tight synchronization in time/frequency is possible
    - Framing is scalable to allow for different number of antennas and channel bandwidths
  - I/Q transfer is standardized and flexible (number of bits, sampling rate, etc.)
  - RF control allows for proprietary signaling to control RF (biggest issue for developers in order to adapt to different RRH vendors)

Cloud RAN
CPRI-based RRH

- **CPRI-based RRH are usually built using FPGA (Xilinx/Altera) platform with small embedded system**
  - Coupled with RF cleanup (upsampling/downsampling filters, TX predistortion)





EURECOM Sophia Antipolis

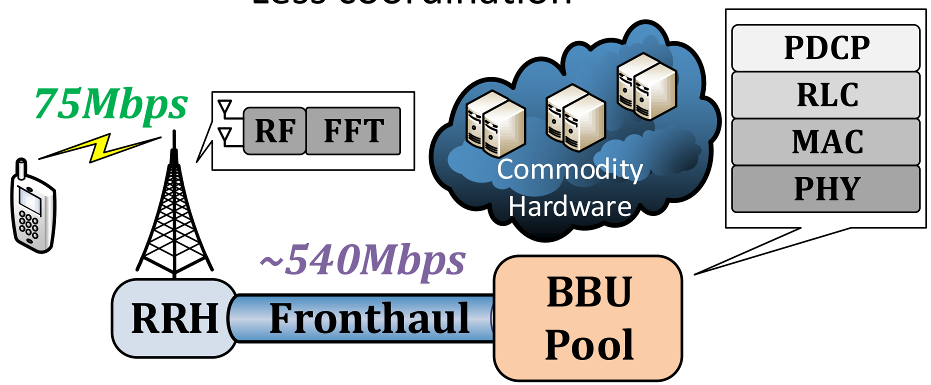
Cloud RAN
Active RRH and Ethernet Frontahul

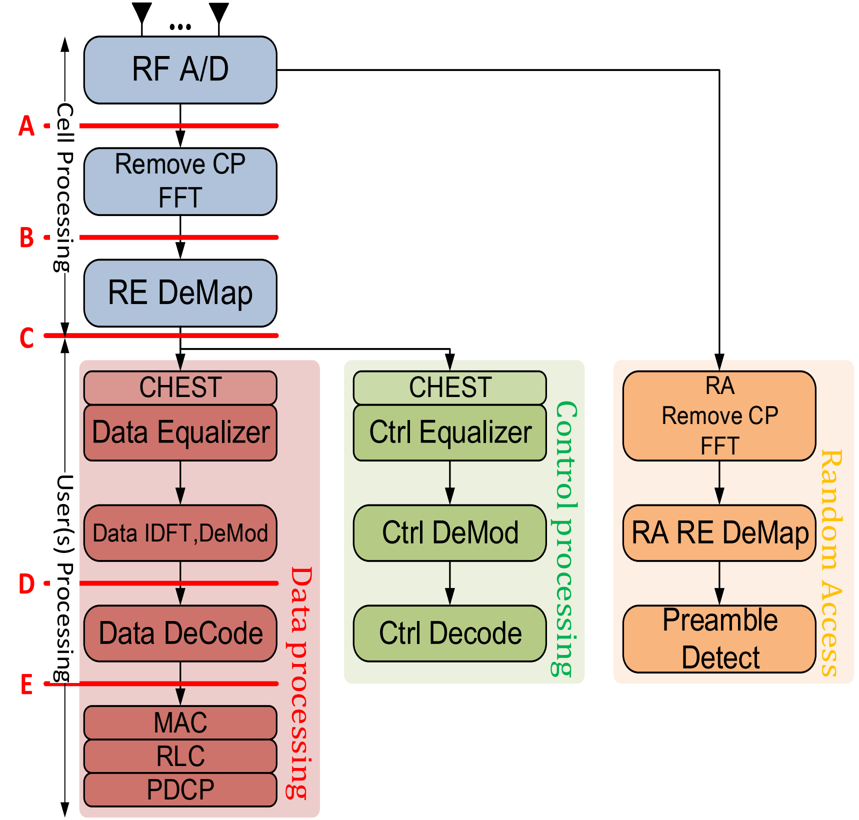
- **CPRI-gateways (switches)**
  - One end is Ethernet (connection with BBU-pool) other is CPRI for commercial RRH
  - Possibility to use a CPRI-GW to deliver synchronous I/Q to group of RRH (P2MP or multi-hop) from a common Atomic reference and provide generic Ethernet to BBU-pool
- **“Cheap” RRH (e.g. large indoor networks)**
  - Regular Ethernet or (syncE) +1588v2 (even copper!)
  - Low-power (<20W), cheap I/Q transport to BBU (i.e. not CPRI) with copper or cheap-fiber Ethernet
  - Some DSP in RRH to reduce transport speed/cost
  - Low-cost RF (e.g. Existing Lime microsystems-based PCIe solution)
  - Open architecture synchronization solution
- **BBU is slave to network of RRH**

Cloud RAN

Active RRH and Flexible Functions Split

- **Place more BBU processing at the edge of the network**
  - Reduce FH capacity requirement
  - Add FFT and remove CP at RRH almost halves the FH bandwidth
    - From 1Gbps to 540Mbps
  - However, some disadvantage…
    - Expensive RRHs
    - Less coordination





EURECOM

Cloud RAN

Active RRH and Flexible Functions Split

- China Mobile/NGFI approach
- Similar to small-cell forum

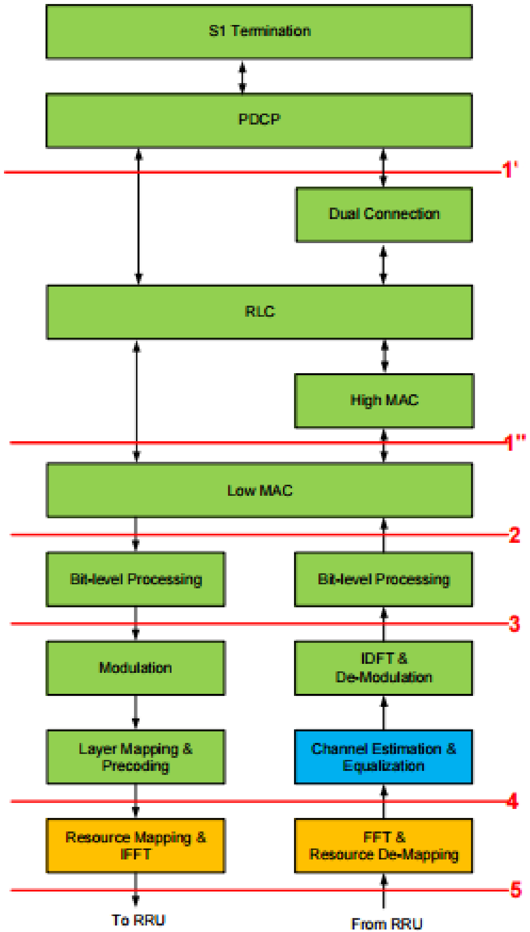


Table 3-1: Maximum Interface Bandwidth

	Interface 1		Interface 2		Interface 3		Interface 4		Interface 5	
	Bandwidth	Ratio	Bandwidth	Ratio	Bandwidth	Ratio	Bandwidth	Ratio	Bandwidth	Ratio
Downlink	174 Mb/s	1	179.2 Mb/s	1	125.2 Mb/s	1	498 Mb/s	3	9,830.4 MB/s	66
Uplink	99 Mb/s	1	78.6 Mb/s	1	464.6 Mb/s	6	2,689.2 Mb/s	36	9,830.4 MB/s	131

Table 3-2: Interface Delay

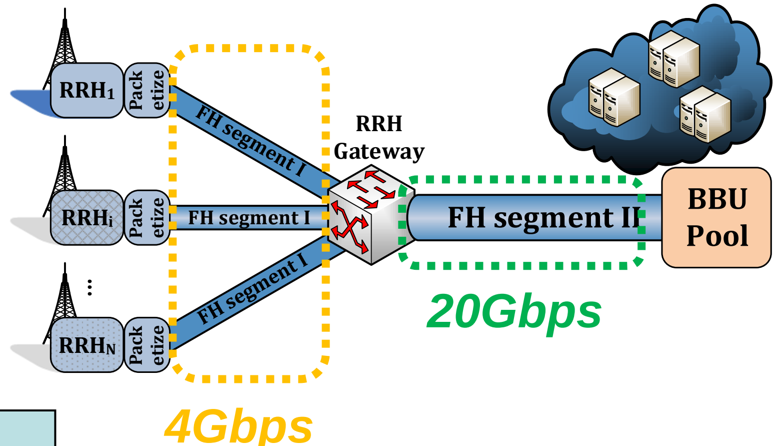
Interface 1		Interface 2		Interface 3		Interface 4		Interface 5	
Delay	Ratio	Delay	Ratio	Delay	Ratio	Delay	Ratio	Delay	Ratio
Less than 100 ms	1	Less than 1 ms	100	Less than 1 ms	100	Less than 1 ms	100	Less than 1 ms	100

Cloud RAN

Where to split?

