



2015 Wireless Innovation Forum European  
Conference on Communications Technologies  
and Software Defined Radio  
(WinnComm-Europe 2015)



5-9 October • Fraunhofer IIS • Erlangen, Germany

**Proceedings of  
WinnComm- Europe 2015  
Wireless Innovation European Conference on Wireless  
Communications Technologies and Software Defined Radio  
6-8 October 2015, Erlangen, Germany**

*Edited by: Lee Pucker, Gerald Ulbricht, Stephanie Hamill*

**Event Sponsors:**



**Copyright Information**

Copyright © 2015 The Software Defined Radio Forum, Inc. All Rights Reserved. All material, files, logos and trademarks are properties of their respective organizations.

Requests to use copyrighted material should be submitted through:

[http://www.wirelessinnovation.org/index.php?option=com\\_mc&view=mc&mcid=form\\_79765](http://www.wirelessinnovation.org/index.php?option=com_mc&view=mc&mcid=form_79765).

## WinnComm-Europe 2015 Organization

### Conference Chair:

Dr. Albert Heuberger, *Fraunhofer IIS*

### Thank you to our Technical Program Committee:

Marc Adrat, *Fraunhofer-Institut*

Onur Altintas, *Toyota*

Claudio Armani, *Selex ES*

Dr Kamran Arshad, *University of Greenwich*

Shaswar Baban

Fabio Casalino, *Selex ES*

David Chester, *Harris Corporation*

Francois Delaveau, *Thales*

Antonio Di Rocco, *Selex ES*

Ismael Gomez, *CTVR*

David Hagood, *Aeroflex*

Albert Heuberger, *Fraunhofer-Institut*

Vincent Kovarik, *PrismTech*      James Neel, *Cognitive Radio Technologies*

David Renaudeau, *Thales*

Charles Sheehe, *NASA*

Sarvpreet Singh, *Fraunhofer-Institut*

Rahul Sinha, *Tata Consultancy Services*

Chayil Timmerman, *MIT Lincoln Laboratory*

Manuel Uhm, *Ettus Research*

Gerald Ulbricht, *Fraunhofer-Institut*

## Table of Contents

### **DFC++ - A novel framework approach for flexible signal processing on embedded systems**

Dominik Soller, Thomas Jaumann and Gerd Kilian (Fraunhofer IIS, Germany); Joerg Robert (Friedrich-Alexander University Erlangen-Nuremberg, Germany); Albert Heuberger (Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany)

pp. 1-5

### **Exploiting Cyclic Features for Jammer Detection in Wideband Cognitive Radios**

Tassadaq Nawaz and Muhammad Ozair Mughal (University of Genova, Italy); Lucio Marcenaro (Università degli Studi di Genova, Italy); Carlo S Regazzoni (University of Genoa, Italy)

pp. 6-10

### **Cross-layer Resource Allocation for 5G Heterogeneous Software Defined Networks**

Giulio Bartoli and Dania Marabissi (University of Florence, Italy); Renato Pucci (CNIT - University of Florence, Italy); Luca Simone Ronga (CNIT, Italy)

pp. 11-15

### **Licensed Shared Access (LSA) field trial using LTE network and self organized network LSA controller**

Seppo Yrjölä (Nokia Networks, Finland); Vesa Hartikainen and Lucia Tudose (Nokia Solutions and Networks, Finland); Jaakko Ojaniemi (Aalto University, Finland); Arto Kivinen and Jarkko Paavola (Turku University of Applied Sciences, Finland); Marko Palola (VTT Technical Research Centre of Finland, Finland); Tero Kippola (Centria University of Applied Sciences, Finland)

pp. 16-25

### **Cellular Baseband Development Platform with an open RF Interface**

Benjamin Weber (ETH Zurich, Switzerland); Harald Kröll (ETH Zurich, ACP AG, Zurich, Switzerland); Stefan Altorfer (ACP AG, Zurich, Switzerland); Qiuting Huang (ETH Zurich, Switzerland)

pp. 26-30

### **Tactical Radio Coalition Interoperability Solution Facilitated by ANW2 AND NINE**

Igor Spivak (Harris Corporation, USA)

pp. 31-37

### **On the Design of Hierarchically Modulated BICM-ID Receivers with Low Inter-Layer Interferences**

Matthias Tschauner and Md. Farhan Tasnim Oshim (Fraunhofer FKIE, Germany); Marc Adrat (Fraunhofer FKIE / KOM, Germany); Markus Antweiler (Fraunhofer FKIE, Germany); Benedikt Eschbach (RWTH Aachen University, Germany); Peter Vary (RWTH Aachen, Germany)

pp. 38-47

### **Advanced Low Power, High Speed Nonlinear Signal Processing: An Analog VLSI Example**

Giuseppe Oliveri (Ulm University, Germany); Mohamad Mostafa (Deutsches Zentrum für Luft- und Raumfahrt (DLR), Germany); Werner G. Teich (Ulm University, Germany); Juergen Lindner (Uni Ulm, Germany); Hermann Schumacher (Universität Ulm, Germany)

pp. 48-56

### **Adopting WINNF transceiver facility for spectrum sensors**

Tomaz Solc (Jozef Stefan Institute, Slovenia)

pp. 57-64

### **Using OPENCL to increase SCA application portability**

Steve Bernier, François Levesque and Martin Phisel (Nordiasoft, Canada); David Hagood (Aeroflex, USA)

pp. 65-70

## **DFC++ - A NOVEL FRAMEWORK APPROACH FOR FLEXIBLE SIGNAL PROCESSING ON EMBEDDED SYSTEMS**

Dominik Soller (Fraunhofer IIS, Erlangen, Germany; dominik.soller@iis.fraunhofer.de);  
 Thomas Jaumann (Fraunhofer IIS, Erlangen, Germany; thomas.jaumann@iis.fraunhofer.de);  
 Gerd Kilian (Fraunhofer IIS, Erlangen, Germany; gerd.kilian@iis.fraunhofer.de);  
 Jörg Robert (Friedrich-Alexander-Universität, Erlangen, Germany; joerg.robert@fau.de);  
 Albert Heuberger (Friedrich-Alexander-Universität, Erlangen, Germany; albert.heuberger@fau.de).

### **ABSTRACT**

Development of modern Software Defined Radio (SDR) based communication systems can be accelerated significantly by the use of processing frameworks. The evolution of SDR and the involved departure from digital representations of classical radio architecture towards more abstract software systems raises new requirements of increased flexibility and versatility. The Data Flow Control for C++ (DFC++) processing framework addresses those requirements by employing modern programming concepts and flow control mechanisms to allow for variable data rates, dynamic paths, and flexible component designs.

Another important trend is the integration of specialized embedded platforms in the software radio domain. The rapidly increasing performance and efficiency of embedded processors enables the deployment of SDR systems in more space and power constrained environments. By relying exclusively on C++ and minimizing external dependencies, DFC++ is specifically designed for excellent portability and adaptability to support current and future designs while maintaining high performance and ease of use.

This paper introduces the key aspects of the DFC++ processing framework focusing on the reference pointer based data transport mechanisms responsible for the propagation of user data between different processing components.

### **1. INTRODUCTION**

Over the last decade Software Defined Radio (SDR) has been the driver of an ongoing revolution within the radio sector. The shift from hardware based signal processing to the software domain allows for increasingly complex designs by employing high level programming languages and software engineering techniques. Framework assisted development has proven particularly beneficial to unleash the full potential of SDR. Common interfaces leverage reusability of components and greatly simplify the

adaptation of radio systems to changing conditions. Furthermore designers can focus on algorithmic issues while the framework handles all peripheral and auxiliary tasks in the background.

Currently available framework solutions focus on the classic idea of signal processing chains with statically defined data streams. Examples can be found in the very popular GNU Radio project [1] or the Software Component Architecture (SCA) [2] based Open Source SCA Implementation Embedded (OSSIE) [3]. Both provide design flows and utilities to quickly assemble and adapt SDR systems and also offer advanced configuration options. Though experience with past SDR projects raised the demand for highly runtime dynamic, yet fast and portable setups. Particularly static paths and fixed data rates have been identified as a limitation for further evolution of the software radio concept.

The Data Flow Control for C++ (DFC++) framework takes a different approach in the attempt to grant a maximum of flexibility and versatility without sacrificing performance or usability. System components are highly independent and links between them can be created and removed dynamically. Instead of predefined data rates the adaptable flow control tolerates source, sink, or throughput driven paths without any reconfiguration. Components can also operate fully asynchronous without engaging flow control mechanisms, useful for monitoring or debugging facilities. Ultimately this allows SDR systems to advance towards modularized general purpose applications which can extend beyond pure signal processing. For example high level system controls or platform management tasks can be covered within the domain of the framework.

### **2. FRAMEWORK STRUCTURE**

The basic tasks for a processing framework include providing a concept for modularized components, managing data exchange between components and establishing configuration and monitoring interfaces. DFC++ distributes this functionality to a set of core elements consisting of

modules, inputs, outputs, and parameters. Those elements are designed to be mostly self-contained and encapsulate the assigned responsibility. This helps to gain a maximum of flexibility while keeping the complexity on top levels as low as possible. The following sections will briefly introduce the core elements and their purpose within the framework.

### 2.1 Modules

Modules are the basic organizational entity in the DFC++ framework. Other framework elements can be attached to modules in order to add certain features. A module can contain any number of inputs, outputs, and parameters including none. Additionally every module is equipped with a message interface for text based debugging and status information. An option to add other modules as submodules enables support for hierarchical structures and multilevel abstraction. Figure 1 provides a graphical overview of the module structure and data flow.

Every user designed DFC++ processing component is implemented as a module by deriving from a common module base class. This introduces all interfaces and functionality to user modules without exposing core framework mechanics. The user domain is limited to configuring the module according to the requirements, filling in the processing algorithm and optionally providing custom initialization and updating instructions. Initialization is performed once when the module is started, processing and updating are executed periodically by a module work loop. The updating routine is intended to synchronize module settings and respond to external changes of the configuration. It is invoked at a lower frequency than the processing routine to reduce the overhead but remains active when the module is disabled.

By default every module spawns its own thread for the work loop and all maintenance tasks. This grants modules full autonomy, not requiring any centralized management facilities after they have been configured and started by a setup procedure. At the same time scheduling capabilities of the platform operating system like multiple priority levels and real time options can be exploited. A non-threaded mode for submodules is also available to offer an alternative for scenarios where manual execution control is preferred. In this mode the coordination of processing and updating submodules is delegated to the parent module.

### 2.2 Inputs and Outputs

Inputs and outputs implement standardized connection terminals for data exchange between different modules. As the structure of an SDR setup is defined by linking modules together, the flexibility of inputs and outputs is crucial for the overall flexibility of the framework. At the same time achieving high data throughput at low processing effort is

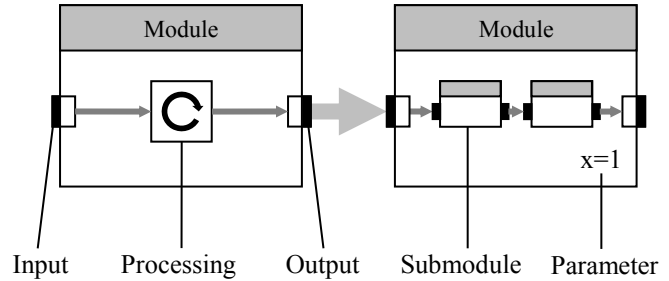


Figure 1: Overview of the DFC++ module structure and data flow through modules.

the most performance critical task within the framework and greatly influences the overhead associated with the use of the framework.

Following the principle of encapsulation the entire data transport functionality including buffering and flow control is contained within the inputs and outputs. Details about the transport mechanism are discussed in section 3. An output can be connected to multiple inputs while each input can only be connected to one output. Connections can be established or terminated at any time for fully dynamic rerouting of data paths. Special input ports and output ports allow linking to submodule inputs and outputs respectively in order to directly expose their terminals at parent level instead of manually forwarding the data.

Synchronization of a request status from input to output allows modules to disable or enable data processing based on the demand for the generated data. Entire paths can be deactivated by propagating the request status from one module to another along a reverse channel. This permits to add debugging facilities or alternate processing chains to a setup with minimal impact on the system performance when they are currently not in use.

### 2.3 Parameters

The primary task of parameters is to provide a standardized interface to internal values for configuration and monitoring purpose. Configuration can be done statically during setup or dynamically at runtime. An integrated sharing mechanism allows connecting parameters of different modules, extending their capability to inter-module communication and signaling. This also enables the use of common configurations for multiple modules and simplifies configuration updates for submodule hierarchies.

A separation of the internal working copy and the externally accessible variable ensures consistent datasets for the processing routine. The user can explicitly control when the external value is loaded to the internal value or vice versa. Furthermore write permissions allow the restriction of either external or internal write access to avoid accidental manipulations for certain parameters. Information whether a

parameter has been changed externally since the last synchronization is also available to simplify the detection of reconfiguration requirements.

### 3. DATA TRANSPORT MECHANISMS

The design of the data transport mechanisms needs to consider runtime connection and disconnection of inputs and outputs as well as fully dynamic data rates. DFC++ employs a reference pointer based solution to gain the necessary flexibility while maintaining a high performance level. Reference pointers take control over a dynamically allocated memory segment or object and allow sharing this object without ownership issues.

#### 3.1 Data Organization

The user data is organized in packets which are managed by reference pointers. Every packet is represented by one reference pointer. The structure of the packets is adaptable to specific applications and allows for arbitrary payload data types as well as attachment of meta information like sample indices or time stamps. Alignment of payload data prepares standard packet types for efficient processing with vector instructions like SSE [4] or Neon [5]. The packet size can be freely chosen, though the reference pointer concept is better suited for larger packets to reduce management overhead.

A packet is transferred from an output to all connected inputs by propagating the according reference pointer instead of the packet data itself as shown in figure 2. This method avoids the necessity to copy any payload data in the process and limits the effort to duplications of the reference pointer. At the same time the ownership for the packet memory is not bound to one module which eliminates synchronization requirements for the deallocation. It is important to notice that received packets must not be manipulated as the data is shared with all other inputs connected to the same output. Reference pointers can also be directly forwarded to other modules to pass through unmodified data as used for example by multiplexers.

#### 3.2 Packet Queue

Every input has a queue to store packet reference pointers until the processing routine of the module requests the data. Reference pointers are received via a callback function used by the connected output. This places the queue at the boundary between two module threads and requires the consideration of concurrent access. Write operations to the queue are within the context of the transmitting module thread while read operations are executed in the receiving module thread. To avoid ensuring mutually exclusive access to the queue, a static array with wrapping write and read pointers is used as a ring buffer. This allows writing and

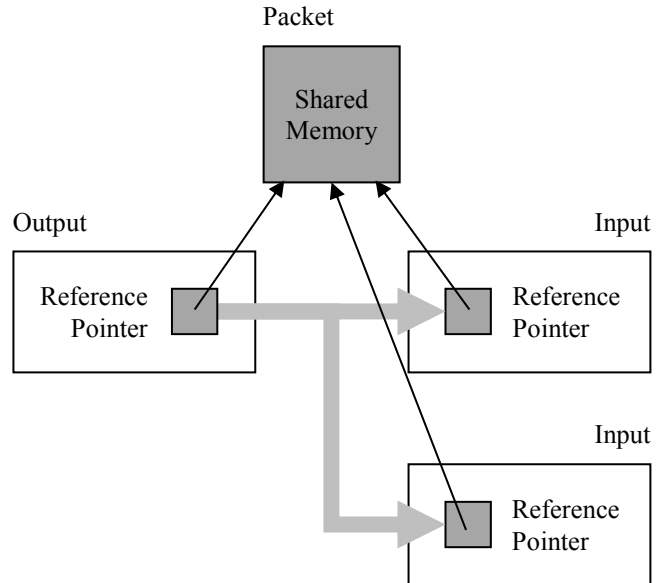


Figure 2: Transport of a DFC++ packet in shared memory via propagation of reference pointers.

reading independently at different positions in the queue. A lock free synchronization of the fill level via atomic operations prevents buffer overflows or underruns.

The queue length of each input can be chosen individually and represents an important optimization parameter for the module performance. The time it takes a module to process any input queue or fill any output queue determines the maximum time the module thread can operate without the upstream or downstream modules processing further data. In a single thread CPU scenario a context switch to another module thread becomes inevitable at this point. Depending on the system architecture a context switch is rather expensive and should not be forced to frequently [6]. Multi-core CPUs permit the simultaneous operation of several modules, though as the module count tends to exceed the number of concurrently executable threads, the general principle remains unaffected. Consequently the selected queue length needs to be a compromise between performance and the maximum latency a system can tolerate.

For delay critical sections the previously mentioned non-threaded mode can help to evade the performance trade off. Placing the involved modules under the control of a parent module which coordinates the processing chain within one thread allows reducing the queue lengths significantly while avoiding the context switching penalty.

#### 3.3 Flow Control

In many cases the source or the sink of an SDR system is some kind of hardware interface like a radio frontend. It dictates a certain data rate by producing respectively

consuming a fixed amount of samples per second. For paths not directly connected to a hardware component or in the case of simulation and testing the data rate might also be determined by the maximum throughput any participating module can achieve. Either way a flow control mechanism is required to synchronize and adapt the data rate to the limiting component.

The demand for full runtime flexibility requires the framework modules to remain responsive for incoming rerouting and configuration requests. Therefore any lock based blocking technique is problematic and involves error prone unblocking backdoors. DFC++ evades those issues by implementing the flow control mechanism via a two sided queue query procedure. Writing to the output is only permitted if free queue slots are available for every connected flow controlled input. Likewise reading from an input requires probing for available data previously. In between those queue boundaries the modules are allowed to operate asynchronously to avoid enforcing excessively frequent context switches among processing threads.

When no further data can be processed, either caused by an empty input queue or a full output queue, the next work cycle is delayed by briefly suspending the module thread as illustrated in figure 3. Thus modules consuming queued data faster than it is replenished are periodically suspended, granting more CPU resources to slower modules to compensate for variations in the processing effort of different components. To reduce the number of futile queue checks in the cases of large throughput deviations, the delay period is dynamically extended with the number of consecutive failed queue access attempts. The maximum duration of the suspension can be configured or the suspension can be disabled completely to optimize for specific module constellations or platform characteristics.

#### 4. PLATFORM SUPPORT

Extending the versatility of software radio systems requires the operation on various hardware platforms to adapt to different conditions. Especially environments with limited space or reduced power envelopes can be challenging for SDR applications. Rapidly evolving embedded ARM or x86 based systems finally start to offer well suited general purpose processor options for such scenarios. On the other hand specialized hardware like dedicated DSPs or even a combination of different processors might still be the only choice for more demanding tasks.

The ability to employ the same framework solution and associated design processes on different platforms can greatly enhance the development efficiency. Therefore a high level of portability is crucial for an adaptable and future proof framework. DFC++ is written entirely in C++ and does not require any external utilities for system setup or configuration. C++ is very well supported by almost any

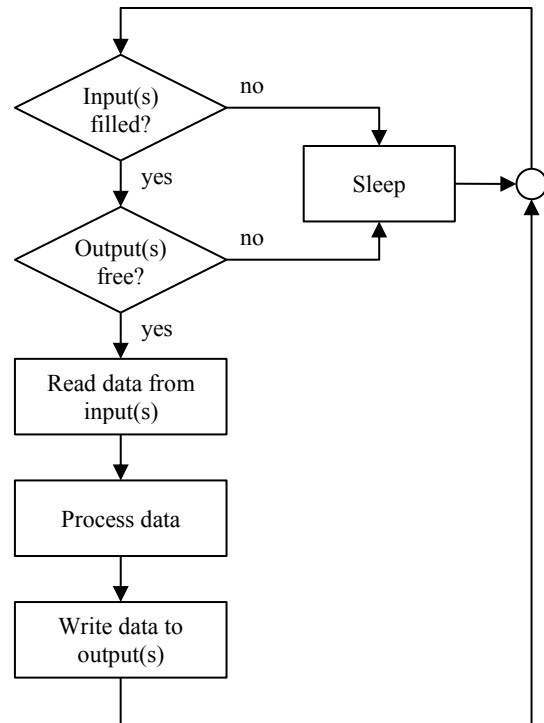


Figure 3: Flow chart of a typical DFC++ module process with the queue level based data flow control mechanism.

processing platform and provides outstanding performance in combination with modern high level programming features.

All dependencies exceeding the C++ 2003 [7] standard library are isolated on an internal abstraction layer for simplified maintenance. Low level dependencies are limited to standard concepts, available in most system APIs, like locking mechanisms, atomic operations, threads or sockets. Furthermore the abstraction layer offers user modules an integrated option to gain platform independent access to those features. This ensures not only excellent portability of the DFC++ core framework but also encourages the design of cross platform user modules.

The majority of general purpose processing platforms are supported by a Linux based operating system. Especially embedded solutions regularly provide customized Linux distribution as a foundation for user applications. This makes Linux the primary target platform for the DFC++ framework. In this case dependencies are mostly based on the POSIX [8] compliant system interfaces of Linux. Windows operating systems are also fully covered by the abstraction layer via the according Microsoft APIs [9], though no optimization has been conducted and testing is limited to Windows 7. Additionally an experimental port for the Texas Instruments SYS/BIOS [10] environment has been implemented to gain access to a variety of DSP platforms.

## 5. CONCLUSION

DFC++ introduces a new processing framework approach with clear focus on flexibility and versatility in order to help expanding the boundaries of the software radio concept. Dynamic data paths with adaptable flow control and variable rates allow integrating many peripheral tasks and management functionality within the framework domain. Runtime reconfiguration and monitoring options can greatly simplify debugging and accelerate the development process. The reference pointer based transport mechanism combines this high level of flexibility with efficient data propagation between modules. Encapsulating internal complexity and employing modern programming techniques improves the user experience and eases the learning curve.

The DFC++ framework has been successfully used on various Linux powered platforms, ranging from Intel x86 PCs to embedded ARM solutions. For example the implementation of the receiver nodes in an asset tracking sensor network [11] was assisted by the framework. Carefully selected and isolated dependencies reduce the adaptation effort for different platforms. Additional support for Windows 7 environments and initial tests on a Texas Instruments C6670 DSP with SYS/BIOS demonstrate the excellent portability of the framework. With this superior versatility the DFC++ processing framework is well prepared to fuel the development of current and future SDR projects.

## ACKNOWLEDGEMENT

This contribution was supported by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology as a part of the Bavarian project “Leistungszentrum Elektroniksysteme (LZE)”.  
<http://www.leistungszentrum-elektroniksysteme.de>

## REFERENCES

- [1] M. Braun et al., *Core concepts of GNU Radio*, May 2014, <http://gnuradio.org/redmine/projects/gnuradio/wiki/TutorialsCoreConcepts>
- [2] J. Bard and V.J. Kovarik, *Software Defined Radio - The Software Communications Architecture*, Wiley, 2007.
- [3] C.R. Aguayo González et al., “Open-Source SCA-Based Core Framework and Rapid Development Tools Enable Software-Defined Radio Education and Research”, *IEEE Communications Magazine*, pp. 48-55, October 2009.
- [4] Intel, *Intel 64 and IA-32 Architectures Software Developer’s Manual*, Volume 1: Basic Architecture, September 2014, <http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-1-manual.pdf>.
- [5] ARM, *Cortex-A9 NEON Media Processing Engine*, Revision: r3p0, July 2011, [http://infocenter.arm.com/help/topic.com.arm.doc.ddi0409g/DDI0409G\\_cortex\\_a9\\_neon\\_mpe\\_r3p0\\_trm.pdf](http://infocenter.arm.com/help/topic.com.arm.doc.ddi0409g/DDI0409G_cortex_a9_neon_mpe_r3p0_trm.pdf).
- [6] F.M. David et al., *Context Switch Overheads for Linux on ARM Platforms*, June 2007, <http://choices.cs.uiuc.edu/contextswitching.pdf>.
- [7] ISO/IEC, 14882:2003: *Standard for Programming Language C++*, October 2003.
- [8] IEEE, *Standard for Information Technology - Portable Operating System Interface*, Base Specifications, Issue 7, April 2013, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6506091>.
- [9] Microsoft, *MSDN - Windows API Index*, August 2015, <https://msdn.microsoft.com/en-US/library/windows/desktop/ff818516%28v=vs.85%29.aspx>
- [10] Texas Instruments, *SYS/BIOS Real-time Operating System User’s Guide*, v6.33, December 2011, <http://www.ti.com/lit/ug/spruex3k/spruex3k.pdf>.
- [11] H.-M. Tröger et al., “Time and Frequency Synchronization of a Wireless Sensor Network with Signals of Opportunity”, *Proceedings of the 46th Annual Precise Time and Time Interval Systems and Applications Meeting*, pp. 117-123, December 2014.



## EXPLOITING CYCLIC FEATURES FOR JAMMER DETECTION IN WIDE-BAND COGNITIVE RADIOS

Tassadaq Nawaz <sup>\*</sup>, Muhammad O. Mughal, Lucio Marcenaro and Carlo S. Regazzoni  
 (Department of Electrical, Electronic, Telecommunications Engineering and Naval  
 Architecture – DITEN, University of Genoa, Genoa, Italy;  
<sup>\*</sup>Email: tassadaq.nawaz@ginevra.dibe.unige.it)

### ABSTRACT

Cognitive radios (CRs) have been proposed as a promising solution for improving spectrum utilization through opportunistic spectrum sharing. However, security issues pertaining to cognitive radio technology are still a research topic. One of the prevailing issue is radio frequency jamming attack, where adversaries are able to exploits on-the-fly reconfigurability potentials and learning mechanism of cognitive radios in order to devise and deploy advanced jamming tactics. In this paper, a new algorithm is proposed for jammer detection in wide-band (WB) cognitive radio networks. The propose approach considers a WB scenario which is structured into multiple fixed length sub-bands (SB). These sub-bands are occupied by narrow-band (NB) signals which can be legitimate users or jammers. Cyclostationary-based classifier is here applied to identify and classify the received signals in different sub-band (SBs) as a legitimate user or a jammer. The performances of the proposed algorithm is shown with the help of Monte-Carlo simulations for different signal-to-noise ratios (SNR).

### 1. INTRODUCTION

Studies have shown that most of the licensed radio-wave spectral bands are under-utilized in time and space, resulting in unused "white spaces" in time-frequency grid at any particular location. The Federal Communications Commission (FCC) of the U.S. has also reported in [1] that the temporal and geographic variations in spectrum utilization range from 15 % to 85 %. On the other hand, fixed spectrum allocation policies do not allow unlicensed users for reusing of rarely used spectrum allocated to licensed users. This problem coupled with the rapidly increasing demand for wireless services and radio spectrum has led to spectrum scarcity for wireless applications. This has required new communication standards that allow unlicensed (secondary) users to utilize the vacant bands, which are allocated to licensed (primary) users. As a consequence, cognitive radios [2] attained much popularity over the last decade. Thus,

cognitive radio has emerged as the enabling technology for Dynamic/Opportunistic spectrum access (DSA/OSA), hence significantly increasing the spectrum utilization. In order to make it possible a cognitive radio to work, it must be aware of its surrounding environment. Cognitive radio encompasses this awareness by dynamically interacting with the environment and altering the operating parameters with the mission of exploiting the un-used spectrum without interfering with the primary users. This spectrum awareness comes from either radio frequency maps [3] or spectrum sensing.

