ULLA-X: A Programmatic Middleware for Generic Cognitive Radio Network Control

Avishek Patra, Andreas Achtzehn, and Petri Mähönen
Institute for Networked Systems
RWTH Aachen University
Kackertstrasse 9, D-52072 Aachen, Germany
Email: {avp,aac,pma}@inets.rwth-aachen.de

***Abstract*—Classical cognitive radio concepts such as the cognitive radio manager or the cognitive engine aim to cleverly optimize radio configuration and networking setups to maximize performance throughout the network. However, these concepts regularly requires experimental verification and testing in complex, real-time and signal processing focused SDR frameworks, e.g. directly in FPGAs or in a software like GNURadio. This hinders rapid prototyping and deepens the gap between implementation-focused lower-layer and cross-layer/control plane oriented research. In this demo, we present a novel, fully-networked middleware for centrally controlling distributed radio link setups of SDRs and reconfigurable legacy devices. As an intermediate layer, ULLA-X allows technology-agnostic device parameter monitoring and reconfiguration based on an easy-to-learn programming language. Through its functionality, ULLA-X mitigates the complexities arising from using different radio platforms, and makes various configuration APIs readily accessible to standard research tools. In this demonstration, we showcase the capabilities of ULLA-X, and present by means of examples its adaptation capabilities for custom radio implementations and its programming concept for different SDR platforms.**

I. Introduction

Research activities in dynamic spectrum access and cognitive radio technologies often follow either of two goals: online optimization of links and networks to maximize instantaneous spectrum exploitation, or asynchronous optimization which aims at selecting transmission waveforms, link control algorithms and their parameters, or network protocols, to acquire best performance throughout the lifetime of a radio network [3], [8]. Unfortunately, particularly the latter is often difficult to research, because complex PHY/MAC implementations on diverse technology platforms need to be orchestrated. Orchestration is often done, for example, by a centralized cognitive engine [6] or radio resource management unit [2], [5], which require individual, technology-specific implementations that are difficult to develop, scale, and maintain.

With the aim to bridge this gap between control-plane and signal processing oriented research, we have developed ULLA-X, a generic middleware that provides access to SDR configurations and legacy control APIs for higher-layer researchers in a central, unified manner. Based on our previous exploration into unifying wireless media access [7], the ULLA-X platform offers a framework and a unified declarative programming language to change the behavior of radio devices during runtime based on asynchronous decision

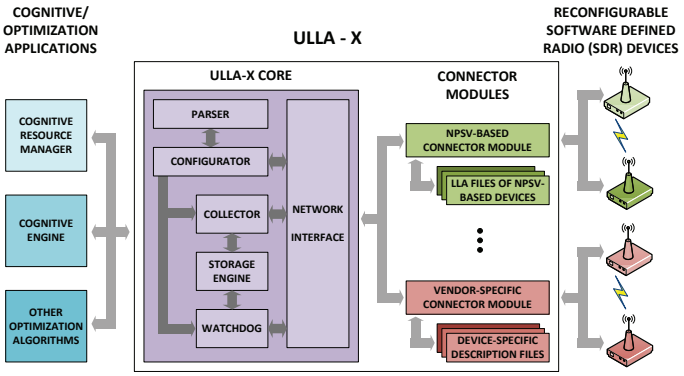

Fig. 1. ULLA-X Architecture Block Diagram.

making algorithms that operate in the central coordination unit. By establishing an adaptation layer between the radio device control APIs and the coordination unit-specific IDE, ULLA-X eradicates the need for deploying technology-specific optimization tools. Instead, it allows researchers to focus on the fundamental optimization tasks and algorithms for cognitive radio networks.

While the concept of a generic interface between optimization modules and cognitive radio network has been proposed earlier [1], [4], to the best of our knowledge we are the first to offer a practical prototype-based setup that can accomplish the decision-to-implementation bridging. In the following demonstration, conducted for a test-network of commercial SDRs (USRP-RIO and PXIe platforms from National Instruments) running various wireless standards, we illustrate the ULLA-X capabilities of direct, coarse real-time interactions with ULLA-X-controlled devices. Apart from the basic device-interaction primitives, necessary for monitoring and reconfiguration, the ULLA-X framework defines a programming language for conditional data collection, and automated event- and statistics-based reconfiguration. In the spirit of Mitola's original cognitive cycle, ULLA-X thereby resembles an "autonomous nervous system" for radios, primed by complex decision making algorithms, acting at a speed sufficient for optimizing long-term objectives. By facilitating such functionalities, ULLA-X provides a new class of control mechanisms architecturally located between comprehensive reasoning and decision-making algorithms and the instantaneous closed-loop control of wireless setups.

978-1-4799-7452-8/15/$31.00 ©2015 IEEE

II. ULLA-X Architecture and Programming

The ULLA-X middleware resides in between generic optimization modules and technology-dependent SDRs. It can be logically decomposed into the two parts: the ULLA-X core and a number of connector modules. The functionalities of these two parts are summarized as follows, and is shown are Figure 1:

1) ULLA-X core: The ULLA-X core is the operational center of ULLA-X. The configurator module inside the core receives commands in the ULLA-X programming language from research applications such as R, MATLAB, etc., where they are parsed by the parser module. In a real product, commands would be received from different processes, subsystems and resource managers of the system. Based on the command type, a collector periodically samples network states variables and, upon triggering by a watchdog, reconfiguration commands are sent to ULLA-X enabled devices. These may be bypassed for direct interaction with radio devices. For post-processing purposes, monitored values are kept in a storage module, where they can be accessed by the watchdog or from the user application.