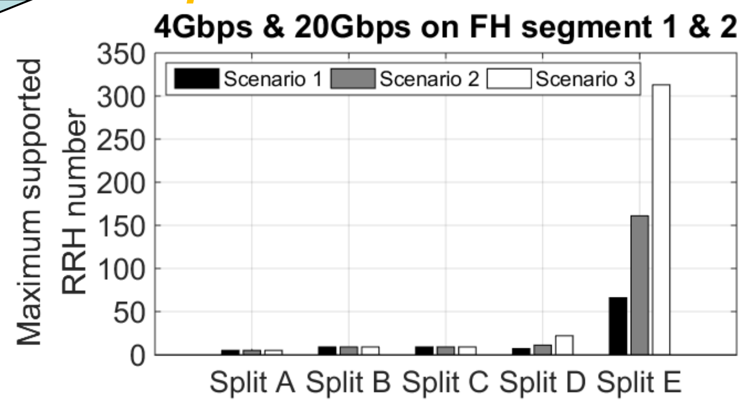
- **Derive maximum supported RRHs based on achievable peak-rate**

Scenario	1	2	3
Bandwidth	20 MHz		
Oversampling Ratio	1		
Rx Antennas	4		
Cyclic prefix length	Normal		
MIMO	4 Layer		
PUCCH RB	4		
SRS BW Config	7		
SRS SF Config	9		
Control Overhead	4.3%		
RA Config	0		
RA Overhead	0.3%		
Modulation	64 QAM	16 QAM	QPSK
TBS index	26	16	9
Time sample bitwidth	16		
Frequency sample bitwidth	16		
LLR bitwidth	8		



Based on achievable peak-rate on all RRHs

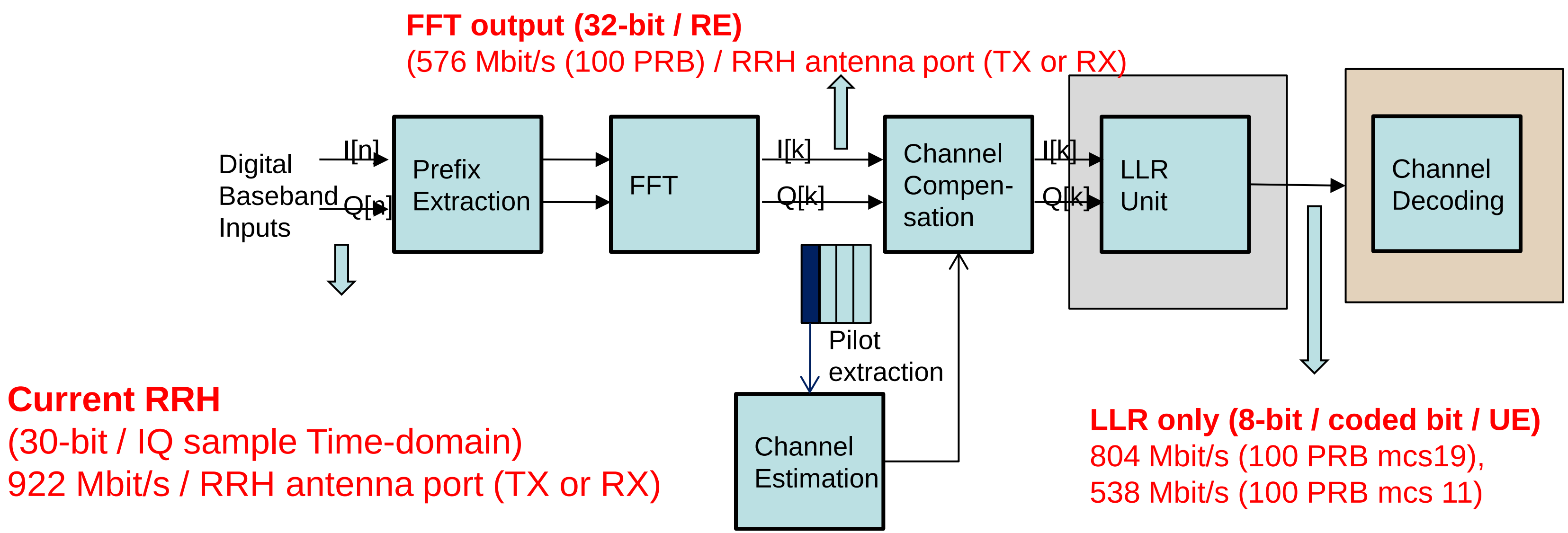
Scenario	1	2	3
Split A	5		
Split B	8		
Split C	9		
Split D	7	11	22
Split E	66	161	313



EURECOM

Cloud RAN
Where to split?

- **TX**
  - Full L1 TX in RRH
  - MAC (scheduler) must provide
    - Transport channel SDUs (common and dedicated)
    - Any precoding information for TM7-10
- **RX split is still under investigation**
  - Depends on number of UEs / RE / RRH (i.e. MU detection per RE)
  - And on models for realistic uplink resources (average MCS) in dense deployments



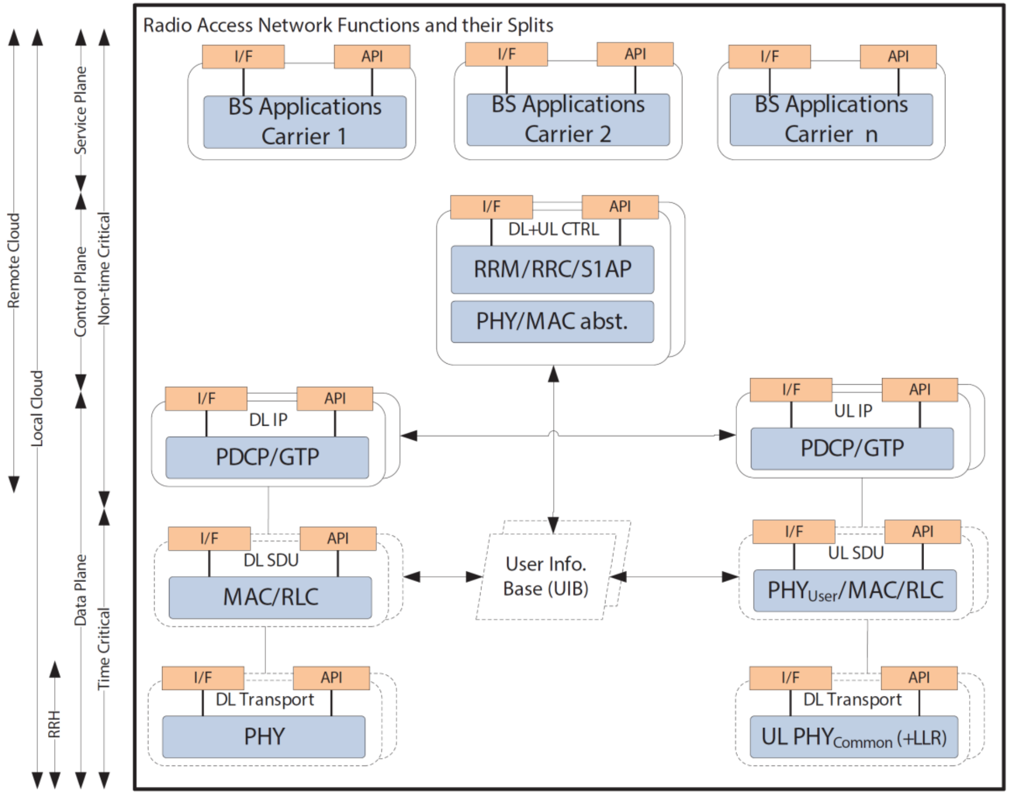
EURECOM

Cloud-RAN
Where to split?

- **It is not clearly evident that transport of the quantized LLR provides significant savings**
- **However**
  - The assumption here is that no further compression is required
    - Because of the quasi-discrete nature of the LLRs, further compression could bring savings
  - If compression can bring us below 8-bit/coded bit/UE then
  - Also, we can trade-off some performance by quantizing LLRs to 4-bits, then there would be significant fronthaul savings
- **TX fronthaul rates can be significantly reduced if baseband TX is performed in RRH**
  - Could be interesting for densely-deployed DL-only RRH

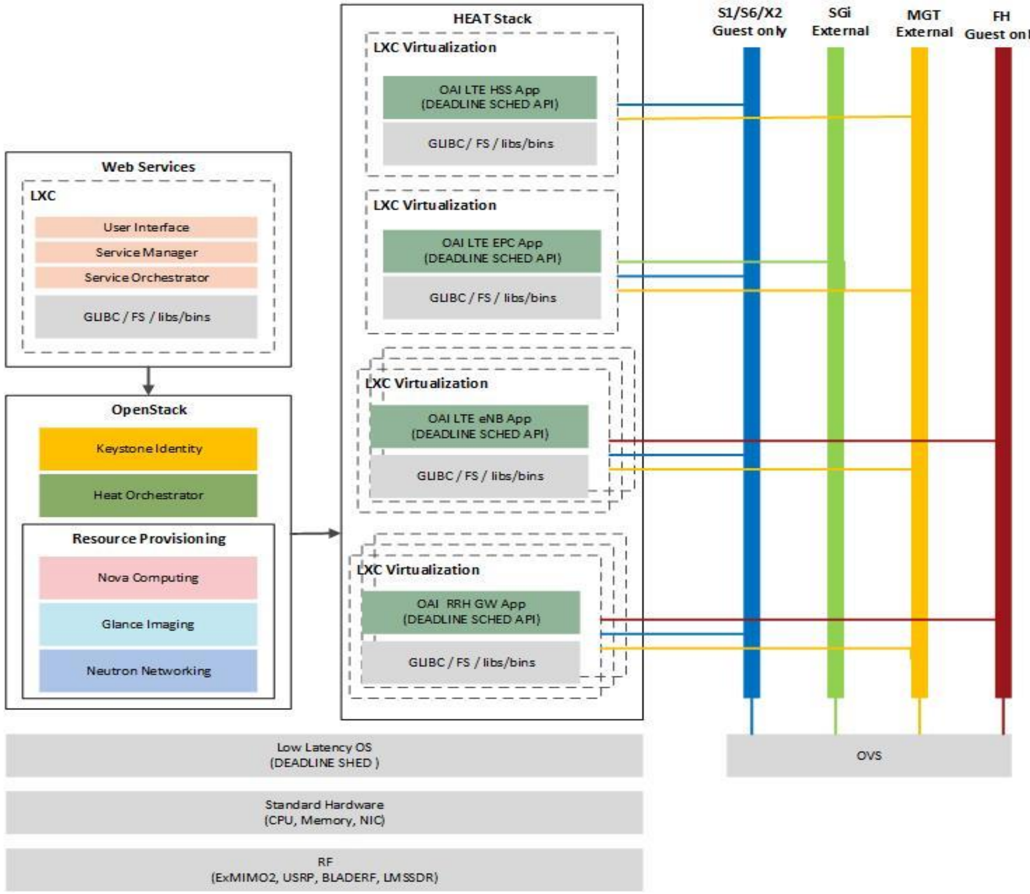
Cloud-RAN
Where to split?