In literature, many spectrum sensing methods have been proposed for cognitive radio, such as energy detection, cyclostationary detection and match filtering [4]. Cyclostationary detection based spectrum sensing is capable of detecting the primary signal from the interference and noise even in very low SNR region [5]. Therefore, the FCC has suggested cyclostationary detectors as an alternative to improve the detection sensitivity in CR networks.

During the last two decades, research efforts exploit the cyclostationary features of signals as a method for classification [6]-[8]. This has been found to be more efficient than simply considering energy detectors and matched filtering. Energy detection is easy to implement and does not require prior signal parameters such as signal bandwidth or carrier frequency, but its performances are highly dependent on noise levels and in-band interference [9]. Moreover, an energy detector can only be used to detect presence or absence of signals, but cannot differentiate between different types of signals. In previous research work, cyclic spectral analysis has been used as powerful tool for signals classification when the carrier frequency and bandwidth information is unavailable [10], [11]. The most prominent reason for selection of this technique as a classifier is its reduced sensitivity to noise, which leads to excellent classification at low SNRs.

Radio frequency (RF) jamming is defined as illicit RF transmission on one or more channels with the aim of disrupting the communication of targeted channels. The RF jamming and anti-jamming concepts are as old as wireless communication itself. However, the recent progress in

cognitive radio technology has enabled devising and deploying of more advanced, self-reconfigurable jamming [12] and anti-jamming solutions [13]. Spectrum sensing information play main role in an anti-jamming system. It can be used to detect potential jamming entities, and to take proactive measures to ensure communication continuity and security. Furthermore, a history of observation can be collected and used to adopt anti-jamming strategies with high success rate. For example, in frequency hopping spread spectrum (FHSS) based system, cognitive radios can modify their strategy to avoid hopping on the channels frequently occupied by jammers [14]. In order to design proper anti-jamming strategies, jamming entity needed to be detected in cognitive radios networks.

Our work introduces a cyclostationary-based technique for jammer detection in WB cognitive radios. WB is divided into multiple fixed length SBs, which are occupied by various narrow-band signals. These narrow-band signals can be classified as legitimate signals or jammer signals. We consider signals as cyclostationary. The spectral correlation function (SCF) is used to extract the cyclic features of modulated signals, such as cyclic frequency ( $\alpha$ ), spectral correlation density ( $S_x$ ) and spectral frequency ( $f$ ). Then, for each occupied SB, these features are compared with the features of the considered licit waveforms which are stored in a database. Based on this comparison, each narrow band signal is classified either as a licit waveform or a jammer. The performance of the proposed technique is evaluated with the help of Monte Carlo simulation. To best of our knowledge, cyclostationary-based jammer detection algorithm in cognitive radios has not been introduced so far in the open literature.

Section 2 reviews the cyclic spectral analysis. Section 3 describes the system model and problem formulation. Experimental results are discussed in Section 4. Finally, paper is concluded in Section 5 along with some future directions.

## 2. SPECTRAL CORRELATION

Many communications signals cannot be accurately modeled as stationary, but can more properly be described as cyclostationary [6]. This cyclostationary property can be utilized to extract several key attributes of a signal, including its modulation scheme. Cyclo-stationary signals statistics vary periodically in time, or equivalently, its spectral components exhibit temporal correlation.

The second-order statistics are considered in this paper. The autocorrelation function of a cyclostationary signal can be expressed in terms of its Fourier series components [15], [16].

$$R_x(t, \tau) = E \left\{ x \left( t + \frac{\tau}{2} \right) x^* \left( t - \frac{\tau}{2} \right) \right\} \quad (1)$$

$$R_x(t, \tau) = \sum_{\{\alpha\}} R_x^\alpha(\tau) e^{j2\pi\alpha t} \quad (2)$$

In (1),  $E\{\cdot\}$  is the expectation operator,  $\{\alpha\}$  is the set of Fourier components, and  $R_x^\alpha(\tau)$  is known as cyclic autocorrelation function (CAF) and give Fourier components. CAF is given by

$$R_x^\alpha(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} R_x(t, \tau) e^{-j2\pi\alpha t} \quad (3)$$

When  $R_x(t, \tau)$  is periodic in  $t$  with period  $T_0$ , (2) can be expressed as

$$R_x^\alpha(\tau) = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} R_x(t, \tau) e^{-j2\pi\alpha t} \quad (4)$$

The Fourier Transform of the CAF is known as Spectral Correlation Function (SCF) and given by

$$S_x^\alpha(f) = \int_{-\infty}^{+\infty} R_x^\alpha(\tau) e^{-j2\pi f \tau} \quad (5)$$

This is equivalent (assuming cyclo-ergodicity) to [6]

$$S_x^\alpha(f) = \lim_{T \rightarrow \infty} \lim_{\Delta t \rightarrow \infty} \frac{1}{\Delta t} \int_{-\frac{\Delta t}{2}}^{\frac{\Delta t}{2}} \frac{1}{T} X_T \left( t, f + \frac{\alpha}{2} \right) X_T^* \left( t, f - \frac{\alpha}{2} \right) \quad (6)$$

Where,

$$X_T(t, f) = \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} x(u) e^{j2\pi f u} du \quad (7)$$

Here,  $S_x^\alpha$  is the true measure of correlation between the spectral components of  $x(t)$ . The resulting SCF is a three dimensional image, which can be seen in Figures 1(a) and 1(b) at 0 dB SNR. The most exciting property of the SCF is its insensitivity to additive white noise. The noise is wide sense stationary process and the spectral components of white noise are uncorrelated, i.e., it does not contribute to resulting SCF for any value of  $\alpha \neq 0$ . This is even true when noise power exceeds signal power, where energy detector have very poor performance

SCF computation requires large amount of data, which make it unreasonable for classifier to operate on in real time, in [10] authors proposed that cycle frequency profile can be used for classification. In [15] authors used both frequency profile (f-profile) and cycle frequency profile for classification of signal, which increased computational complexity.

$$\vec{\alpha} = \max_f [S_x^\alpha] \quad (8)$$

In [10] author's showed excellent classification results; therefore, in this work we also consider  $\alpha$ -profile. The  $\alpha$ -profile of BPSK, QPSK and WB signals at 0dB SNR is shown in Figures 2(a), 2(b) and 2(c) respectively. In Figure 2(c) WB is comprised of three narrow signals; BPSK, QPSK and jamming signal (Sine Wave).

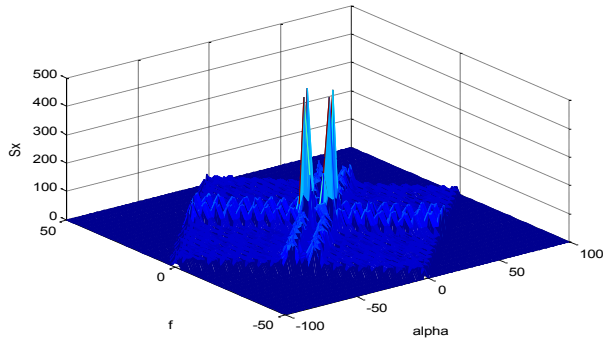


Fig. 1(a): SCF of BPSK

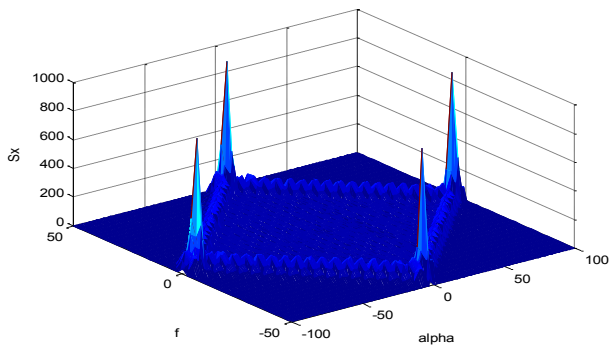
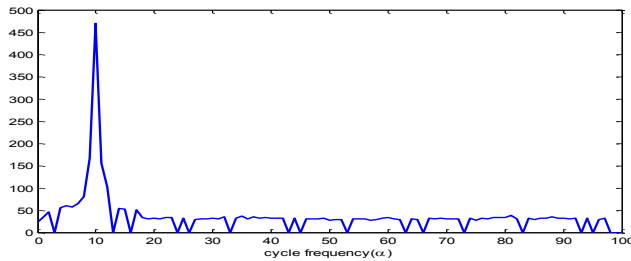
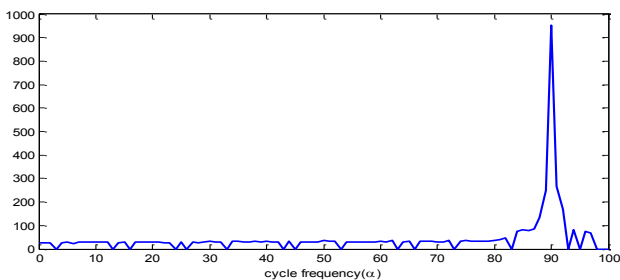
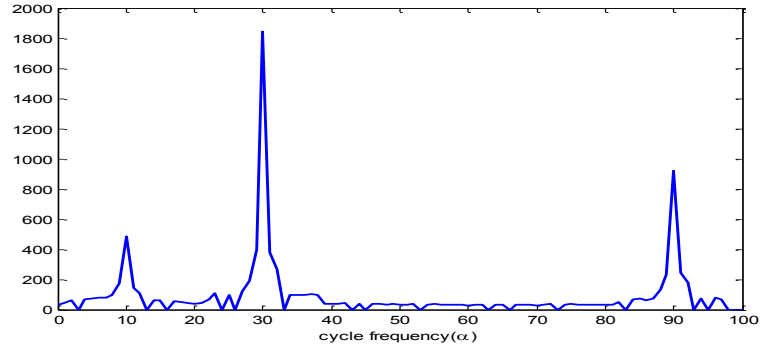


Fig. 1(b): SCF of QPSK


 Fig. 2(a):  $\alpha$  profile of BPSK

 Fig. 2(b):  $\alpha$  profile of QPSK

 Fig. 2(c):  $\alpha$  profile of Wide-band signal

### 3. SYSTEM MODEL AND PROBLEM FORMULATION

A received WB spectrum is considered of  $\Delta$  Hz. We suppose that this WB is occupied by various NB signals  $s_n(t)$ ,  $n \in \{1, 2, 3 \dots N\}$ , with different carrier frequencies and modulation types that we want to identify. The received WB signal is an aggregated time-domain signal which can be presented as

$$r(t) = \sum_{n=1}^N h_n(t) * s_n(t) + w(t), \quad (9)$$

where,  $h_n(t)$  is the channel coefficient between  $n$ -th transmitter and receiver,  $*$  is convolution operator, and  $w(t)$  is AWGN with zero mean and power spectral density  $\sigma_w^2$ .

In our work, these NB signals can be generated by different type of modulation schemes, such as, binary frequency shift keying (BFSK), binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), or any other modulation scheme as show in Figure 3(a). The WB is divided into multiple SBs of equal bandwidths and each of these NB signals can occupy one SB with no spill over energy into the neighboring SBs.

A single tone (sine wave or cosine wave) is here considered as a jamming signal. This jammer can jam any of the SBs, if it has higher power than the legitimate signal as shown in Figure 3(b). The tone jamming allows to concentrate all of power on a single data channel. Therefore, tone jamming has high success rates against NB signals, and often be the best strategy for jammers with limited transmission power. Suppose that the targeted signal is QPSK-modulated and uncoded, and that the targeted receiver is coherent receiver. Then, the error probability  $P_e$  to jam either in-phase component ( $I$ ) or quadrature component ( $Q$ ) of the targeted signal can be calculated as in [16].

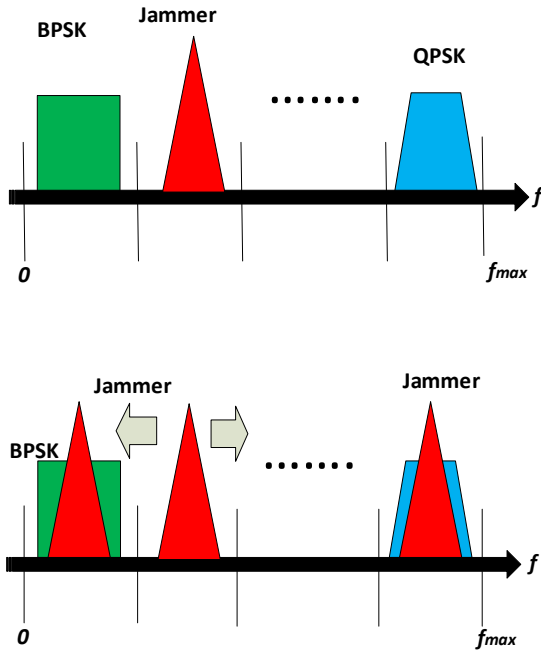


Fig. 3: (a) Wide-band spectrum divided into multiple sub-SBs and each SB is occupied by a narrow-band signal. (b) Narrow-band jammer jumps to neighboring SB to jam licit (BPSK or QPSK) signal.

$$P_e^I = Q \left( \sqrt{2 \frac{P_R}{P_N} \left( 1 - \sqrt{2 \frac{P_J}{P_R} \sin(\theta^J)} \right)} \right) \quad (10)$$

$$P_e^Q = Q \left( \sqrt{2 \frac{P_R}{P_N} \left( 1 - \sqrt{2 \frac{P_J}{P_R} \sin(\theta^J)} \right)} \right) \quad (11)$$

where  $P_R$  is the received power of targeted signal,  $P_N$  is thermal noise power,  $P_J$  jamming signal received power,  $\theta^J$  is the phase of jamming signal, and  $Q$  is the Gaussian  $Q$ -function. To make our analysis simpler, we assume that  $P_J \gg P_R$ , resulting in  $P_e \approx 100\%$  whenever jammer transmits on the same channel as the targeted transmitter-receiver pair.

#### 4. EXPERIMENT RESULTS AND DISCUSSION

A WB spectrum is considered of  $50\Delta\text{Hz}$ . This WB is divided into 5 equal bandwidth SBs. Each SB is either occupied by narrow-band signal or free. In order to test our proposed technique, BPSK and QPSK are considered as legitimate

signals and sine wave as a jammer. The received signals are considered to be affected by AWGN. Three different configurations of system is considered: (a) BPSK signal is using SB-1, QPSK signal is using SB-5 and jamming signal in free band SB-2; (b) BPSK signal is using SB-1, QPSK signal is using SB-5 and jammer jumps to SB-1 to jam BPSK signal; (c) BPSK signal is using SB-1, QPSK signal is using SB-5 and jammer jumps to SB-5 to jam QPSK signal. The Nyquist rate is set to  $f_s = 100\Delta\text{Hz}$ . The Monte-Carlo simulation is run for 1000 runs to evaluate the performance of system.

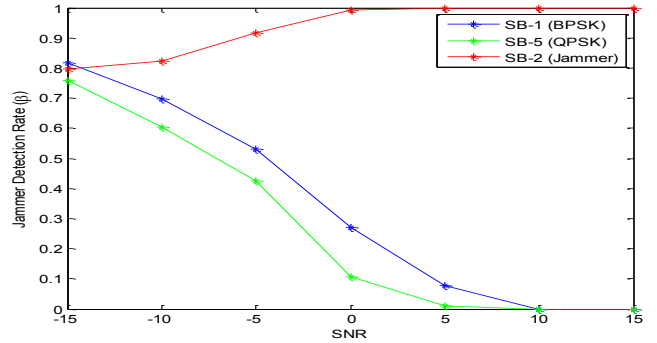


Fig. 4: Jammer Detection Rate ( $\beta$ ) vs. SNR

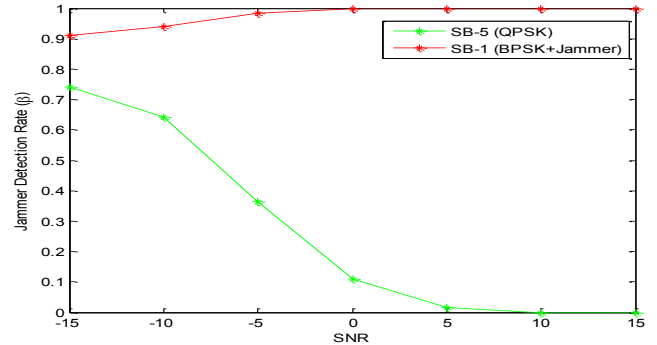


Fig. 5: Jammer Detection Rate ( $\beta$ ) vs. SNR

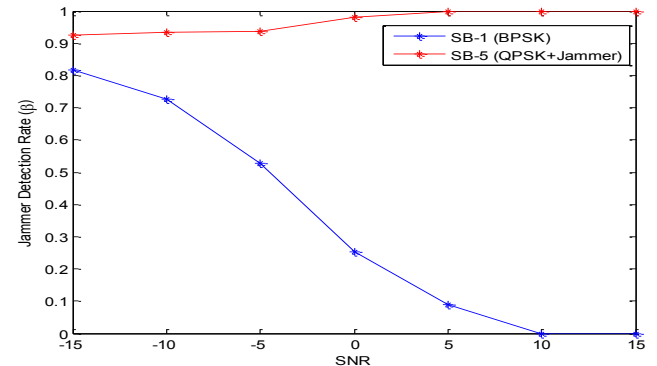


Fig. 6: Jammer Detection Rate ( $\beta$ ) vs. SNR

Figure 4 shows jammer detection rate ( $\beta$ ) versus SNR for occupied SBs. It can be observed that this technique perform well under low SNR conditions. In overall analysis, when SNR increases  $\beta$  is significantly increased for SB2 and decreased for SB-1 and SB5. The  $\beta$  is approximately 0.92 for SB-2 at -5dB. However,  $\beta$  is 0.53 for SB1 and 0.42 for SB-5, due to miss classification of legitimate signals at -5dB. The  $\beta$  is significantly increased for SB-2 and it reaches near to 1 at 0dB. On other hand  $\beta$  is reduced to 0.17 and 0.1 for SB-1 and SB-5 respectively at 0dB. Beyond, 0dB  $\beta$  is 1 for SB-2 and its value significantly reduces for SB-1 and SB-5.

After that we configure our system in such that SB-5 is occupied by QPSK signal while jammer jumps into SB1 thus jamming the SB1 which is occupied by BPSK. The plot for this case is given in Figure 5. In this configuration  $\beta$  is 0.98 for SB-1 (due to abnormal changes in  $S_x$  value) and 0.36 for SB-5 at -5dB. The  $\beta$  values are 1, 0.1 for SB-1 and SB-5 respectively at 0dB.

In third system configuration, SB-1 is occupied by BPSK signal while the jammer jumps from SB-2 to SB-5 in order to jam the QPSK signal. Figure 6 shows results for this system configuration. In this case  $\beta$  is 0.93 for SB-5 (due to abrupt changes in  $S_x$  value) and 0.53 for SB-1 at -5dB SNR. The  $\beta$  values are approximately 1, 0.15 for SB-5 and SB-1 respectively at 0dB SNR. The results show that this technique can be used for detection of jammer with high detection rate at low SNR of 0 dB, due to its insensitivity to AWGN noise.

## 5. CONCLUSION AND FUTURE WORK

In this paper, a cyclostationary-based jammer detection algorithm is proposed to be used for WB cognitive radios. The WB is considered to be comprised of several NB signals. It is important to note that we have assumed no prior knowledge about signals, other than their presence. The cyclic features of received WB signal are extracted by spectral correlation function (SCF). The SCF produces large amount of data therefore, we used  $\alpha$ -profile for the purpose of classification.

The peaks obtained in  $\alpha$ -profile are compared with licit waveforms database to identify jamming waveforms in each SB. In the end, results are evaluated for different configurations of system with the help of Monte-Carlo simulation and plotted as jammer detection rate ( $\beta$ ) versus SNR.

The technique is shown to perform well at low SNR value within the limitations imposed for using simple classification based on plain comparison of parameters from database. In future, more sophisticated classification schemes such as neural network or support vector machine (SVM) can be used to improve the performance of this technique. This technique can provide bases for jamming/anti-jamming system based on cognitive radio technology, where both transmitter and jammer can be equipped with spectrum sensing capabilities.

## 6. REFERENCES

- [1] "Spectrum policy task force," Rep. ET Docket 02-135, Federal Communications Commission, Nov. 2002.
- [2] J. Mitola and G. Q. Maguire Jr., "Cognitive radio: Making software radios more personal," *IEEE Personal Commun.*, 6(4):13–18, Aug. 1999.
- [3] S. Haykin, "Cognitive radio: Brain-empowered wireless Communication," *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, Feb. 2005.
- [4] H. B. Yilmaz, T. Tugcu, F. Alag'oz, and S. Bayhan, "Radio environment map as enabler for practical cognitive radio networks," *Communications Magazine, IEEE*, 51(12):162–169, December 2013.
- [5] S. Haykin, D. J. Thomson, and J. H. Reed, "Spectrum sensing for cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 5, pp.849–877, 2009.
- [6] W. Gardner, "Statistical Spectral Analysis: A Nonprobabilistic Theory," New Jersey: Prentice Hall, 1987.
- [7] C. Spooner, "Automatic radio frequency environment analysis," in *Proceedings on the Thirty-Fourth Asilomar Conference*, October 2000.
- [8] Spooner, C.M. "On the utility of sixth-order cyclic cumulants for rf signal Classification," in *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol.1, November 2001, pp. 890–7.
- [9] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Conference Record of the 38th Asilomar Conference Signals, Systems, and Computers*, 2004, pp.772–776.
- [10] A. Fehske, J. Gaeddert and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," *IEEE DySPAN*, pp. 144-150, Baltimore, 2005
- [11] Eric Like, Vasu Chakravarthy and Zhiqiang Wu, "Reliable Modulation Classification at Low SNR Using Spectral Correlation," *IEEE CCNC*, pp. 1134-1138, 2007.
- [12] D. J. Thuento and M. Acharya, "Intelligent jamming in wireless networks with applications to 802.11b and other networks," In *Proceedings of the 2006 IEEE Conference on Military Communications*, MILCOM'06, pages 1075–1081, Piscataway, NJ, USA, 2006. IEEE Press.
- [13] K. Dabcevic, M. O. Mughal, L. Marcenaro, and C. S. Regazzoni, "Spectrum intelligence for interference mitigation for cognitive radio terminals," in *2014 Wireless Innovation Forum European Conference on Communications Technologies and Software Defined Radio (WInnComm-Europe 2014)*, Nov. 2014.
- [14] M.J.A. Rahman, M. Krunz, and R. Erwin, "Interference mitigation using spectrum sensing and dynamic frequency hopping," in *Communications (ICC), 2012 IEEE International Conference on*, pages 4421–4425, June 2012.
- [15] Eric Like, Vasu D. Chakravarthy and Zhiqiang Wu, "Modulation Recognition in Multipath Fading Channels Using Cyclic Spectral Analysis," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008 on*, pages 1-6, Nov 2008.
- [16] K. Dabcevic, "Intelligent Jamming and Anti-Jamming Techniques using Cognitive Radios," PhD thesis, University of Genova, Genova, Italy, 2015.

# Cross-layer Resource Allocation for 5G Heterogeneous Software Defined Networks

Giulio Bartoli<sup>1</sup>, Dania Marabissi<sup>1</sup>, Renato Pucci<sup>1</sup> and Luca Simone Ronga<sup>2</sup>

<sup>1</sup>Department of Information Technology, University of Florence

<sup>2</sup>CNIT, Florence Research Unit

**Abstract**—The demand for pervasive wireless access and requirements for high data rates are expected to grow significantly in the near future. In this context, the deployment of Heterogeneous Networks will enable important capabilities such as high data rates and traffic offloading, providing dedicated capacity to homes, enterprises, and urban hotspots. Despite HetNet technology will be beneficial for future wireless systems in many ways, the massive cell diffusion has as consequence an exponential increase of the backhaul traffic that can create congestion and collapse of the backhaul network. In this paper we consider a Software Defined Network (SDN) architecture where different access nodes are part of an infrastructure layer that is connected and managed by the SDN control software through OpenFlow APIs. An appropriate high level SDN application is responsible for the efficient coordination among the network layers of access nodes and to take dynamic joint decisions about the resource usage. Due to the complexity of the decisional process in the SDN controller, also subject to tight time constraints, a cognitive parallel logic is proposed. SDN abstraction provides a simplified view of underlying network to the RAN SDN application, passing to it all the required augmentations. In particular we propose a cross-layer approach that takes into account both the data traffic load of each cell in the HetNet and the capacity of the backhaul links that are available to connect the cells. The proposed iterative procedure tries jointly to optimize the distribution of the traffic in the backhaul network and the UEs cell association with the goal of minimizing the unsatisfied UEs requests.

**Index Terms**—Cross-layer, SDN, HetNets

## I. INTRODUCTION

The volume of mobile data is expected to grow exponentially over the next few years and the challenge for the fifth generation (5G) networks will be to overcome the fundamental limits of the existing cellular networks with the aim of guaranteeing high quality and high data rate services to increasing numbers of users with limited resource availability. 5G wireless communication technologies are expected to satisfy stringent requirements in terms of bit rate per square kilometer, energy consumption and latency [1]. To this end several important features must be addressed [2]. Among these access point densification is considered to be the key approach to boost capacity [3], [10]. In particular the deployment of Heterogeneous Networks (HetNets) is based on a multi-layer architecture consisting of macrocells overlaid with smaller cells to serve users with different Quality of Service (QoS) requirements in a spectrum- and energy-efficient manner. Indeed, the capacity and the coverage of any wireless network can

be improved in terms of efficiency by moving the transmitter and the receiver closer together. Moreover, with the massive Multiple Input Multiple Output (MIMO) and millimeter wave communication technologies emerging into 5G networks, the cell size of 5G networks has to become smaller [4].

HetNets encompass a broad variety of cells, including microcells, picocells, metro cells, and femtocells, as well as advanced wireless relays and distributed antennas that can be deployed anywhere. This massive cell diffusion has as consequence an exponential increase of the backhaul traffic that can create congestion and collapse of the backhaul network. In addition the backhaul network will be characterized by heterogeneous links, indeed small cells will be used in different environments, such as homes, small offices, hotspots and enterprises, where the available backhaul connections will be different (i.e., optical fibre, subscriber broadband communications links, etc.). The forward of massive traffic is a great challenge for future 5G backhaul networks whose architecture and protocols has to be properly designed [5].

Specific problem either in RAN or core network can be solved by using Software Defined Network (SDN) principle.

Network Virtualization [6] allows the adoption of centralized redundant decisional entities able to perform complex optimization goals with an abstract view of the underlying network. If compared to traditional approaches the main advantages are found in the complexity of decisions and larger computational resources available. However, due to the abstraction from real network devices, several radio access details are hidden to the SDN controller resulting in poor radio optimization performances. The proposed approach try to mitigate this effect by creating two independent decisional cores exchanging information. A decisional engine aware of physical layer aspects and a higher decisional point controlling pure network resources. The former is responsible for the collecting of UE resource requests and for the optimized association between each UE and its serving cell. The latter try to distribute the aggregated traffic flows in order to meet the requirements and constraints imposed by the available links.

