

GRANT



AGREEMENT NO.: 732174

Call: H2020-ICT-2016-2017

Topic: ICT-13-2016

Type of action: RIA



Orchestration and Reconfiguration Control Architecture

D6.3: Advanced integration of SDR platforms and initial implementation of SDR experimentation tools

Revision: v.1.0

Work package	WP6
Task	Task 6.1, 6.2, 6.3
Due date	31/12/2018
Submission date	21/12/2018
Deliverable lead	IMEC
Version	1.0

Authors	Joao F. Santos (TCD), Diarmuid Collins (TCD), Wei Liu (IMEC), Pieter Becue (IMEC), Ingrid Moerman (IMEC), Andrea Guevara (KUL), Seyed Ali Hassani (KUL), Zhitao Lin (TUD)
Reviewers	Joao F. Santos (TCD), Sofie Pollin (KUL)

Abstract	This deliverable mainly describes the actions taken towards the further integration of SDR platforms into the ORCA facility, and further improvement of Fed4Fire compliance among the ORCA facilities.
Keywords	Testbeds, SDR

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable*:		R
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CI	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to ORCA project and Commission Services	

Disclaimer

The information, documentation and figures available in this deliverable, are written by the ORCA (Orchestration and Reconfiguration Control Architecture) – project consortium under EC grant agreement 732174 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Confidential - *The information contained in this document and any attachments are confidential. It is governed according to the terms of the project consortium agreement*

Copyright notice

© 2017 - 2020 ORCA Consortium

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc

EXECUTIVE SUMMARY

This deliverable reports on the efforts of testbed development in the second year of the ORCA project. The ORCA facility consists of 5 testbeds: (i) the w-iLab.t testbed from IMEC, (ii) the KUL testbed, (iii) the ORBIT testbed from RUTGERS, (iv) the IRIS testbed from TCD and (v) the OWL testbed from TUD. Among the five testbeds, there are three testbeds (w-iLab.t, IRIS, ORBIT) already Fed4Fire (F4F) compliant before the ORCA project. The advantage for a testbed to be F4F compliant enables users to access different testbeds via a unified interface and a single user account. Making all ORCA testbeds F4F compliant is needed to have a common portal for ORCA facilities. KUL and TUD have taken the necessary steps to make their testbeds also compliant with the F4F standards in the 1st year of ORCA, and in this year these two testbeds also added the support for F4F monitor functionality. In addition, all partners have brought in new functionalities for Software-defined Radio (SDR) related experimentations, ranges from deployment of new hardware, and/or support for new experimentation tools on the existing SDR platforms. More specifically, some main improvements in each testbed are listed as follows: IMEC w-iLab.t has included new hardware devices (i.e. USRP B200/B210), and built up new tutorials for using USRP with RFNoC framework or Zynq SDR with TAISC framework; IRIS testbed has deployed new Dell servers and commercial LTE terminals in terms of hardware improvements, and has deployed OpenStack and srsLTE EPC regarding software improvements; TUD OWL testbed has provided OS image selection functionalities and various hardware and software components for the operation of MultiRAT platform; KUL testbed has included Digital Self-interference Cancellation module for the full-duplex operation of nodes in the testbed, and various hardware and software improvements are made regarding Massive MIMO operation in the testbed; Finally ZYNQ SDR has been deployed on the IMEC portable testbed.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	4
LIST OF FIGURES	5
LIST OF TABLES	6
ABBREVIATIONS	7
1 INTRODUCTION	8
2 FED4FIRE COMPLIANCE OF ORCA TESTBEDS	9
2.1 KUL F4F federation monitor	9
2.2 TUD OWL F4F federation monitor	9
3 SDR PLATFORM INTEGRATION INTO ORCA TESTBEDS	11
3.1 Overview of supported SDR platforms.....	11
3.2 SDR platform integration into testbeds.....	12
3.2.1 w-iLab.t.....	12
3.2.2 IRIS	15
3.2.3 TUD OWL testbed.....	16
3.2.4 KUL testbeds	21
3.2.5 Portable testbed.....	23
4 CONCLUSIONS	25
REFERENCES	26

LIST OF FIGURES

Figure 1 The ORCA testbeds on the F4F federation monitor.....	9
Figure 2 Using USRP B200/B210 via USB hub in w-iLab.t.....	13
Figure 3 Using JTAG to program USRP X310	13
Figure 4 Integrated Logical Analyser (ILA) support in w-iLab.t.....	14
Figure 5 Multi-AM in OWL Testbed.	18
Figure 6 MultiRat Platform	19
Figure 7 Hardware set up of MultiRat Platform.....	20
Figure 8 Resource node with USRP b205 mini.....	20
Figure 9 Resource node with two USRP 2920	21
Figure 10. External power amplifier for users in KUL Massive MIMO testbed.	22
Figure 11 Zync SDR in portable testbed.....	23
Figure 12 2D design of Zync SDR enclosure.....	24

LIST OF TABLES

Table 1 Overview of supported SDR platforms.	11
Table 2 User power amplifier for KUL MaMIMO.	22

ABBREVIATIONS

AM	Aggregate Manager
CBTM	Cloud Based Testbed Manager
DiSIC	Digital Self-Interference Cancellation
DLS	Dithered Linear Search
DNS	Domain Name System
DPM	Data Protection Manager
EBD	Electronic Balance Duplexer
eNB	eNodeB
F4F	Fed4Fire
FPGA	Field-Programmable Gate Array
IBFD	In-Band Full Duplex
ILA	Integrated Logical Analyzer
JTAG	Joint Test Action Group
OS	Operating System
RAN	Radio Access Network
RFNoC	Radio Frequency Network-on-Chip
SDR	Software-defined Radio
SDN	Software-defined Networking
UHD	USRP Hardware Driver
USRP	Universal Software Radio Peripheral
VLAN	Virtual Local Area Network
VM	Virtual Machine

1 INTRODUCTION

This deliverable reports on the effort of testbed development in the second year of ORCA project. The ORCA facility consists of 6 testbeds: (i) the w-iLab.t testbed from IMEC, (ii) the KUL testbed, (iii) the Orbit testbed from RUTGERS, (iv) the IRIS testbed from TCD, (v) the OWL testbed from TUD, and (vi) the portable testbed from IMEC. All six testbeds achieved Fed4Fired (F4F) compliance in the 1st year of ORCA. The F4F federation monitoring function was not yet implemented for the testbeds at TUD and KUL last year, this was added in Y2 of ORCA. More details about the effort to bring this added feature is detailed in Section 2.