- **RRC and MME Placement**
- **PDCP as a convergent layer**
- **PHY_{user} as a variable**
  - W and W/O MAC/RLC
- **Allow split across RRH, local, and remote cloud**
- **I/F**
  - Orchestration logic
- **API**
  - Controller logic



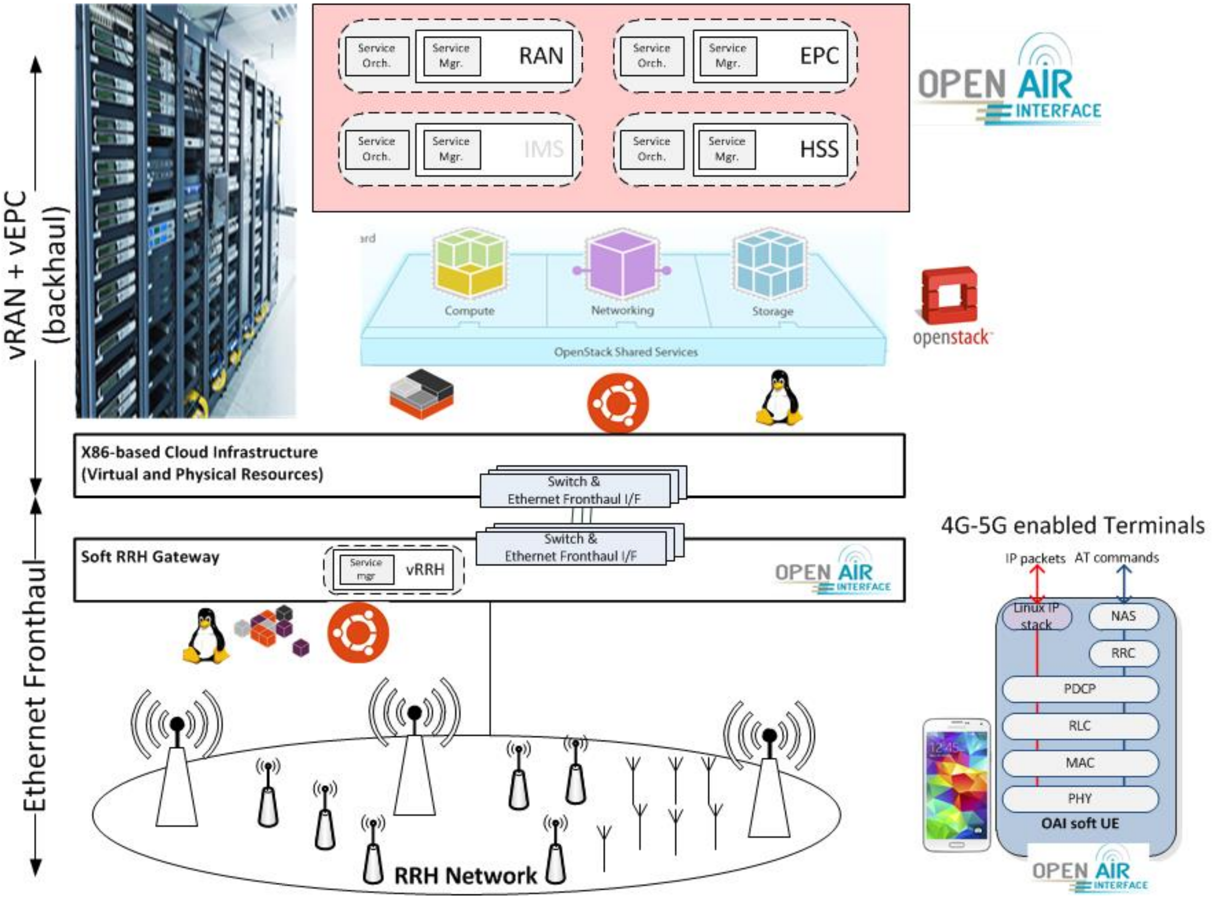
C-RAN Testbed on Sophia Antipolis Campus

- **Three components**
  - web service
  - OpenStack
  - Heat stack
- **Heat Template describes the virtual network deployment**
- **Linux Container**
- **Open vSwitch**
- **Low latency kernel**
- **RF frontend HW**



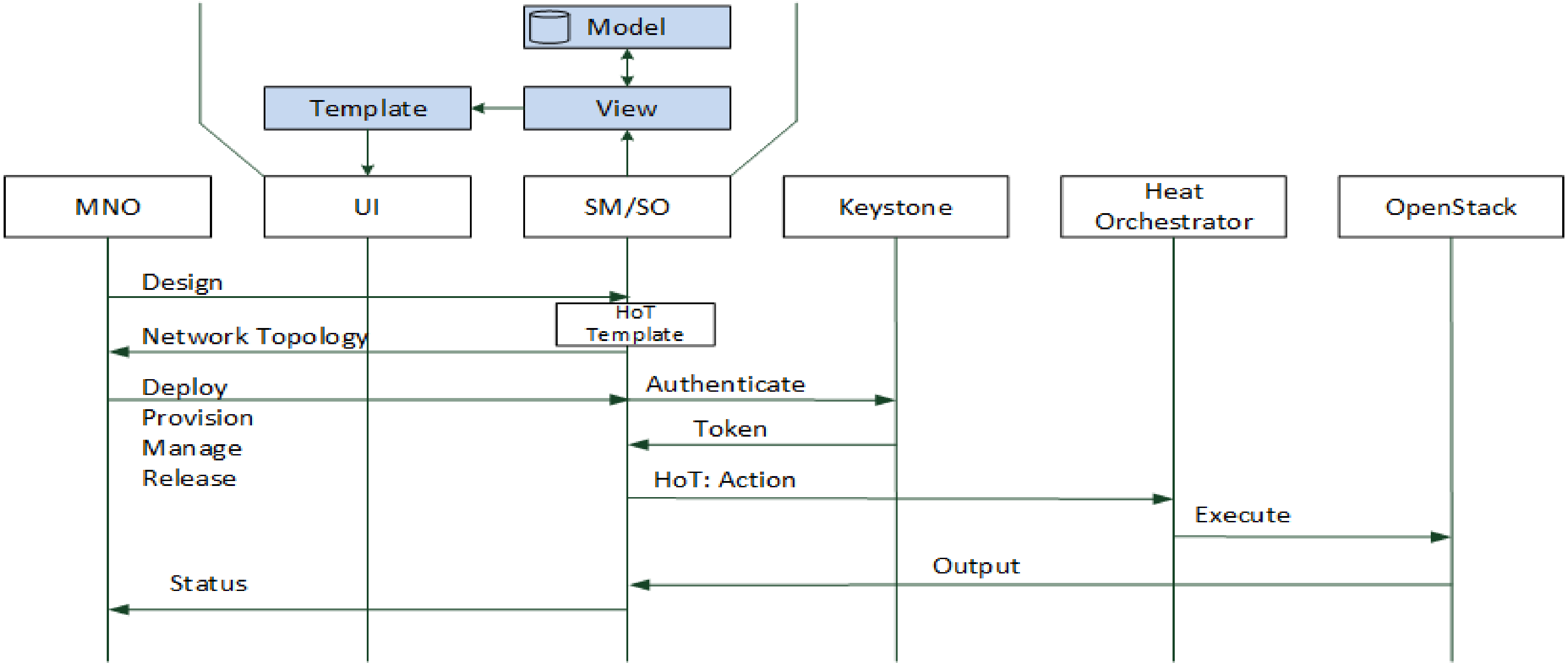
EURECOM
Sophia Antipolis

C-RAN Testbed on Sophia Antipolis Campus



C-RAN Testbed on Sophia Antipolis Campus Message Sequence (Openstack)