In particular we propose a cross-layer approach that takes into account both the data traffic load of each cell in the HetNet and the capacity of the backhaul links that are available to connect the cells. The procedure is iterative, at each step the User Equipments (UEs) cell association procedure receives as input the ability of the backhaul network to satisfy the amount of data transfer request by each cell. At the same time the

backhaul network receives as input the requests of the cells and tries to adapt to those.

The paper is organized as follows. Sec. II introduces the considered network model, while the proposed cross-layer solution is described in Sec. III and its performance is evaluated in Sec. IV. Finally conclusions are drawn in Sec. V.

## II. NETWORK MODEL

### A. Access Network

This paper focuses on a HetNet where a macrocell is overlapped by  $S$  small cells (i.e., micro and femto-cells) that are randomly placed in the macrocell area following a uniform distribution. The cells have different transmission power and coverage. The total number of cells is  $C = S + 1$ . In particular, we consider a dense deployment scenario where many small cells are densely deployed to support huge traffic over a relatively wide area [7]. The number of UEs,  $U$  is chosen following a Spatial Poisson Point Process (SPPP) that is extensively used for modelling heterogeneous networks [8].

Each UE in the area can be served by a small cell or the macrocell. Using the traditional association policy, each UE would select as serving cell the one with the highest Signal-to-Interference plus Noise-Ratio (SINR) measured on specific Downlink (DL) reference signals. However, small cells are characterized by a low transmission power, hence by a reduced coverage that is also limited by the interference from the MBS (Macrocell Base Station) that has a significant higher transmission power. This means that, by adopting a traditional approach, only the UEs in close proximity would select the small cell as a serving cell, which makes this approach not suitable for HetNets. Indeed, the UEs may connect to distant high-power MBS rather than nearby small cell, thus causing an inefficient load distribution. In order to reduce the macrocell load and to maximize the cell splitting, we assume here that the coverage area of the small cell is extended by using the Cell Range Extension (CRE) [9]. This method is based on the use of a bias (i.e., Range Extension Bias - REB) that is a positive value added to the measured signal power received by the UEs from the small cell.

The average SINR of the  $u$ -th UE served by the  $\bar{c}$ -th cell can be written as

$$\Gamma_u = \frac{P_{u,\bar{c}}}{N + \sum_{c=1, c \neq \bar{c}}^C P_{u,c}} \quad (1)$$

where

- $P_{u,c}$  represents the power received<sup>1</sup> by the  $u$ -th user from the  $c$ -th cell
- $N$  is the AWGN noise power

Actually, the interference experienced by the  $u$ -th user that communicates on the  $k$ -th RB depends on how many and which cells are simultaneously transmitting on the same  $k$ -th RB towards different users. However, in order to decouple our problem from the resource assignment problem we focus

<sup>1</sup>This value takes into account also the transmitting and receiving antenna gains and the pathloss.

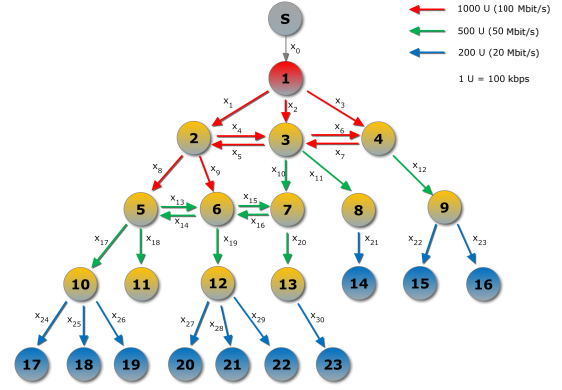


Fig. 1. Backhaul network topology

on the worst interference condition, meaning that all the cells not involved in the joint transmission act as interferers for a given UE.

Moreover, each UE,  $u$ -th, is characterized by a data rate request  $Q_u$ . As a consequence the amount of physical resources (called here *Resource Blocks*, RBs) requested by the  $u$ -th UE,  $N_u$ , depends on  $Q_u$  and on the experienced SINR and can be expressed as

$$N_u = \frac{Q_u}{B \log_2(1 + \Gamma_u)} \quad (2)$$

where  $B$  represents the bandwidth of a physical RB.

### B. Backhaul network

In an actual scenario each small cell BS (SBS) is connected with other SBSs or with the MBS by heterogeneous links with different capacity, while, it is reasonable to assume that the aggregated backhaul traffic at the MBS from and towards the core network is transmitted by high capacity fibre links. Hence, we are interested in optimizing the traffic inside the local backhaul network composed by the macrocell and the small cells in its area. This network has hierarchical topology, as the one represented in Fig. 1. In particular, we consider a tree topology where the macrocell represents the root, and the nodes of the higher levels of the tree are connected with a meshed structure. The small cells represent intermediate nodes and leaves of the tree. The hierarchical structure reflects the capacity of the links. In particular, the links are coloured depending on their capacity, chosen among the values of 100 Mbps, 50 Mbps and 20 Mbps for core optical links, outer xDSL links and femto-links respectively. The capacity of each link is expressed in units, called  $U_B$ , of 100kbps.

Fig. 2 shows the geographical map of the considered area. Cell  $n.1$  is the single *macro* cell, cells from  $n.2$  to  $n.13$  represent *micro* cellular coverages, scattered around the macro. Cells  $n.14$  through  $n.23$  are *femto* cells located in private premises.

### C. SDN

The Network Virtualization adopted in the backhaul network allows instantaneous routing decisions based on bandwidth

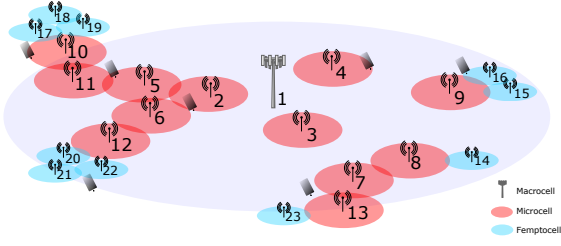


Fig. 2. Heterogeneous coverage map

requests from the radio access network. Each backhaul link is considered structural, in the sense that the arcs and the capacity does not vary over time. Each node performs two main tasks:

- 1) it collects the inbound traffic flow to be delivered through its air interface to the associated UEs;
- 2) it forwards the received traffic flows to the appropriate adjacent nodes as instructed by SDN controller.

As in the SDN paradigm, the traffic routing decisions are taken centrally by the controller, following the load balancing and optimization objectives imposed by the network intelligence.

### III. ITERATIVE PROPOSED SOLUTION

The benefits of CRE cell association strategy for HetNets can be improved if it is combined and jointly optimized with resource allocation and backhaul traffic distribution procedures. Indeed, the CRE association simply forces UEs to select low power nodes by adding a fixed bias to the received signal power, but it does not take into account the load of the selected cell and the capability of the relative backhaul link to support all the small cell traffic.

We express the total cell load as the aggregated data requests in terms of requested RBs, and we have that the  $\bar{c}$ -th cell is overloaded if the amount of requested RBs,  $R_{\bar{c}}$ , exceeds the number of available RBs,  $K$ . Hence, the amount of unsatisfied data rate requests at  $\bar{c}$ -th cell is

$$O_{\bar{c}} = R_{\bar{c}} - K = \sum_{u \in \mathcal{U}_{\bar{c}}} \left\lceil \frac{Q_u}{N_s \log_2(1 + \Gamma_u)} \right\rceil - K \quad (3)$$

where  $\lceil x \rceil$  indicates the smallest integer value greater than  $x$  and  $\mathcal{U}_{\bar{c}}$  is the set of UEs associated with the  $\bar{c}$ -th cell. If  $O_{\bar{c}} > 0$  the cell is not able to satisfy all the UEs requests.

For what concerns the load of the backhaul link of the  $\bar{c}$ -th cell, it is approximated as the aggregated requested data rate managed by the cell as

$$B_{\bar{c}} = \sum_{u \in \mathcal{U}_{\bar{c}}} Q_u. \quad (4)$$

The ability of the backhaul link of the  $\bar{c}$ -th cell to support the required load depends not only on the capacity of the  $\bar{c}$ -th link, but also on the distribution of the traffic in the network tree.

The goal of the proposed system is to minimize the amount of the total unfulfilled data rate requests taking into account both the ability of the cell to serve the associated UEs and

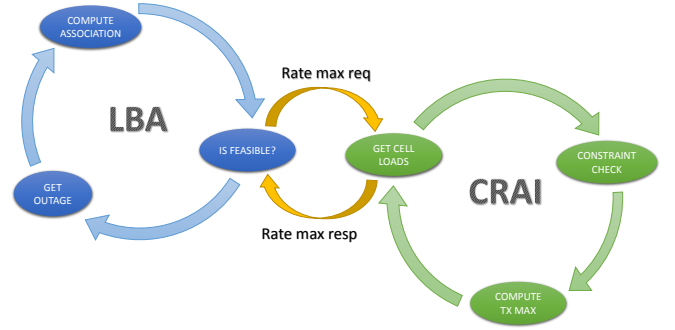


Fig. 3. PHY-NET cycle of the proposed procedure.

the backhaul network capacity. In particular, we propose an iterative cross-layer algorithm that tries jointly to optimize the distribution of the traffic in the backhaul network and the UEs cell association.

The algorithm works as depicted in Fig. III. At the physical (PHY) layer a new cell association procedure that takes into account CRE and the cell load, called Load Balancing Association (LBA) is performed producing as output the "Rate Max Request", that is the amount of data traffic that should be supported by the backhaul link of each node of the network. This request is sent to the network (NET) layer that using a suitable traffic distribution algorithm called CRAI (Cognitive Radio artificial Intelligence) tries to satisfy the PHY layer requests. The NET layer produces as output the "Rate max Response" that is a proposal on how to distribute the traffic in the backhaul network, indicating whenever the link is overloaded or under-loaded. This cyclic continues for several iterations. The procedures starts and ends at the PHY layer.

#### A. Load Balancing Association

At the PHY layer a new cell association procedure is performed (i.e. LBA) that takes the traffic load into account. In particular, in the network initialization phase, each UE is associated to the BS (SBS or MBS) that has the highest SINR taking REB into consideration. Then, when at each iteration the following steps are executed:

- 1) each BS determines if the amount of requested RBs is lower of the number of the available RBs (as in (1));
- 2) each BS determines if its backhaul link is able to support the total traffic amount generated by its potential associated UEs,  $B_{\bar{c}}$ ;
- 3) if at least one of the two previous conditions<sup>2</sup> is not satisfied some UEs are moved towards another serving cell as explained below;
- 4) RBs are allocated to the UEs in each cell following a Proportional Fairness (PF) policy.

When the access and/or the backhaul links of some cells of the network are overloaded, the cells are not able to serve some UEs even if these cells are those received with the highest SINR value by the UEs. Hence, the new LBA procedure is

<sup>2</sup>At the first iteration only the access capacity load is considered.



performed, whose aim is to select the UEs that can be served by different cells and the new serving cells. In particular, let us indicate with  $\mathcal{S}$  (with  $\mathcal{S} \subset \mathcal{C}$ ) the set of cells that are not overloaded (it means that they are able to satisfy all the UEs requests in terms of either access capacity or backhaul capacity). Each overloaded BS determines for each of its associated UE the neighbour cell belonging to  $\mathcal{S}$  that is received with the highest SINR. Then the UEs are sorted in ascending order with the increase of the RBs requested to the new potential serving cell (i.e., the new cell is received with a lower SINR value, and hence the amount of requested RBs increases). Starting from the first UE in the ordered queue, a new association is performed if the new selected cell is able to support this new UE without overloading. The UEs association with the new serving cell continues until the original cell is not more overloaded or when there are not neighbour cells for the new association of the UEs.

At the end of the association procedure the PHY calculates the aggregated data rate of its associated UEs and sends this information to the NET layer.

### B. CRAI (Cognitive Radio Artificial Intelligence)

Taking into account the PHY layer data rate requests per node, the NET layer is responsible for the network optimization. This process is executed by the CRAI and it is aimed at optimizing the distribution of the data flows among arcs and nodes in order to remove possible loops and avoid congestion in the network. Such an optimization can be defined as a minimum-cost network flow problem where the goal is to find a flow that satisfies all arc capacity and node data rate requests, while minimizing total cost. Defining a flow as a function  $x : A \rightarrow \mathbb{Z}_{\geq 0}$ , the minimum-cost flow problem can be formulated as follows:

$$\min \left( z(x) = \sum_{(i,j) \in A} c_{ij} x_{ij} \right) \quad (5)$$

subject to the following two conditions:

$$l_{ij} \leq x_{ij} \leq u_{ij}, \forall (i,j) \in A \quad (6)$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = b_i, \forall i \in N \quad (7)$$

where  $N$  is the set of nodes,  $A$  is the set of directed arcs,  $l : A \rightarrow \mathbb{Z}_{\geq 0}$  is the lower capacity function on the arcs,  $u : A \rightarrow \mathbb{Z}_{\geq 0}$  is the upper capacity function on the arcs,  $c : A \rightarrow \mathbb{Z}$  is the flow cost-per-unit function on the arcs and  $b : A \rightarrow \mathbb{Z}$  is the node mass balance function on the nodes [12].

The minimum-cost flow problem is solved using the NetworkFlow constraint tool of the Java Constraint Programming solver (JaCoP) [12] in an iterative way. Basing on the request vector from the PHY layer as input parameter, the approach is the following:

- if a solution for the minimum-cost flow problem is found, the algorithm stops.

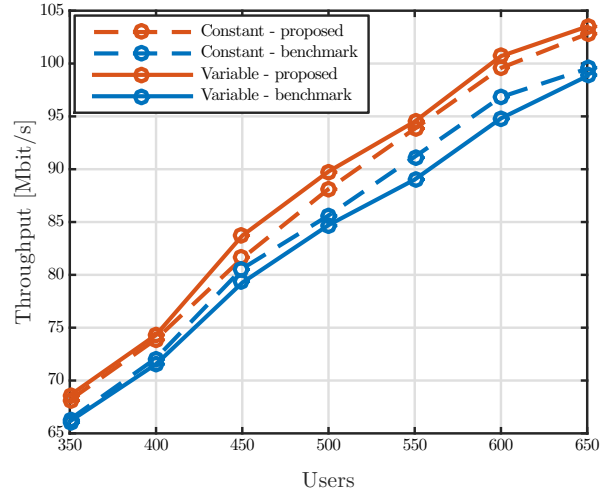


Fig. 4. Network throughput

- otherwise an iterative approach is used to determinate the additional minimum capacity for the arcs thanks to which input requests can be fully satisfied.

For both the cases, once a solution is found, the NET layer notifies to the PHY layer for each node the quantity of units  $U_B$  available for additional traffic or exceeding the maximum imposed limit causing cell outage.

## IV. NUMERICAL RESULTS

This section presents the numerical results, obtained by means computer simulations that are provided in order to validate the effectiveness of the proposed cross-layer method.

The performance is expressed in terms of throughput and unsatisfied data rate requests as a function of the mean number of UEs in the area. The advantages of the proposed iterative method are showed in comparison with a benchmark resource allocation method that does not perform any cross-layer optimization: the PHY layer determines the cell association based on the RBs availability and then sends its rate request to the NET layer that produces a consequent allocation of the resources to the backhaul links. The proposed method is iterative, but in order to have a predefined resolution time, the number of iterations has been assumed fixed. In particular we have verified that significant gains are obtained up to the 5-th iteration, then the gain is limited and does not justify additional delays.

Figs. 4 and 5 show the system throughput and the unsatisfied data rate requests, respectively. The curves refer to the cases of constant (i.e., @194kbps) and variable data rate requests (i.e., @18kbps, @160kbps and @460kbps.) of the UEs. In both the cases, numerical results prove that the proposed cross-layer method performs better than the benchmark resource allocation method.

The gain of the proposed method is achieved thanks to a more fair distribution of the traffic load on nodes, and hence, on backhaul links. This is shown in Fig. 6 where the *Jain Index* [11] is shown. This index measures the fairness and is defined as the ratio between the square mean and the mean square value of the traffic load distribution.

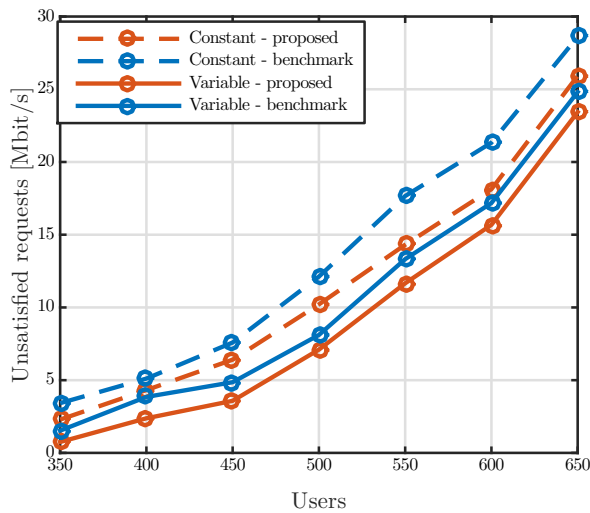


Fig. 5. Total unsatisfied data rate requests

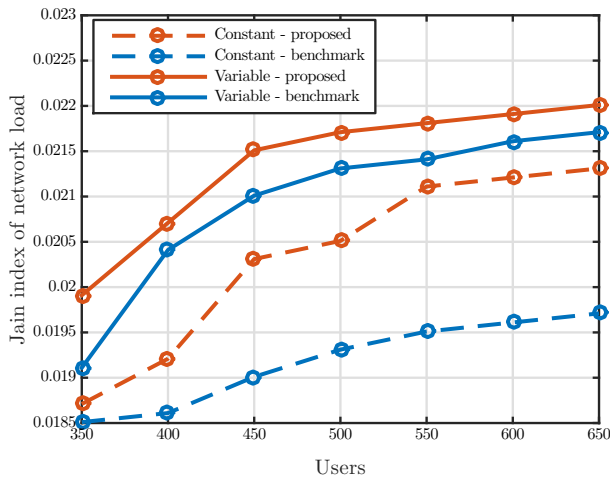


Fig. 6. Fairness of the load distribution among the network nodes.

## V. CONCLUSIONS

HetNets represent a novel networking paradigm based on the concept of access point densification and a multilayer architecture. It is considered one of the main enhancements of 5G networks to boost capacity and coverage. However, the massive diffusion of access points leads to an exponential increase of the backhaul traffic and to the need of a suitable management of the backhaul network. This paper presented a new cross-layer approach that allows to jointly allocate the resources in the access and the backhaul networks. The proposed iterative procedure is based on a new cell association procedure that is performed at the PHY layer and an optimization of the backhaul network. The results show that the joint decision allows to achieve better results in comparison with a non-iterative benchmark method.

## REFERENCES

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast," *white paper*, 2014.

[2] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 74–80, February 2014.

[3] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5g," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. 82–89, February 2014.

[4] G. Bartoli, R. Fantacci, K. Letaief, D. Marabissi, N. Privitera, M. Pucci, and J. Zhang, "Beamforming for small cell deployment in lte-advanced and beyond," *Wireless Communications, IEEE*, vol. 21, no. 2, pp. 50–56, April 2014.

[5] X. Ge, H. Cheng, M. Guizani, and T. Han, "5g wireless backhaul networks: challenges and research advances," *Network, IEEE*, vol. 28, no. 6, pp. 6–11, Nov 2014.

[6] C. Liang, F. Yu, and X. Zhang, "Information-centric network function virtualization over 5g mobile wireless networks," *Network, IEEE*, vol. 29, no. 3, pp. 68–74, May 2015.

[7] 3GPP, "Scenarios and requirements for small cell enhancements for E-UTRA and E-UTRAN," TR36.932 V12.1.0, Tech. Rep., 2013.

[8] H. ElSawy, E. Hossain, and M. Haenggi, "Stochastic Geometry for Modeling, Analysis, and Design of Multi-Tier and Cognitive Cellular Wireless Networks: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 996–1019, Third Quarter 2013.

[9] Qiaoyang Ye and Beiyu Rong and Yudong Chen and Al-Shalash, M. and Caramanis, C. and Andrews, J.G., "User Association for Load Balancing in Heterogeneous Cellular Networks," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 2706–2716, 12 June 2013.

[10] Andrews, J.G. and Claussen, H. and Dohler, M. and Rangan, S. and Reed, M.C., "Femtocells: Past, Present, and Future," *IEEE Commun. Surveys Tuts.*, vol. 3, pp. 497–508, 30 April 2012.

[11] Jain, R. and Chiu, D. and Hawe, W., "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," *Digital Equipment Corporation*, vol. TR-301, 1984.

[12] <http://jacop.osolpro.com/>

## LICENSED SHARED ACCESS (LSA) FIELD TRIAL USING LTE NETWORK AND SELF ORGANIZED NETWORK LSA CONTROLLER

Seppo Yrjölä (Nokia, Oulu, Finland; seppo.yrjola@nokia.com); Vesa Hartikainen (Nokia, Espoo, Finland; vesa.hartikainen@nokia.com); Lucia Tudose (Nokia, Espoo, Finland; lucia.tudose@nokia.com); Jaakko Ojaniemi (Fairspectrum, Helsinki, Finland; jaakko.ojaniemi@fairspectrum.com); Arto Kivinen (Turku University of Applied Science, Turku, Finland; arto.kivinen@turkuamk.fi); Jarkko Paavola (Turku University of Applied Sciences; jarkko.paavola@turkuamk.fi); Marko Palola (Technical Research Centre of Finland, VTT, Oulu, Finland; marko.palola@vtt.fi) and Tero Kippola (Centria University of Applied Sciences, Ylivieska, Finland; tero.kippola@centria.fi)

### ABSTRACT

This paper presents the results from the over the air field trial of the new Licensed Shared Access concept utilizing a TD-LTE radio access network in the IMT spectrum band 40 (2.3-2.4 GHz) in Finland. In the field trial, the LTE network shared the spectrum with Program Making and Special Events incumbent. New LSA concept elements, LSA Repository for incumbent protection information and LSA Controller for controlling the mobile broadband network in the same spectrum band were implemented in the trial environment. The trial utilized commercially available network elements like multimode multiband terminals, LTE base stations, core network and network management system. Incumbent spectrum usage data was collected to the LSA Repository, which further converts it to spectrum availability information for the LSA controller. The developed LSA Controller consists of Minimum Separation Distance and Protection Zone Optimization algorithms to analyze and optimize base station parameters according to the spectrum availability information and used network management system to configure the radio network accordingly. This was the first LSA trial which has LSA controller implemented as Self Organizing Network solution fully integrated into commercial Operational Support System. Incumbent users' rights were protected by evacuating the overlay LSA TD-LTE band and handing users over to coverage FDD LTE network when requested by the incumbent spectrum user. Numerical results are presented to quantify the duration of the LSA work flow steps in particular in emergency evacuation phase. The trial showed that the LSA concept can be implemented with commercial available network elements and a minimum amount of new software and hardware components. The performance results on the LSA system workflow indicated that in the PMSE use case the usage of the LSA band can be

managed timely manner and the incumbents' rights can be protected.

### 1. INTRODUCTION

Spectrum is one of the most in-demand resources in our digitalizing information economy. We have witnessed the exponential growth of wireless services to access information, enjoy content, and conduct commerce from mobile devices anywhere, anytime. The number of mobile broadband subscribers and the amount of data used per user is set to grow significantly over the coming years [1] leading to increasing spectrum demand. In addition to traditional exclusively licensed spectrum with long term licenses, there are globally allocated exclusive International Mobile Telecommunications (IMT) bands that are currently restricted by the incumbent use, but are mostly unused in time and location. These underutilized spectrum bands have recently been considered as an opportunity by regulators, industry and research community in finding sufficient supply of spectrum resource to meet the growing demand of the mobile broadband (MBB) communication on time. The most prominent recent spectrum sharing concepts under study in the technology, policy and business domains are the 3 tier Spectrum Access System (SAS) from the US [2] [3] and the Licensed Shared Access (LSA) [4] from Europe.

LSA is a novel spectrum sharing concept introducing spectrum sharing between a Mobile Network Operator (MNO) and another type of incumbent spectrum user. The LSA concept has received interest in both the European regulation and standardization for coordinating the spectrum access of both the incumbent and MNO in the same IMT spectrum band 40 (2.3-2.4 GHz). European Commission defines LSA as [5], *“a regulatory approach aiming to facilitate the introduction of radio communication systems*

operated by a limited number of licensees under an individual licensing regime in a frequency band already assigned or expected to be assigned to one or more incumbent users. Under the Licensed Shared Access (LSA) approach, the additional users are authorized to use the spectrum (or part of the spectrum) in accordance with sharing rules included in their rights of use of spectrum, thereby allowing all the authorized users, including incumbents, to provide a certain QoS”.

In the LSA concept [5] and [4] spectrum sharing is allowed between an incumbent spectrum user and a licensee in a binary way so that both have exclusive individual access to a spectrum at a given time and location. The spectrum regulator is responsible for identifying LSA spectrum to be licensed, defining the *sharing framework* consisting of rules and conditions for sharing as well as granting the license to the LSA licensee. Based on the national framework the incumbent and LSA licensee negotiate the private commercial *sharing agreement* under the permission and governance of the regulator. In the voluntary LSA framework and agreement the incumbent spectrum user defines the part of its spectrum that can be used for sharing with the LSA concept, the license duration and geographical area.

This paper focuses on demonstrating the LSA concept and validating key performance parameters for sharing between an MNO and another type of incumbent, in particular Program Making and Special Events (PMSE) spectrum user in the Finnish use case.

The LSA concept has been field trialed the first time in Finland by the Cognitive Radio Trial Environment (CORE) project consortium in April 2013 [6] followed by iteratively updated features demonstrated in April 2014 [7] and December 2014 [8]. To authors knowledge no other LSA field trial environment has been reported to date. The field trial presented in this paper enhances the CORE environment further by introducing first time LSA controller implemented as a part of Self Organizing Network (SON) solution fully integrated into commercial Operational Support System’s (OSS) with advanced incumbent protection algorithms needed to optimize protection zones to protect the incumbent’s business while maximizing availability for the licensee.

The rest of this paper is organized as follows. The Finnish LSA trial environment and key elements are introduced in Section 2. Section 3 presents field trial set up, work flow and operations. Performance evaluation and measurement results from the LSA trial using live commercial LTE network in the 2.3 GHz band are summarized in Section 4. Finally, conclusions are drawn in Section 5.

## 2. LSA FIELD TRIAL ENVIRONMENT

The LSA concept introduces two new elements on top of the radio access network architecture, LSA Controller (LC) and LSA Repository (LR), both under consideration in standardization [9], [10] and [11]. The LSA trial environment consists of the following key elements as shown in Fig. 1:

- Commercial available heterogeneous LTE network of TDD and FD LTE macro and small cell Evolved Node B (eNB) Base Stations (BSs), Evolved Packet Core (EPC) core network, network management system (NMS) and end user equipments (UEs),
- PMSE incumbent spectrum users with LR and the Incumbent Spectrum Manager,
- LC utilizing commercially available OSS NMS and SON platforms and interfaces with incumbent protection algorithms and SON features

The stakeholder roles, individual system elements, their operations and connections are discussed in more detail in the following sub-sections.