In this year, each of the ORCA testbed continues to improve its SDR related functionalities, by improving the functionalities of existing SDR platforms and also by deploying new state-of-the-art Software-Defined Radio (SDR) devices. In Section 3, an overview of supported SDR platforms is given, followed by a more detailed description of each of the partner's SDR integration.

2 FED4FIRE COMPLIANCE OF ORCA TESTBEDS

The FED4FIRE project provides facility monitoring to all testbeds in the federation. Every testbed is checked regularly if it can still be accessed and if it is still possible to create an experiment with at least one node. Behind the scenes, the experiments are created by jFed. The results are visualized on a web page [1], and a screenshot of some of the WiSHFUL testbeds is shown in Figure 1.

Wireless Testbeds

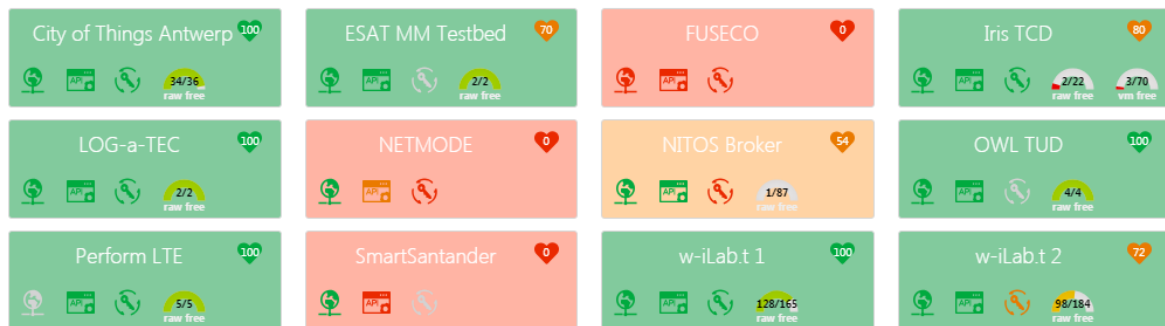


Figure 1 The ORCA testbeds on the F4F federation monitor.

During the second year of ORCA, the testbeds at KUL and TUD were added to the F4F federation monitor. The federation monitor can thus be used by the experimenters of ORCA as a first reference point in case of issues with a particular testbed. Especially for facility providers this is a useful tool, because they are notified in case of issues with the testbed and can inform their users about potential problems during their experiments.

2.1 KUL F4F federation monitor

The testbed at KUL has been added to the F4F federation monitor in a simplified way. The connectivity towards the testbed is checked, followed by a functional test of the F4F Aggregate Manager (AM), which serves as an entry point to the testbed for all users of the jFed tool. Because all nodes in the testbed are Windows machines to which a remote desktop connection has to be setup, only the connectivity towards those nodes are tested. The node login tests are skipped by the federation monitor, which currently only supports SSH connections.

2.2 TUD OWL F4F federation monitor

In last year, most of the F4F compliant functionalities were implemented in the AM. We improved the AM to combine the experiment-managing functionalities better to the jFed GUI as well as to the federation monitor webpage.

In order to enable the F4F federation monitor function, we have fixed the public reachable IP and the DNS name of OWL testbed and configured the firewall to allow the F4F monitoring connections. The AM is configured to reply the calls from the F4F Federation Monitor with complete and correct information. By replying the calls, the AM enable the Monitor to get the status of the testbed, the availabilities and the location of the resources.

The AM accepts the automated tests of the federation monitor. However, it is worth mentioning that our testbed does not completely support the “node login”, since this login test requests access to the resources via SSH connection. In our testbed, we provide graphical software such as LabVIEW to the user, and choose Remote Desktop as the connection between the user and the resources instead of SSH.

3 SDR PLATFORM INTEGRATION INTO ORCA TESTBEDS

3.1 Overview of supported SDR platforms

The Table 1 gives an overview of the SDR platforms supported in ORCA facilities.

Table 1 Overview of supported SDR platforms.

Testbed	w-iLab.t (IMEC)	ORBIT (RUTGERS)	IRIS (TCD)	TUD Testbed	KUL Testbed	Portable Testbed
Nutaq ZeptoSDR		X				
Nutaq picoSDR		X				
PicoZed Xilinx ZYNQ®-7000 SoC		X			X	
USRP B200-mini	X	X		X		X
USRP B200	X					
USRP B210	X					
USRP E310		X				
USRP N210	X	X	X	X		
USRP X310	X	X	X			
USRP 2920				X		
USRP 2921					X	
USRP RIO 2942R					X	
USRP RIO 2943R	X				X	
USRP RIO 2952R (+ GPS)					X	
USRP RIO 2953R (+ GPS)				X		
WARPv2	X	X				
Xilinx ZC706 Evaluation Kit - ZYNQ® 7000 SoC + AD FMCMM radio frontend	X					X
ZedBoard Xilinx ZYNQ®-7000 SoC + AD FMCMM radio frontend	X	X				X
BB – NI PXI 7975 Module				X		
BB – NI PXI 7965 Module				X		

FE - NI PXI 5644				X		
FE – NI PXI 7976R					X	
PXIe 8135				X		
PXIe 8880				X		
NI PXI based mmWave base band + Sibeam V-band transceiver				X		

*Legend: BB (baseband motherboard), FE (Frontend, RF daughterboard)

3.2 SDR platform integration into testbeds

3.2.1 w-iLab.t

3.2.1.1 USRP B200/B210

This year w-iLab.t has added 2x USRP B200 and 2x USRP B210 devices. The B series USRP are needed by experimenters of the 1st open call, since they need to use the srsLTE software to collect statistics of LTE traffic. The srsLTE requires the USRP to run at a specific master clock rate, which were not supported by USRP N210 devices.

A particular difficulty we faced when deploying the B series USRP is that they are connected over USB 3.0 interface to the host node, however this connection prevents the USRP to be remotely power cycled, because the host USB port is always powered even when it is shut down. To resolve this issue, a remotely controllable USB 3.0 hub is placed in between the USRP and the host PC. The driver of this USB hub allows to power cycle each USB port, which is ideal for our purpose.