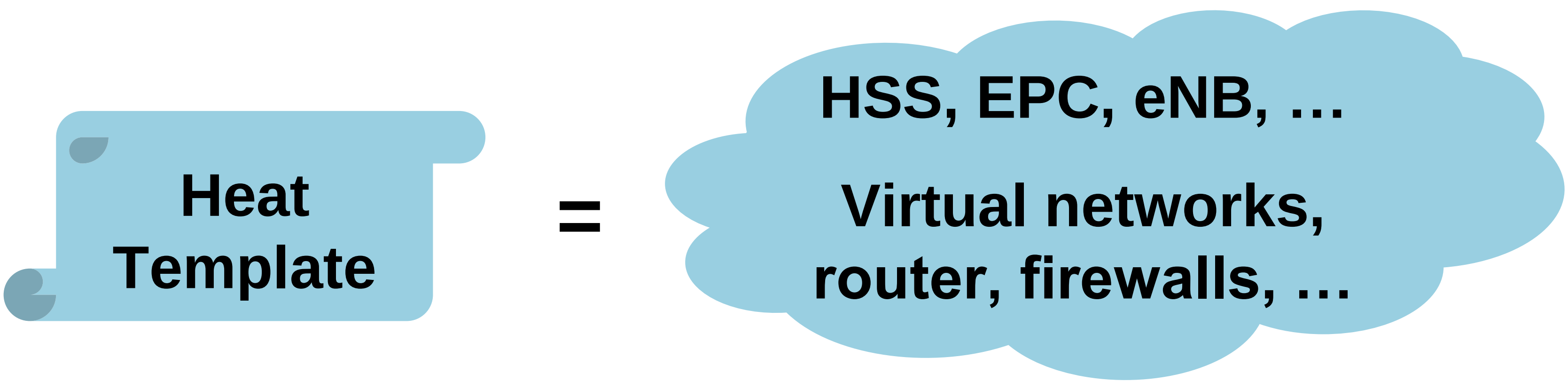
- **Orchestrator is key in the life cycle management**



EURECOM Sophia Antipolis

C-RAN Testbed on Sophia Antipolis Campus
Heat Orchestration Template (HOT)

- **The instantiation of a whole system (e.g., an LTE ecosystem) can be easily achieved with HOT**
  - virtual components of the communication network defining a network slice
- **Different level of abstractions are required**



C-RAN Testbed on Sophia Antipolis Campus

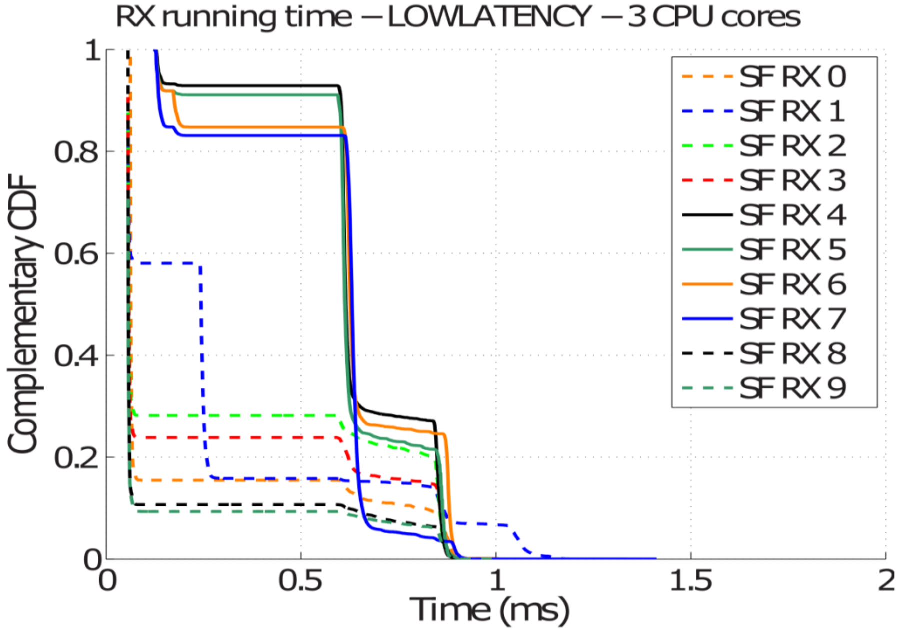
Heat Orchestration Template (HOT) - Example

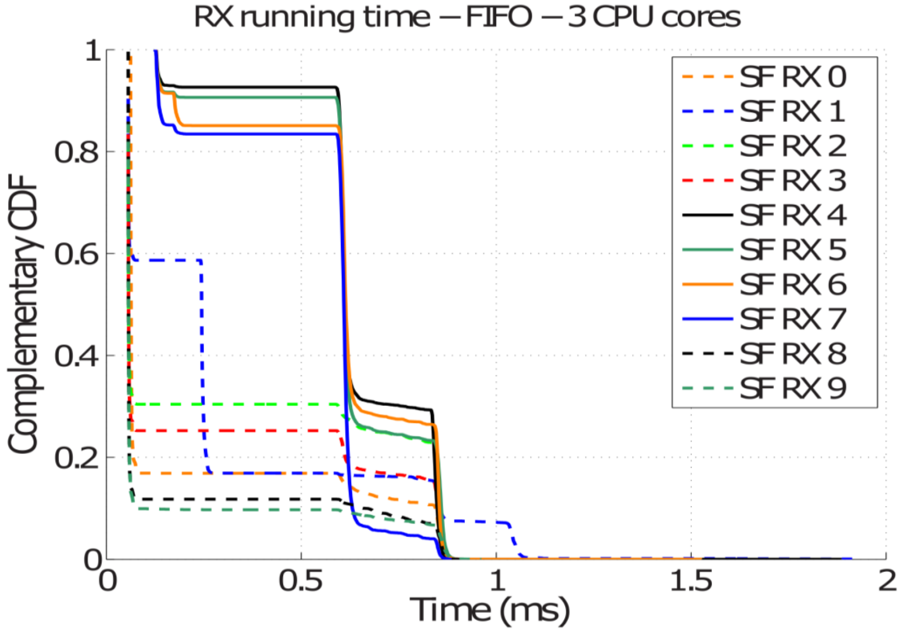
```
description:   LTEaaS,
parameters: {   key_name: {
                type: string, description: Name of a KeyPair to enable SSH access to the instance, default : cloud,          }},
resources: {    HSS: {...    } },
                EPC: { ...    } },
                ENB: {
                        type: OS::Nova::Server,
                        properties: {
                                image: enb-1,
                                flavor: eNB.med,
                                key_name: cloud,
                                networks: [{network: S1, }],
                                user_data: {
                                        #!/bin/bash\n
                                        MY_IP_S1=`ip addr show dev eth0 | awk -F'[ /]*' '/inet /{print $3}'`\n
                                        sed -i 's/MY_IP/'$MY_IP_S1'/g' /etc/hosts\n
                                        sed -i 's/EPC_IP/'$EPC_IP'/g' /etc/hosts\n
                                        sed -i 's#MY_IP#'$MY_IP_S1'/24#g' enb.band7.exmimo2.lxc.conf\n
                                        sed -i 's#EPC_IP#'$EPC_IP'#g' enb.band7.exmimo2.lxc.conf\n
                                        build_oai.bash -l ENB -t SOFTMODEM -D --run -C enb.band7.exmimo2.lxc.conf > /tmp/oai.log\n,

                                params: {
                                        $EPC_IP: {get_attr: [EPC, first_address],}
                                }
                }
} }
```

C-RAN Testbed on Sophia Antipolis Campus
Impact of the OS scheduler

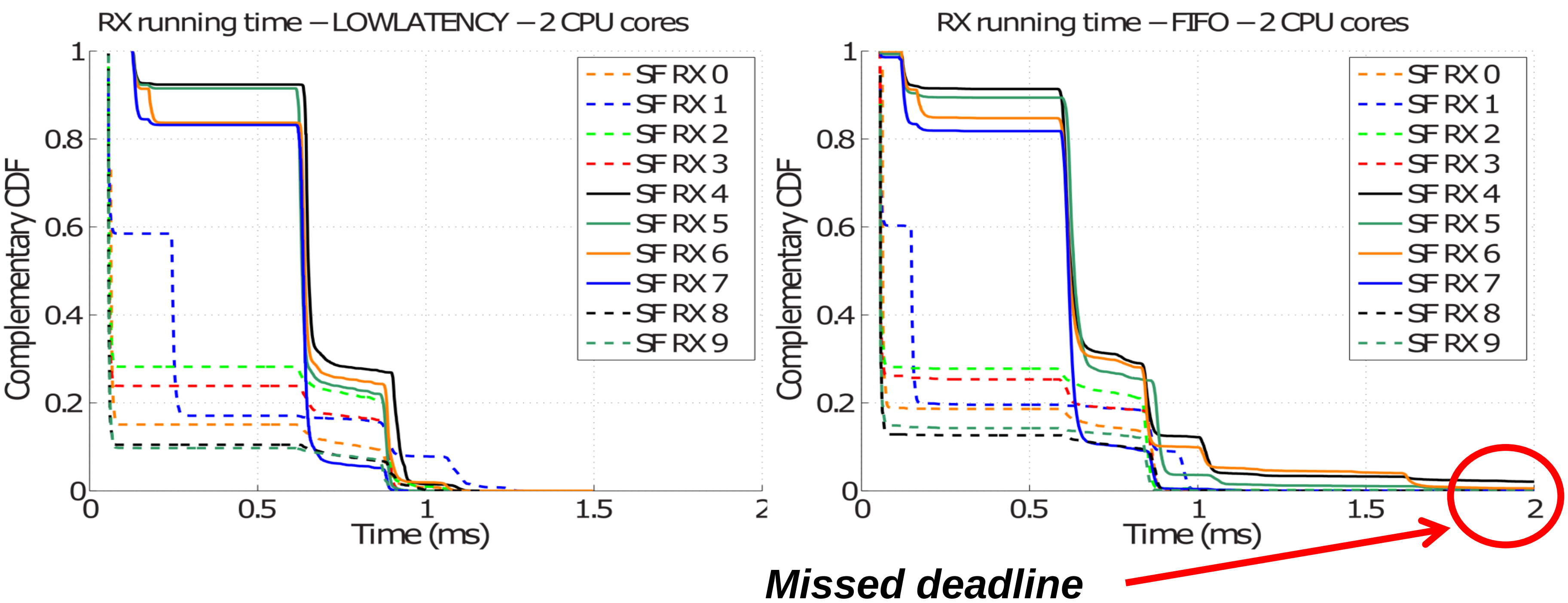
- **FDD, 10MHZ, SISO, with EXMIMO RF**
- **UL Processing time: Only 4 uplink sub-frames**
  - SF #0, 1, 2 and 3, allowing UL transmission to occur in SF # 4, 5, 6, 7.
  - Full uplink traffic





