Experimenting Cognitive Radio Communication on FIT/CorteXlab

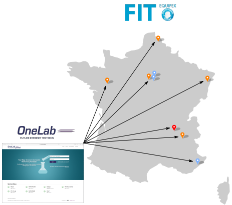
Tanguy Risset
and the FIT Team: Leonardo Cardoso, Jean-Marie Gorce, Guillaume Villemaud, Florin Hutu, Matthieu Imbert, Yasser Fadlallah.
www.cortexlab.fr

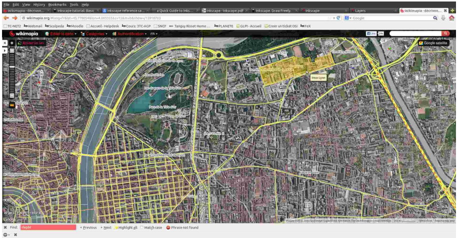
Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France

R2Lab Inauguration November 8, 2016

Context and geography

FIT/CorteXLab developed at Citi laboratory by INSA-Lyon and INRIA



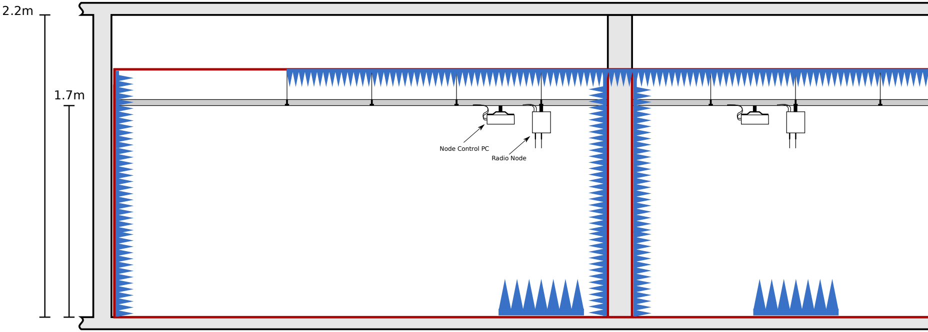


- CorteXlab is deployed by the Inria Socrate, guided by Jean-Marie Gorce and Tanguy Risset.
- Socrate research team (11 permanent members) works on software and cognitive radio.
- CorteXlab is one of the platforms of the FIT Equipex.

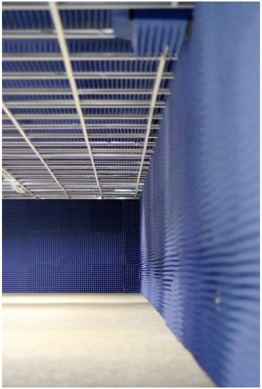
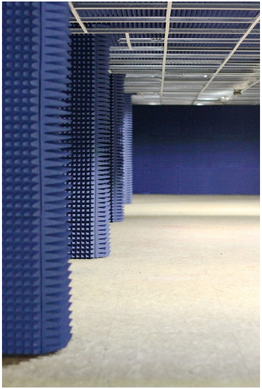
Table of Contents

Experimentation Room

INSA Lyon - Claude Chappe building - basement

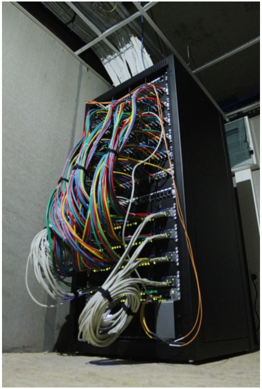


Experimentation Room
INSA Lyon - Claude Chappe building - basement


Experimentation Room

After Node installation



CorteXlab In numbers

- $\sim$ 200 m^2 in experimentation room area
- $\sim$ 500 m^2 of electromagnetic isolation material (50 dB)
- $\sim$ 300 m^2 of radio absorbers
- Aprox. 40 SDR nodes (MIMO, SISO, BB)
- Operating between 300 MHz - 3 GHz (for SDR cards)
- 28 MHz of bandwidth
- $\sim$ 1 km (copper) and 600 m (fibre) network cables
- 3 high perf. servers, 7 switches and routers
- 3 years of deployment, 7 years of exploitation
- Total investment of about 1M€