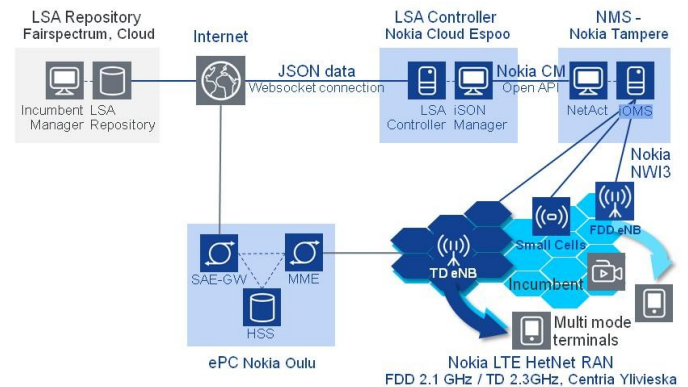


Figure 1. Field trial environment for LSA

### 2.1. LSA repository and Incumbent manager

Incumbent spectrum user in the trial is selected according to the national Finnish LSA use case to be an employees of a media or broadcasting company using PMSE services in program making on the IMT band 40 (2.3-2.4 GHz), as defined in [12] and [13].

The LR and the Incumbent spectrum manager tools are developed in the Tekes Trial White space test environment for broadcast frequencies (WISE2) project [14]. The LR is a database with the following key functions [11]:

- supports the entry and storage of information describing Incumbent’s usage and protection requirements,
- conveys availability information to authorized LCs,

- receives and stores acknowledgement information received from the LCs,
- provides means for National Regulatory Authority (NRA) to monitor operation of the LSA System, and to provide the system with information on the Sharing Framework and the LSA Licensees and
- ensures that the LSA system operates in conformance with the Sharing Framework, and may in addition implement any non-regulatory details of the Sharing Arrangement

Based on this information, protected areas are defined based on the underlying regulatory requirements. These protected areas are *exclusion zones* within which LSA Licensees are not allowed to have active radio transmitters, *protection zones* where incumbent receivers will not be subject to harmful interference caused by LSA Licensees' transmissions or *restriction zones*, where LSA Licensees are allowed to operate radio transmitters, under certain restrictive conditions e.g. maximum effective isotropic radiated power (EIRP) limits and/or constraints on antenna parameters [10].

Incumbent user is requested to make a specific LSA *Spectrum Resource Availability Notification* to enable the LR to send LSA spectrum resource availability information to the LC. It can be used to send either specific immediate notifications, or periodic updates of the overall LSA spectrum resource availability information related to this LC. In addition Licensee could send the LSA *Spectrum Resource Availability Information Request* to make a request for LSA spectrum resource availability information. This procedure can be used to initiate LSA operation, or to synchronize information between LR and LC during LSA operation.

It is essential that these procedures should not increase incumbent user's operational load while ensuring Quality of Service (QoS) and security, robustness, reliability and fault management functional requirements. Two tools are developed in the WISE2 project for collecting incumbent reservations: The *LSA Incumbent Manager* (IM) and the PSME Location application. Tools are illustrated in Fig. 2. With these applications the incumbent user sets a spectrum band, informs availability term, request multiple protection areas, define a protection type and remove the protection from the LR.

The LSA Incumbent Manager tool allows incumbent user to request multiple protections in time, for example, to register coming sport venue casting event. The incumbent user sets a location to the map interface, a protection type, a transmit frequency range, antenna height, and a casting time. The LSA Incumbent Manager processes the protection information and sends data to the LR. The LSA Incumbent Manager connected PMSE location application uses mobile

phone's location services such as GPS to track the location of incumbent, which can be updated to LR accordingly.

The incumbent user can select between different PMSE use case protections such as Cordless camera, a Mobile video link and a Portable video [13]. PMSE use case information consists a protection distance, protection height threshold, transmit power limits and a reference to the broadcasting event setup and is used by the LC protection algorithms to calculate accurate protection areas for each PMSE use. Different event setups include clear line-of-sight used with *Cordless cameras*, suburban area with no line-of-sight used with *Portable video* and cameras on-board vehicles used with *Mobile video link*.

The LR stores information describing Incumbent's usage and conveys availability information to authorized LCs when the information changes. The LR is able to communicate with several LCs. In this trial, the LC located within the LSA Licensee's domain has the main responsibility of computing the protection criteria and implementing the terms of the sharing agreement between licensee and the incumbent. Based on the availability information, reservations and possible incumbent movement, licensee should act upon, and decide whether it can use LSA base station resources at certain location or not.

Regulators may monitor spectrum usage via the LR, which monitors the LSA system for possible exception situations such as the unavailability of LC or unconfirmed protection request. Notification will be sent to regulator immediately if failure occurs.

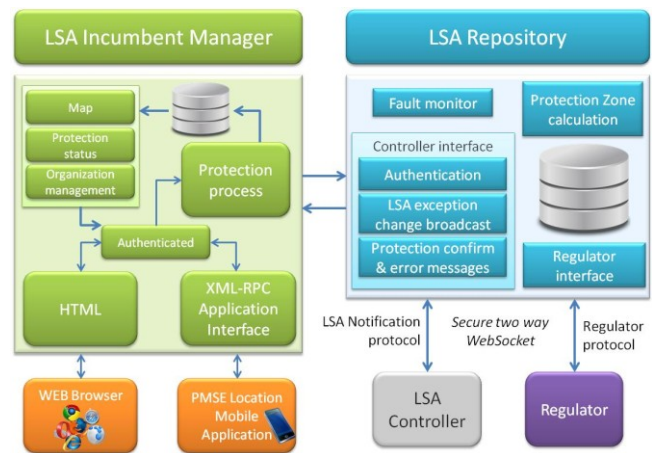


Figure 2. The LSA Repository and LSA Incumbent Manager tool for the PMSE users

## 2.2. LSA controller

The LC developed in the Tekes Trial Cognitive Radio to Business (CRB) and Local Area Spectrum Sharing (LASS) projects by the Nokia in collaboration with the CORE+ consortium [15] provides the MNO licensee with means to

access the LSA spectrum and to react on the incumbent user activity. The LC located within the LSA Licensee's domain:

- enables the LSA Licensee to obtain spectrum resource availability information from the LR
- enables the LSA Licensee to provide acknowledgment information to the LR
- interacts with the Licensee's mobile network in order to support the mapping of availability information into appropriate radio transmitter configurations and to receive the respective confirmations from the mobile network

In the trial, the MNO licensee is able to receive de-activation and activation requests from the LR based on incumbent's frequency reservation reports. The incumbent's reservations include *location, frequency bands, time range, a PMSE protection type and the emergency evacuation*.

### 2.2.1. LC User Interface

The key features of the LC in the trial are incumbent protection algorithms and MNO user interface (UI). The MNO LC trial UI has three main sections: trial network status, LSA network control and protection algorithms. The LC controller gathers information on LTE BSs and cells and their *statuses* through the NMS. The LTE base stations, sectors of LTE network and sector states are depicted in the map view as shown in Fig 3. The current LTE sector state of five LTE macro cells and one small cell with their statuses in the trial is shown with different colors; green sector is active, yellow sector is being evacuated and grey sector has been evacuated. LSA spectrum band information and possible existence of incumbent(s) is also collected, and presented in the same section of the UI. In the example, there is a single incumbent in an active mode and operating in the same band and area as the licensee, so LSA spectrum resource de-activation process is currently ongoing for the impacted cell.

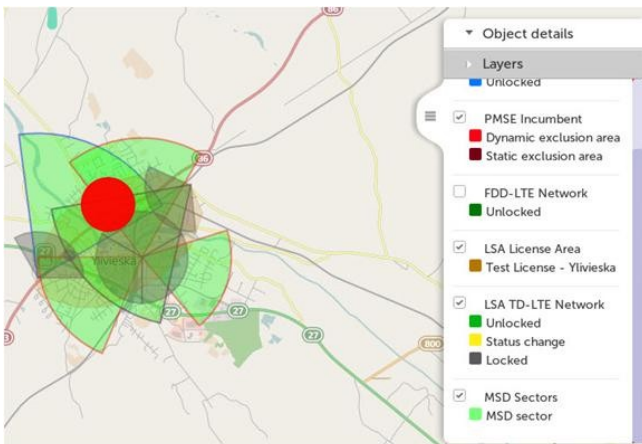


Figure 3. LSA trial network status view in the LC UI.

In the LSA *network control UI* view shown in the Fig. 4 the MNO licensee's is able to control set of features with respect of LSA network, its usage, and incumbents' protection. In the first selection, called *LSA network enabled*, MNO can prevent the LC of using the LSA BSs even though it could have spectrum resource available. Similar manner, if LSA network is currently in use, and MNO un-checks this option, the LC will lock automatically the respective BSs.

In the control menu the *graceful shutdown* option enables BS transmitters to lower its power level step-by-step during the selected time period before locking the air interface instead of shutting the transmitter in the cell down abruptly during de-activation. This allows terminals to detect another network and carry out a seamless handover avoiding a cell reselection, causing potentially a connection break to the ongoing session.

The *emergency evacuation* enables locking all the LSA network BS cells according to pre-defined emergency plan with automatic plan activation operation for the whole network as in the case of public safety access class barring NMS feature [16].

Ylivieska LSA Network	Administrative Status	LSA Operation Started	LSA Operation Ended
Ylivieska LSA Network			
Fouahala LSA 70			
Sector 29201	unlocked	2015-05-15 12:03:36	2015-05-15 12:04:47
Ratakatu LSA 71			
Sector 29457	unlocked	2015-05-15 12:03:36	2015-05-15 12:04:47
Sector 29458	unlocked	2015-05-15 12:03:36	2015-05-15 12:04:47
Sector 29459	unlocked	2015-05-15 12:03:36	2015-05-15 12:04:47
Alpumi LSA 72			
Sector 29713	unlocked	2015-05-15 12:03:36	2015-05-15 12:04:47
Small cell LSA 75			

Figure 4. Mobile network operator UI for selecting the trial features and controlling the LSA network use.

### 2.2.2. LC algorithms

An essential part for allowing the coexistence between the LSA network and the incumbent is to define a criterion which guarantees an interference-free operation of the LSA licensee and incumbent transmissions. In the implemented LC band evacuation can be based on two algorithms. *Minimum Separation Distance* (MSD) protection algorithm calculates the minimum required distance between the incumbent and the LSA transmitter taking into account both the Incumbent and Licensee radio transmission parameters and in particular the cell sector antenna configuration, such as direction and down tilt angles, to calculate the MSDs to specific geographical directions. The Incumbent protection distances are consistent with the methodology presented in [13] corresponding to the worst case scenarios.

However, since the mobile broadband network (MN) is an interference-limited system where multiple spatially separated BSs are transmitting simultaneously on the same frequency band, the aggregate field strength created by the

MN at the incumbent receiver can result in intolerable interference. Therefore, a more advanced protection criterion is developed. The second algorithm, demonstrated here first time, is the *Protection Zone Optimization (PZO)* algorithm. Even if the MSDs of all individual BSs are satisfied the interference created by the MN can be higher than allowed, resulting in MSD shorter than MSD of any single LSA transmitter, that is, the aggregate interference from all BSs of the network can exceed the Protection zone limit even if none of the BSs exceeds it alone. This limit is defined by the incumbent receiver sensitivity, noise floor, and additional interference margin.

The PZO method computes the cumulative interference created by the MN. Specifically, linear optimization methods and accurate propagation modeling is used to determine the individual cells which are required to be switched off so that the resulting aggregate field strength at the Incumbent receiver remains below the Protection zone limit. This allows the MNO to operate its network at full viable capacity while satisfying the criteria for interference-free operation of the co-existing Incumbent. Example of the calculated MSD protection areas in the case where incumbent was on adjacent frequency and all the macro cells had been evacuated except the cell pointing to South-East and small cell is shown in the Fig. 5.

The LC algorithm outputs two lists: 1) BS cells which cause interference and should be evacuated if sectors are active and 2) cells that are not interfering with at least one of the incumbent users and are possible candidates for activation. However, a cell can be activated only if the same cell is not included to the other incumbents' lists and the sector is currently off air.

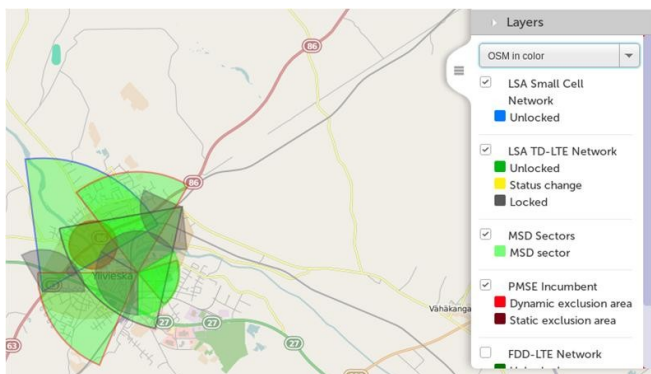


Figure 5. LSA incumbent protection algorithm view in the LC UI

The *Load balancing* [17] is an additional SON feature of the trial system, allowing monitored and controlled terminals to switch between the coverage layer FDD-LTE and the LSA TD-LTE networks on demand. Load balancing is LTE SON self-optimization feature that aims to even out the load generated across the network by moving users from one cell to another [18]. LSA enabled BSs can be used as an additional capacity layer, providing more capacity to

balance the load and optimized connectivity experience for users. The nature of LSA spectrum availability leads to considerations on which users groups can be best served and are least affected by possible evacuation.

### 2.2.3. LC architecture

The LC used in the trials is developed in the Tekes Trial CRB and Tekes 5G LASS projects. Demo controller fully utilizes commercially available OSS solutions and related *Intelligent Integrated SON (iSON)* architecture and interfaces.

Nokia Networks implements SON functions within and beyond standards, and controls them in a single tool called iSON Manager. iSON Manager simplifies daily network operations by providing closed loop automation, at the same time enabling experts to maintain control and visibility of what is going on in the network. iSON Manager provides content packs for self-configuration, self-optimization and self-healing. Each content pack includes support for specific end-to-end process for SON.

Radio access networks comprise complex combinations of cells, frequencies, technologies and layers that require smart optimization and network management. iSON Radio automates the configuration, healing and optimization of such networks. By automating the management of Heterogeneous Networks (HetNets), iSON enhances their interworking and mobility. With tools to manage and interoperate multiple layers and technologies, iSON ensures *small cells interwork* with the macro layer, even in a multivendor environment. Other key SON functions are traffic steering and mobility management. *Traffic steering* directs traffic to a particular radio access technology (RAT) or layer to enable operators to optimize their resources, improve the way users experience services and minimize power consumption. Traffic steering works hand-in-hand with *mobility management* to ensure a reasonable number of handovers and eliminate radio link failures. It also considers other factors such as the capabilities of the terminals and network and the load in different RATs and layers. Today, most network operating processes are well established. It can be hard to identify the right time and place to intervene by starting to implement automation to raise efficiency and reduce complexity. Process integrated SON addresses exactly that - it enables SON support for operator process sub-entities like *site creation* and LSA [19].

In the trial, LSA1 interface is utilizing existing Protocol to Access White-Space (PAWS) protocol [20] and JSON data websocket connection interface. The LSA\_OAM interface uses available Nokia proprietary CM Open API based on Web Services within the OSSii OSS interoperability initiative between different OSS vendor's equipment [21] as the Third Generation Partnership Project (3GPP) Service and System Aspects Telecom Management working group (SA5) standardization work has just started

as 3GPP Work Item study on operations, administration and management (OAM) support for the LSA [22].

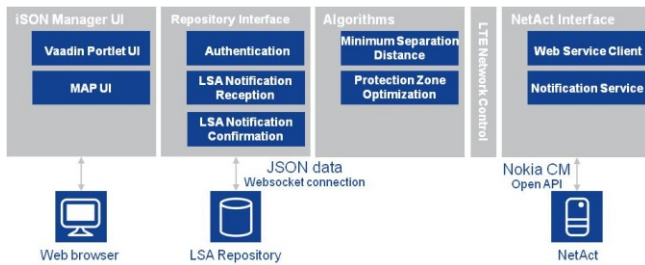


Figure 6. LC architecture and interfaces

## 2.2. LTE network

LTE network in the LSA trial environment consists of commercial available 3GPP Release 8+ compliant radio accesses and core network as shown in the Fig.7. NET's commercially available LTE-Advanced capable Flexi Multiradio 10 Base Stations are used. Three macro TD-LTE base stations and one small cell at IMT band 40 (2.36-2.40 GHz) are located in the vicinity of the city and Centria University of Applied Sciences campus area in Ylivieska, Finland. In addition two FDD-LTE macro Flexi Multiradio 10 Base Stations provide primary LTE coverage layer to the same area in the IMT band 1 (2.1 GHz) and remains available also should the LSA spectrum resource become temporarily unavailable.

All the LTE eNB BSs are connected to LTE ePC core network at NET Oulu and are managed from a single point by the multi-technology, multi-vendor NetAct OSS NMS platform located in Tampere. The developed SON LSA demo controller discussed above runs in NET Espoo and interfaces with the NetAct NMS in order to exchange network information and to execute management operations.

Commercial LTE multi-mode (FDD and TD) multi-band (band 1 and 40) UEs are used supported by major chip set vendors. In the trials Samsung S4 terminals supporting seamless TD-FDD handover are used.

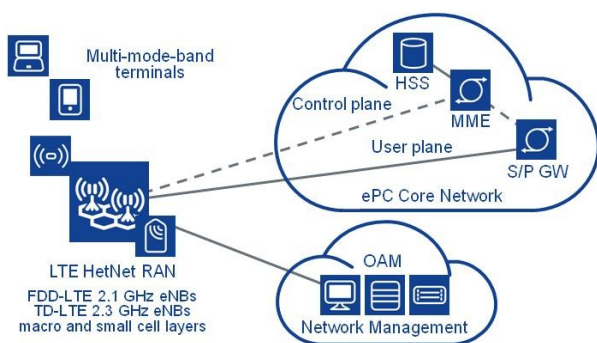


Figure 7. LTE trial network elements and connections

## 3. LSA FIELD TRIAL SET UP

The first LSA field trial using commercially available over-the-air LTE network in the 2.3GHz IMT band 40 was shown in the CORE+ trial environment in Finland April 2013 documented in [6] followed by iteratively enhanced trials utilizing research LSA controller developed by Finnish National Research Center VTT [7], [8] and [23]. A new enhanced trial introduced in this paper with SON LSA controller was first trialed in Finland with CORE++ consortium in May 2015 using the trial environment described in Section 2.

### 3.1. Key features

The key features of this LSA trial include the following:

- Incumbent manager tools to track incumbent spectrum data and store into LR
- Incumbent spectrum user's data stored in LR
- Commercial FDD and TD LTE eNBs at IMT band 1 and 40.
- Commercial ePC core network and OSS NMS with advanced SON features
- MSD and PZO protection algorithms to maximize the LSA spectrum resource usage while ensuring agreed incumbent protection.
- SON LC to manage TD-LTE BSs via the NMS according to LR's LSA spectrum resource availability information.
- SON LC emergency evacuation feature to evacuate LSA band on demand.

This trial goes beyond [23] by introducing novel protection algorithms implemented into SON LSA controller integrated with the commercial OSS NMS.

### 3.2. Trial workflow

The overall flow of the LSA procedures and the trial specific demonstration workflow are discussed next. Based on the LSA system architecture [11] we have divided LSA operative flow into the three main phases:

- 1) LSA provisioning,
- 2) LSA operations and
- 3) LSA release phase.

In the *provisioning* phase, the sharing framework, sharing agreement and licensing will take place between regulator, incumbent and licensee. In the trial Finnish NRA FICORA has granted the trial license to use the 2.36 – 2.40 GHz band in the trial test license area in Ylivieska, Finland.



Based on this information licensee identify, configure and optimize radio network for the LSA spectrum resources.

In the second *operations* phase, the incumbent spectrum user starts *activation* by informing the LSA band availability to the licensee via the LR. The licensee activates and configures BSs to use vacant band based on the spectrum availability information and reports on LSA band usage. During the LSA operation licensee estimate interference, optimize LSA spectrum resource usage and maintain QoS and Quality of Experience (QoE). If requested by the incumbent, the licensee needs to *de-activate* spectrum resources and return it to the incumbent. Actions include cell de-activation, interference estimation, maintaining QoS and QoE and confirming usage to LR. Now the incumbent can re-start to use the band and inform the LR when the LSA spectrum resource could be handed back for the licensee.

In the final *release* phase, the licensee fully releases the LSA spectrum resource when the LSA license expires and the incumbent can start to use it for its own purposes.

The technical LSA trial in this paper is focused on validating performance of the second operations phase. The key components of the trial was discussed in the section 2 and shown in Fig. 1 and Fig. 2. The trial of the LSA procedure flow in which the incumbent updates usage and protection requirements via LR notification of new LSA spectrum resource availability information consists of the following steps [11]:

- LR activation
- LC registers with LR
- LR configures itself with up to date set of incumbent requirements
- Incumbent modifies the usage and protection requirements and inputs them to the LR
- LR checks compliancy with the Sharing Framework and Sharing Arrangement
- LR processes the changes taking into account other possible Incumbent requirements
- LR informs the LC about the updates
- LC acknowledges (ACK) modified LSA spectrum resources availability information after LC has processed the changes and requested the NMS to update the cell configuration(s).
- Once the changes have been applied by the MBB network LC ACK the updated LSA spectrum resources availability information to LR
- LC has an up to date set of LSA spectrum resources availability information from the incumbent
- MBB network operates on the available LSA spectrum resources accordingly

The exchange of data between the LR and the LC is supported by a procedure at LSA1 interface as shown in the Fig. 8. In this trial use case as soon as new LSA spectrum

resource availability information is configured in the LR, the LR notify immediately the LC.

1. The LR sends a LSA Spectrum Resource Availability Notification message to the LC, containing new or updated LSA Spectrum Resource Availability information.
2. The LC will upon reception of the LSA Spectrum Resource Availability Notification message check the consistency of the information provided.
3. If consistency check is successful, the LC will respond with a LSA Spectrum Resource Availability Notification ACK message to confirm the reception of new spectrum resource availability information.
4. Upon successful configuration of the LSA spectrum resources, the LC sends a LSA Spectrum Resource Confirmation message to the LR to confirm execution of changes in the mobile network.
5. Upon reception of the LSA Spectrum Resource Confirmation Request message, LR acknowledges the reception of the confirmation by sending a LSA Spectrum Resource Confirmation Request Response message to the LC.

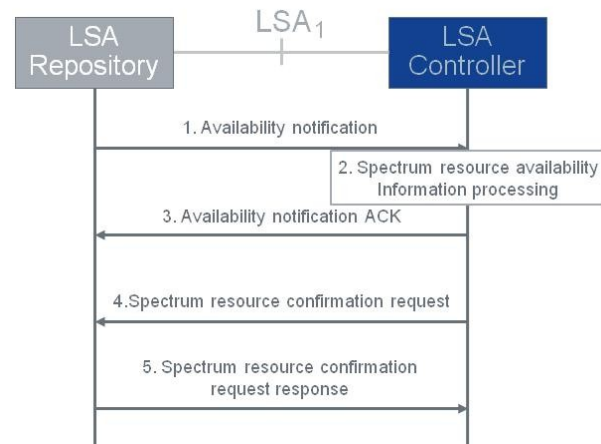


Figure 8. LSA message flow for the LR notification of new LSA spectrum resource availability information

#### 4. PERFORMANCE VALIDATION

The most important performance indicator is the evacuation time from the Incumbent evacuation request to the time the affected LSA base station cells are locked and off-the air.

##### 4.1. Measurement system

The LSA procedures and functions of the system elements described above can be presented as the different phases of the LSA spectrum resource evacuation process for the trial performance validation measurements as follows:

- 1) The incumbent makes an evacuation request via LSA Incumbent Manager. The LSA process starts as the incumbent spectrum user makes an evacuation request to the LSA Incumbent Manager. The LSA Incumbent Manager submits the information to the LR which forwards the information to the LC.
- 2) LC receives Incumbent information from LR. Based on the Incumbent user information, the LC calculates which BSs or cells on the LSA network are impacted and submit de-activation command to the NMS accordingly.
- 3) NMS receives the de-activation command from LC and executes de-activation radio plan for the affected BSs and cells on the LSA network. Two de-activation radio plans are used. In urgency, the MBB network locks i.e. turns off transmitters of the impacted BSs and cells and UEs will automatically start a cell re-selection procedure. Alternatively when evacuation is known in advance *graceful shutdown* enables the power of the LSA BSs or cells to be decreased gradually so that UEs will do a seamless handover when the signal level at the serving cell drops below the signal level of the available FDD cell.
- 4) BS or cell in the LSA network is de-activated followed by the no LTE signal detected in the LSA spectrum. The NMS finishes the radio plan execution, begins the LSA cell status check and sends cell off-the-air status update to the LC.
- 5) As soon as all needed LSA cells have reached off-the-air status confirmation from NMS, LC ends evacuation and submits evacuation completed information to LR.
- 6) Incumbent user receives a confirmation on the evacuation to the LSA Incumbent Manager.

In the case of Graceful Shutdown, the time for the shutdown as well as the step size for the eNB power decrease can be specified. This time will be added to the band evacuation time. The Anite's Nemo Outdoor drive test tool was used to record LTE signaling information from the UE in order to record the time stamp for the BS off-the-air.

#### 4.2. Measurement results and discussion

Described LSA band evacuation process utilizing the enhanced SON LC was implemented into the Finnish CORE++ LSA trial environment and initial performance measurement studies have been conducted to evaluate the involved time scales. Corresponding time stamps are reported in Tables 1 and 2.

The evacuation performance measurement results in Table 1 indicate that, in the case of the evacuation of the

first cell, it takes in average 24 seconds from making the evacuation request until the LSA band has been cleared and 27 seconds until the confirmation of the evacuation of all the cells is visible to the incumbent in the LSA Incumbent Manager. As discussed in the section 2 the trial environment consists of both commercial network elements and specific demo elements developed for the LSA trials. In the table 2 the measurement results have been divided accordingly into a LSA SON Demo Controller platform and commercial LTE NMS delays. The LTE NMS was in average 49 seconds for both algorithm cases. The NMS configuration plan provisioning time was minimized by utilizing pre-validated radio plans in which case the execution time of provision operation is shorter and not linearly dependent on the LSA network size. In the NMS performance tests utilizing the *Access Class Barring* feature [16], NMS could be commanded to skip plan validation and even country wide emergency plan download with automatic activation operation for the whole network could be completed below 3 minutes. In the trial provisioning a multi-site de-activation radio plan takes only about 3 more seconds to complete than a single site command indicating promising results for larger LSA network management.

The LSA system demo platform execution time consists of the LR and the LC delays. The LR delay was approximately 0,5 s while LC was 2,6 s for the MSD and 1,9s for the PZO algorithm. The shorter execution time for the more complicated PZO case was due to additional pre-processing phase in the work flow implementation. Demo platforms shorter operational delays compared to CORE+ research platform mainly results of integrating four of six distributed locations as a part of the OSS SON LC, leaving LR and Incumbent Manager tools as only external systems. In the trial, the research platform delay was reduced approximately 90% compared to [23].