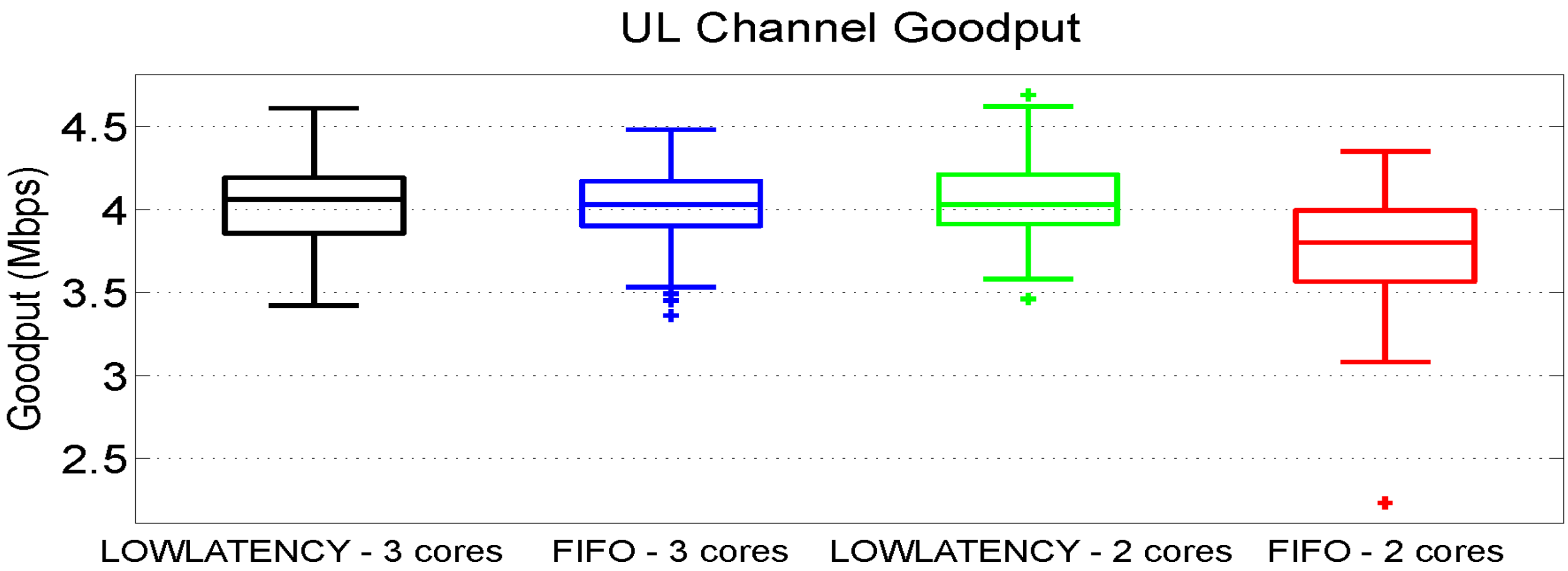
C-RAN Testbed on Sophia Antipolis Campus
Impact of the OS scheduler

- **FDD, 10MHZ, SISO, with EXMIMO RF**
- **UL Processing time: Only 4 uplink sub-frames**
  - SF #0, 1, 2 and 3, allowing UL transmission to occur in SF # 4, 5, 6, 7.
  - Full uplink traffic



C-RAN Testbed on Sophia Antipolis Campus
Impact of the OS scheduler

- **FDD, 10MHZ, SISO, with EXMIMO RF**
- **UL Processing time: Only 4 uplink sub-frames**
  - SF #0, 1, 2 and 3, allowing UL transmission to occur in SF # 4, 5, 6, 7.
  - Full uplink traffic

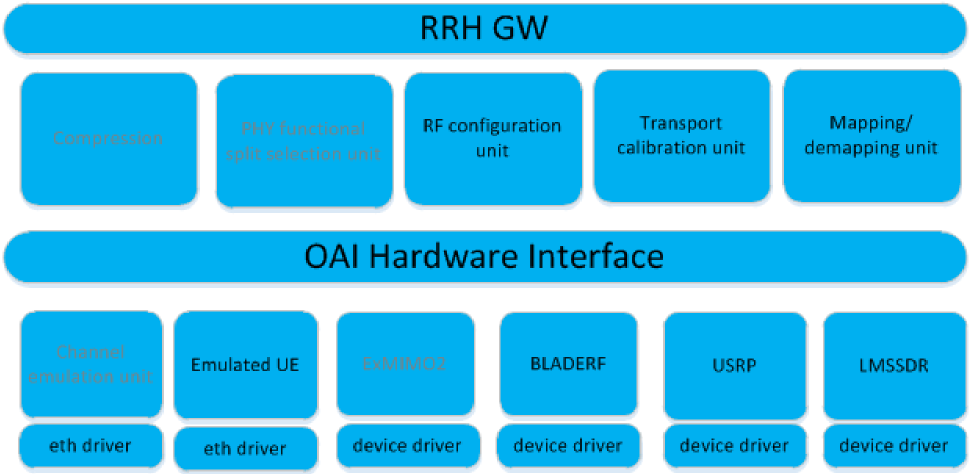


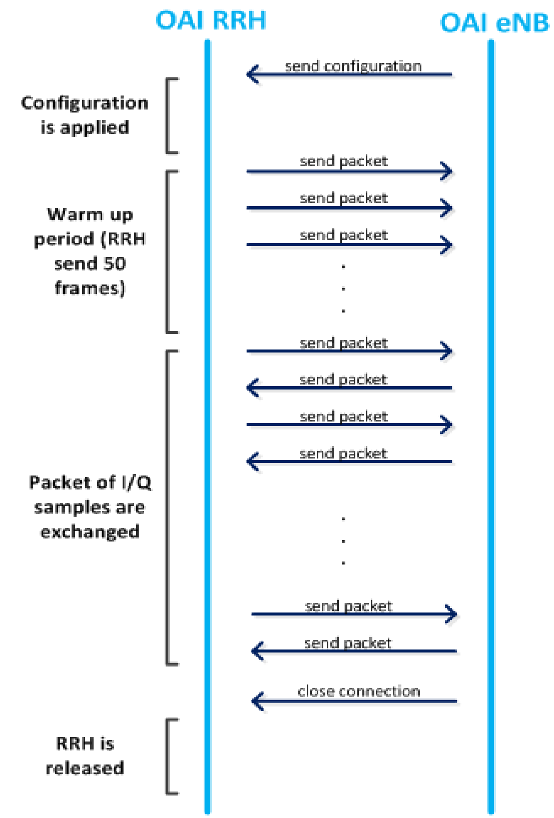
EURECOM

C-RAN Testbed on Sophia Antipolis Campus

OAI RRH

- **OAI eNB**
  - eNB is a client able to initiate a connection with the RRH GW
  - eNB is configuring the RF (DL/UL frequency, TX/RX gain_ and managing the data path(I/Qsamples)
- **OAI RRH**
  - RRH is a I/Q sample server waiting for incoming BBU client connections.
  - RRH is the RF front end and provides the timestamp





EURECOM

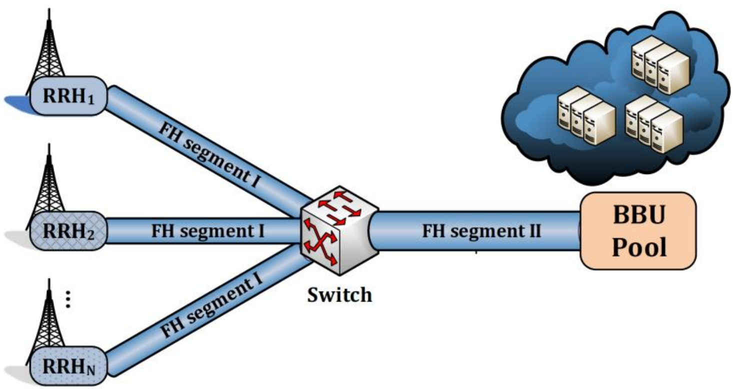
C-RAN Testbed on Sophia Antipolis Campus OAI RRH