It is also verified that by placing the USB hub, the bus speed is still fast enough to support srsLTE software, so it is transparent to the users. Detailed instructions are provide in the tutorial of w-iLb.t documentation, “Using the USRP B200/B210 devices”. The following screenshot shows part of the tutorial content.

To enable fast USB3.0 connectivity and provide enough processing power to perform the srsLTE experiments, a new node type was added to the w-iLab.t testbed. In total 10 Intel NUC nodes were added to the testbed, hosting an Intel i5 processor and 8GB of DDR3 RAM. Two of these Intel NUC nodes were deployed on mobile nodes, in combination with a USRP B200-mini. This provides the experimenters with the possibility of performing mobile srsLTE experiments, or any other type of experiment that requires a USRP device.

Activate the USB ports

The USRPs are connected through a USB3.0 hub, of which the ports can be remotely switched ON/OFF. By default, all USB ports are turned off.

To turn ON the USB port and thus enable the USRP, please run:

```
#check if a USB Hub is present
ykushcmd ykush3 -l
#bring up port 1
ykushcmd ykush3 -u 1
#run UHD test
uhd_usrp_probe
#try a UHD benchmark (should result in 26 Msps)
/usr/lib/uhd/examples/benchmark_rate --args="master_clock_rate=52e6" --rx_rate 30.72e6 --tx_rate 30.72e6
#if no devices are found, try to toggle the USB port OFF / ON
ykushcmd ykush3 -d 1
ykushcmd ykush3 -u 1
```

Figure 2 Using USRP B200/B210 via USB hub in w-iLab.t

3.2.1.2 USRP X310 JTAG programming support

Although USRP X310s were available in the previous year, the Joint Test Action Group (JTAG) programming functionality is added only in this year. This is driven by the specific need of Opencall 2 for extension – Partial reconfiguration of Field-Programmable Gate Array (FPGA) on an SDR device. The USRP X310’s 10G Ethernet port is connected to a switch, which can then be connected to any of the micro servers defined by jFed topology. Though its JTAG programming cable is statically mapped to the USB port of a nearby node. This means users whoever wants to use JTAG programming interface will need to include the specific node in the jFed experiment. Xilinx Vivado Labtools are installed in a common network mounted storage, which contains necessary tools for JTAG programming of the USRP X310 device. Despite the “successfully programmed” print out of the software tool, we notice that initially the FPGA image downloaded by the JTAG interface was not effectively loaded. Through trial and error, eventually we notice the 10G Ethernet port must be deactivated during JTAG programming. These instructions are now part of a tutorial in w-iLab.t documentation. An screenshot of the relevant part is given below:

- If you need to use jtag to program the USRP X310, please follow these steps:
 - In the ssh terminal of server, execute `sudo ifconfig enoX down`, modify X accordingly, you can bring the 10G interface connected to USRP down.
 - APU-U2 and APU-V2 are connected to USRPX310-1 and USRPX310-2 respectively, double click on the APU node corresponding to your USRP in jFed experiment GUI to log in, in the ssh terminal, execute the following:
 - `source /share/xilinx/rfnoc/src/uhd-fpga/usrp3/top/x300/setupenv.sh --vivado-path=/share/xilinx/Vivado`, this will setup necessary environment to do jtag programming.
 - `viv_jtag_program /path/to/your/bitstream/filename.bit`, use the `viv_jtag_program` tool to download bit stream by specifying it as the first argument. **WARNING: This command takes several minutes to return, likely because it calls Vivado behind the scene which is installed on a network mounted storage, please wait till it finishes.** When it returns, you should observe output similar to the following on the terminal:


```
INFO: [Labtools 27-2285] Connecting to hw_server ur1 TCP:localhost:3121
INFO: [Labtools 27-2222] Launching hw_server...
INFO: [Labtools 27-2221] Launch Output:
- Target: localhost:3121/xilinx_tcf/Digilent/251635F5C0FFA
- Device: xc7k410t_0
- Filename: /tmp/usrp_x310_fpga_rfnoc_H6.bit
```
 - After `viv_jtag_program` returns, go back to the server and bring the 10G Ethernet interface up by execute `sudo ifconfig enoX up`, similar as before, replace X accordingly.
 - Ping the USRP or use `uhd_usrp_probe` to test if the image is loaded correctly.

Figure 3 Using JTAG to program USRP X310

3.2.1.3 USRP X310 RFNoC

The Radio Frequency Network-on-Chip (RFNoC) is a branch of UHD driver for USRP, that enables users to design its processing blocks running on the FPGA of the 3rd generation USRP devices, such USRP X310. To ease the usage of USRP X310 with RFNoC in the w-iLab.t testbed, an Operating System (OS) image is prepared with a recent RFNoC version installed. Although this seems very straightforward, the OS image turned out to be quite big. The compiled RFNoC software takes up approximately 8GB, in addition, some prerequisite libraries also take up more than 2GB. The large size of the image causes the loading process to time out. This issue is reported by the Open call 2 for extension’s users. As a resolve, the RFNoC source files and compiled files are removed from the image,

a back-up of the complete source file is made available on the network mounted storage. After this intervention, the image loads smoother.

3.2.1.4 TAISC SDR functionality support

TAISC is a MAC design tool originally for sensor node, however has been ported to ARM and now running on ZYNQ SDR. In Year 2 new ZED boards with FMCOMMS2 frontend are deployed in the testbed. We prepared two tutorials: one is more general, it is about how to program the ZED boards in the testbed or connect to serial terminal so on; while the other tutorial is about how to use TAISC tools on the SDR. The 1st tutorial is combined with the ZC706 board, as both carrier boards are programmed in a similar way. The 2nd tutorial is so far focusing on the software environment setup, such as how to setup TAISC and Contiki, this tutorial will be continuously improved when more and more advanced SDR features are integrated into TAISC.

3.2.1.5 Integrated Logical Analyser (ILA) support in w-iLab.t

During the support of Open call 1 for extension, the users have requested tools other than printing out debug messages on the serial terminal. As a result, we have installed Vivado Web Edition 2017.4. This software is license free, however still allows the user to connect hardware manager to the board, downloading a bitstream, and then import a probe file to visualize the waveform of signals running in FPGA. The usage of this tool is almost identical as if the tool is running locally, with the only difference to upload probe file/bitstream that are generated elsewhere. Note that the testbed does not support generation of bitstream mainly due to Xilinx software licencing issue. This new feature is documented in the testbed tutorial, a screenshot of relevant part is given below.