2) Connector modules: Connector modules coordinate between the ULLA-X core and the network radio devices. Connector modules are device- and implementation-specific wrappers that serve two purposes: First, the modules map the generic access commands (parsed in ULLA-X core) to design-specific parametric details. Such mappings are enabled through simple implementation description files, called Link Layer Adapter (LLA) files, generated by the SDR-implementation designers. When parameters are requested for access through generic commands, the generic-to-specific mapping ensures seamless access. Second, following the mapping, the actual parameters are retrieved/updated by the connector modules and passed on to the ULLA-X core. Furthermore, connector modules are also capable of automatic device discovery and arbitration.

All interfaces between components of ULLA-X use standard network message passing protocols which enables distributing the logical units of over multiple network entities.

Commands to control ULLA-X-enabled devices are specified in the ULLA-X programming language. The commands, with easily comprehensible syntaxes, can be classified based on their functionalities: discovery & management, data definition, collection and & manipulation (parameter read and write), and triggered execution (conditional reconfiguration). The ULLA-X programming language closely follows other declarative programming language, e.g. SQL.

III. Demonstration Flow

For the demonstration, our test network (shown in Figure 2) will consist of (1) NI PXIe Chassis with FlexRIO FPGA core and RF transceiver, and (2) NI USRP-RIO (embedded FPGA core and RF frontend). The NI PXIe will be deployed with a OFDM-SISO based LTE-like PHY while the NI USRP-RIOs will run a variant of an IEEE 802.11 a/g/n/ac PHY.

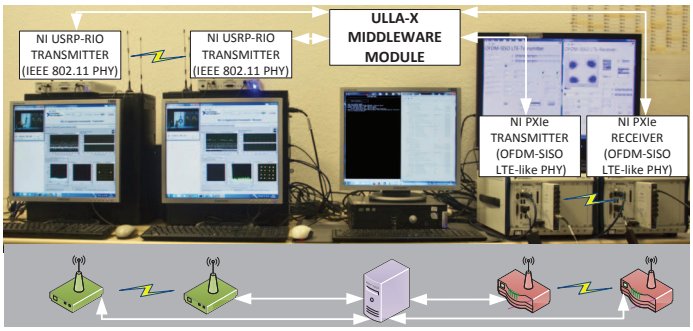

Fig. 2. ULLA-X Demonstration Set-up.

ULLA-X is developed using ANSI C and runs on Windows. We note that ULLA-X may use Linux as underlying operating system as well. The technology-specific connector modules allow access to Network Published Shared Variables (NPSVs), which are means to externally configure National Instrument LabVIEW based SDRs. For the implementation description, i.e. the list of available parameters and parameter formats, text files (referred to as Link Layer Adapter (LLA) files for NPSV-based devices) are used. For information exchange between research applications, ULLA-X core and the connector modules, TCP/IP based network sockets are employed. We have implemented a control unit using standard industrial and academic research tools, which will let the audience see the interaction between the different ULLA-X components and the radio hardware, and learn about the underlying programming language. As an example for a R&D application using ULLA-X, we provide a demo interface and access library in MATLAB. Direct interaction with the ULLA-X is also shown with a command-line interface implemented in C from which more complex statements can be issued.

Acknowledgement

We would like to thank National Instruments for sharing hardware- and software-related expertise towards the SDR-based wireless network setup and for providing valuable feedbacks towards the ULLA-X developments.

References

[1] M. Ahmed *et al.*, "A component-based architecture for cognitive radio resource management," in *Proc. IEEE Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, June 2009.

[2] K. Arshad *et al.*, "Resource management for QoS support in cognitive radio networks," *IEEE Commun. Mag.*, vol. 52, no. 3, pp. 114–120, March 2014.

[3] A. de Baynast *et al.*, "ARQ-based cross-layer optimization for wireless multicarrier transmission on cognitive radio networks," *Comput. Netw.*, vol. 52, no. 4, pp. 778–794, March 2008.

[4] V. Kolar *et al.*, "A case for generic interfaces in cognitive radio networks," in *Proc. ICT-Mobile Summit*, June 2009.

[5] M. Petrova and P. Mähönen, "Cognitive Resource Manager," in *Cognitive Wireless Networks*. Springer Netherlands, 2007, pp. 397–422.

[6] T. W. Rondeau, "Application of artificial intelligence to wireless communications," Ph.D. dissertation, Virginia Polytechnic Institute and State University, September 2007.

[7] M. Sooriyabandara *et al.*, "Unified link layer API: A generic and open API to manage wireless media access," *Comput. Commun.*, vol. 31, no. 5, pp. 962–979, March 2008.

[8] R. Thomas *et al.*, "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives," *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 51–57, December 2006.