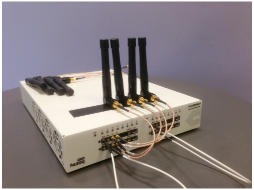
USRP Nodes from Ettus Research (National Instrument)

- The room contains 22 NI USRP 2932 with Gigabit Ethernet link to PC

+ Large community support
+ Full open-source toolset (GnuRadio)
+ Known IF-to-RF connection
– PC-Computing power
– No (easy) FPGA programming

Nutaq PicoSDR Nodes

- The room also contain 16 Nutaq Pico-SDR
  - Gigabit Ethernet and 8Gb PCIe link to PC
  - Xilinx Virtex6 SX315T FPGA
  - 4 of the 16 Pico SDR have 4x4 MIMO capabilities



+ Standard IF-to-RF connection
+ MIMO option available
+ Realtime operation

– "Non-open" development tools (licenses needed)

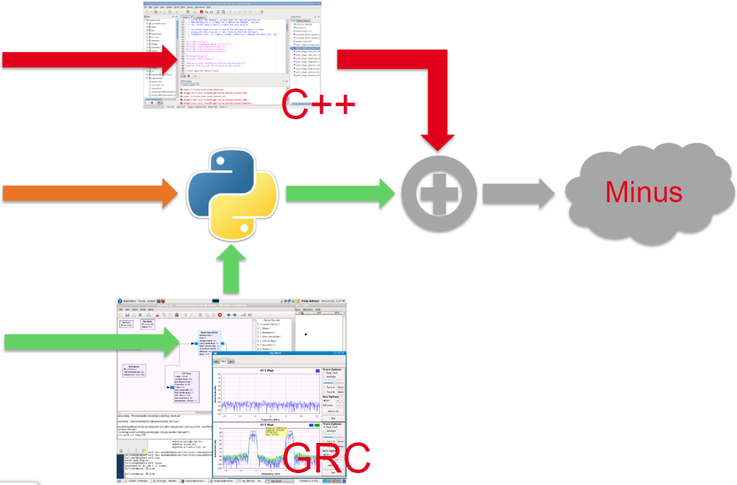
– "Off-road" development not so easy

Control PCs

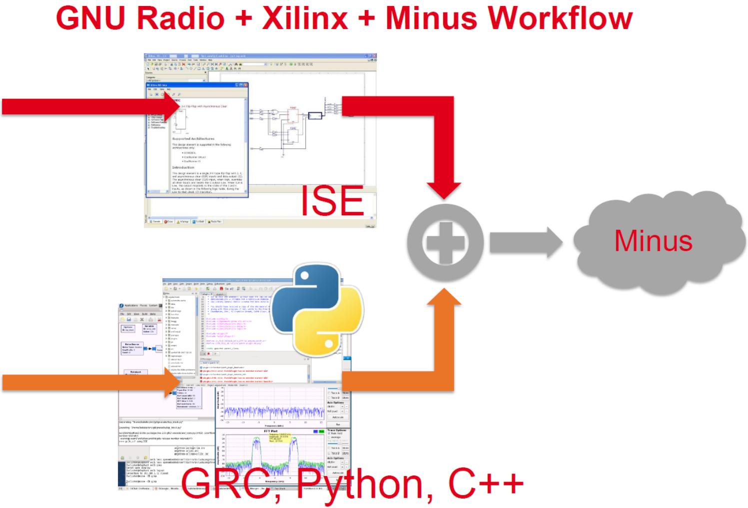
- One Industrial PC (no Fan) for each node.
- Debian linux OS.
- Ethernet controlled power switch

