HyDRA: A Hypervisor for Software Defined Radios to Enable Radio Virtualization in Mobile Networks

Maicon Kist*, Juergen Rochol*, Luiz A. DaSilva‡, Cristiano Bonato Both†
*Federal University of Rio Grande do Sul, Brazil, ‡Trinity College Dublin, Ireland,
†Federal University of Health Sciences of Porto Alegre, Brazil
{mkist, juergen}@inf.ufrgs.br*, dasilval@tcd.ie‡, cbboth@ufcspa.edu.br†

***Abstract*—In this article, we present HyDRA, an SDR virtualization layer that enables the execution of multiple air-interfaces. HyDRA multiplexes raw IQ signal samples of multiple virtual RF front-ends into a single stream that can be transmitted. We also describe a proof-of-concept demonstration of HyDRA multiplexing IQ samples generated by LTE and NB-IoT SDRs into a single waveform.**

I. Introduction

The next generation of mobile networks is envisioned to provide connectivity services to a multitude of devices with vastly different requirements, ranging from mobile subscribers with high-bandwidth services (*e.g.,* video-streaming) to IoT devices with bursty traffic and low-bandwidth services (*e.g.,* utilities metering). To efficiently support such services future mobile networks should be flexible, providing different air-interfaces for particular users and applications [1], *e.g.,* with base stations capable of providing connectivity to LTE mobile subscribers and NarrowBand-IoT (NB-IoT) devices. In this context, radio virtualization is a promising solution to enable multiple virtual radios, *i.e.,* air-interfaces, to coexist on top of one base station [2] [3].

Radio virtualization requires the development of an hypervisor [4]. The hypervisor must abstract the base station RF front-end into a number of virtual RF front-ends, which are accessed by SDRs. As SDRs generate a stream of raw IQ samples to the virtual front-ends, the hypervisor needs to multiplex it into a single waveform that will be transmitted by the physical RF front-end. Although radio virtualization is currently being considered in state-of-the-art architectures, an implementation is still lacking in the literature. Such implementation is essential to correctly analyze its benefits and drawbacks on future mobile wireless networks, *e.g.,* decrease in the SDR throughput due to the overhead of using a virtual RF front-end.

In this paper, we describe and demonstrate the implementation of HyDRA (HYpervisor for software-Defined RAdios). The basic idea of our design is to receive raw IQ samples from multiple virtual RF front-ends coexisting in the same RF front-end and use software baseband processing to multiplex these samples into a single transmitted waveform. HyDRA is implemented purely in software and fully extensible to receive raw IQ samples from any SDR.

II. HyDRA Design Choices and Implementation

The core function of the hypervisor is to multiplex the incoming raw IQ samples of each virtual RF front-end into a single waveform that is transmitted by the physical RF front-end. This process must satisfy the following requirements:

- the multiplexing must be air-interface agnostic to support different access technologies. This means that the multiplexing can modify the IQ samples of virtualized RF front-ends only using generic signal processing algorithms;
- modifications in the original signal caused by the multiplexing cannot be distinguishable from well-known wireless disturbances, *e.g.,* path-loss, frequency shift, and phase distortion. This requirement ensures that receiving devices can recover the data transmitted using common physical layer equalization mechanisms;
- the multiplexing must add as little overhead as possible.

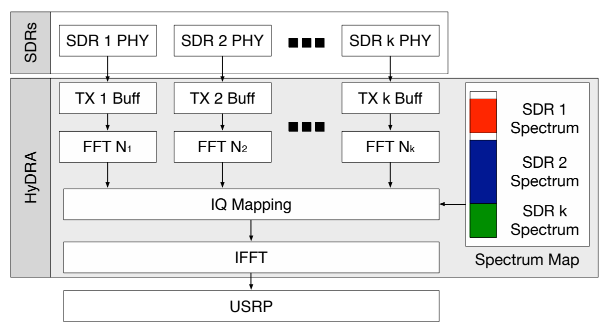

Fig. 1. Main architectural blocks of HyDRA.

Based on these requirements, we designed the architecture of HyDRA as shown in Fig. 1. As HyDRA abstracts the RF front-end, SDRs send a stream of baseband samples that represent the intended radio signal to our hypervisor. Moreover, HyDRA enables SDRs to configure on-the-fly the central frequency, bandwidth, and sampling rate with ensured isolation, *i.e.,* only the SDR that requested a configuration will be affected. To keep track of these configurations for each virtual radio, HyDRA makes use of a spectrum map.

Our multiplexing is based on FFT/IFFT operations, as shown in Fig. 2. First, the incoming raw IQ samples are transformed from time to frequency domain in an FFT with N_i points (N_i is a function of the bandwidth of the virtual RF

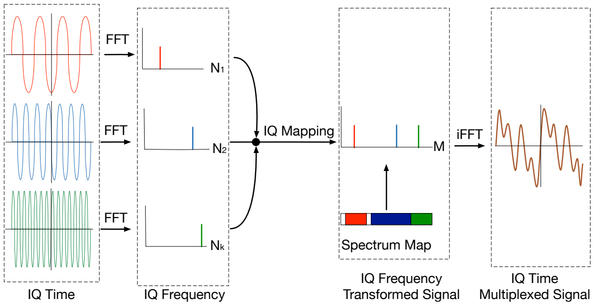

Fig. 2. Multiplexing process performed by HyDRA.