- If you want to use the ILA (Integrated Logic analyzer) of Xilinx for debugging, you also need to upload the probe file to the server. It is usually located in the `.run` folder of Vivado project, with extension `.ttx`, in the example project this is the `debug_net.ttx` file.
 - You may open a new terminal on the server, becoming root by executing `sudo bash`
 - Set the environment variable of Xilinx by executing `source /share/xilinx/2016_2/Vivado_Lab/2016.2/settings64.sh`
 - Please note that there is also Vivado 2017.4 installed, in case you need version 2017.4, please execute `source /share/xilinx/Vivado/2017.4/settings64.sh` instead of the above
 - Launch the vivado lab tool by executing `vivado_lab`
 - You should observe the vivado lab tool GUI pops up, the following refers to operation on the GUI
 - Click on **Open Hardware Manager**, **Open target**, and then **auto connect**
 - Suppose you already programmed the device as indicated in the previous step, simply specify the probe file by clicking on the **Specify the probes file** and **refresh the device** in the *Trigger Setup* window
 - Select the `.ttx` file you have uploaded to the server and click on **Refresh**
 - You should now be able to view the probe names in the *Waveform* window.
 - Feel free to define a trigger condition, and click on trigger button, you can then observe the waveform to debug
 - In the example project, a debug probe is inserted on the 4x LED peripheral, you may trigger on the LED port and observe one of the LED pin goes high, as indicated in the following screenshot.

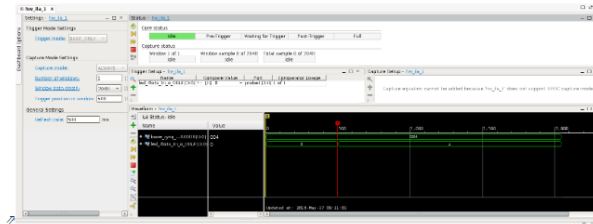


Figure 4 Integrated Logical Analyser (ILA) support in w-iLab.t

3.2.1.6 Pending testbed issues reported by Open Call partners

During the support of Open call 1 for extension, we noticed some hardware stability issues, as one server connected to the ZYNQ SDR randomly crashed. After some investigation, we believe the `xsd` binary of Xilinx Vivado Lab tool could cause the Linux kernel to panic. This binary is used to download FPGA bitstream or firmware to the board, which is often executed.

The servers which are currently used in the w-iLab.t testbed are supplied by SuperMicro and host an Intel Xeon processor which can support up to 16 threads. While these servers are quite powerful, and host 2x 10G Ethernet interfaces, they don't seem suited to run the Vivado Lab software. The testbed administrators are still investigating the issue. Possibly the errors are related to memory incompatibility, or other memory related bugs in the Vivado software. The issues are expected to be resolved at latest by the start of the second open call of ORCA.

3.2.2 IRIS

The IRIS testbed operates a premier open access reconfigurable 5G radio testbed in Europe, which is hosted at the CONNECT centre in TCD. Currently, we make 22 USRPs available for remote use by experimenters. Users can instantiate Virtual Machines (VMs) in our cluster of servers with tools that realise many different SDR, and Software Defined Networking (SDN) research and testing configurations for 4G and 5G networks. In the past two years, we have had more than 2000 instances of VMs provisioned by external experimenters, and a further 1000 from internal TCD staff and students. Using these configurations, users have performed tens of thousands of experiments. Typical external experimenters are from SMEs and academic research institutions in Ireland, Spain, France, Italy, Poland, Greece, Colombia, Brazil, and the USA.

3.2.2.1 Updates in Year 2

The IRIS team regularly update and upgrade our hardware, and SDN and SDR images. This section documents some of the more notable software/hardware upgrades in 2018.

The following equipment was added to the IRIS testbed in 2018:

- 2 x Dell PowerEdge R440 Servers, each with two Broadcom 57416 Dual Port 10 GbE BaseT Network LOM Mezz Network Cards.
- 1 x ASUS RT-AC53 Dual-Band Wireless AC750 Gigabit Router, Superfast 802.11ac Wi-Fi Router with 3 External Antenna, Wider Coverage, VPN Support.
- 20 x DELTA7A/X/SMAM/S/S/11 - Hinged Mount WiFi / ISM Antenna, 2.4/5.8GHz, Direct Mount, SMA Male.
- 1 x Xiaomi Redmi 5 Plus [Official Global Version] 5.99 Inch 4G LTE Smartphone 18:9 Full Screen MIUI 9 4GB 64GB Snapdragon 625 Octa Core 12.0MP Cam Touch ID 4000mAh Battery - Black.
- 10 x sysmoUSIM-SJS1-4FF SIM USIM Java 64k nano (10-pack) (901-70 MCC-MNC).

The following software elements were added/upgrade in 2018:

- srsLTE EPC with srsLTE eNodeB: The srsLTE SDR image was upgraded to include the srsLTE EPC, and srsLTE eNB software. This image is compatible with the Xiaomi Redmi 5 Plus device and with any one of the ten Sysmocom USIM SIM cards.
- Migration to OpenStack: To provide continuous testbed upgrade, improvement, and stability for experimenters, the IRIS testbed has started moving towards the OpenStack cloud platform. Currently, the IRIS team has deployed OpenStack (Rocky) with KeyStone, Glance, Nova, Neutron, and Horizon, supported by two Dell PowerEdge R440 servers (acting as Controller and Compute nodes) with 10GB Ethernet Connection (SR-IOV) to four N210 USRPs on Dell S4048T-ON SDN 1.3 switch support by ONOS. The goal is to migrate all IRIS's Cloud Based Testbed Management (CBTM) functionality in jFed to OpenStack in 2019.

3.2.2.2 Issues experienced by Open Call partners

During the ORCA OC EXP-1 and EXT-1, the experiments and extension providers encountered some issues related to the stability of the IRIS testbed via jFed. These issues were primarily caused by the static nature of IRIS's Cloud Based Testbed Management (CBTM) system, and the number of new VM images required by experimenters/extension providers. Currently at IRIS, adding new VM images requires a restart of the jFed Aggregate manager. This caused some users to lose their experiment sessions.