Programming USRPs with GnuRadio

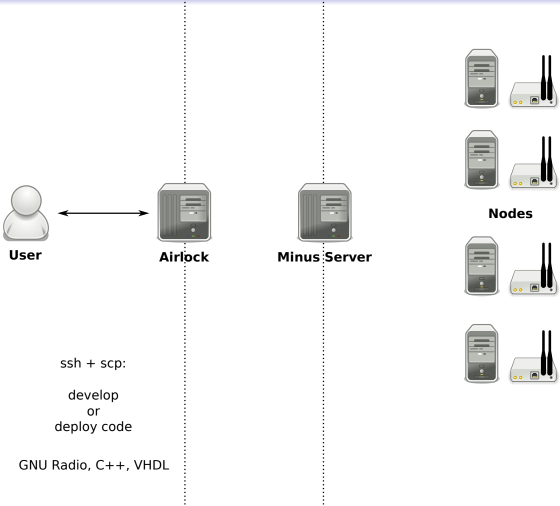
GNU Radio + Minus Workflow



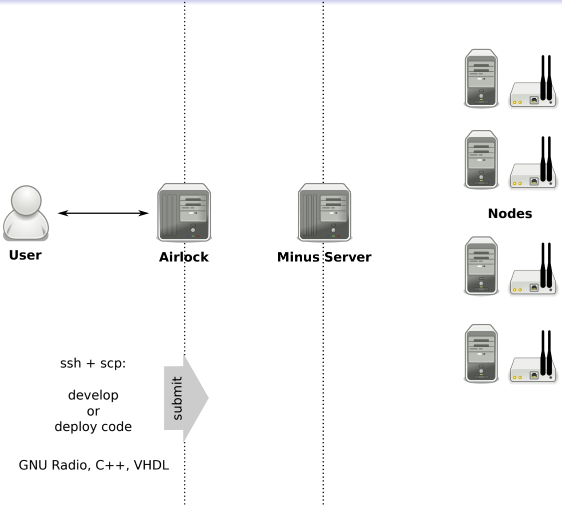
Programming PicoSDR with VHDL



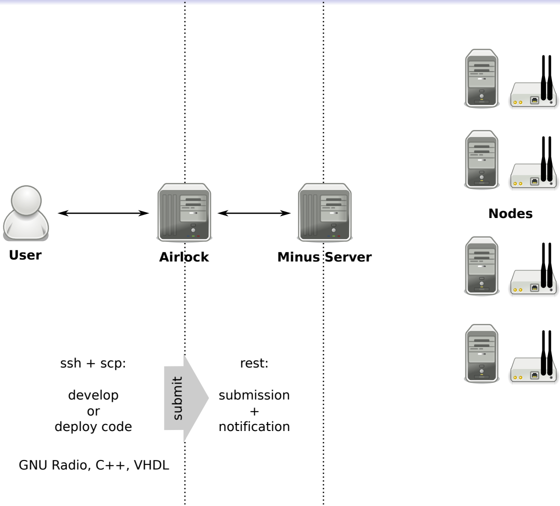
Experiment Start



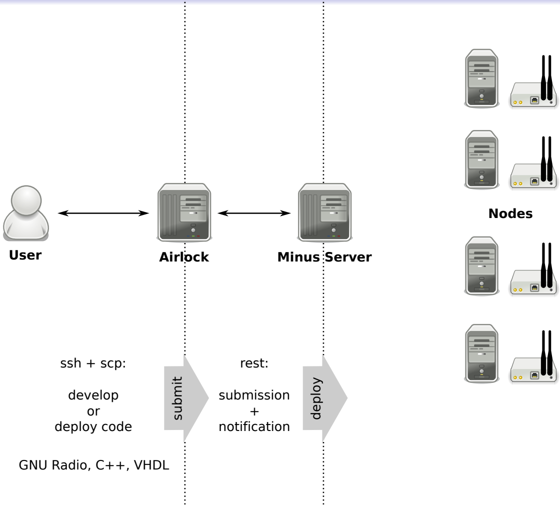