The results presented in Tables 1 and 2 reveal that the developed concept works in realistic scenarios with live network. The average evacuation time of 51 seconds is an adequate result for LSA evacuation time for most incumbents and particularly for the Finnish PMSE use case. On the other hand locking or unlocking large live networks may activate wide load balancing and self-optimization routines [24] which could take hours before mobility and cell selections are again fully optimized in the adjacent cells. This process as well could be speed up by pre planning LSA use cases as a part of initial network planning in particular for the static and semi static LSA use cases.

The future research includes studies and trial how to ensure consistent QoS and QoE for the end users when LSA spectrum resource availability changes abruptly. In this trial the tests were made in optimal conditions without e.g. the real network congestion, which could slow down the evacuation process considerably. The users connected to interfering evacuated cells will experience a Radio Link

Failure (RLF) if the cells are locked abruptly hard shutdown. In order to reduce the number of RLF, the shutdown or the modification of the Tx power and or antenna downtilt could be done during a certain Graceful Shutdown period, allowing users to be handed over to other cells. The third thing to consider is which alternative network to use for the back off handovers. For example, the alternative network could be of lower capacity or it could be congested, leading to lowered QoS for the end users after the LSA evacuation process. This is an active research topic for carrier aggregation, load balancing, traffic steering and handover optimizations in a HetNet MBB network management.

Table 1. LSA band evacuation measurement results

	Meas point	Evacuation MSD		Evacuation PZO	
		Time[s]	SD[s]	Time[s]	SD[s]
		1. Incumbent makes evacuation request via LSA Incumbent Manager (IM)	LSA IM	0	
2. LC receives incumbent information from LR	LC	0,27		0,27	
3. OAM starts de-activation command	OAM	2,35	1,74	1,17	0,75
4. eNB/cell on LSA band is deactivated	LSA band	24,40	1,53	24,19	2,13
5. OAM notify LC that plan commission is completed	LC	51,30	1,54	50,88	0,75
6. Incumbent user receives confirmation on evacuation to LSA IM	LSA IM	51,57	1,73	51,14	0,68

Table 2. Total measured execution times of each trial system element

Total execution time [s]	e2e		component	
	MSD	PZO	MSD	PZO
NMS	48,49	48,78	48,49	48,78
LC	51,03	50,61	2,55	1,83
LR	51,57	51,14	0,54	0,54
Algorithm calculation	0,30	0,09	0,30	0,09

## 5. CONCLUSIONS

The paper has presented a field trial demonstration of the novel LSA concept that allows a mobile network operator to share spectrum resource from other type of incumbent spectrum users. The trial successfully demonstrated that a TD-LTE network licensee can take the IMT band 40 (2.3-2.4 GHz) into LSA use and vacate it when requested by the Incumbent spectrum user. The load between bands were balanced utilizing Load Balancing method and in the case of

evacuation end users proactively did hand over to FDD LTE networks to maintain their connection, enabled by Graceful Shut down feature.

The trial showed that the dynamic availability of the LSA spectrum resource can be managed with commercially available network elements and a minimum number of additional components, namely the LSA Repository and the LSA Controller. Furthermore the trial demonstrated first time the LSA Controller developed as a SON feature integrated with the commercial OSS system. Advanced protection algorithms were tested to maximize LSA spectrum resource availability for the licensee while ensuring incumbent protection.

Performance validation was conducted by measuring the duration of the spectrum evacuation workflow steps in releasing the LSA band due to Incumbent's immediate spectrum resource availability notification. The measurement results revealed that the evacuation operation can be done in a way that fulfills typical PMSE service incumbent's requirements in the Finnish sharing use case and wider in a static and a semi-static LSA use cases. Comparing results to previous research platform based LSA controller demonstrations OSS integrated LSA controller reduced overall LC operations delay approximately 90%.

In the future, the SON LSA Controller will be enhanced by exploiting further features from LTE Advanced and self-organizing networks, e.g. Carrier Aggregation, more sophisticated Load Balancing and Traffic Steering. Interference measurements will be conducted to measure real interference levels in the trial environment to further develop algorithms and to help regulation and standardization in defining the actual rules and conditions for sharing framework and arrangements. In particular, to further develop dynamic protection algorithms to maximize LSA resource availability on the border area between radio systems. This will be done by means of dynamically configuring base station cell radio parameters like transmission power, antenna tilts and antenna beams. Furthermore CORE++ trial environment will be extended to cover 3.5 GHz spectrum band to study LSA concept's evolution paths towards the US 3 tier Spectrum Access System (SAS) and 5G.

## 6. ACKNOWLEDGMENT

This work has been done in the CRB, LASS, CORE++ and WISE2 research projects within the Trial and 5G programs of Tekes - the Finnish Funding Agency for Technology and Innovation. The authors would like to acknowledge the project consortium members: VTT Technical Research Centre of Finland, University of Oulu, Centria University of Applied Sciences, Turku University of Applied Sciences, University of Turku, Aalto University, Nokia, Fairspectrum, Anite, Finnish Communications Regulatory Authority and Tekes.

## 10. REFERENCES

- [1] Cisco white paper, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019,” [Online]. Available: [https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white\\_paper\\_c11-520862.pdf](https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf), Feb. 2015.
- [2] The White House, President’s Council of Advisors on Science and Technology (PCAST) Report, “Realizing the Full Potential of Government-Held Spectrum to Spur Economic Growth,” July 2012.
- [3] The FCC, “The 3.5 GHz report and order and second further notice of proposed rulemaking,” [Online]. Available: [http://transition.fcc.gov/Daily\\_Releases/Daily\\_Business/2015/db0421/FCC-15-47A1.pdf](http://transition.fcc.gov/Daily_Releases/Daily_Business/2015/db0421/FCC-15-47A1.pdf), April 2015.
- [4] ECC Report 205, “Licensed Shared Access,” 2014.
- [5] European Commission, Radio Spectrum Policy Group, “RSPG Opinion on Licensed Shared Access,” RSPG13-538, November 2013.
- [6] M. Matinmikko, M. Palola, H. Saarnisaari, M. Heikkilä, J. Prokkola, T. Kippola, T. Hänninen, M. Jokinen, S. Yrjölä, “Cognitive Radio Trial Environment: First Live Authorized Shared Access-Based Spectrum-Sharing Demonstration,” *IEEE Vehicular Technology Magazine*, vol. 8, no. 3, pp. 30-37, Sept. 2013.
- [7] M. Palola, M. Matinmikko, J. Prokkola, M. Mustonen, M. Heikkilä, T. Kippola, S. Yrjölä, V. Hartikainen, L. Tudose, A. Kivinen, J. Paavola, and K. Heiska, “Live field trial of Licensed Shared Access (LSA) concept using LTE network in 2.3 GHz band”, in the 7th IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), McLean, Virginia, USA, Apr. 1st-4th, 2014.
- [8] ETSI workshop on Reconfigurable Radio Systems [Online]. Available: <http://www.etsi.org/news-events/events/807-etsi-rrs-workshop-2014>, Dec. 2014
- [9] ETSI, “Mobile broadband services in the 2 300 MHz – 2 400 MHz frequency band under Licensed Shared Access regime, “ ETSI System reference document. TR 103 113, V. 1.1.1. July 2013.
- [10] ETSI, “System requirements for operation of Mobile Broadband Systems in the 2300 MHz - 2400 MHz band under Licensed Shared Access” TS 103 154, Oct. 2014.
- [11] ETSI, “System Architecture and High Level Procedures for operation of Licensed Shared Access (LSA) in the 2300 MHz-2400 MHz band” ETSI System Architecture document, TS 103 235, V0.0.9, April 2015.
- [12] ERC Report 38, “Handbook on Radio Equipment and Systems Video Links for ENG/OB use,” May 1995.
- [13] ECC Report 172, “Broadband Wireless Systems Usage in 2300-2400 MHz,” Mar. 2012.
- [14] WISE2 project web page [Online]. Available: <http://wise.turkuamk.fi>.
- [15] CORE+ project web page [Online]. Available: <http://core.willab.fi>.
- [16] 3GPP technical report, “Access Class Barring and Overload Protection,” TR 23.898, 2007.
- [17] 3GPP, “SON Policy and Optimization Function Definitions,” TS 32.522 V11.7.0, Sept. 2013.
- [18] M. Mustonen et al., “Cellular Architecture Enhancement for supporting Licensed Shared Access (LSA) Concept,” *IEEE Wireless Commun. Mag.*, vol. 21, no. 3, pp. 37–43, June 2014.
- [19] Nokia whitepaper, “Intelligent Self Organizing Networks (iSON),” [Online]. Available: [http://networks.nokia.com/sites/default/files/document/nokia\\_ison\\_white\\_paper.pdf](http://networks.nokia.com/sites/default/files/document/nokia_ison_white_paper.pdf)
- [20] IETF draft-ietf-paws-protocol-20, “Protocol to Access White-Space (PAWS) Databases,” Nov. 2014
- [21] OSSii OSS interoperability initiative web page [Online]. Available: <http://www.ossii.net/>
- [22] 3GPP Work Item, “ 670028 (FS\_OAM\_LSA) Study on OAM support for Licensed Shared Access (LSA) [Rel-13],” April 2015.
- [23] CEPT, “Technological and regulatory options facilitating sharing between Wireless broadband applications (WBB) and the relevant Incumbent services/applications in the 2.3 GHz band,” CEPT Report 56, March 2015.
- [24] S. Hamalainen, H. Sanneck and C. Sartori (Eds.), “LTE Self-Organizing Networks (SON),” John & Wiley Sons, Ltd., 2012.

# Cellular Baseband Development Platform with an open RF Interface

Benjamin Weber\*, Harald Kröll\*<sup>†</sup>, Stefan Altorfer<sup>†</sup>, Qiuting Huang\*

\*Integrated Systems Laboratory, ETH Zurich

<sup>†</sup>ACP AG, Zurich

**Abstract**—Cellular modems have different architectural demands, depending on which features of the cellular standard they support. This leads to a variety of baseband processing architectures ranging from dedicated VLSI implementations to processor-centric architectures with accelerators. In this paper we present a cellular modem SDR development platform based on an RF-board with an open FMC interface. The FMC interface opens a variety of FPGA, DSP, and CPU resources for baseband algorithm investigation and implementation. As an example we present 2G, 3G, and 4G RF FMC modules and a GSM/Evolved EDGE modem based on the platform.

## I. INTRODUCTION

A cellular modem consists of RF, analog baseband, and digital processing blocks as shown in Fig. 1. Analog signal processing comprises up/down conversion, analog baseband processing, and conversion between the digital and analog domains. Digital processing comprises computationally demanding Digital Baseband (DBB) and less demanding protocol stack software. In many implementations, the digital processing is physically distributed over DSPs with attached accelerators and one or more processing units. The analog and digital processing blocks can be implemented on a single die or on separate chips. The control and data plane interface between analog and digital processing is standardized in the DigRF [1] and RBDP [2] specifications for inter-chip communication.

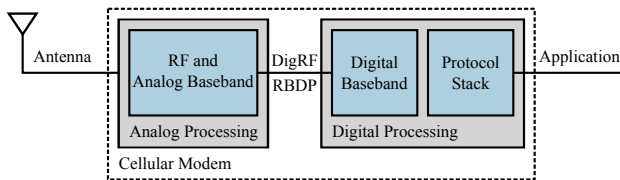


Fig. 1. A cellular modem consists of analog and digital processing which can be distributed over separate chips. Standard interfaces such as DigRF or RBDP are used for inter-chip communication.

Different applications for cellular modems require different baseband architectures, because every application has its own demands regarding data rates and power consumption. The vast number of 3GPP specifications spans a large design space for cellular modems which can be exploited to build e.g. area and power efficient cellular Internet of Things (cIoT) nodes or high data rate modems. A cellular modem has a number of features it can support. They influence the baseband architecture significantly. The following items are options for

3GPP 2G cellular modems [3], [4] (a similar number of options exists for 3G cellular modems):

- GSM only, GPRS, EDGE (EGPRS), or Evolved EDGE (EGPRS2)
- Receive diversity
- Downlink dual-carrier
- Latency reductions
- Multislot class type
- Voice enhancements (VAMOS, wideband)
- Simultaneous data and voice connections
- EC-GSM, coverage enhancements for cIoT applications

By the same token, 3GPP 4G LTE modems can support a number of different UE categories ranging from high-throughput full-duplex MIMO (e.g. UE category 4) to single receiver half-duplex UE category 0 for cIoT [5].

The support of options has to be carefully made. It would not make sense to equip a cIoT node, which reports the humidity on a weekly basis, with receive diversity or MIMO and high data rate support. On the other hand, a mobile broadband residential router should support the highest possible data rate and power consumption is of secondary importance. Therefore, the digital processing after the analog processing blocks needs to be composed according to the application and the chosen 3GPP release and options the modem shall support.

SDR development platforms can be used for prototyping wireless transceivers [6]. A vast number of SDR development platforms have been introduced in the past, where computational power is provided by CPUs, DSPs, and FPGAs. The use of the FPGA Mezzanine Card (FMC) standard [7] for SDR development platforms brings modularity and interoperability between components from different providers. In [6], the FMC concept for SDR was thoroughly studied, where versatile RF FMC modules from Analog Devices [8], [9], called FMCOMMS, are used. The FMCOMMS1 module [8] is well suited for a number of applications which require e.g. multi-carrier transmission and reception from 400 MHz to 6 GHz and high bandwidth. This versatility comes at the cost of moderate performance in terms of noise figure. The FMCOMMS2 [9] has a better noise figure but is tailored for 2.4 GHz only. A cellular modem needs to support a small specific set of frequencies and requires a particular good noise figure for those frequencies. RF boards and analog components tailored for cellular standards have better power efficiency and noise figure as multi-purpose RF boards.

In this paper, we reuse the concept of a versatile SDR development platform based on FMC carriers and modules from [6]. A cellular communication specific platform is presented. It consists of an FMC carrier card (FPGA, DSP, or CPU) for digital processing and a dedicated FMC module with cellular specific RF components with an open interface for analog processing. This setup allows verification and prototyping of cellular baseband algorithms jointly with the cellular specific RF and analog baseband parts before ASIC fabrication or deployment. The use of cellular specific RF components even allows single-chip mixed signal prototyping. As a proof of concept, a 2G GSM/Evolved EDGE modem is presented.

## II. PLATFORM ARCHITECTURE

The concept of an FMC based SDR development platform was thoroughly studied in [6]. The same concept is reused here and the cellular SDR platform consists of an FMC carrier card and a custom FMC module. Together they provide capacities for analog and digital processing as shown in Fig. 2. A general purpose wireless communication FMC module is used in [6]. Here, cellular communication specific FMC modules are presented.

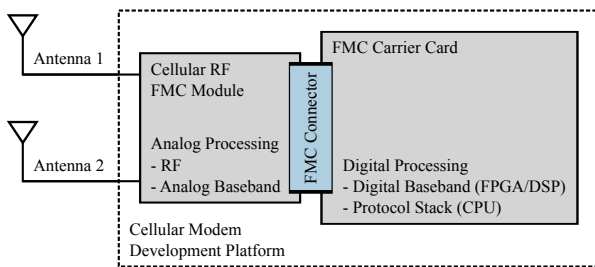


Fig. 2. FMC based cellular baseband development platform with two antennas. Analog processing is provided by an FMC module whereas digital processing is done on an FMC carrier card.

### A. FMC Interface

The standard for FMC [7] provides electrical and mechanical specifications as well as a common connector for FPGA extension modules. A large ecosystem of FMC carrier cards ranging from pure FPGAs, combined with DSPs, or combined with CPUs exists. On the other hand, a large amount of FMC modules is available. The FMC standard comes in two flavors, a Low Pin Count (LPC) and High Pin Count (HPC) connector which are mechanically and electrically compatible. The LPC signals are a subset of the HPC signals. The HPC version provides (LPC in parenthesis):

- 160 (68) single-ended or 80 (34) differential signals
- 10 (1) multi-gigabit transceiver pairs
- 2 (1) multi-gigabit transceiver clocks
- 4 (2) differential clocks
- 159 (61) ground pins
- 15 (10) power connections

This set of connectivity allows to perfectly connect an RF IC standard interface such as DigRF v1.12 for 2G or RBDP for 3G and 4G. The presence of multi-gigabit transceiver pairs

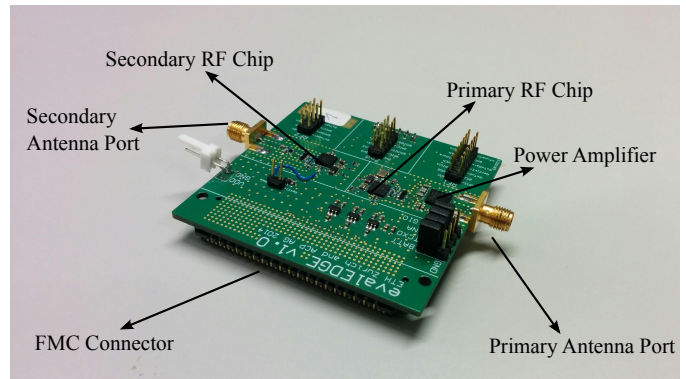


Fig. 3. Cellular single-mode 2G RF FMC module, called evalEDGE, capable of Evolved EDGE receive diversity and downlink dual-carrier operation.

provides the physical and electrical necessities to support the DigRF v4 standard often used in 4G RF ICs.

### B. Single-Mode 2G RF FMC Module: evalEDGE

A single-mode 2G RF FMC module is depicted in Fig. 3. As it offers full 2G Evolved EDGE support it is named evalEDGE FMC module. It hosts two RF ICs from ACP AG<sup>1</sup> with 2G support, a power amplifier, and power supply circuitry. The primary RF IC can receive and transmit. The secondary RF IC supports reception only. Two antennas can be attached via SMA connectors. Both RF ICs share a common clock. They can be configured independently from each other using their respective control plane DigRF v1.12 interfaces. This evalEDGE FMC module enables downlink dual-carrier and receive diversity reception. The DigRF control and data plane signals are routed directly to the FMC connector. The pin mapping is omitted here as the same protocols are used in the multi-mode 2G, 3G, and 4G RF FMC module in the next section, where a detailed mapping table is provided.

### C. Multi-Mode 2G, 3G, and 4G RF FMC Module: evalLTE

A multi-mode 2G, 3G, and 4G RF FMC module is depicted in Fig. 4. As it offers 4G LTE support it is named evalLTE FMC module. It hosts an RF IC from ACP AG with 2G, 3G, and 4G support, a power amplifier, and power supply circuitry. The RF IC comprises a dual-receiver for single-carrier LTE receive diversity or downlink MIMO operation. Two antennas can be attached via SMA connectors. Control and data plane signals run over DigRF v1.12 for 2G and RBDP for 3G and 4G. The DigRF and RBDP control and data plane signals are routed directly to the FMC connector. Table I shows the mapping of DigRF, RBDP, and control signals onto the FMC connector. The mapping supports HPC and LPC FMC. A global reference clock is provided by the RF FMC module and is mapped to one of the reference clock pins of the FMC connector. DigRF v1.12 and RBDP control plane share a 3-wire SPI interface and one reset line which are mapped onto 4 single-ended signals. The DigRF v1.12 data plane consists

<sup>1</sup>www.newacp.ch

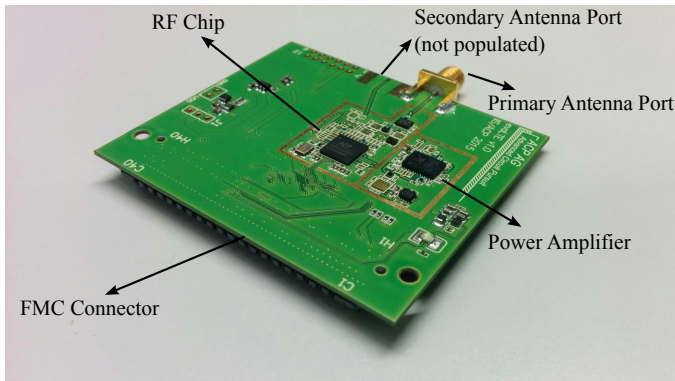


Fig. 4. Multi-Mode 2G, 3G, and 4G cellular RF FMC module, called evaLTE, capable of single-carrier LTE dual-reception. The secondary receive path is not populated.

of 2 single-ended bidirectional signals. The RBDP data plane consists of 2 clocks, 1 in each direction and 2 bidirectional 12 bit wide buses.

#### D. Digital Processing

A regular FMC carrier card is required. Depending on the application, processing requirements vary from simple control to expensive signal processing including protocol stack software. Available FMC carrier cards provide an FPGA and are often enhanced with DSP and CPU chips. Listing all possible candidates here is beyond the scope of this paper. The next section provides a sample implementation with one possible carrier card.

A driver written in HDL code on the FMC carrier card is required to facilitate RF FMC module access for the user. It provides register access for receive and transmit

- frequency,
- bandwidth,
- gain, and
- data buffers.

Additionally, registers to start and stop the receiver and transmitter, respectively, are required. Lastly, register access to control the FMC reference clock and an actual clock signal

Signal Description	FMC Signal Name	Comments
Global reference clock	CLK0_M2C	provided by FMC module
DigRF and RBDP control plane	LA00, LA02	3-wire SPI and reset signals
DigRF data plane	LA04	2 bidirectional signals
RBDP data plane	LA01, LA03, LA05-LA16	2 clocks, 2 bidirectional 12 bit buses
Power control, auxiliary signals	LA17-LA25	18 single ended signals

TABLE I

MULTI-MODE RF FMC MODULE PIN MAPPING WHICH SUPPORTS HPC AND LPC FMC CONNECTORS. FMC SIGNAL NAMES COMPRISE ONE DIFFERENTIAL OR 2 SINGLE-ENDED SIGNALS. POWER CONNECTIONS ARE PREDEFINED BY THE FMC STANDARD AND ARE NOT LISTED.

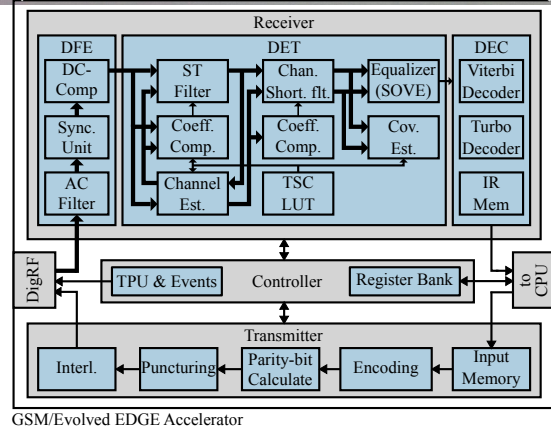
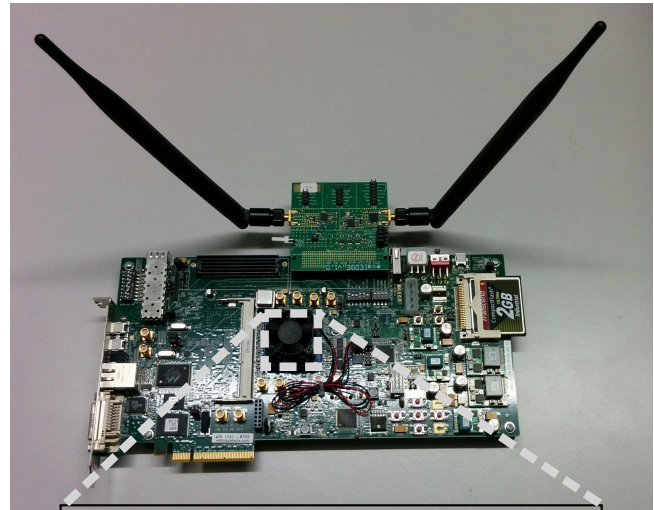


Fig. 5. FPGA based cellular modem consisting of FMC RF module (top board), FMC carrier card (bottom board). A GSM/Evolved EDGE accelerator is mapped onto the FPGA. A CPU (not shown) provides software support.

are required. Drivers for a number of FMC carrier cards are currently under development.

### III. GSM/EVOLVED EDGE IMPLEMENTATION

The single-mode 2G RF FMC module was used to test and prototype a GSM/Evolved EDGE modem. The ML605 board [10] from Xilinx is used as FMC carrier card. It hosts a Virtex 6 FPGA. The assembled platform is depicted in Fig. 5.

#### A. GSM/Evolved EDGE Implementation

Fig. 5 shows a block diagram of a GSM/Evolved EDGE accelerator implemented in HDL. The accelerator block has been mapped onto the FPGA of the FMC carrier card and thoroughly tested. Its algorithms are described in detail in [11]. It comprises three logical blocks which are *Controller*, *Transmitter*, and *Receiver*. It communicates with the single-mode 2G RF FMC module using DigRF data and control plane signals.

An FPGA version of a CPU based on [12] can be used for protocol stack software and configures the Controller block of the accelerator by register write accesses. Uplink and Downlink data are written/read to/from memories in the

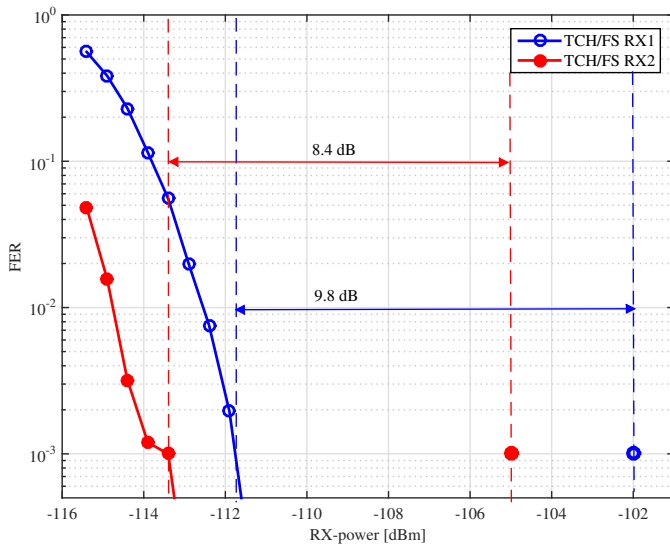


Fig. 6. Measured sensitivity performance [11] in terms of FER for GSMK modulated GSM voice in a static channel environment without (RX1) and with (RX2) receive diversity. Frequency hopping is not used.

Transmitter and Receiver blocks, respectively. The Controller holds a time processing unit which is responsible for exact timing of configuration data to the other processing blocks and the RF FMC module.

The transmitter is responsible for encoding, puncturing, and interleaving of user data according to the GSM/Evolved EDGE standard. The Receiver is divided into Digital Front End (DFE), DETector (DET), and DECoder (DEC). The DFE is responsible for synchronization and filtering. The DET provides equalization and interference cancellation. Lastly, the DEC provides decoding and incremental redundancy services.