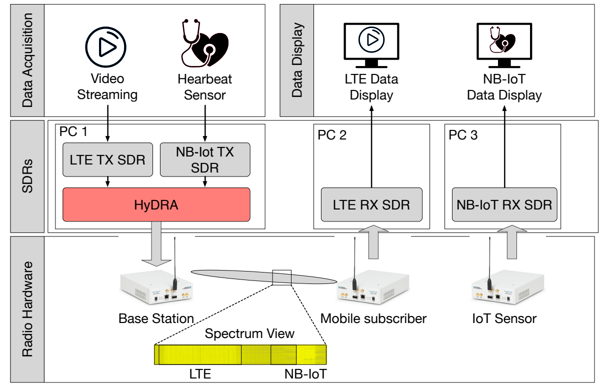

Fig. 3. Demonstration framework.

front-end i and the sampling rate of physical RF front-end). After that, the resulting N_i frequency components are mapped into the buffer of an IFFT M according to the spectrum map. After the mapping, we perform the IFFT to convert the frequency components of all virtual RF front-ends into the resulting multiplexed time domain signal, which can be transmitted by the physical RF front-end. The main advantage of adopting FFT/IFFT as the basis for multiplexing is its efficient execution on modern processors, *e.g.,* by making extensive use of Single Instruction Multiple Data (SIMD).

III. Demonstration Flow

We adopted as the scenario for our demonstration the vision in which future mobile base station may provide connectivity services to LTE mobile subscribers and NB-IoT devices, as illustrated in Fig. 3. It consists of (I) one USRP acting as the base station multiplexing raw IQ samples LTE and NB-IoT SDRs, (II) one USRP acting as the LTE mobile subscriber, and (III) one USRP acting as the NB-IoT healthcare data receiver. At the base station side, a dedicated computer executes HyDRA with two virtual RF front-ends; one is assigned to LTE transmitter SDR and another to the NB-IoT transmitter SDR. Similarly, we have one dedicated computer for the LTE and NB-IoT receivers. Table I summarizes the principal parameters for HyDRA, the LTE SDR, and the NB-IoT SDR.

To demonstrate HyDRA multiplexing the LTE and NB-IoT SDRs into a single waveform, we selected two services aligned with the capabilities of each air-interface: a constant high-bandwidth video-streaming for the LTE, and a low-bandwidth heartbeat rate notification for the NB-IoT. We have two displays to show the video-streaming received in the mobile subscriber and the heartbeat notification in the NB-IoT receiver. The heartbeat rate transmitted over the air is collected in real-time from participants willing to interact with the demonstration. We control some parameters of HyDRA and the LTE and NB-IoT SDRs using WiSHFUL Unified Programming Interfaces (UPIS) [5].

The source code of HyDRA is developed in C++ as an out-of-tree module for GNURadio, *i.e.,* it executes on any computer with GNURadio binaries and is recognized by GNURadio as a standard "block". It is publicly available online at GitHub (github.com/maiconkist/gr-hydra). The source code is accompanied with several examples with configurable parameters such as the number, central frequency, and bandwidth of virtual RF front-ends, and GUI elements such as spectrum waterfall and plotters to see data transmitted and received.

IV. Concluding Remarks

We demonstrate the feasibility of radio virtualization as a mechanism to easily and fast provide connectivity services in the next generation of mobile networks. To this end, we designed an hypervisor that capable to create multiple virtual RF front-ends and to multiplex IQ samples into a single waveform that can be transmitted by a physical RF front-end. Our demo is aligned with standardizations efforts made by 3GPP that consider a base station providing connectivity services to LTE and NB-IoT devices.

TABLE I
HyDRA, LTE and NB-IoT configurations

Parameter	Value(s)
HyDRA	**CF**: 5.5 GHz **BW**: 4 MHz, **FFT**: 4096
LTE VR	**CF**: 4.999 GHz, **BW**: 1.4 MHz, **FFT**:128, **CP**: 7 symbols (short), **MOD**:QPSK
NB-IoT VR	**CF**: 5.5001GHz, **BW**: 200 KHz, **FFT**:64, **CP**: 7 symbols (short), **MOD**:BPSK

Acknowledgements

This work has received funding from the European Union's H2020 for research, development, and demonstration under grant agreement no. 645274 (WiSHFUL).

References

[1] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: A Software Defined Networking Architecture for 5G Wireless Systems," *Computer Networks*, vol. 85, no. 7, pp. 1–18, 2015.

[2] T. Taleb *et al.*, "EASE: EPC As a Service to Ease Mobile Core Network Deployment Over Cloud," *IEEE Network*, vol. 29, no. 2, pp. 78–88, 2015.

[3] J. Liu *et al.*, "CONCERT: A Cloud-Based Architecture For Next-Generation Cellular Systems," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 14–22, 2014.

[4] C. Liang and R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, 2015.

[5] WiSHFUL project home page: http://www.wishful-project.eu.