Experiment Start



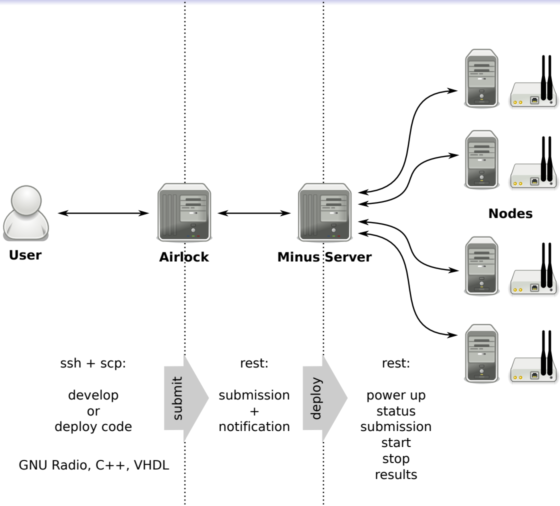
Experiment Start



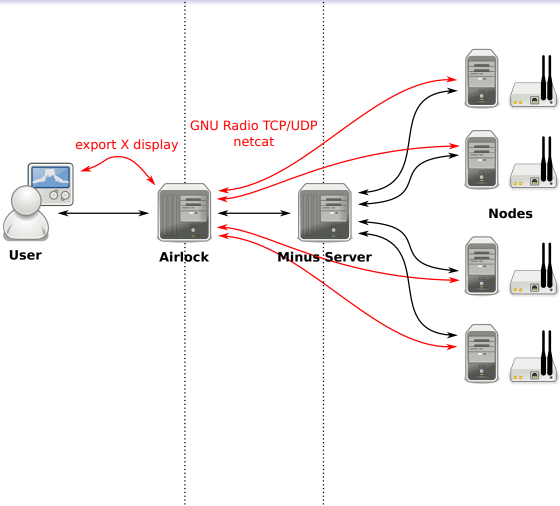
Experiment Start



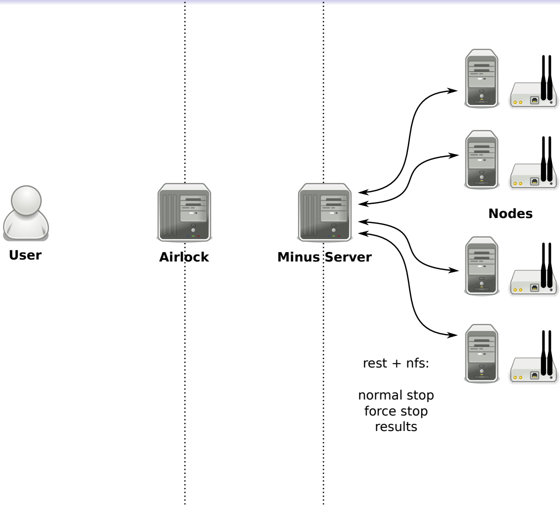
Experiment Start



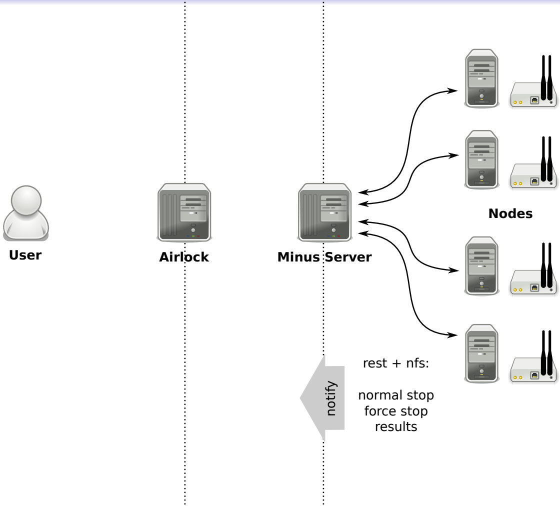
Experiment Run (Debug)