### B. Measurements

The testbed has been used in [11] to perform 2G sensitivity performance measurements<sup>2</sup> as required by the 3GPP standard [14]. To this end, a channel emulator (PropSim C8) and a wireless protocol tester (Agilent 8960) have been connected to the SMA connectors of the RF FMC module. Fig. 6 shows the measured sensitivity for GSMK modulated GSM voice in a static channel environment. The sensitivity performance surpasses requirements by 9.8 dB and 8.4 dB for single stream and receive diversity reception, respectively. Evolved EDGE sensitivity has been evaluated for 16-QAM (DAS8) and 32-QAM (DAS10) modulation schemes, see Fig. 7. Sensitivity performance surpass requirements by 7.3 dB and 7.4 dB for 16-QAM and 32-QAM schemes, respectively.

## IV. CONCLUSIONS

An FPGA based prototyping platform for cellular modems is presented. It uses a regular FMC carrier card and a custom single-mode 2G or a multi-mode 2G, 3G, and 4G RF FMC

<sup>2</sup>For the measurements the ZedBoard [13] from Avnet has been used instead of a CPU based on [12].

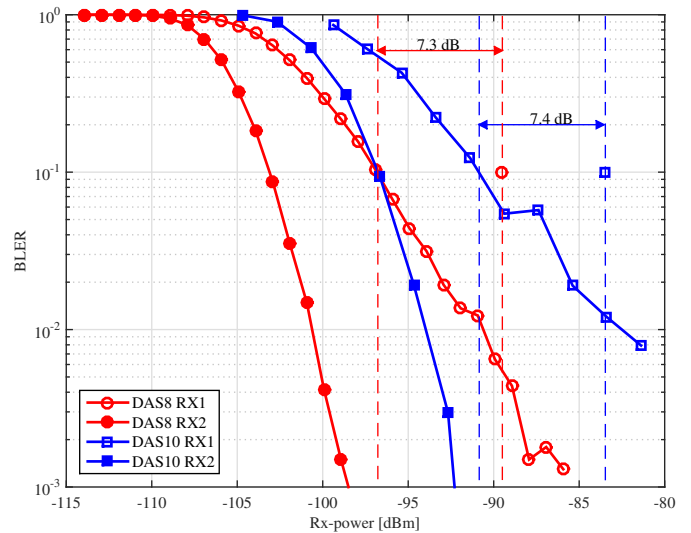


Fig. 7. Evolved EDGE measured sensitivity performance [11] in terms of BLER for the 16-QAM modulated DAS8 scheme and the 32-QAM modulated DAS10 scheme in a TU50 channel environment without (RX1) and with (RX2) receive diversity. Frequency hopping is not used.

module with an open interface. The platform with the single-mode RF FMC module is used to demonstrate a GSM/Evolved EDGE modem prototype implementation. It enables low-cost development and prototyping of cellular modems.

## V. OUTLOOK

While the single-mode FMC module has been used for receiver performance evaluation, the suitability of the multi-mode RF FMC module for a cellular modem testbed remains to be thoroughly analyzed. In particular extreme modem configurations such as high-throughput on the one side and low-power and low-area IoT options remain to be analyzed. Drivers for a variety of FMC carrier cards are under development or remain to be implemented.

## REFERENCES

- [1] *DigRF(SM) Specifications*, MIPI Alliance Std., Jun. 2015. [Online]. Available: <http://mipi.org/specifications/digrfsm-specifications>
- [2] *Radio Front End - Baseband Digital Parallel (RBDP) Interface*, JEDEC Std. JESD207, Mar. 2007. [Online]. Available: <https://www.jedec.org/standards-documents/docs/jesd-207>
- [3] M. Saily, G. Sebire, and E. Riddington, *GSM/EDGE: Evolution and Performance*. John Wiley & Sons, 2011.
- [4] *Cellular System Support for Ultra Low Complexity and Low Throughput Internet of Things*, 3GPP TR 45.820, Rev. 2.1.0, Aug. 2015. [Online]. Available: <http://www.3gpp.org/DynaReport/45820.htm>
- [5] *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities*, 3GPP TS 36.306, Rev. 12.5.0, Jul. 2015. [Online]. Available: <http://www.3gpp.org/DynaReport/36306.htm>
- [6] R. Machado and A. Wyglinski, "Software-defined radio: Bridging the analog-digital divide," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 409–423, March 2015.
- [7] *FPGA Mezzanine Card (FMC) Standard*, American National Standards Institute, Inc. ANSI/VITA 57.1, Rev. 2010, Feb. 2010. [Online]. Available: <http://www.vita.com/fmc>
- [8] "AD-FMCOMMS1-EBZ FPGA Mezzanine Card for Wireless Communications," Sep. 2015. [Online]. Available: <http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/EVAL-FMCOMMS.html>



- [9] “AD-FMCOMMS2-EBZ - AD9361 Software Defined Radio Board (2.4GHz Optimized),” Sep. 2015. [Online]. Available: <http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcomms2.html>
- [10] “Virtex-6 FPGA ML605 Evaluation Kit,” Jun. 2015. [Online]. Available: <http://www.xilinx.com/products/boards-and-kits/ek-v6-ml605-g.html>
- [11] H. Kröll, S. Zwicky, B. Weber, C. Roth, D. Tschopp, C. Benkeser, A. Burg, and Q. Huang, “An Evolved GSM/EDGE Baseband ASIC Supporting Rx Diversity,” *Solid-State Circuits, IEEE Journal of*, vol. PP, no. 99, pp. 1–12, 2015. [Online]. Available: <http://dx.doi.org/10.1109/JSSC.2015.2417802>
- [12] D. Rossi, A. Pullini, M. Gautschi, I. Loi, F. Gurkaynak, P. Flatresse, and L. Benini, “A -1.8V to 0.9V Body Bias, 60 GOPS/W 4-Core Cluster in Low-Power 28nm UTBB FD-SOI Technology,” in *The IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (IEEE S3S)*, Oct. 2015.
- [13] “ZedBoard,” 2015. [Online]. Available: <http://zedboard.org/product/zedboard>
- [14] *Radio transmission and reception*, 3GPP TS 45.005, Rev. 12.1.0, Dec. 2013. [Online]. Available: <http://www.3gpp.org/DynaReport/45005.htm>

## **TACTICAL RADIO COALITION INTEROPERABILITY SOLUTION FACILITATED BY ANW2 AND NINE**

Igor A.(Tony) Spivak (Harris Corporation, Rochester, NY; e-mail: [ispivak@harris.com](mailto:ispivak@harris.com))

### **ABSTRACT**

Multinational coalitions have become prevalent in military operations over the past several decades. In many instances, communications issues have hindered effective dissemination of information at various levels of military command structures to all required parties. These communications issues have been attributed to a variety of factors including: proprietary, non-interoperable communications devices, lack of common waveform interoperability standards, lack of common COMSEC/TRANSEC key management, generation and distribution infrastructure, etc. To mitigate these issues, temporary stop-gap solutions have been developed; however, a comprehensive coalition interoperability strategy must still be defined.

Harris Corporation, RF Communications division has developed the Adaptive Networking Wideband Waveform (ANW2). The ANW2 is a wideband networking waveform that supports simultaneous secure voice, data and video services for up to 30 networked nodes. The ANW2 has been implemented in NSA certified, Type 1 high grade radio platforms, as well as commercial, non-Type 1 tactical radio platforms. This paper presents a potential candidate solution to solve the coalition interoperability problem in the tactical radio domain. The proposed solution is based on ANW2, integrated with Network and Information Infrastructure (NII) Internet Protocol Network Encryption (NINE) specification to achieve a standards based, interoperable high grade radio mode that can be implemented as an SCA waveform and ported to a variety of tactical radio platforms.

### **1. INTRODUCTION**

Radio communications coalition interoperability remains an important objective for military leaders. Many of the recent global conflicts involved formation of multi-national coalitions in order to achieve the desired military objectives. Such coalitions have unique communications challenges that must be overcome in order to achieve these objectives. One recent example of the coalition communications challenge and how it was overcome is the Afghanistan Mission Network (AMN). The AMN was created over the period of

several years in order to support the 45 member nation International Security Assistance Force (ISAF) coalition in Afghanistan. Prior to the creation of the AMN, coalition members relied on individual countries' communications systems. These systems were primarily stove piped, custom communications systems that had their own security policies and constraints. It was not easy to securely share information between coalition partners in such an environment. The AMN was designed to allow coalition members to securely share services such as e-mail, chat, VOIP telephone connectivity, web browsing and secure video teleconferencing (SVTC) over a common network. Though the AMN was ultimately successful in achieving the information sharing objectives of the ISAF, it was based largely on existing US DOD and NATO communications infrastructures that have been updated in the areas of policies, procedures and governance. Going forward, the development of technologies that have been designed with secure coalition interoperability in mind will greatly enhance the effectiveness of future multinational coalition military operations.

Many initiatives have been kicked off by the United States Department of Defense (DoD) and NATO to create standard protocols and interface specifications to facilitate radio platforms interoperability. These initiatives include Joint Tactical Radio System (JTRS) program, which leverages SCA technology and develops/maintains a set of interoperable waveforms. Within the NATO community NINE and SCIP specification development working groups have defined interoperability specifications for secure IP based networked and secure voice communications, respectively. Another important activity has been initiated by the NATO Communications and Information (NCI) Agency to develop a common Key Management (KM) Interface Specification (ISPEC), which is designed to define the common interface between the NATO Key Management System and NINE and SCIP compatible End Cryptographic Units (ECUs). All of these activities are intended to contribute to the future improvement of coalition communications, however, in the tactical radio domain, the most efficient method for achieving coalition interoperability is by defining an SCA radio waveform that can be developed, tested and ultimately ported to variety of tactical radio platforms. This paper shows how the Harris

SCA compliant ANW2 integrated with NINE based IP network security protocol can be considered as a potential solution to improve coalition tactical radio communications.

## 2. ANW2 TECHNICAL OVERVIEW

Harris’ Adaptive Networking Wideband Waveform (ANW2) is a wireless mobile ad-hoc networking (MANET) waveform that provides high-throughput internet protocol (IP) data and simultaneous combat net radio (CNR) voice. ANW2 provides wireless IP connectivity directly to the warfighter, providing common real-time access to critical information to mobile, dismount, and command units. ANW2 is a fully adaptive waveform that automatically optimizes route selection and data transmission rate to provide the best possible IP connectivity without user intervention.

ANW2 supports a wide variety of applications including:

- Situational Awareness
- Voice over Internet Protocol (VoIP) Telephony
- Video / Video Teleconference (VTC)
- File Transfer
- IP Chatting
- Targeting Applications
- Remote Database Access
- Integration with third party IP capable radios

Individual wireless domains may include up to 30 full member nodes and up to 255 guest member nodes. Since the ANW2 supports a standard IP interface, the radio that implements ANW2 can be used in conjunction with standard networking infrastructures and equipment.

ANW2 operates in the 225 MHz to 2GHz frequency range at bandwidths of either 1.2 MHz or 5 MHz. This allows ANW2 to operate in the standard UHF frequency and bandwidth ranges used by the military. The use of the UHF spectrum provides for better performance in urban environments than existing legacy radio systems. The UHF spectrum also has a smaller terrestrial jamming footprint. Additionally, UHF antennas are of a smaller size than VHF or HF antennas.

The radio’s physical layer implements a library of Signals-in-Space (SIS) to accommodate different channel bandwidths (1.2 MHz, 2.5 MHz, and 5 MHz) and modem burst data rates (85 Kbps to 10 Mbps max). ANW2 makes use of nine different wireless modems to support the actual

RF waveform. These modems provide for wireless connectivity between ANW2 radios. The lower the number of the modem, the greater the distance it will cover, but also the lower the data throughput that it supports (i.e. WF1 supports the longest range, and WF9 supports the highest rate). More recent improvements to the waveform include Gaussian Minimum Shift Keying (GMSK) modulation technique, which significantly increased the capacity of the 1.2 MHz and 5 MHz channels that support high-rate link conditions.

The robust connectivity and extended range offered by GMSK comes without much compromise to data rate, and even offers increased capacity in some conditions. In addition, the high-rate modems introduced in the latest ANW2 release yield a significant increase in performance under supporting channel conditions.

All SIS are implemented using an efficient single carrier high baud rate modulation. The receiver utilizes advanced RAKE and Data Directed Equalization (DDE) channel equalization methods plus FEC coding (including code combining techniques). Channel quality measurements are used to continuously adapt the modem burst rate at each end of an active link in order to maintain the expected received packet error rate below a prescribed threshold.

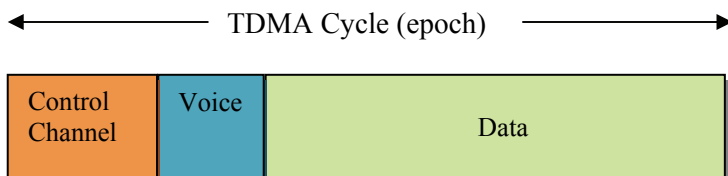
ANW2 peak and average output power depends on the platform that implements the waveform. In the Harris RF-7800M-MP platform, the ANW2 has a peak output power of 5W in a dismount manpack scenario and 50W when used with a Vehicular Adapter Amplifier (VAA). Each of the modems used by ANW2 has a different RF propagation range. The range that can be expected depends greatly on the terrain. In a clear line-of-sight environment, ANW2 will provide connectivity at distances up to 85 km. In an urban environment that range is approximately 2-10 km. Table 1 provides estimates of the typical range in km as a function of peak output power and terrain.

	Urban	Rural	Hilltop
<b>5W Manpack</b>	2.5	4.0	10
<b>50W Vehicle</b>	6.0	12	25
<b>Air to Ground</b>	85	85	85
<b>Ship-to-Shore</b>	20	35	60

**Table 1 - Typical Ranges (km)**

ANW2 is configurable to have a maximum of 1 to 30 member radios in the sub network. In addition, ANW2 provides flexibility for support of up to 255 nodes in a single network with guest users, allowing for CNR voice capabilities and receipt of multicast data. ANW2 creates a

mobile ad-hoc network by using a distributed Time Division Multiple Access (TDMA) channel access wireless MAC sub network layer. The guaranteed channel access of a TDMA MAC is the basis for providing simultaneous voice and data service as well as allowing multiple radios to gain simultaneous access to a single frequency radio channel. The time domain is divided up into units called “epochs”. A high-level diagram of an epoch is shown in Figure 1.



**Figure 1 – ANW2 Normal Mode Epoch**

Figure 1 illustrates an epoch within TDMA cycle. Each epoch is divided into three areas: control channel, voice and data.

- **Control Channel** – The control channel provides each network radio with guaranteed access to a control plane channel on a regular basis. The deterministic, non-collision based channel access method used to support control plane communications adds to the robustness of the mobile network as well as minimizes implementation complexity.
- **Digital Voice (DV)** – This portion of the epoch supports 2400bps MELP based tactical voice service. This voice service is half-duplex and heard by all radios within user configurable 0-9 hops of the source of the digital voice. The voice hopping occurs all within a single epoch and incurs no detectable latency.
- **Data** –The data portion of the epoch is where the proactive routing protocol packets and IP data packets are transmitted to the network as a whole. Each radio gets a dedicated data interval in which to transmit data. Each of the data intervals is further sub-divided to allow transmission to up to 4 unique destinations in a single epoch.

The use of a TDMA MAC protocol allows channel bandwidth to be allocated on a dedicated basis to voice and data services. This means that a radio can simultaneously transmit (and/or receive) data and DV traffic. The ANW2 currently supports all-informed digital voice (i.e., from a source radio to all radios within the specified hop radius) configurable from 0 to 9 hops. Advanced FEC coding and code combining techniques are used to achieve a high

probability of synchronization for encrypted DV transmissions out to the maximum range of the radio. In order to maintain error-free data connectivity, ANW2 will adjust the data rate at which IP data is transmitted.

The effective per radio user data rate (i.e. net user data rate), with all radios transmitting simultaneously depends upon the network size and the channel conditions, but not on how many radios are transmitting.

ANW2 maintains its view of the network through the use of a proactive Layer 2 ad-hoc routing mechanism. Each radio sends out periodic messages to announce its presence in the network and pass various routing statistics. Using the cumulative messages received from all of its neighbors, a radio can construct an operational picture of the network. A distributed and co-operative algorithm is used to support reliable network formation under adverse channel conditions, i.e. impulsive noise, fading, asymmetrical links, and movement of the radios. Use of this ad-hoc routing mechanism allows ANW2 to be a self-forming, self-healing network that does not require a centralized control radio in order to manage the network. Network formation times are generally less than 20 seconds following a radio power-up. When a radio joins an existing ANW2 network, the join time is less than 5 seconds. Radios can leave and rejoin the network at any time without impacting guaranteed single frame network performance.

An ANW2 network does not have a central control station. The information required for network control is collected and maintained by the Global Server Node (GSN). The GSN is a radio that is elected by the network rather than being pre-configured before the network forms. Use of the GSN reduces the time it takes for radios to join or re-join the network if for some reason they have become disconnected. Typical networking ad-hoc scenarios that are enhanced by this mechanism in the mobile environment are network bifurcations and network merge. If the radio that is the GSN leaves the network for any reason a new GSN is elected by the network without impacting the performance of the network.

### 3. ANW2 DEPLOYMENT SCENARIOS

ANW2 can be easily applied to a wide variety of mission scenarios. ANW2's advanced planning tools and ad-hoc routing mechanism make planning and deployment substantially easier than with legacy and non-networking waveforms.

ANW2 automatically routes IP data and adapts to changes in physical topologies, late-net-entry by new radios, and merging and splitting of the ANW2 network. ANW2 requires no special configuration and automatically supports all of the described example scenarios.

#### Mesh Network

In a mesh network all radios are distributed in an operational area with all radios having comparable connectivity to each other. Although the physical topology of this network may take many forms, the radios form one logical network. Figure 2 and Figure 3 show examples of logical mesh networks with varying ad-hoc physical topologies.

As the number of radios in an ANW2 network increase the network routing possibilities also increase. All combinations of complex radio topologies are supported by the self-healing ad-hoc routing mechanisms in ANW2. These routing mechanisms provide the network with the ability to seamlessly move data from any radio in the network to any other radio. Since this robust routing is active for the entire mission, it automatically supports rapid changes in the topology, providing communications on the move.

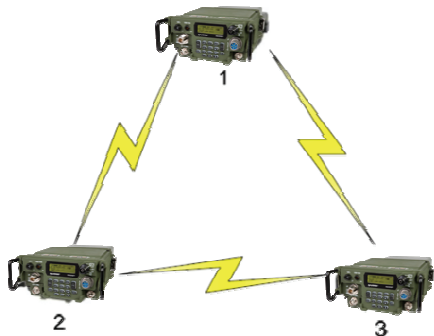


Figure 2 - Physical and Logical Mesh Topology



Figure 3 - Physical Line and Logical Mesh Topology

#### Advantaged Radios

ANW2 allows for several general scenarios which utilize advantaged radios:

- *Relay* – One radio in an advantaged position acts as a relay between nodes that do not have sufficient connectivity. This type of deployment is often achieved by placing a radio at the top of a hill, on a tower, or on a building while the remaining radios are distributed below. This deployment is illustrated in Figure 4.

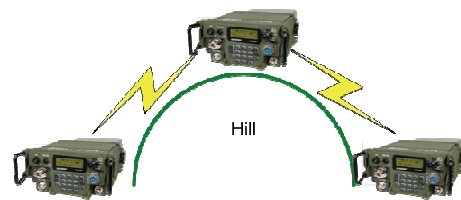


Figure 4 – Data Relay

- *Area Coverage* – One or more radios in advantaged position providing an area or “cell” of data coverage to nodes that are in a disadvantaged environment. This deployment provides a high level of data service to a small number of disadvantaged radios. Figure 5 illustrates the area coverage scenario.



Figure 5 – Area Coverage

- *UAV Feed (Download)* – Figure 6 shows an intelligence source (UAV) providing a data feed to a number of ground radios.

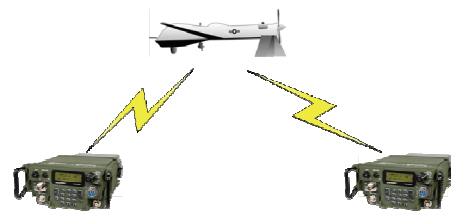


Figure 6 – UAV Feed

### Multi-connected ANW2 Networks

In a multi-connected ANW2 network scenario, a number of radios that are part of a single wireless network can also have a connection to a larger network to extend the reach of voice and data services across echelons in the battlefield architecture. Figure 7 below illustrates direct interconnection of ANW2 subnets to form a larger radio wide area network.

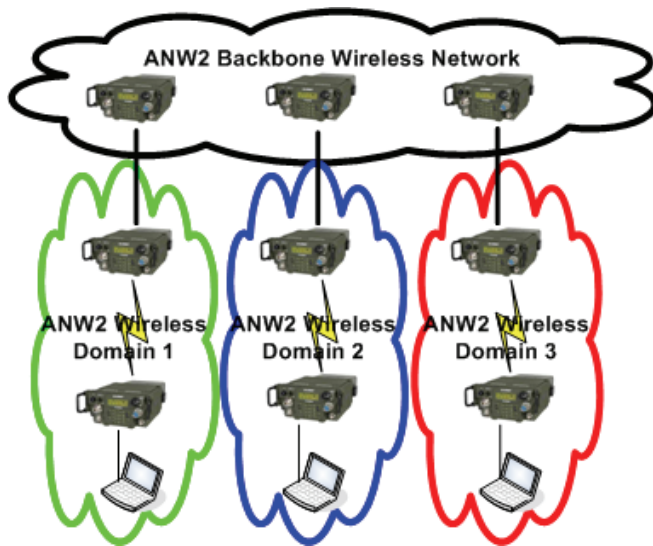


Figure 7 - ANW2 Sub Networks

In this scenario, one ANW2 subnet provides the backbone capability to link several other ANW2 subnets into a seamless wide area network. The use of a backbone is advantageous in that the distance between any pair of subnets in the command hierarchy is at most one intermediate subnet. It is also worth noting that subdividing a larger network into several smaller sub networks provides significant advantages in terms of the performance of the individual radio networks, specifically in terms of net formation/maintenance, network timing, throughput, provision of QoS, simplifying channel access and overall network scalability.

### 4. ANW2 AND NETWORKING

The ANW2 is normally implemented in a standard tactical radio security architecture that separates red (plain text) and black (cipher text) functions. A cryptographic subsystem is responsible for encryption/decryption, as well as other information assurance related functions and is typically placed between the red side and the black side to provide the required isolation between unprotected and

protected domains. The ANW2 IP networking functionality is accomplished via a full IP stack on the red side of the radio that provides the tunneling and encryption required to work with a black side IP networking stack. The black side stack is used to interface to the ANW2 black side waveform components, which format the information into the data portion of the ANW2 epoch, as shown on Figure 1. A full network stack (i.e. transport and IP layers) is implemented on both the red and black sides of the radio. The system supplies a black (i.e. target radio) IP address corresponding to the target host's attached radio or a gateway radio. The black side ad-hoc routing functionality discovers a route and forwards the encrypted packet to this black address. An external router can be used to advertise an ANW2 network to other ANW2 or non ANW2 networks. Preprogrammed security associations can be provided to identify gateway radios to access these external networks. The ANW2 supports IP V4 standard unicast and multicast best effort delivery of IP packets. Multicast and broadcast IP traffic is sent out at the highest common data rate throughout the wireless network. This is done to provide the greatest likelihood that the multicast traffic reaches all radios, which is important in systems that use Situational Awareness and messaging applications.

### 5. ANW2 AND NINE NETWORK SECURITY

One key advantage of ANW2 IP network based architecture is the ability to integrate the waveform with the standards based network security protocols, such as IPSec or NINE. NINE Interoperability Specification (NINE IS) is a set of requirements to provide traffic protection, networking, and management functions for networking devices supporting IPv4 and IPv6 networking protocols. Development of NINE and SCIP are efforts within the NATO alliance aimed at creating interoperable equipment for secure communications over IP and secure voice connections, respectively. NINE IS is a NATO standard intended to ensure that IP networking devices manufactured by different vendors will securely interoperate with each other in support of NATO missions. In order for such devices to be deemed NINE IS compliant they must successfully pass NINE Interoperability Conformance Evaluation (NICE) testing. The NINE IS compatible devices provide confidentiality service at the IP layer, ensuring no contamination of data amongst enclaves operating at the same classification level. The IP layer encryption also allows the packets to be routed on the black side of the radio for true end-to-end secure communications. The NINE IS compatible devices support both manual IP configuration and auto-negotiated configuration. The protocol is capable of automatically negotiating data rate and mode with all interfacing network

devices when set to the auto-negotiating option. Another important advantage to leveraging standard network security specification, such as NINE, is that the key management support infrastructure is well defined, as part of the specification. NINE IS specifies both Type A and Type B (a.k.a. Suite A and Suite B) algorithms and associated key management requirements. For coalition interoperability scenarios, it is likely that Type B based communication security will be specified. For Type B key management, NINE specifies a number of key generation methods including:

**Authenticated Pre-Placed Keys (APPK)**

This is a symmetric key distribution method. APPK traffic keys are generated and authenticated by the Key Management System. Community of Interest (CoI) unique Trust Anchors are used to manage distribution of the APPK to appropriate ECUs.

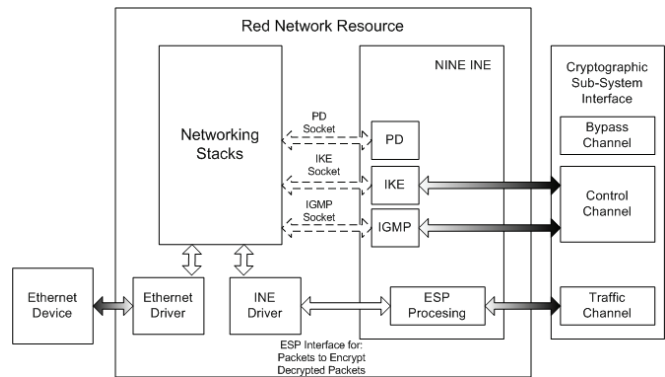
**Device Generated Shared Key (DGSK)**

This is a symmetric key distribution method. One of the ECUs is designated a net controller and is responsible for key generation. The generated traffic keys are distributed to the remaining net members either manually or via specific peer-to-peer key transfer methods.

**Elliptic Curve Diffie-Hellman (ECDH)/IKE V2 Key Exchange**

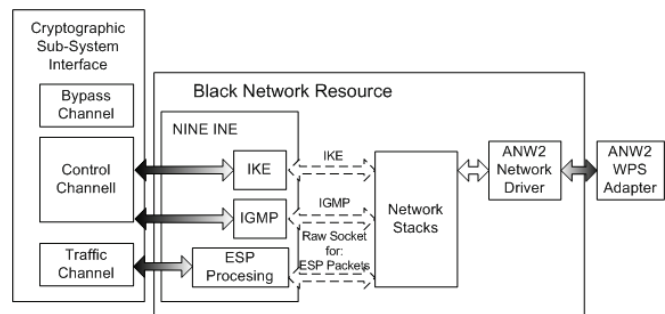
This is an asymmetric key distribution method. ECDH is a key agreement protocol that allows parties that have the proper, pre-loaded Public Key Infrastructure (PKI) credentials to establish a shared traffic key.