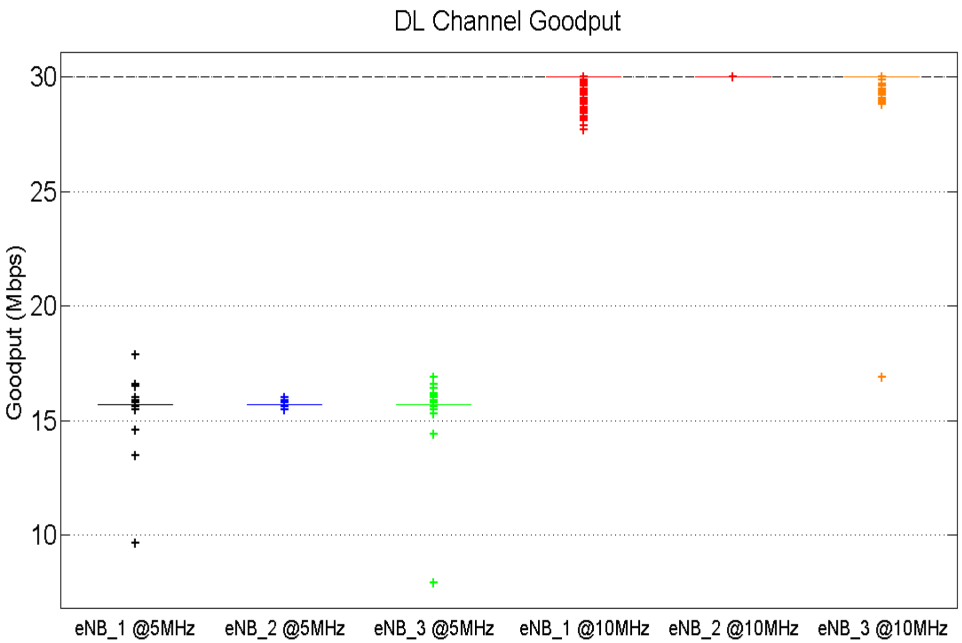
- **The FH interface is divided logically into two streams:**
  - Data: transports payload,packet length is a function of BW and MTU.
  - Control: in-band or out-of-band; eNB configures and manages RRHs
- **Two flavors of FH protocol are supported:**
  - UDP transport protocol: offers statistical multiplexing (multiple simultaneous communication on the same medium) at the cost of one additional layer in the protocol stack.
  - RAW Ethernet: offers minimal protocol stack but unable to support statistical multiplexing
- **Header format (no split case)**

Fields	Size (bits)	Description
Control Flag	1	This flag is used to specify whether the payload of the packet is either data or control data. value=0 ->data value=1 ->control data
Timestamp	64	The timestamp of the packet is the time that the payload was generated by the radio equipment.
Antenna ID	16	Antenna ID is a number used to map a packet to the appropriate antenna of the radio front-end equipment.

C-RAN Testbed on Sophia Antipolis Cam
DL Performance

- **Three setting (FDD, SISO, with USRP B210 RF, Eth fronthaul network)**
  - eNB_1: No RRH
  - eNB_2: Local RRH
  - eNB_3: Remote RRH



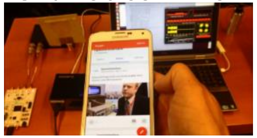


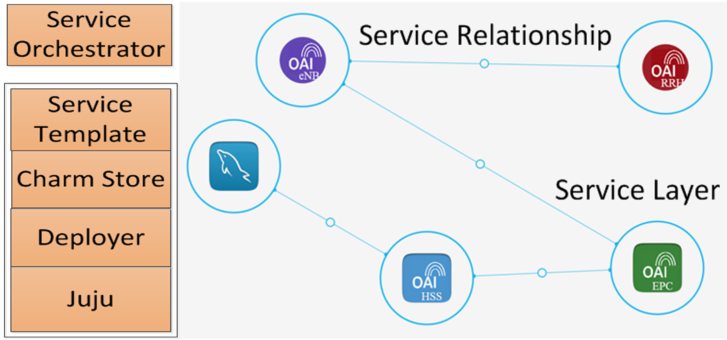
EURECOM

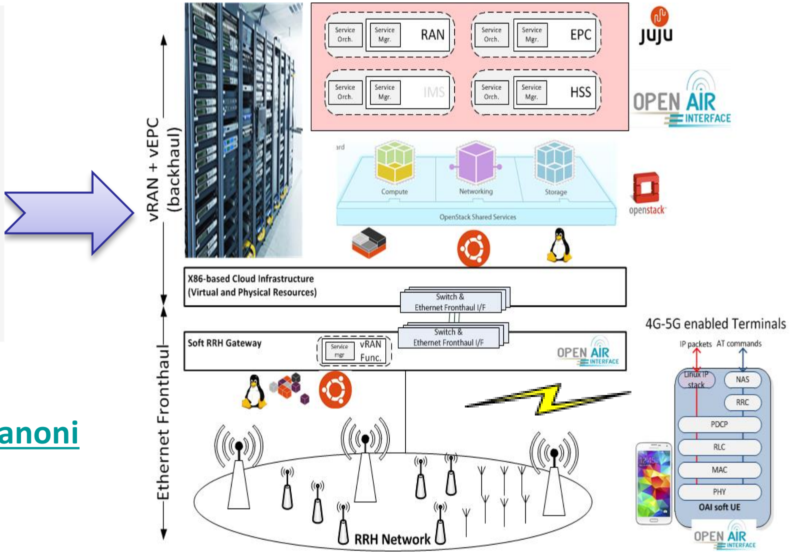
Deployed CRAN NFV Service Template Juju

MWC 2016

MCN/Mobicom Demo





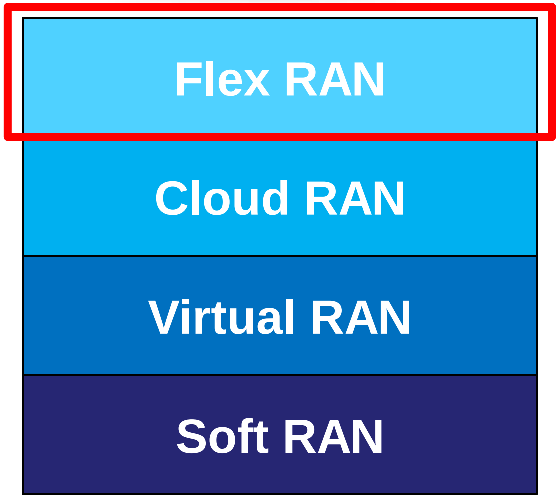


- **Demo @ MWC 2016 w/ Canonical**
- https://insights.ubuntu.com/2016/02/22/canonicals-vnf-pil-for-nfv-scale-out-architectures/
- https://jujucharms.com/q/oai

Life Cycle KPI	Unit	KPI measurements
Installation	Time(s)	600 seconds
Configuration	Time(s)	4 seconds
Disposal	Time(s)	< 1 seconds
Service upgrade duration	Time(s)	122-300 seconds

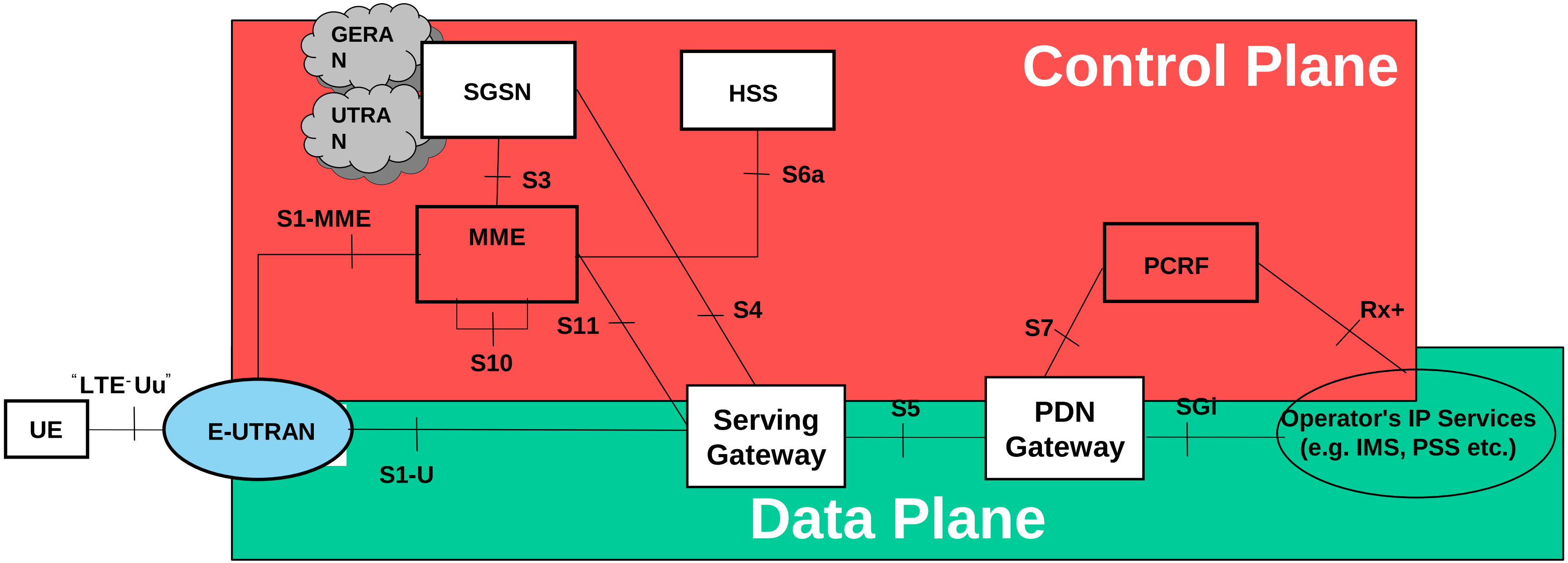
This Tutorial – Part II

- Technology
- Challenges
- Results
- Conclusion



Software defined networking

- **Simplify network control and coordination**
  - Separation of the control from the data plane with a well-defined API
  - Consolidation of the control plane
  - Network abstraction and programmability