Experiment Closing



Experiment Closing



Experiment Closing

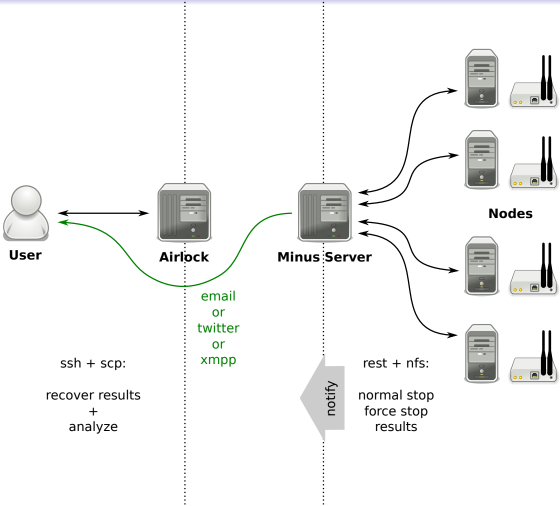
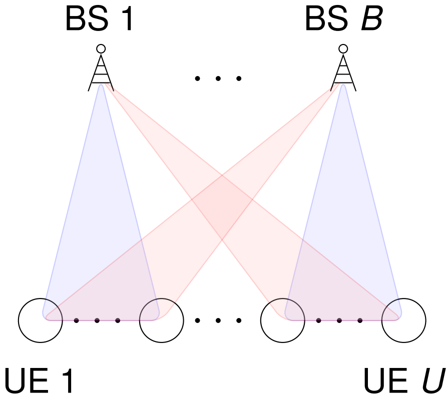


Table of Contents

Exp 1: Broadcast Channel interference Alignment

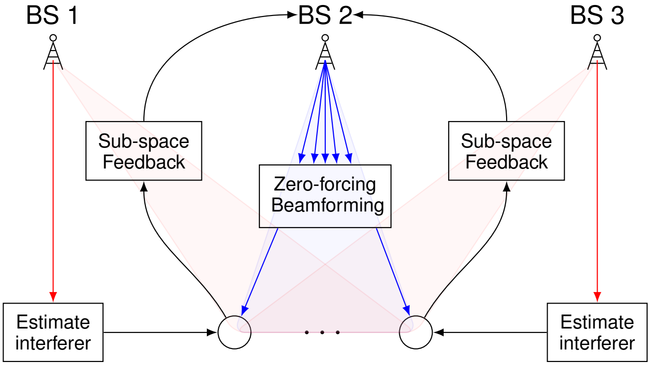


- B base stations and U users
- $\mathcal{U}_b$ is the set of users attached to BS b
- Bandwith W divided in F frequency sub-bands, and power per sub-band $p_b^(f)$
- M antennas on the BSs, N on the UEs

$$\hat{\mathbf{s}}_u = \mathbf{D}_u^\dagger \mathbf{H}_{1,u} \mathbf{C}_u \mathbf{s}_u + \sum_{\substack{v \in \mathcal{U}_1 \\ v \neq u}} \mathbf{D}_u^\dagger \mathbf{H}_{1,u} \mathbf{C}_v \mathbf{s}_v + \sum_{b \geq 2} \sum_{v \in \mathcal{U}_b} \mathbf{D}_u^\dagger \mathbf{H}_{b,u} \mathbf{C}_v \mathbf{s}_v + \mathbf{D}_u^\dagger \mathbf{z}_u$$