Within the radio platform, the NINE components can be easily integrated with the required red side and black side platform networking components. Figure 8 shows the typical red side architecture of the NINE based network security service. The Red Network Resource component integrates the required NINE control and traffic services with the networking stacks and the interfaces to the cryptographic sub-system.



**Figure 8 - Red Side Network Components**

The black side network components can be integrated with the required black side NINE components as shown on Figure 9 below.



**Figure 9 - Black Side Network Components**

Similar to the red side, the black side NINE INE components provide control and traffic interfaces to/from the Cryptographic Sub-System and black side network stacks. The platform black networking components interface with the ANW2 Wireless Protocol Stack (WPS) component, which transforms the encrypted network packets to the epoch format ANW2 will use to send the data over the air.

## 6. CONCLUSION

### “Take Away Messages”

- Most recent and ongoing military conflicts involve the support of multinational coalitions. These coalitions require standards based, secure, interoperable radio communications solutions in order to meet mission requirements.
- The most effective way to enable tactical radio coalition interoperability is to define an SCA compliant radio waveform that can be developed, validated and ultimately ported to a variety of tactical radio platforms.
- ANW2 is a high performance, robust and flexible tactical networking waveform that can accommodate a variety of the required tactical radio operational use cases.
- ANW2 can be easily integrated with NINE to provide secure and robust solution that can support coalition operation communications requirements.

## 7. REFERENCES

- [1] Serena, Chad C., Isaac R. Porche, Joel B. Predd, Jan Osburg and Brad Lossing. Lessons Learned from the Afghan Mission Network: Developing a Coalition Contingency Network. Santa Monica, CA: RAND Corporation, 2014..
- [2] Adaptive Networking Wideband Waveform (ANW2) for the RF-7800M Wideband Multiband Radio, Harris, RF Communications, Rev. B, Oct. 2011.
- [3] Network and Information Infrastructure (NII) Internet Protocol Network Encryption (NINE) Concept of Operations, Version 0.5, 16 April 2013
- [4] NIST Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March, 2007
- [5] RFC4306, Internet Key Exchange Version 2 (IKE V2) Protocol, December, 2005



## ON THE DESIGN OF HIERARCHICALLY MODULATED BICM-ID RECEIVERS WITH LOW INTER LAYER INTERFERENCES

M. Tschauner (Fraunhofer FKIE, Wachtberg, GER, matthias.tschauner@fkie.fraunhofer.de)  
 M.F.T. Oshim (Fraunhofer FKIE, Wachtberg, GER, farhan.oshim@rwth-aachen.de)  
 M. Adrat (Fraunhofer FKIE, Wachtberg, GER, marc.adrat@fkie.fraunhofer.de)  
 M. Antweiler (Fraunhofer FKIE, Wachtberg, GER, markus.antweiler@fkie.fraunhofer.de)  
 B. Eschbach (IND RWTH Aachen University, GER, eschbach@ind.rwth-aachen.de)  
 P. Vary (IND RWTH Aachen University, GER, vary@ind.rwth-aachen.de)

### ABSTRACT

In this paper, we present a novel methodology to optimize hierarchically modulated Bit Interleaved Coded Modulation with Iterative Decoding (HM-BICM-ID). This methodology allows designing a receiver which supports several configurations. Each configuration is able to decode the same transmitted signal over the air with different fidelity. This concept permits using radios with varying processing capabilities, e.g. handhelds, vehicular based etc. However, earlier simulation results have shown that HM-BICM-ID loses, if compared to non-hierarchical schemes, in BER performance due to *Inter-Layer Interferences* (ILI). Our proposed iterative tunable algorithm optimizes hierarchical modulation schemes considering several criteria by moving critical constellation points towards the optimal direction. A novel modulation scheme has been found with minimized ILI and simulation results show an improved asymptotic BER performance in a wide range of channel conditions and for a two layered HM-BICM-ID.

### 1. INTRODUCTION

A transceiver system based on *Bit Interleaved Coded Modulation* with *Iterative Decoding* (BICM-ID) has been introduced the first time by X. Li and J. A. Ritcey in [1][2]. In their work they have shown that a BICM-ID system significantly outperforms *Trellis-Coded Modulation* (TCM) which has been introduced much earlier by G. Ungerboeck in [3]. A BICM system is given by a serial concatenation of channel coder, bit-interleaver, and modulator. At the receiver side, the arrangement of a demodulator, de-interleaver and channel decoder is used to reconstruct the originally transmitted signal. Compared to G. Ungerboeck's TCM scheme, BICM improves in terms of the *Bit Error Rate* (BER) especially under fading channels. The asymptotic performance of BICM depends mainly on the modulation scheme. For example, a Gray symbol labeling is often used in BICM because neighbored symbols differ at only one bit position which yields a small number of bit errors. The bit-interleaver as another key element in BICM helps to remove long burst errors and correlations between

neighbored bits in such a way that the decoder is able to improve decoding. The channel decoder further uses the redundant information introduced at the transmitter side for protection and reconstructs the net bits with high reliability. With the introduction of the well-known turbo principle of digital signal processing [4] it was possible to improve BICM by extending it to BICM-ID. In BICM-ID an additional feedback line is used to exchange information between decoder and demodulator. In [1], a hard-decision feedback version of BICM-ID has been introduced. Additional improvements can be made when extending BICM-ID with soft-decision [5]. Then, reliability information in terms of so-called extrinsic information or log-likelihood ratios (LLR) is exchanged between channel decoder and demodulator. In a BICM-ID system, the BER curves get better with the number of iterations and converge until the *bit error floor* is reached. The *bit error floor* is the part of the BER curve that can be interpreted as the lowest achievable bound for the BICM-ID system and can be simulated according to an *Error Free Feedback* (EFF) scheme. To measure the EFF, *a priori* knowledge is relayed as error free and reliable LLR values to the demodulator. Another typical observation in performance measurements for BICM-ID is the *waterfall region* where the BER is reduced considerably in a small range of  $E_s/N_0$ . For a good performance of BICM-ID the channel code and symbol labeling must be optimized together. But significant improvements in BICM-ID, e.g., reaching a very low *bit error floor*, can only be realized when a symbol labeling with a high *Harmonic Mean*  $d_h^2$  [2] is used. In other words, the *Euclidean Distance* of labels that differ only in one bit position shall be maximized. Such an optimization is contrary to the optimization of the modulation scheme in BICM. However, the improved performance for good channel conditions comes with the cost of complexity at the receiver side.

In the literature, *Hierarchical Modulation* [6][7] or *Layered Modulation* has been introduced to give a transceiver system

the possibility to receive data under different circumstances. *Hierarchical Modulation* allows the operator to send multiple data streams modulated to a single symbol stream. The different data streams are called *base layer* (BL) or *enhancement layers* (EL). Depending on the channel condition and computational capability of the receiver the modulated symbols can be demodulated in such a way that all or a subset of data streams are recovered. Therefore, the BL provides the most important information being transmitted in a robust way to all radio devices. In addition, each EL carries optional information used to provide further valuable information. This information can be used at the receiver to improve the communication in several ways, e.g., a higher data rate, reliability, or range. One major challenge of hierarchical systems is the design of the different layers which is equivalent to the minimization of *Inter-Layer Interference* (ILI). However, hierarchical modulation is very attractive because of its possibility to switch between different receiver configurations. Therefore, it is used in broadcast systems like *Digital Video Broadcasting over Satellite* (DVB-S2) [8] or over *Terrestrial Antennas* (DVB-T2) [9].

In the literature, digital communication systems exploiting both the benefits of BICM-ID and the advantages of *Hierarchical Modulation* have not been extensively discussed so far. In [10][11], first solutions to a hierarchically modulated BICM-ID (HM-BICM-ID) have been introduced. In our previous work [12], we proposed a HM-BICM-ID system based on a hierarchical 8x8-PSK where each constellation point is composed by a group of eight labels with *Hamming Distance*  $d_{ham} = 2$ . This was a systematic way to reduce the influence of ILI. Now, we propose a novel algorithm to give the system designer the possibility to develop novel hierarchical modulation schemes with reduced ILI and improved BER performances.

This paper is structured as follows: Chapter 2 introduces the two layered HM-BICM-ID system. The effect of ILI is described in details and reasons for typical performance degradations in hierarchical systems are given. In Chapter 4 we propose a novel algorithm which moves the constellation points of a certain modulation scheme to a proper direction considering two main criteria, the *Harmonic Mean* and the *Bit Error Probability*. Then, the novel algorithm is used in Chapter 4 to develop a novel hierarchical modulation scheme. This scheme is used in a HM-BICM-ID system to perform a BER simulation and the results are presented and discussed. The paper concludes with Chapter 5.

## 2. HIERARCHICALLY MODULATED BICM-ID

The block diagram of a two configuration transmitter and receiver based on HM-BICM-ID as proposed in [12] is depicted in Fig. 1. It shows two different transmitters, an additive white Gaussian noise (AWGN) channel, and two different receivers. Assuming that two radios of different

capabilities can be chosen at the transmitter and receiver side independently, four different configurations for a communication link can be considered for analysis.

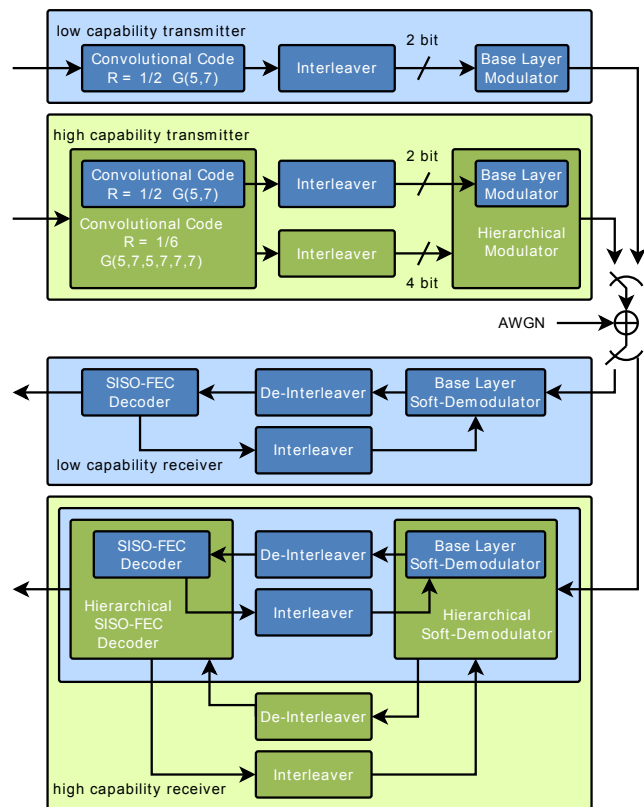


Figure 1: Block Diagram of a BICM-ID (BL) and HM-BICM-ID (EL) considering different configurations for transmitter and receiver.

The BL signal processing boxes used by a low capability transmitter and receiver are shown in Fig. 1 within the blue boxes. The situation is equivalent to signal processing in BICM-ID. Due to the higher computational power available for a high capability radio a HM-BICM-ID transmitter and/or receiver can be used as EL. They are shown in Fig. 1 by green boxes. Please note that the EL implies the blue signal processing blocks from the BL. However, only three of the configurations in Fig. 1 are relevant. Receiving signals, sent by a low capability radio, with the high capability device does not yield any improvements and is therefore irrelevant. Just as well, using a low capability radio at transmitter and receiver side corresponds to a state of the art BICM-ID scheme and thus is not considered in our work. Anyways, to support reliable communication while being flexible in the choice between specific configurations, a fully hierarchical system is used to guarantee interoperability over the air for the other three cases. Therefore, HM-BICM-ID is a suitable solution to manage such communication scenarios. In the following, we

consider a high capability transmitter based on HM-BICM-ID. A fully hierarchical transceiver as depicted in Fig. 1 uses the concept of *incremental redundancy* [13][14] in *forward error correction*, e.g., a setup of different convolutional codes as mentioned in [12]. For the BL the generator polynomial  $G(5,7)_8$  with a *free distance* of  $d_f = 5$  and a coding rate of  $R_c = 1/2$  is used. The EL uses the  $R_c = 1/6$  convolutional code  $G(5,7,5,7,7,7)_8$  with  $d_f = 16$ . Both convolutional codes with *constraint length*  $K = 3$  are proposed by [15][16] due to the maximum *free distance* and *optimal distance spectra*. Because the first code is a subset of the latter one, a separation of the encoded data stream into BL and EL information is possible. Then, each data stream is independently bit-interleaved and merged together to groups of bits. The hierarchical modulator is designed in such a way that the BL modulator is a subset of the EL modulator. A typical hierarchical modulation scheme with 2 bits for the BL and 4 extra bits for the EL is shown in Fig. 2 (a) and (b).

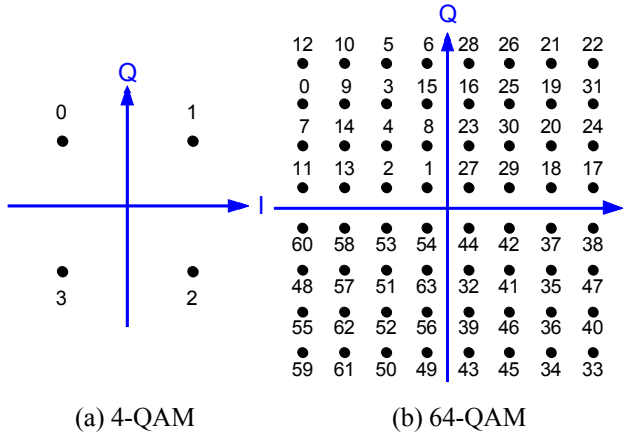


Figure 2: Labelling of signal constellation points for the BL (a) and the EL (b).

Thus, groups of 6 bits are mapped to one single symbol by the hierarchical modulator. This keeps the overall rate of the system constant. Such a HM-BICM-ID system guarantees the use of high capability radios at receiver side while also keeping signal interoperability over the air with a low capability receiver. In the case where a high capability receiver is used, HM-BICM-ID helps to demodulate and decode the additional information carried by the EL. In our work HM-BICM-ID is designed to provide additional information to improve robustness and range. This is novel compared to broadcast systems where the throughput is increased under good channel conditions [8][9].

## 2.1 System Design

A big challenge in HM-BICM-ID is the design of the modulation scheme and channel coder in the EL. The BL is often state of the art and somehow fixed. Therefore,

especially in the case where a high capability transmitter and low capability receiver are used, interoperability must be kept in mind during the design process. This can be solved when both the BL modulator and BL decoder are subsets of the corresponding EL modulator and EL decoder. With this side constraint so-called ILI is introduced which reduces the overall performance of the communication link. To prevent such negative effects, ILI has to be minimized. ILI can be characterized by two main effects. The first effect is caused by a design constraint of the labels in the EL modulator. Due to the hierarchical nature, the choice for the labels and positions of the constellation points are restricted to a given range. This gives the operator with the low capability radio the possibility to receive the BL information even when the high capability transmitter has been used. Unfortunately, restrictions in the design of the modulation scheme lead to a reduced *Harmonic Mean*  $d_h^2$ . A high  $d_h^2$  causes BICM-ID systems to improve the overall BER performance under good channel conditions. Therefore, we expect a degradation of the BER performance for HM-BICM-ID systems. The second effect of ILI affects the BL itself. The use of an EL modulator at the transmitter side and a BL demodulator at the receiver side results in a mismatch between the symbols and introduces additional noise at the receiver. For example, to transmit the EL modulator symbol  $49_{10} / [\text{MSB}]110001_2$  as depicted in Fig. 2 (b) causes the BL demodulator to decide for the correct symbol  $3_{10} / [\text{MSB}]11_2$  when AWGN is absent. However, the probability for a wrong decision increases because some EL symbols are close to the BL decision bounds. This causes the BL demodulator to decide wrong when AWGN is present. This is illustrated in Fig. 3 where EL symbol  $49_{10}$  is noisy received (green arrow) and demodulated by the BL to  $2_{10}$ .

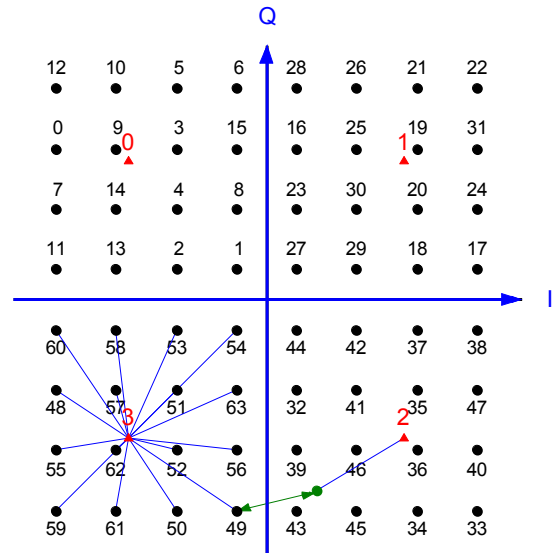


Figure 3: Constellation points mismatch between EL and BL modulation scheme causes additional interferences.

### 3. NOVEL ALGORITHM FOR JOINT MULTI-LAYER OPTIMIZATION OF HIERARCHICAL MODULATION SCHEMES

Due to the fact that ILI causes performance degradations in hierarchical receivers, our main idea is to minimize ILI by minimizing the underlying effects that cause ILI. Therefore, we propose a novel algorithm that starts from a hierarchical modulation scheme whose labels have already been optimized for BICM-ID. The algorithm moves the constellation points towards the direction of interest in a serial iterative manner. We propose to optimize two main criteria for each layer. On the one hand, we want to increase the *Harmonic Mean*  $d_{h,layer}^2$  of each layer. On the other hand, the algorithm decreases the *Bit Error Probability*  $P_{b,layer}$  in each layer. To prevent an uncontrolled growth of the energy per symbol  $E_S$  during the movement of the constellation points a normalization is done after each algorithm step. Thus, the energy per symbol is kept constant during the process of the algorithm and a convergence of the algorithm is guaranteed. A description for a comprehensive optimization of a hierarchical modulation scheme is given by Algorithm 1.

---

#### Algorithm 1: Comprehensive optimization

---

```

1: Initialize constellation points and labels
2: normalize
3: measure  $d_{h,layer}^2$  &  $P_{b,layer}$  for each layer
4: set optional stop criteria for lower (LB) and upper
   bound (UB), e.g.  $d_{h,layer,LB}^2$ ;  $d_{h,layer,UB}^2$ ;  $P_{b,layer,LB}$ ;  $P_{b,layer,UB}$ 
5: set flag as true
6: set maximum number of iterations
7: loop until stop criteria fulfilled or flag equals false
   or number of iterations exceeded
8:   set for each layer:  $d_{h,layer,old}^2$  as  $d_{h,layer}^2$ 
9:   set for each layer:  $P_{b,layer,old}$  as  $P_{b,layer}$ 
10:  optimize  $d_{h,layer}^2$  with Algorithm 2 for each layer
11:  optimize  $P_{b,layer}$  with Algorithm 3 for each layer
12:  measure  $d_{h,layer}^2$  &  $P_{b,layer}$  for each layer
13:  if  $d_{h,layer}^2 < d_{h,layer,old}^2$  then
14:    set flag as false
15:  end if
16:  if  $P_{b,layer} > P_{b,layer,old}$  then
17:    set flag as false
18:  end if
19: end loop
    
```

---

First, the initial modulation scheme, i.e., constellation points and labels, is normalized and the *Harmonic Mean* and the *Bit Error Probability* of each layer are calculated. Additionally, optional stopping criteria for the algorithm are defined, e.g., an upper and a lower bound for the main criteria. For example, a very small *Harmonic Mean* for the

EL might be a good choice to optimize BICM-ID performance for the EL. The algorithm optimizes for each layer two criteria. The parameters are measured once again and compared with the old values. This is important because the optimization of the first parameter, e.g., the *Harmonic Mean*, might degrade the *Bit Error Probability* and vice versa. A comparison between the two constellations is done to recognize an improvement during the iteration process. If one parameter degrades, the algorithm stops immediately and outputs the final constellation. To maximize the *Harmonic Mean*, we propose to use an algorithm as described in Algorithm 2.

---

#### Algorithm 2: Maximize $d_{h,layer}^2$ of a specific layer

---

```

1: Initialize normalized modulation scheme
2: for k = 1 to number of constellation points do
3:   compute direction vector  $\vec{r}_{k,layer}$  using Eq. (3)
4: end for
5: for k = 1 to number of constellation points do
6:   move constellation point to direction of  $\vec{r}_{k,layer}$ 
7: end for
8: normalize
    
```

---

The algorithm initializes the normalized modulation scheme which has been committed by algorithm 1. Then the direction vector  $\vec{r}_{k,layer}$  for a specific layer is computed. A more detailed explanation of the derivation and definition of  $\vec{r}_{k,layer}$  including Eq. (3) are given in sub-section 3.1. The computation is performed for each symbol of the modulation scheme. Finally, the constellation points are moved towards the computed direction and normalized. The concept of Algorithm 2 can easily be adapted to the *Bit Error Probability* in the same way and is described in Algorithm 3. The direction vector is here identified as  $\vec{s}_{k,layer}$  which is derived and explained in more details in sub-section 3.3.

---

#### Algorithm 3: Maximize $P_{b,layer}$ of a specific layer

---

```

1: Initialize normalized modulation scheme
2: for k = 1 to number of constellation points do
3:   compute direction vector  $\vec{s}_{k,layer}$  using Eq. (9)
4: end for
5: for k = 1 to number of constellation points do
6:   move constellation point to direction of  $\vec{s}_{k,layer}$ 
7: end for
8: normalize
    
```

---

#### 3.1 Maximization of the *Harmonic Mean* $d_{h,layer}^2$

To maximize the *Harmonic Mean*  $d_h^2$  of both layers in a HM-BICM-ID we first need to define it for hierarchical modulation schemes. From [2] the definition of  $d_h^2$  can be modified and rewritten in such a way that  $d_{h,layer}^2$  is given

by the invers of the sum of the *Harmonic Mean Costs*  $H_k^{cost}$  for each Symbol  $x_k$  with  $k \in [1..M]$  of an  $M$ -ary modulation scheme with predefined labels and constellation points:

$$d_{h,layer}^2 = \left( \frac{1}{M} \sum_{k=1}^M H_k^{cost} \right)^{-1} \quad (1)$$

To improve a HM-BICM-ID system  $d_{h,layer}^2$  needs to be maximized which corresponds to the minimization of all *Harmonic Mean Costs*  $H_k^{cost}$ :

$$H_k^{cost} = \frac{1}{m_{layer}} \sum_{l=1}^{m_{layer}} \left( \frac{2^{m_{layer}}}{M} \sum_{z \in X_{b(x_k)}^l} \frac{1}{\|x_k - z\|^2} \right) \quad (2)$$

$H_k^{cost}$  is defined as the sum of the inverse *Euclidean Distances* between the symbol  $x_k$  and all neighbors  $z$  defined by the subset  $X_{b(x_k)}^l$ . The subset includes all symbols  $z$  with an inversed bit at the  $l^{th}$  bit position and equal bit patterns for the remaining BL. For example, the EL symbol  $32_{10} / [\text{MSB}]10000_2$  is related to the BL symbol  $2_{10} / [\text{MSB}]10_2$ . Considering the 2<sup>nd</sup> bit position  $l=2$  all EL symbols with binary labelling  $[\text{MSB}]11xxxx_2$  are neighbors because they are all related to the BL symbol  $3_{10} / [\text{MSB}]11_2$ . This is shown in Fig. 4 by the green lines. The second group with inversed bit at bit position  $l=1$  is defined by the BL symbol  $0_{10} / [\text{MSB}]00_2$  and related to all EL symbols with bit pattern equal to  $[\text{MSB}]00xxxx_2$ . This is shown in Fig. 4 by blue lines. As a consequence, the number of neighbors for a hierarchical modulation scheme for a specific layer increases compared to the definition of the *Harmonic Mean* in non-hierarchical modulation schemes.

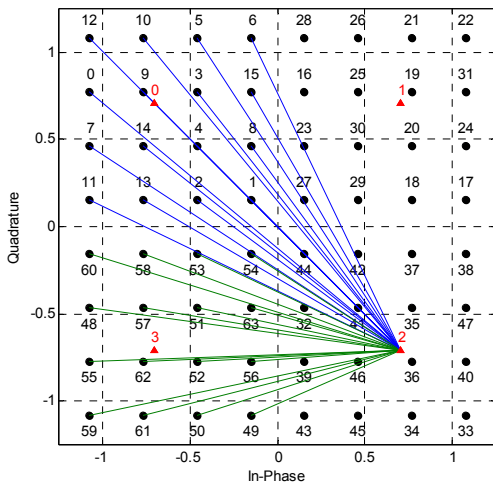


Figure 4: Constellation diagram of a QPSK (BL) and 64-QAM (EL) with corresponding pairs of neighbors in terms of the optimization of the *Harmonic Mean* of the BL.

With the definition from Eq. (1) and Eq. (2), we propose the maximization of the *Harmonic Mean* by moving the constellation points far away from all neighbors with high influence. As a consequence, the vector for the movement considering a specific layer can be identified as follows:

$$\vec{r}_{k,layer} = \frac{\xi_{layer}}{m_{layer}} \sum_{l=1}^{m_{layer}} \left( \frac{2^{m_{layer}}}{M} \sum_{z \in X_{b(x_k)}^l} \frac{e^{j \cdot \angle(x_k - z)}}{\|x_k - z\|^2} \right) \quad (3)$$

The vector  $\vec{r}_{k,layer}$  describes the vector to move the  $k^{th}$  symbol of an  $M$ -ary modulation scheme in a proper direction. The definition is similar to Eq. (2) and introduces an additional term for the direction  $e^{j \cdot \angle(x_k - z)}$ . The inverse of the *Euclidean Distance* is used to weight the direction vector. A lower distance between neighbors in the terms of the *Harmonic Mean* causes an increased value and influence of the movement. The direction component in  $x_k$  for the movement away from  $z_l$  is defined by the vector subtraction. The final direction vector is the superposition of the direction vectors of each neighbor. For hierarchical modulation schemes the number of neighbors differs significantly. The scaling factor  $\xi_{layer}$  is introduced to further control the convergence behavior of Algorithm 1 and shall be a positive real value  $\xi_{layer} \in \mathbb{R}^+$ . Considering a two layered modulation scheme with a BL of 2 bits per symbol and an EL with 6 bits per symbol values for  $m_{layer}$  are given by  $m_{BL} = 2$  and  $m_{EL} = 6$ . Please note, in case of the EL the definition of the *Harmonic Mean* from Eq. (1) and (2) falls back to the definition of [2] because the number of neighbors reduces to  $m_{EL}$ .