Furthermore, the volume of experimenters, both from the ORCA OC EXP-1 and EXT-1, internal TCD staff and students, and from other H2020 projects including FUTEBOL and Fed4FIRE+ using the IRIS testbed in parallel, caused some load issues on the servers. Occasionally these issues required server/full system restarts. In general, these issues had minor impacts on experimenters, as they were notified in advanced, and restarts were scheduled in the evenings/nights and at weekends.

In general, most of the experimenters were satisfied with the standard IRIS testbed capabilities and equipment available via jFed. However, several experimenters needed more powerful processing capabilities, and additional wireless experimentation equipment. These included:

- Wi-Fi UE access to WiFi router
- Access to multiple UE nodes
- Use of B200 USRPs
- Much more powerful CPU for processing Radio Access Network (RAN) sharing mechanisms on the eNodeB.

At IRIS, we provided access to these resources with laptop computers. This temporary solution seems to have resolved the requirements of the experimenters. However, it raises the question if we can provide this functionality via our cloud platform in the future. Towards resolving many of these issues, we have started to migrate our CBTM platform to OpenStack, as discussed above.

3.2.3 TUD OWL testbed

In order to increase the performance, security and robustness of the OWL testbed, we brought some hardware and software updates into the testbed.

3.2.3.1 Online user tutorial

The OWL testbed uses the Remote Desktop Protocol to provide access to the resource nodes. In order to make it easier for the experimenters to use our testbed, we provide the online user tutorial on our webpage¹ and a video tutorial in our YouTube channel². The experimenters can follow the tutorial to setup an experiment and access into the resource nodes systematically.

3.2.3.2 OS image selection function

In last year, each raw PC in our testbed had only one fixed OS. Which means that, if the users wanted to use another OS, they needed to reserve another resource node.

¹ <http://owl.ifn.et.tu-dresden.de/testbed/tutorial/>

² <https://www.youtube.com/watch?v=wld5MbXm4U>

In order to improve the functionality, one raw PC can now be used for different OS image. If the AM can write another image onto the same raw PC, the users just need to select the image, instead of to reserve an additional resource node. The image selection function makes the use of resource node more efficient.

In order to support the OS image selection function, we adjusted the advertisement RSpec of our AM, which enables the image selection function in jFed. The “ListResource” call on the F4F federation monitor webpage can get the reply containing the information of the available images in our AM. We have prepared Windows 7, Ubuntu 16.04 as well as the NI real-time Linux OS in the local server, the AM can restore the by user in jFed selected image to the resource node.

3.2.3.3 Multi virtual server and VLAN

Considering the privacy and security, in this year we implemented more virtual machines as virtual experiment servers and completed the Virtual Local Area Net (VLAN) management in the testbed, to separate the experiment behaviours of different F4F project groups.

The OWL testbed stores the specific disk image of the resources in local server for each F4F project group. The user should not be able to visit the disk images of the other project group. Each individual virtual server works only for one F4F project group and stores the disk images for its user group. The separated virtual servers keep the disk images not visible by the user from another project group.

During the experiment, the operation of one group should not disturb the other groups. For this purpose, we install individual AM in each virtual server, where we have developed the AM gateway. The AM gateway receives all the jFed experiment commands, checks the F4F username in the commands, separate the commands and forwards them to the respective virtual servers. The AM in the virtual server executes the jFed commands, without any affect to the other project groups.

The users should not be able to access the resource nodes reserved by another group. For these purposes, we implemented the VLAN manager and employed a programmable network switch. The VLAN manager checks the reservation status of the resource nodes, reconfigures the ports on the network switch, based on the reservations. The ports on the switch allocate the connected resource nodes to different VLANs. The VLANs separate the remote accesses of different experiments and avoid unexpected access to the resources in use. The diagram of the multi-AM is depicted in Figure 5.

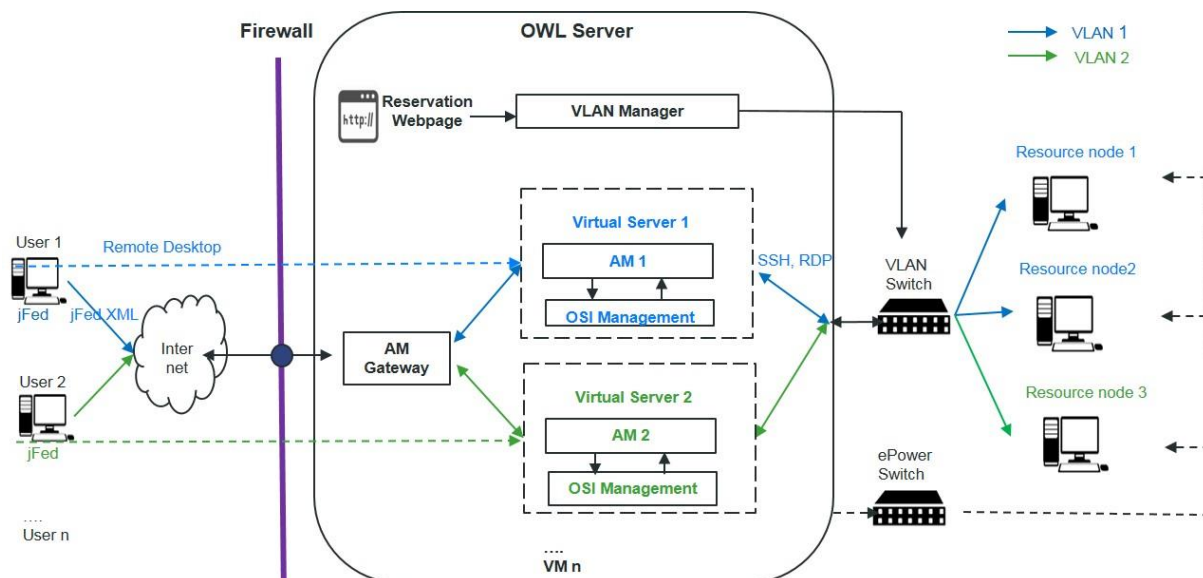


Figure 5 Multi-AM in OWL Testbed.