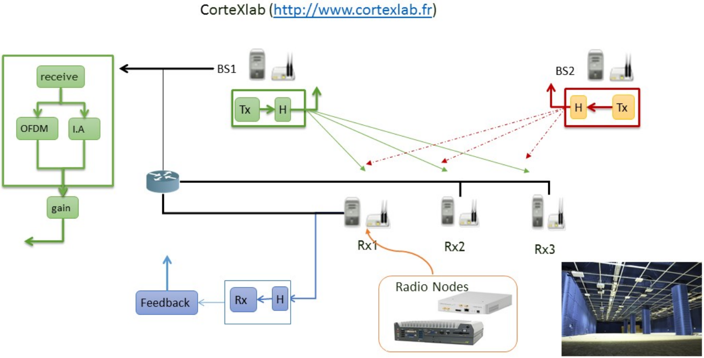
Broadcast Channel IA basic Idea

- Remove all intra-cell and some inter-cell interferers (Suh *et al.*, 2011, Bayesteh *et al.*, 2011)
- Key idea : reduce the actual signal space used by the BS



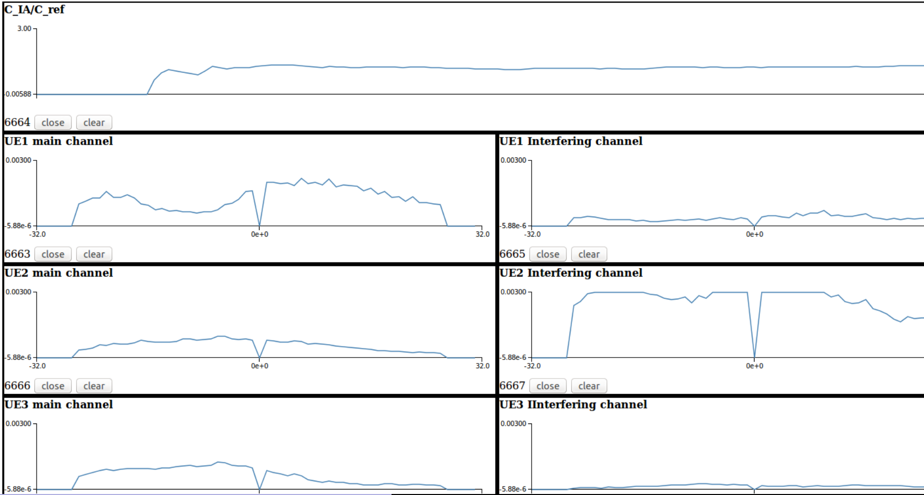
Implementation in CorteXlab

- See https://arxiv.org/abs/1511.01276 and publication in IEEE Communication Magazine



Interference Alignment in CorteXlab

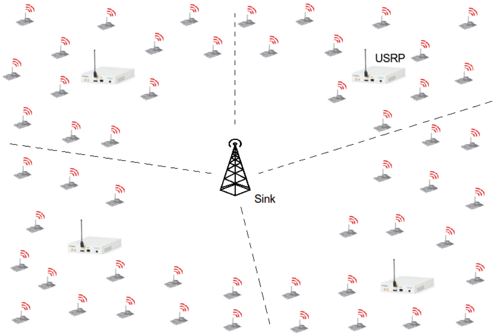
Demonstrated at Green Touch final meeting June 2015 (http://www.bell-labs.com/greentouch/)

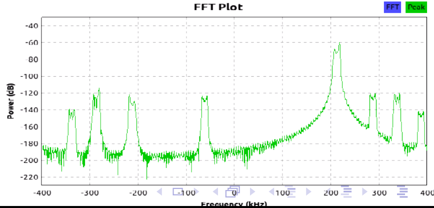


Exp 2: IoT Spectrum emulation

Collaboration with Orange Labs: emulate IoT networks spectrum

- Several (thousands) nodes are transmitting asynchronously
- Several independent communication protocols.