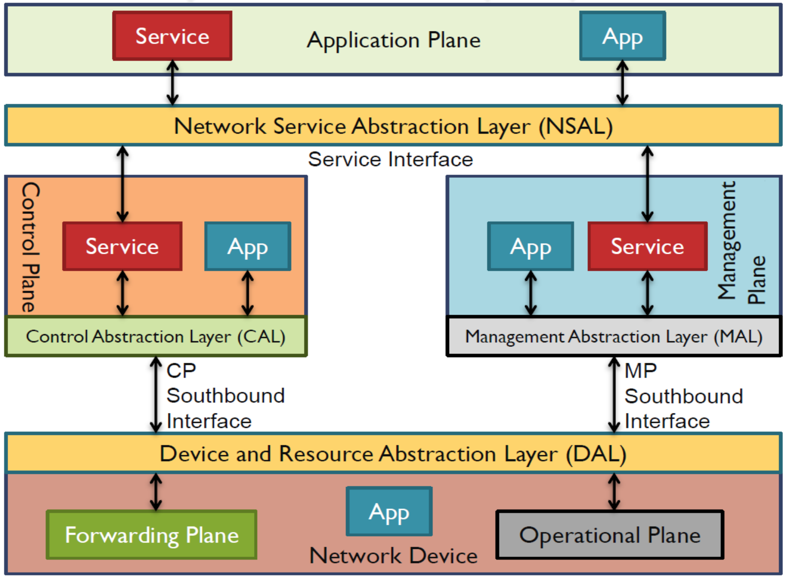


EURECOM

SDN for Wireless

- **Standard flow-level abstraction is not enough**
  - Stochastic nature of wireless links
  - Resource allocation granularity and time-scale
  - Heterogeneity of RAN

IEEE Architecture (RFC7426)



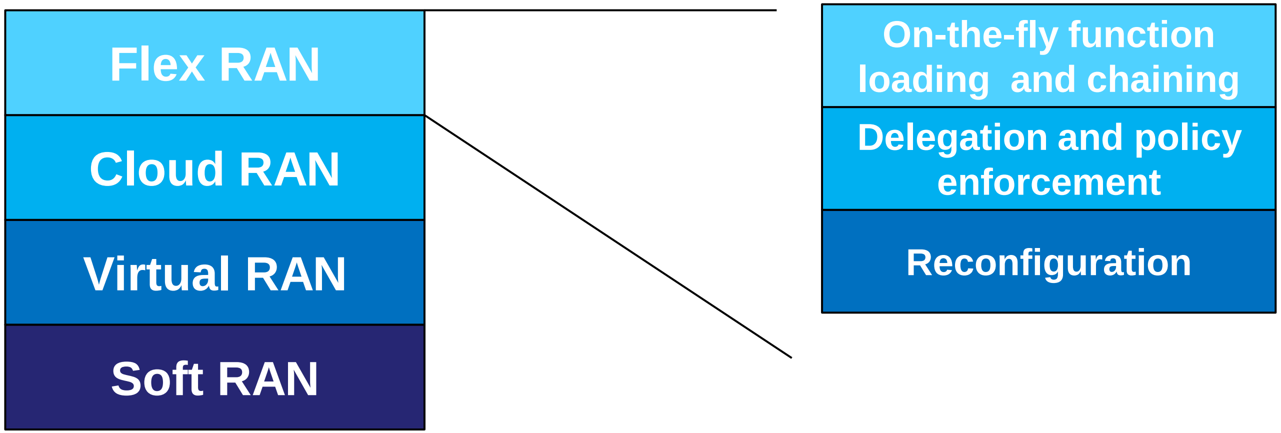
- **Requirements for wireless network abstraction**
  - State management
  - Resource allocation
  - Network monitoring
  - Network control

- Network Application
- Semantics
- Programming SDK

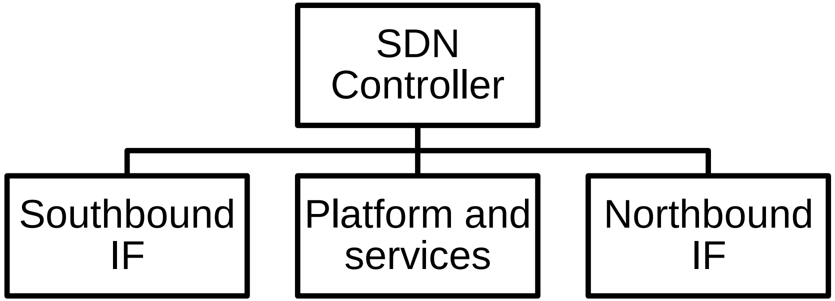
- Northbound Interface
- Network Operating System
- Network Abstraction

- Southbound Interface
- Network Infrastructure

SDN Challenges



- **Radio and core API and Southbound Protocol**
- **Network Abstraction and graphs**
- **Scalability and Control delegation mechanisms**
- **Realtime control**
- Low latency edge packet services
- Cognitive management, self-adaptive, and learning methods
- Northbound Application programming interface



EURECOM

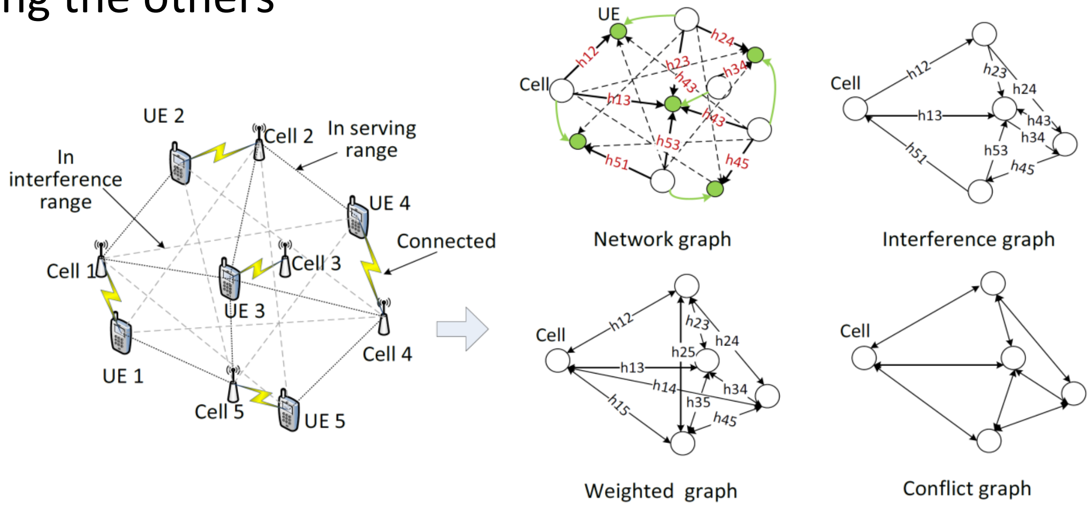
Radio and core API and Southbound Protocol

- **Control plane APIs allowing fine grain radio and core control and monitoring**
- **Platform- neutral and extendable protocol message service**
  - Language agnostic
- **Optimize message footprint**
  - Aggregation
  - (de)serialization
- **Asynchronous control channel**
  - Queue
  - Pubsub communication model
- **Supported network topologies**
  - P2P, P2MP, and possibly others

Message	Field	Usage
Configuration request	Configuration type	Type of configuration, either set or get
	Cell configuration flag	Bit map of the requested cell configuration
	Cell configuration list	List of cells (in IDs) to request configuration
	UE configuration flag	Bit map of the requested UE configuration
	UE configuration list	List of UEs (in IDs) to request configuration
Configuration reply	Cell configuration	Requested cell configuration report
	UE configuration	Requested UE configuration report
Status request	Status type	Can be periodical, one-shot, event-driven
	Status period	Period in Transmission Time Interval (TTI)
	Cell status flag	Bit map for the requested cell status
	Cell list	List of cells (in IDs) to request the status
	UE status flag	Bit map for the requested UE status
	UE status list	List of UEs (in IDs) to request the status
Status reply	Cell status	List of cell including the statistic reports
	UE status	List of UE including the statistic reports

Network Abstraction and graph

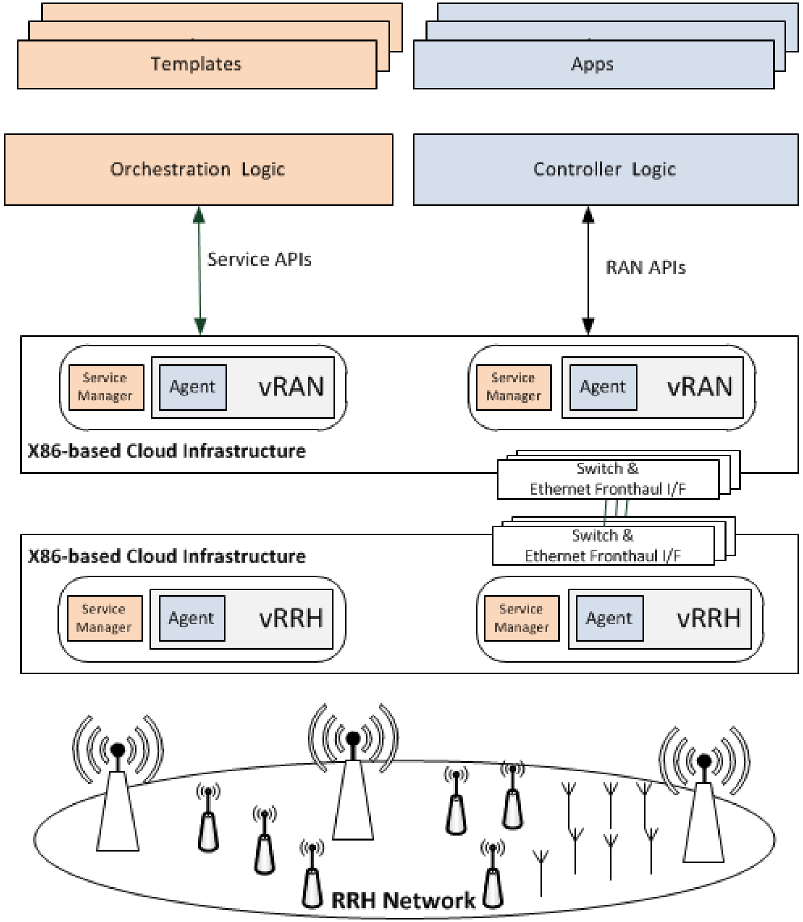
- **Effective representation of the network state at different network levels allowing**
  - fine-grained programmability, coordination and management of atomic or composed services across different domains/regions via
- **Network graphs can be separated based on**
  - Time, Region, carrier, cell among the others
- **Encompass data models**
  - Time-frequency status and resources
  - Spatial capabilities
  - Key performance indicators