3.2.3.4 Server cluster

In our testbed, we provide individual virtual machines and AM servers for each F4F project group. The parallel working virtual machines request high CPU and disk resources, especially when the AMs are loading or saving OS images to or from the testbed resource nodes. To increase the performance of the virtual machines, we added an additional physical server and built a load balancing and failover server cluster, based on the Microsoft hyper-V. The virtual AM servers are operated in this system.

Load balancing in the server cluster can optimize the CPU use and minimize response time. The Failover feature makes sure that the AM services will not be interrupted, in the case that one individual server fails and need maintenance.

We use the Microsoft Data Protection Manager (DPM) to generate backups of the server regularly, and thus, protecting the data of the testbed.

3.2.3.5 MultiRat Platform

In order to enable the prototyping of all the layers of LTE and 802.11 protocol and satisfy the needs of the user from open call 1 for extension, we combined additional controllers and USRPs this year to build the two MultiRat Platforms. Each MultiRat Platform, which is depicted in Figure 6, contains 2 USRPs and 2 controllers: The NI USRP RIO 2953R with controller NI PXIe-8133 were already available in our testbed, then we added an extra controller NI PXIe-8880 and the NI USRP RIO 2954R to build this prototyping platform, whose setup can be listed as:

Hardware:

NI PXIe 1082 with controller NI PXIe-8133, or with controller NI PXIe-8880,

NI USRP RIO 2953R, NI USRP RIO 2954R,

Antenna VERT2450, SMA cables

Software:

OS: Windows 7, Linux (LV Real time)

Other: Labview Communications 2.0, Matlab, ns-3

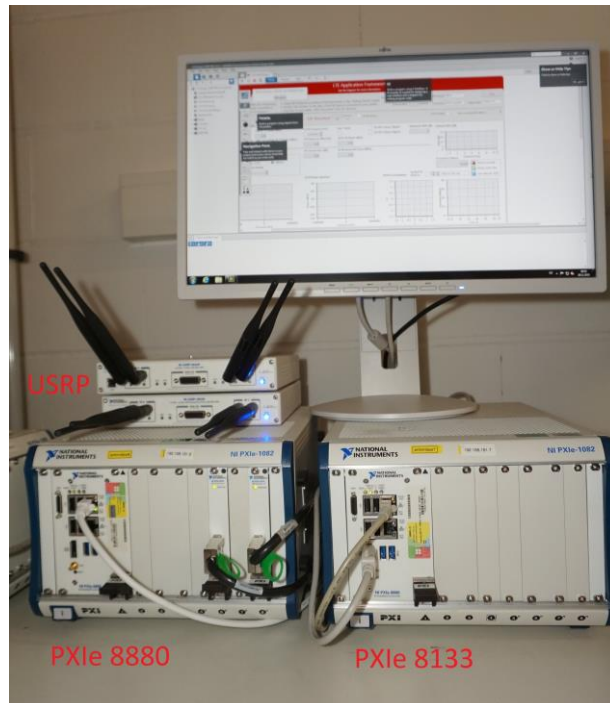


Figure 6 MultiRat Platform

We installed on the controller PXIe-8880 the software packet, which was provided by NI Dresden. Currently the NI PXIe-8880 has the operating system NI real-time Linux, with the network simulator NS-3 installed. NS-3 enables the test of upper-layer protocols, such as LTE Layer 2 and 3 and Evolved Packet Core (EPC) over the real-time simulation. The controller NI PXIe-8133 supports the NI 802.11 and LTE Labview framework, which implements the PHY Layer on the NI USRP RIO. The USRP RIO is composed of FPGA as well as the RF frontend, and enables the over-the-air transmission.

Currently we provide two MultiRat Platforms in our testbed. Figure 7 illustrates the setup of the 2 MultiRat Platforms. Using the platforms, the user can setup a LTE-WLAN Radio Aggregation (LWA) simulation. NI Dresden provides a video tutorial regarding using this platform in our testbed. This video tutorial is attached on the webpage of our testbed.

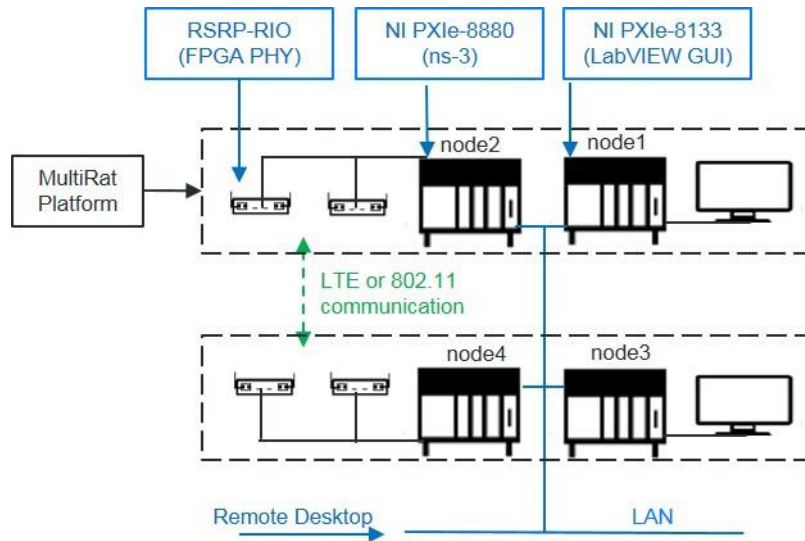


Figure 7 Hardware set up of MultiRat Platform.

3.2.3.6 USRP b205 mini

In year 2, we included one Linux PC Host with USRP b210 mini in the testbed, to satisfy the need of the experiment of the 1st call. The experimenter provides the Ubuntu operating system image (OSI) with necessary development tool software such as Matlab, GNURadio and UHD. We implemented the compatibility of our Aggregate Manager (AM) with Linux Operating System, and configured the experimenter's OSI to adjust it to our testbed



Figure 8 Resource node with USRP b205 mini

3.2.3.7 Node with NI USRP 2920

In order to further support the experiment from the first open call for experiments, we combined one PC with 2 USRP 2920 and 1 USRP b205 mini as a new resource node in our testbed. This node consists of the following components:

Hardware:

PC with intel i7 processor, USRP 2920

SMA cable, MIMO cables

Software:

OS: Windows 7, Linux (Ubuntu 16.04)

Other: Labview Communications 2.0, Labview 2015

The two USRP 2920s are connected to the PC Host by a Ethernet cable. A MIMO cable synchronizes the two USRP 2920s. The two USRP 2920 and the Labview Communications 2.0 or Labview 2015 compose a SDR Platform.