Wireless Caching

- Collaboration with Nokia Bell Labs New Jersey and U. of Naples
- Objective: combine wireless caching in 5G Networks and coded multicasting to serve multiple unicast demands.
- Motivation: wireless users rarely access the same content at the same time
- We evaluate on a prototype implementation the experimental performance of state-of-the-art caching-aided coded multicast schemes compared to state-ofthe-art uncoded schemes
- To be published in IEEE communications magazine

Planned experimentations

- EPHYL ANR project accepted in 2016
  - Supelec Rennes (C. Bader), CEA Leti (V. Berg) and Socrate (J.M. Gorce)
  - investigate coming and future Low Power Wide Area technologies (i.e. "small packet") to improve coverage, data rate and connectivity
  - Planned experimentation: prototype "small packet" waveforms on CorteXlab
- OpenBTS on CorteXlab
- Open-source IEEE 802.15.4 GNURadio receiver on USRP
- Open-source IEEE 802.15.4 transceiver on PicoSDR

Table of Contents

Links with R2Lab

- CorteXlab is built to study any problem where Cognitive Radio and physical layer of wireless communications are concerned. R2Lab is more specifically targeted to 5G MAC and higher layers.
- However, many technical efforts can be shared between the platforms:
  - R2Lab investigates the Open-Air Interface software, CorteXlab investigates on GnuRadio; both skills could be shared between the sites.
  - CorteXlab can contribute with the many GNU Radio designs already available to its users:
    - Zigbee on USRPs (Bastian Bloessl)
    - OFDM on USRPs (GNU Radio, T. Rondeau)
    - OFDM on Pico-SDR (Nutaq design)
    - OFDM with GNU-radio on Pico-SDR (GNU Radio + Nutaq path-through)

Important open Questions

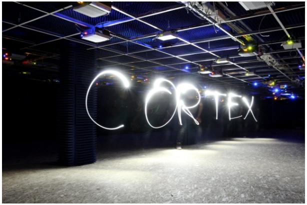
- We need users to:
  - Bring more waveform designs to CorteXlab (Wifi, LTE, BlueTooth, etc.)
  - Validate multi-user communication in a real and reproducible radio communication environment

$\Rightarrow$ Cooperation with R2Lab, Eurecom and the French Telecommunication community is essential.

- Important technical open questions for CorteXlab :
  - Fast compilation for FPGA-based SDR
    - ease the PicoSDR programming
  - Enable dynamic data flow modification in GNU radio
    - specify Cognitive Radio Application in a more natural way

Table of Contents

Conclusion: A unique testbed for Cognitive radio applications

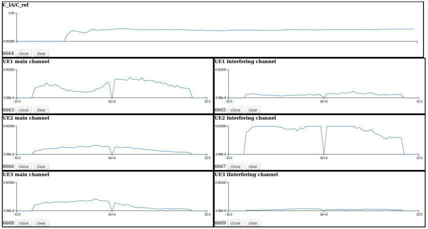


- Powerful and flexible RF front-end
- Powerful programmable baseband (FPGA)
- Current platform usage since march 2016:
  - 51 user accounts
  - 807 tasks launched
- Programmable from everywhere in the world
- Web site: `www.cortexlab.fr`
- Git-hub repository: `https://github.com/CorteXlab`

⇒ Please register, its free! `register@cortexlab.fr`

Recent platform infrastructure improvements

- Improved debugging capabilities: centralized live monitoring of all nodes and platform servers logs.
- Improved platform reliability (reboot of FPGA nodes after each experiment)
- Improved spectrum analyzing tool: FFT-Web



- More tutorials and howtos available or improved
- Continuous bugfixing and maintenance

Platform infrastructure improvements in the near future

- Continuous Improvement of the user-friendliness and documentation based on user feedback
- Improving interactions between platform nodes and the OAR batch scheduler to:
  - automatically switch off / on the nodes and radio nodes when needed (improved reliability and energy efficiency)
  - improve monitoring of node states (to detect faulty nodes with better accuracy)
- Explore new GNURadio features, such as CtrlPort, which would allow better live feedback of experiments, as well as more complex or more interactive experiment workflows.
- Better FPGA support: more documentation, MIMO capabilities.
- Setup sandboxes: small prototyping platforms.