EURECOM

Scalability, Control delegation mechanisms, and Realtime Control

- **Feasible to achieve a realtime RAN programmability at TTI level (1ms)**
  - Realtime control: Guarantee a (quasi-) deterministic reaction time of a control command triggered by the controller
- **Hierarchical controller logic**
  - non-time critical → centralized entity
  - time critical → edge entity
  - May offloaded time critical operation to an agent acting as a local controller
- **Network applications**
  - Proactive based on periodical event
    - Scheduler
  - Reactive based on event-triggering
    - E.g. mobility manager
- **Interplay with the orchestrator**

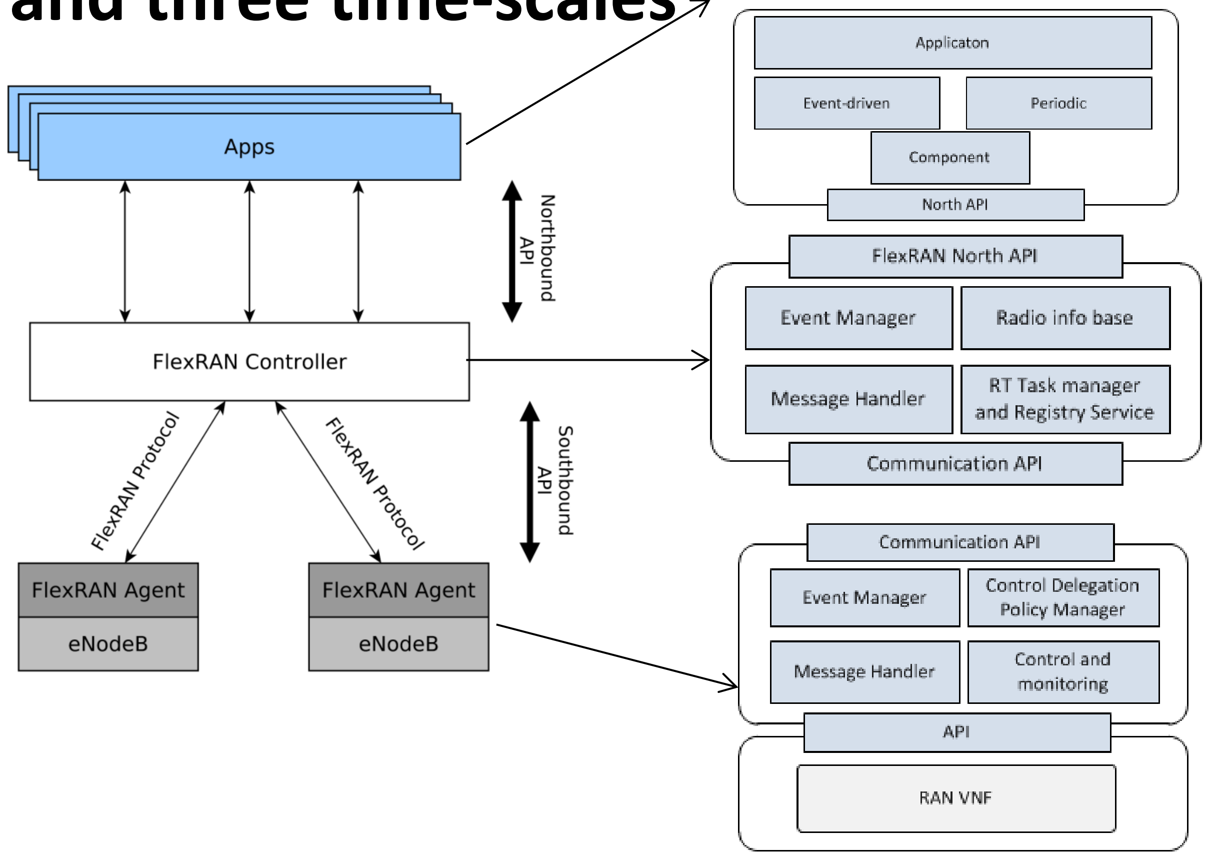


EURECOM

Programmable RAN Controler-Agent Design

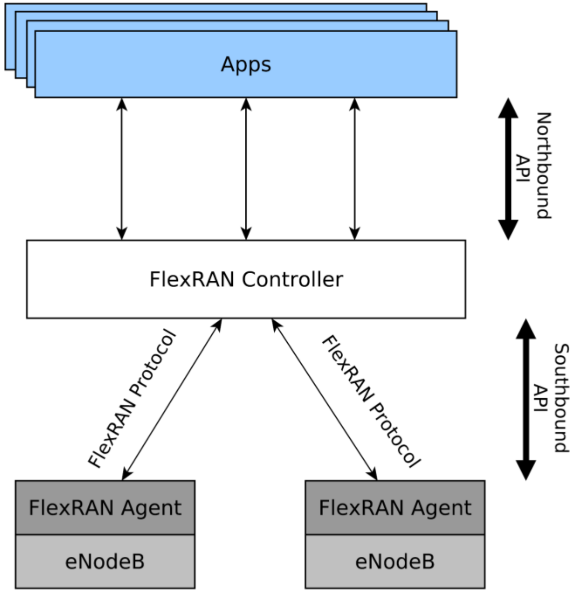
- **Three subsystems and three time-scales**

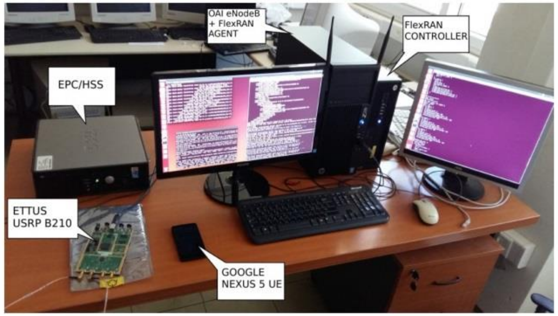
Message	Field	Usage
Configuration request	Configuration type	Type of configuration, either set or get
	Cell configuration flag	Bit map of the requested cell configuration
	Cell configuration list	List of cells (in IDs) to request configuration
	UE configuration flag	Bit map of the requested UE configuration
	UE configuration list	List of UEs (in IDs) to request configuration
Configuration reply	Cell configuration	Requested cell configuration report
	UE configuration	Requested UE configuration report
Status request	Status type	Can be periodical, one-shot, event-driven
	Status period	Period in Transmission Time Interval (TTI)
	Cell status flag	Bit map for the requested cell status
	Cell list	List of cells (in IDs) to request the status
	UE status flag	Bit map for the requested UE status
	UE status list	List of UEs (in IDs) to request the status
Status reply	Cell status	List of cell including the statistic reports
	UE status	List of UE including the statistic reports

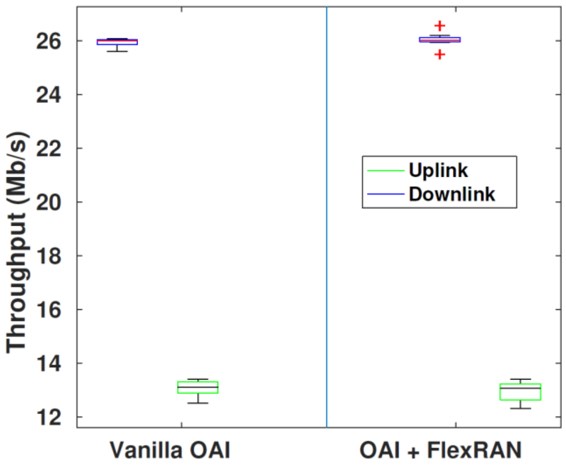


Programmable RAN
DL performance comparison

- **Considered controller apps**
  - DL scheduler
  - Monitoring and analytics







EURECOM Sophia Antipolis

Conclusion

Conclusion

- **4G/4G+ feasible on General Purpose Processors (x86) and Virtualization environment**
  - Exploit hybrid CPUs
- **Gap between virtualization and cloudification**
  - Exploit the microservice and NFV principles
- **Realtime network programmability is feasible at TTI level**
  - Exploit MEC principles for the data-plane programmability
- **Gat between static and cognitive control and management, self-adaptive, and learning methods**
  - Exploit machine learning and data mining techniques

Future Research Topic

- **Functional split in RAN and CORE**
  - What is the optimal split under capacity-limited fronthaul/backhaul and processing-limited compute resources ?
  - How to change the functional split on the fly?
- **Cognitive self network management**
  - What are the right network abstraction and modelling?
  - How the new techniques in machine learning, data mining and analytics can be leveraged in improving network and user experience?
- **Network slicing and mutli-domain E2E service management and orchestration**
  - How to change the E2E service definition on the fly?
  - How to deliver a network service offerings optimized for each and every use case, application and user?


Q&A
Questions
Answers