We provided both WIN7 OSI and Ubuntu 16.04 OSI images for this resource node. One USRP 205 mini is also connected to this PC via USB. If the user select the Ubuntu image in jFed, this node can also work as the node with USRP b205 mini, which is introduced in last sub-section. The setup is shown in Figure 9.

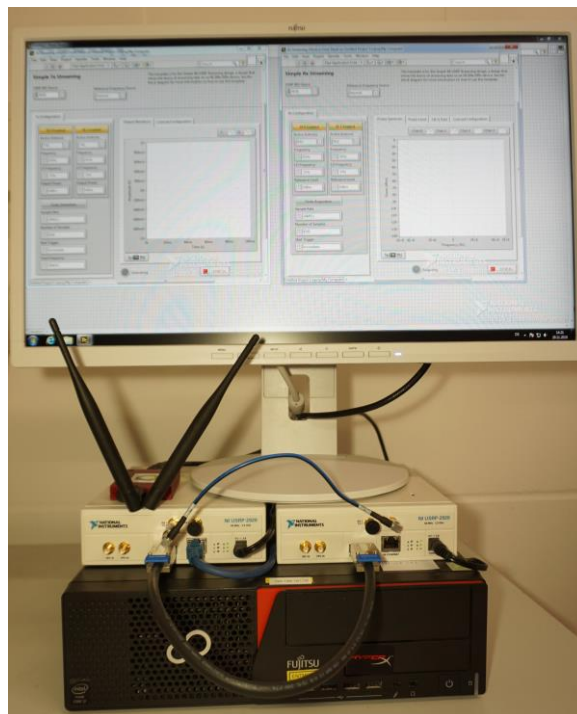


Figure 9 Resource node with two USRP 2920

3.2.4 KUL testbeds

3.2.4.1 Full-Duplex testbed

In ORCA Y2, the KU Leuven in-band full duplex (IBFD) testbed is updated by the enhanced version of the cross-layer PHY-MAC architecture [2] which is implemented by LabVIEW FPGA and enables low-latency and reliable networking. Moreover, hardware and software modules were developed in LabVIEW FPGA and C++, respectively, and they can deliver similar functionality. This allows an advanced level of trade-off between latency performance and flexibility.

The PHY layer is also enhanced to enable realistic applications where the IBFD communication node can be affected by the dynamics in the environment. To this end, we implemented the dithered linear

search (DLS) adaptation algorithm to optimize the analog self-interference canceller, i.e. the electrical balance duplexer (EBD). This approach has reduced the EBD tuning time to 0.124 ms.

Furthermore, the digital self-interference cancellation (DiSIC) module is integrated into the testbed. The DiSIC is realised by Tampere University of Technology (TUT) in the framework of ORCA first open call. More development is required to improve the joint analog and digital self-interference cancellation, achieving the optimum self-interference suppression.

3.2.4.2 KUL Massive MIMO testbed

During the second year the KU Leuven Massive MIMO testbed joined F4F. One indoor experiment was carried out successfully, controlling the base station remotely. For this particular case a person in-situ was required to help with the deployment of the antennas array and the users.

External power amplifier

For the outdoor MaMIMO experiment described in D5.3 a hardware and software extension was implemented to boost the user power in KU Leuven Massive MIMO testbed. SZHUASHI power amplifiers were integrated by pairs to a USRP-2942R, acting as a bridge between the USRP antenna port and each dipole antenna. Figure 10. In addition an external DC power supply set at 12V was connected to the power amplifier.

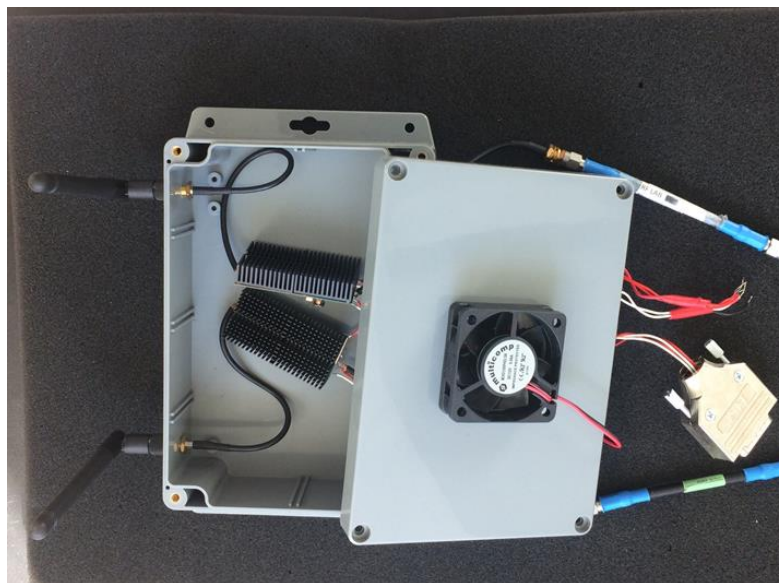


Figure 10. External power amplifier for users in KUL Massive MIMO testbed.

The KUL Massive MIMO testbed is based on a variation of the LTE-TDD standard and operates at 2.6GHz as central frequency with 20MHz bandwidth. As SZHUASHI power amplifier specifications (Table 2) meets all the testbed requirements, it was implemented without any other hardware constrain.

Table 2 User power amplifier for KUL MaMIMO.

Frequency range:	2.4 – 2.6GHz
Operating voltage:	6 - 16V

Received gain:	10dB
Transmission gain:	13dB
Input trigger power:	3dBm – 20dBm
Maximum output power:	36dBm
EVM:	3% at 28dBm 802.11g 54Mbps OFDM 64-QAM BW 20MHz
DC Supply current:	435mA

To control the time switching between transmitter and receiver, the control code was deployed in LabVIEW Communications MIMO Application Framework 1.1 which runs in the USRP-2942R.

3.2.5 Portable testbed

During the second year of ORCA, the portable testbed was extended with some extra SDR hardware. The portable testbed, as an outcome of the WiSHFUL project, already hosted USRPB200, USRPB210 and USRPB200-mini devices. During this year the Xilinx Zynq@-7000 was added to the portable testbed. The integration of the device into the portable testbed is shown in the figure below. The Zync interfaces with gigabit Ethernet towards a switch, which provides connectivity between the Zync board and the Intel NUCs. For now, no other powerful server hardware is added to the portable testbed, because it is believed that the specifications of the Intel NUCs will be sufficient. Feedback from the open call experimenters will be used to confirm or deny this assumption, and other hardware will be added if necessary.

To protect the Zync board during transportation, and to make it easier for the experimenters to handle the device for portable experiments, a new enclosure was designed. The enclosure is made out of plexiglass and is customized by making use of a laser cutting machine. The design can be seen in Figure 12.

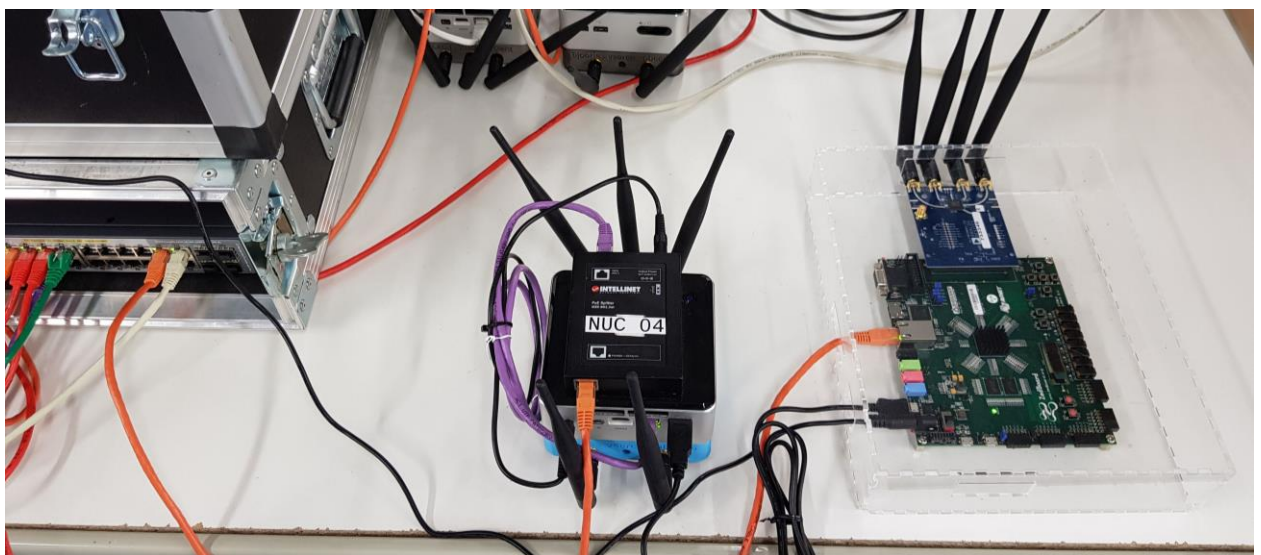


Figure 11 Zync SDR in portable testbed.

The Zync board can reserved through jFed and its usage is identical as the devices installed in the w-iLab.t testbed.

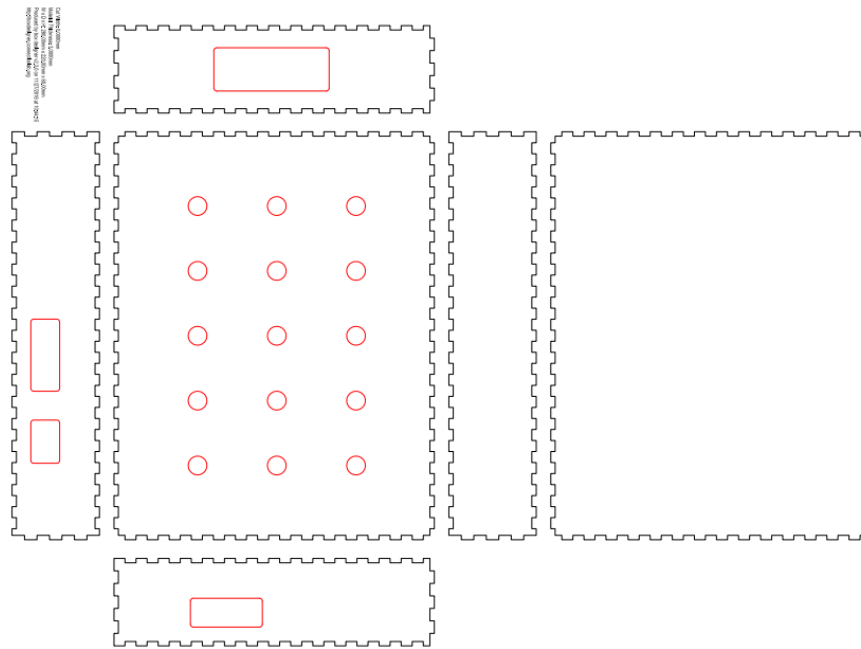


Figure 12 2D design of Zync SDR enclosure.

4 CONCLUSIONS

This deliverable reports the effort of testbed development in the second year of ORCA project. The ORCA facility consists of 5 testbeds: (i) the w-iLab.t testbed from IMEC, (ii) the KUL testbed, (iii) the ORBIT testbed from RUTGERS, (iv) the IRIS testbed from TCD, (v) the OWL testbed from TUD and the (vi) IMEC portable testbed. Although all 6 testbeds have basic features to be F4F compliant in Year 1 of ORCA, in this year TUD and KUL testbeds added the support for F4F monitoring functionality, which further improves the compatibility.

In addition, each of the partners has brought in new functionalities for SDR related experimentations, including the deployment of new hardware, and or support for new experimentation tools on the existing SDR platforms. In Section 3, an overview is given for the supported SDR tools in the ORCA facility, followed by a more detailed description of each of the partner's new SDR facilities. In conclusions, in this year of ORCA, more advanced functionalities of SDR has been integrated in to the ORCA testbed.

REFERENCES

- [1] FED4FIRE Federation Monitor, available at <https://fedmon.fed4fire.eu/overview>, last access December 5th 2018
- [2] Seyed Ali Hassani and Sofie Pollin, "A Low-Latency Wireless Network for Cloud-Based Robot Control". CROWNCOM 2018, September 2018, Ghent, Belgium.