### 3.2 Convergence Behavior during Optimization of the *Harmonic Mean*

In the previous subsection we described the algorithm for optimizing the *Harmonic Mean*  $d_h^2$  for each layer. A high *Harmonic Mean* guarantees a convergence to an asymptotic low BER during the iterative process in BICM-ID and HM-BICM-ID systems. This is mainly because the *a priori* knowledge is going to improve during the iterations and thus the demodulator is able to distinguish much better between two symbols of similar labels (one bit position differs). However, in the first iteration in BICM-ID another challenge has to be mastered because no *a priori* information is available at the beginning. During the development of Algorithm 1, it has been observed for a given constellation, e.g., 16-QAM as shown in Fig. 5 (a), and several algorithmic iterations that the novel constellation may converge to a multi-labeled *Binary Phase-Shift Keying* (BPSK). Two groups of eight constellation points with even and odd parity bits in the labels appear as depicted in Fig. 5 (b). This is caused by the fact that BPSK has the highest *Harmonic Mean* of  $d_{h,BPSK}^2 = 4$ .

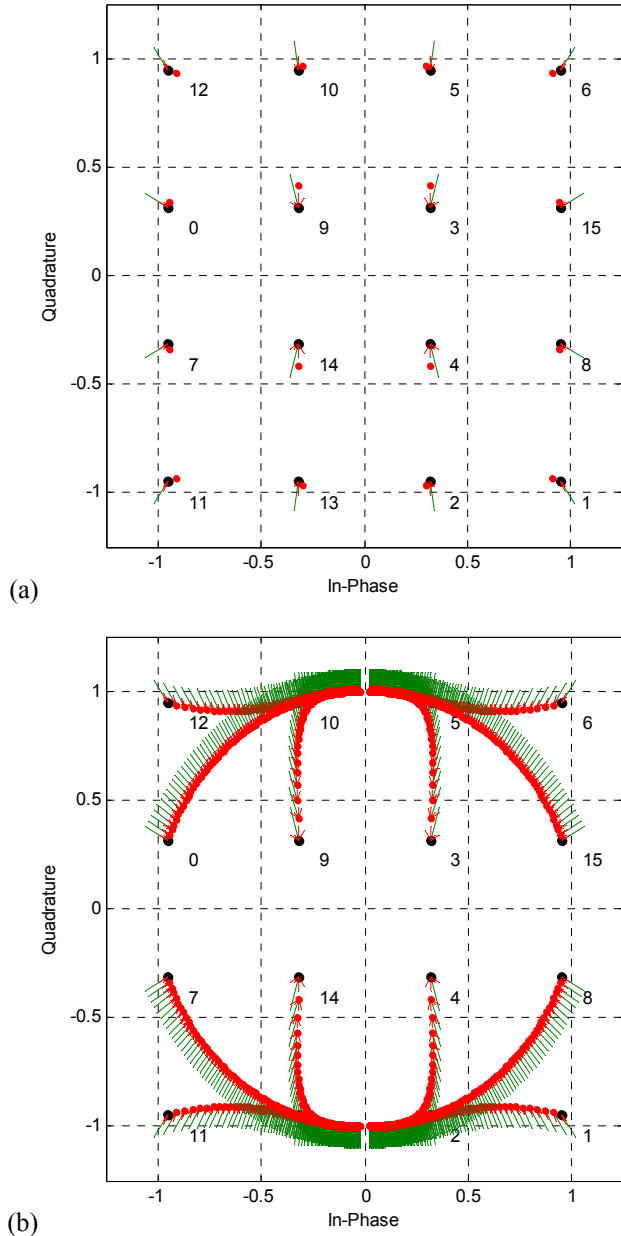


Figure 5: 16-QAM-Ray after 1<sup>st</sup> iteration (a) and 100<sup>th</sup> iteration (b) with Algorithm 2 and  $\xi = 0.1$ .

However, this causes a new challenge. Without any *a priori* knowledge at the first step in a BICM-ID system, the soft-demodulator, based on a multi-labeled BPSK, can only distinguish between the even and odd parity group but not between different labels within the same group. This is because the labels within one group have the same constellation points. Therefore the demodulator cannot produce any valuable extrinsic information and the BER performance will degrade.

In a second simulation we analyzed the behavior of the *Harmonic Mean* under the influence of the number of

iterations and the scaling factor  $\xi$ . The results are shown in Table 1.

Table 1: Influence of the scaling factor  $\xi$  and the number of iterations on the convergence behavior of Algorithm 1

Iteration number	<i>Harmonic Mean</i> $d_h^2$		
	$\xi = 0.01$	$\xi = 0.1$	$\xi = 1.0$
0 <sup>th</sup> (initial)	2.7190	2.7190	2.7190
1 <sup>st</sup> iteration	2.7411	2.9102	3.4454
2 <sup>nd</sup> iteration	2.7625	3.0555	3.6244
5 <sup>th</sup> iteration	2.8232	3.3369	3.8483
10 <sup>th</sup> iteration	2.9136	3.5669	3.9619
100 <sup>th</sup> iteration	3.5856	3.9946	4.0000

For this, the algorithm has been executed with an initial 16-QAM-Ray, as given in [17][18], and without any restrictions, i.e., no partitioning into different layers and no further stopping criteria. 16-QAM-Ray labelling has been chosen because of its *Harmonic Mean* of  $d_{h,16\text{-QAM-Ray}}^2 = 2.719$  which is the highest compared to any other 16-QAM labelling. Within Table 1, a general convergence behavior towards a multi-labeled BPSK scheme can be observed for all parametrizations. But with a higher value for  $\xi$  a reduced number of iterations is needed to reach the multi-labeled BPSK and therefore a faster convergence behavior is observed. But this might also result in the instability of the algorithm when used with a hierarchically modulated scheme. Thus, the movement must be controlled by a carefully chosen  $\xi$  to ensure that no constellation point will cross a decision bound of another layer as this would violate the premise of one layer being a subset of another layer.

### 3.3 Minimization of the *Bit Error Probability* $P_{b,layer}$

As a consequence of the multi-labeling problem, we propose not only to use the *Harmonic Mean*  $d_h^2$  as an optimization criterion, but also the *Bit Error Probability*  $P_b$ . The main idea is to prevent the collapse of several constellation points to groups and thus achieve a minimization of the BER for the first iteration in BICM-ID.

In Algorithm 3, we propose the calculation of the direction vector  $\vec{s}_{k,layer}$  for a given constellation point  $x_k$  based on the *Probability Density Functions* (PDF) of each symbol in a modulation scheme as well as the labels and the decision bounds. The main challenge to calculate the influence is the fact that the symbols are moving and changing during the iteration process of the algorithm and therefore the decision bounds and PDF are changing. This renders the determination of the *Bit Error Probability* in an analytical way, e.g., by integration, very complex or even impossible. Therefore, we propose a numerical way of calculating the *Bit Error Probability* considering a division of the I/Q-plane into sufficiently segments. Assuming an additive white Gaussian noise (AWGN) channel, the PDF  $p(\mathbf{w}, \mathbf{x}_k)$  in  $\mathbf{w}$  with the

mean equal to the  $k^{\text{th}}$  symbol  $\mathbf{x}_k$  and a variance of  $\sigma^2 = N_0/E_s$  can be described by:

$$p(\mathbf{w}, \mathbf{x}_k) = Pr(\mathbf{x}_k) \cdot \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{|\mathbf{w}-\mathbf{x}_k|^2}{2\sigma^2}} \quad k \in [1..M] \quad (4)$$

To fulfill the side constraint that the segment under the sum of the PDF  $p(\mathbf{w}, \mathbf{x}_k)$  in  $\mathbf{x}_k$  over all symbols must be equal to one, the  $p(\mathbf{w}, \mathbf{x}_k)$  must be normalized by the symbol probability  $Pr(\mathbf{x}_k)$ . Considering that all symbols are equally distributed, we can define the symbol probability as  $Pr(\mathbf{x}_k) = 1/M$ . Assuming, that a noisy symbol has been received in the specified small segment with center  $\delta_{i,j}$  and, assuming that the symbol  $\mathbf{x}_k$  has been sent, the *Bit Error Probability*  $Pr(\delta_{i,j}, \mathbf{x}_k)$  is given by:

$$Pr(\delta_{i,j}, \mathbf{x}_k) = \int_{\delta_j - \frac{\Delta}{2}}^{\delta_j + \frac{\Delta}{2}} \int_{\delta_i - \frac{\Delta}{2}}^{\delta_i + \frac{\Delta}{2}} p(\mathbf{w}, \mathbf{x}_k) dw_x dw_y \quad (5)$$

$\Delta$  describes a sufficiently small grid size for a specific segment of the I/Q-plane for both directions  $w_x$  and  $w_y$ .  $\delta_{i,j} = (\delta_i, \delta_j) \in \mathbb{R}^2$  is the center of the specific segment. For all segments which are part of a given border, a slightly modified bound of  $\pm\infty$  for the integral shall be used. In a further step, only those segments where the labels between the nearest and the originally transmitted symbol differs in at least one bit position will have an influence on the *Bit Error Probability* from (5). Therefore, it must be multiplied by the label error rate  $\Lambda_{\text{layer}}$  which is related to the normalized *Hamming Distance*. We can derive:

$$P_{b,\text{layer}}(\delta_{i,j}, \mathbf{x}_k) = \Lambda_{\text{layer}}(\mu(\delta_{i,j}), \mu(\mathbf{x}_k)) \cdot Pr(\delta_{i,j}, \mathbf{x}_k) \quad (6)$$

$\Lambda_{\text{layer}}(\mu(\delta_{i,j}), \mu(\mathbf{x}_k))$  is defined by the *Hamming Distance* considering only a subset of  $m_{\text{layer}}$  bits of the labels divided by the maximum number of layer bits:

$$\Lambda_{\text{layer}}(\mu(\delta_{i,j}), \mu(\mathbf{x}_k)) = \frac{d_{\text{ham}, m_{\text{layer}}}(\mu(\delta_{i,j}), \mu(\mathbf{x}_k))}{m_{\text{layer}}} \quad (7)$$

$d_{\text{ham}}(\mu(\delta_{i,j}), \mu(\mathbf{x}_k))$  is the *Hamming Distance* between the label of  $\mathbf{x}_k$  and the label related to  $\delta_{i,j}$  where  $\delta_{i,j}$  is mapped to the nearest symbol  $\mathbf{x}_l$  with:

$$l = \arg \min_{n \in [1..M]} |\delta_{i,j} - \mathbf{x}_n|^2 \quad (8)$$

A colored projection of the mapping  $\mu$  is shown in Fig. 6. The values in the color bar are arranged from 1 to 64 according to the mapping of the hierarchical 64-QAM modulation scheme from Fig. 2 (b).

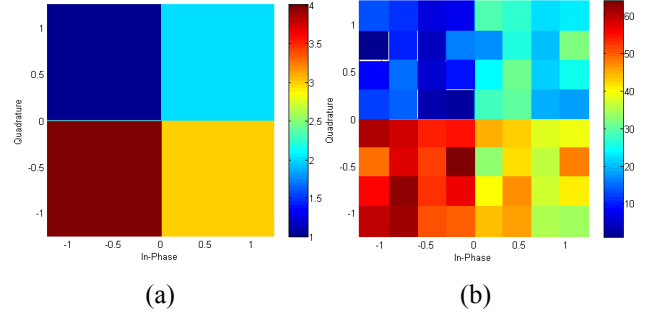


Figure 6: Colored projection of the mapping  $\mu(\delta_{I,Q})$  of a hierarchical 64-QAM modulation scheme from Fig. 2. BL (a) and EL (b).

Similarly, the hierarchical BL part of the 64-QAM modulation scheme which is given by the QPSK from Fig. 2 (a) is also shown in Fig. 6. The colors of the four quadrants are similar to the mean color of those quadrants in Fig. 6 which is typical for a hierarchical modulation scheme. The corresponding *Bit Error Probability*  $P_b(\delta_{i,j}, \mathbf{x}_k)$  of all symbols  $\mathbf{x}_k$  over the I/Q-plane is shown as a colored scatterplot in Fig. 7.

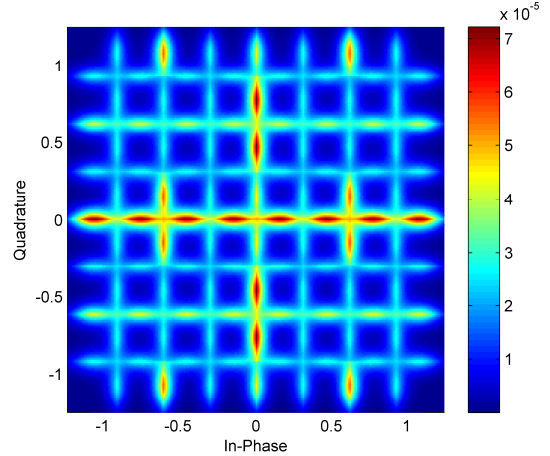


Figure 7: Colored projection of the *Bit Error Probability* for all symbols of a given two layered 64-QAM as depicted in Fig. 2 (b) for  $E_s/N_0 = 15$  dB.

With the given *Bit Error Probability*  $P_b(\delta_{i,j}, \mathbf{x}_k)$  for a small segment in the constellation diagram the direction of influence for  $\mathbf{x}_k$  can be expressed as follows:

$$\vec{s}_{k,\text{layer}} = \beta_{\text{layer}} \sum_{\delta_{i,j} \in \mathbb{R}^2} P_{b,\text{layer}}(\delta_{i,j}, \mathbf{x}_k) \cdot e^{j \cdot z(\mathbf{x}_k - \delta_{i,j})} \quad (9)$$

$\vec{s}_{k,layer}$  is a vector describing the direction of movement as a result of the superposition of the movements expressed by each small segment. For all segments a direction based on the vector subtraction of  $\mathbf{x}_k - \delta_{i,j}$  can be determined. The weight for the direction is directly related to the bit error rate of each segment. Therefore, a low  $E_s/N_0$  and a short distance to a neighbor will result in a higher weight of the segment. Due to the nature of the Gaussian distribution the segments with highest impact are arranged at the borders of the decision bounds. Consequently, the direction of movement is very often similar to the task of increasing the distance to the nearest neighbor. The convergence behavior can be controlled by the positive real value  $\beta_{layer} \in \mathbb{R}^+$ .

#### 4. PERFORMANCE ANALYSIS OF HM-BICM-ID WITH OPTIMIZED MODULATION SCHEME

In the previous sections we introduced our proposed algorithm. Two parameters, the *Harmonic Mean* and the *Bit Error Probability*, have been used as optimization criteria for the algorithm. Now, we want to use the novel algorithm to develop a novel hierarchical modulation scheme with optimized performance in terms of BER for both the BL and EL of a two layer modulation scheme.

##### 4.1 An Optimized Two Layer Modulation Scheme Based on a Hierarchical 64-QAM Modulation Scheme

In [12], HM-BICM-ID has been introduced for several configurations. Both three layer and two layer hierarchical modulation schemes have been investigated and it was shown that a reduced number of layers give the designer more freedom to optimize the modulation scheme. Therefore, we propose a HM-BICM-ID system with one BL and one EL to keep the freedom in the design of the hierarchical modulation as high as possible. Further, we propose the BL modulation scheme to be a QPSK as shown in Fig. 2 (a) with *Harmonic Mean*  $d_{h,QPSK}^2 = 2.6667$ .

The initial EL is a hierarchical 64-QAM which can be constructed by superposition of QPSK mapping (first 2 bits fixed) and a 16-QAM-Ray labelling from [17][18] for each quadrant. The mapping is identical to Fig. 2 (b). The parameter set for the algorithm has been chosen to  $\beta_{EL} = 25.0$ ,  $\xi_{EL} = 0.01$ ,  $\beta_{BL} = 0.2$ ,  $\xi_{BL} = 0.01$ . The maximum number of iterations is 10. There were no further restrictions, e.g., no optional stopping criteria. Executing the algorithm results in a constellation diagram as depicted in Fig. 8.

Due to the look of the scatter plot we proposed to call the constellation *64-QAM Butterfly* (64-QAM-BF). The normalized symbols of all labels (index) are given in details in Table 2.

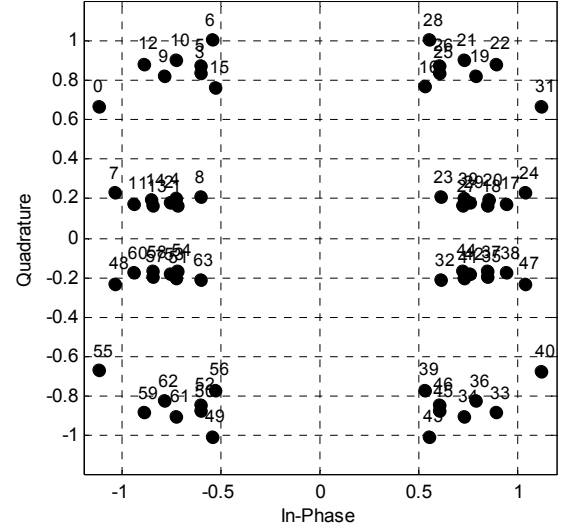


Figure 8: 64-QAM-BF modulation scheme

Table 2: Symbol mapping of the 64-QAM-BF

Map	Symbol	Map	Symbol
0	-0.8910 + 0.8805i	1	-0.7261 + 0.9005i
2	-0.6066 + 0.8749i	3	-0.5489 + 1.0047i
4	0.5500 + 1.0045i	5	0.6072 + 0.8750i
6	0.7262 + 0.9004i	7	0.8909 + 0.8806i
8	-1.1176 + 0.6701i	9	-0.7861 + 0.8201i
10	-0.6058 + 0.8386i	11	-0.5292 + 0.7662i
12	0.5303 + 0.7667i	13	0.6063 + 0.8386i
14	0.7863 + 0.8201i	15	1.1176 + 0.6703i
16	-1.0410 + 0.2303i	17	-0.8521 + 0.1918i
18	-0.7322 + 0.2010i	19	-0.6073 + 0.2088i
20	0.6079 + 0.2086i	21	0.7323 + 0.2009i
22	0.8521 + 0.1917i	23	1.0413 + 0.2301i
24	-0.9419 + 0.1735i	25	-0.8464 + 0.1646i
26	-0.7566 + 0.1831i	27	-0.7242 + 0.1625i
28	0.7243 + 0.1625i	29	0.7567 + 0.1830i
30	0.8463 + 0.1646i	31	0.9420 + 0.1734i
32	-0.9413 - 0.1755i	33	-0.8475 - 0.1665i
34	-0.7569 - 0.1841i	35	-0.7245 - 0.1648i
36	0.7246 - 0.1648i	37	0.7569 - 0.1840i
38	0.8475 - 0.1664i	39	0.9412 - 0.1754i
40	-1.0408 - 0.2326i	41	-0.8509 - 0.1922i
42	-0.7329 - 0.2021i	43	-0.6078 - 0.2108i
44	0.6084 - 0.2106i	45	0.7330 - 0.2020i
46	0.8510 - 0.1921i	47	1.0410 - 0.2324i
48	-1.1191 - 0.6706i	49	-0.7863 - 0.8226i
50	-0.6054 - 0.8405i	51	-0.5290 - 0.7672i
52	0.5301 - 0.7678i	53	0.6060 - 0.8405i
54	0.7864 - 0.8226i	55	1.1191 - 0.6707i
56	-0.8917 - 0.8813i	57	-0.7263 - 0.9021i
58	-0.6061 - 0.8763i	59	-0.5489 - 1.0052i
60	0.5499 - 1.0050i	61	0.6066 - 0.8763i
62	0.7264 - 0.9021i	63	0.8916 - 0.8815i



## 4.2 Performance Analysis for 64-QAM-BF

In a first step, we measured the *Harmonic Mean* for different modulation schemes as depicted in Table 3. The upper schemes are measured for a single layered typical BICM-ID system. The values are given from literature. It can be observed that hierarchical schemes used for BICM-ID systems have poor performance due to the significantly reduced *Harmonic Mean*. Therefore in [12], performance measurements in terms of BER have been proposed. The best performance has been reached by an 8x8-PSK modulation scheme because the *Harmonic Mean* is increased. The reason for this is a relaxed design of the constellation points which is no more fixed to the 64-QAM scheme. Compared to the latter one, the novel 64-QAM-BF modulation scheme has a slightly reduced *Harmonic Mean*. But, considering the *Harmonic Mean* of the BL in a HM-BICM-ID system we find the *Harmonic Mean* of the BL to be increased dramatically to  $d_{h,BL}^2 = 2.847$ . Therefore, we expect the 64-QAM-BF to outperform all other two and three layered hierarchical modulation schemes from [12].

Table 3: *Harmonic Mean*  $d_h^2$  for several modulation schemes

System	Modulation	Layer		
		1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>
BICM-ID	BPSK (upper bound)	4.000		
	$\varphi$ -4PSK; $\varphi=90^\circ$ [19]	2.667		
	16-QAM-Ray [17][18]	2.719		
	64-QAM Non Hierarch. [12]	2.874		
	16-QAM Straightforward [12]	0.853		
	64-QAM Straightforward [12]	0.290		
	8x8-PSK [12]	0.792		
	64-QAM-BF	0.689		
HM-BICM-ID	BL $\varphi$ -4PSK; $\varphi=90^\circ$ [19]	1.407	0.665	0.290
	EL1 16-QAM Straightfor. [12]			
	EL2 64-QAM Straightforward			
	BL $\varphi$ -4PSK; $\varphi=90^\circ$ [19]	1.958	0.792	
	EL 8x8-PSK [12]			
	BL $\varphi$ -4PSK; $\varphi=90^\circ$ [19]	2.847	0.689	
EL 64-QAM-BF				

In a second step, we performed a BER simulation for a two layer HM-BICM-ID system with 64-QAM-BF. BER curves and the corresponding EFF curves have been measured for both the BL and EL. The setup for the simulation environment has been chosen equivalent to [12] and is depicted in Table 4. In Fig. 9 the BER performance depending on the energy per symbol to noise ratio  $E_s/N_0$  is depicted for three systems. The reference system is a non-iterative BICM and uses a QPSK modulation scheme with Gray labeling. The BER is described by the black solid curve. The second system is the HM-BICM-ID from [12] with a hierarchical modulation with relaxed design constraints. The BER performance of the BL and EL referred in [12] are depicted by the blue solid (square marker) and green solid (diamond marker) BER curves. The performance of the EL compared to the reference system

considering a BER of  $10^{-6}$  is a gain of 2.98 dB. But, comparing the reference system with the BL approach from [12] we find a loss of 2.02 dB.

Table 4: Setup of simulation parameters for HM-BICM-ID

Parameter	Value
Frame size	1000 net bits
BL FEC	Feed Forward Convolutional Code G(5,7) <sub>8</sub> ; R = 1/2; K = 3; L = 2 tail bits
BL encoded bits	2004 encoded bits
BL Interleaver	Random Interl.: size of 2004 bits
BL modulation	QPSK: 2 bits / symbol
EL FEC	FFCC: G(5,7,5,5,5,7) <sub>8</sub> ; R = 1/6; K = 3; L = 2 tail bits
EL encoded bits	6012 encoded bits
EL Interleaver	Hierarchical Random configuration Interl. 1: 2004 bits; Interl. 2: 4008 bits
EL modulation	hierarchical 64-QAM: 6 bits / symbol 2 bits for BL; 4 bits for EL
total symbols	1002 symbols / frame
total rate	$\approx 1$ net bit per symbol
Iterations	10

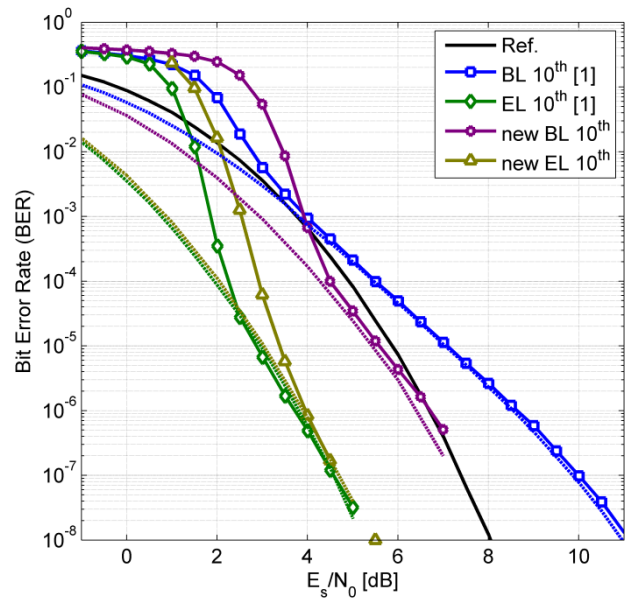


Figure 9: Comparison between HM-BICM-ID with novel modulation scheme and with relaxed design constraint [12]

However, the third system corresponds to our novel approach for a HM-BICM-ID with a modulation scheme as depicted in Fig. 9. The BER performance is given by the violet solid line (star marker) and yellow other solid line (triangle marker). Each dashed colored line describes the EFF curve of the associated HM-BICM-ID system. The EFF describes the lowest possible BER performance that can be reached with the corresponding system. As it can be seen, the novel EL approach has a similar EFF curve compared to the EL from [12]. For a BER of  $10^{-6}$  dB a gain of

approximately 2.71 dB can be reached. Considering now the BL and BER of  $10^{-6}$  dB the novel approach has a very small Loss of 0.02 dB. Therefore, the BL performance of the novel approach outperforms [12]. This remains also valid for  $\text{BER} < 10^{-6}$ . Of course, the better performance in the BL comes with the cost of a shifted *waterfall region* in both the BL and EL. For the BL, the novel HM-BICM-ID already improves for  $E_s/N_0 > 4$  dB and  $\text{BER} < 10^{-3}$ . Other regions of the BER and  $E_s/N_0$  are irrelevant. Therefore, only in the EL the novel approach performs slightly worse because of the shifted *waterfall region*. For higher  $E_s/N_0$  the novel approach performs similar to [12]. Therefore, we can conclude that the novel algorithm gives the designer more freedom to improve specific layers without degrading the performance of another layer in the same way. Thus, the overall BER of HM-BICM-ID is improved. Further, we can additionally deduct that ILI has been greatly reduced. Finally, the novel HM-BICM-ID system outperforms the BICM reference system in the EL for values of  $E_s/N_0 > 2$  dB and in the BL between  $4 \text{ dB} < E_s/N_0 < 7 \text{ dB}$ . Only in the relevant range  $E_s/N_0 > 7 \text{ dB}$  a small loss of the BER performance in the BL can be observed although the EFF curve is below the BER curve of the reference system.

## 5. CONCLUSIONS

In our paper we proposed the concept of HM-BICM-ID. To further improve hierarchical modulation schemes, we developed a novel algorithm to move constellation points of a certain modulation scheme to a direction where critical parameters for a specific layer, i.e., the *Harmonic Mean*, and the *Bit Error Probability* are maximized. For a given parametrization, e.g., the number of iterations and the convergence behavior of each optimization step, we derived a novel modulation scheme termed 64-QAM-BF. To demonstrate the performance of the scheme we designed a HM-BICM-ID with 64-QAM-BF and compared it with those already known from literature. It has been observed that the novel algorithm provides more design freedom to improve BER performance. Finally, it has been shown that the novel HM-BICM-ID outperforms the reference system for a wide range of  $E_s/N_0$  in both the BL and EL. In our future research work, we plan to modify the algorithm to improve the convergence behavior and the balance between the different optimization criteria. This shall help to fine tune modulation schemes. HM-BICM-ID uses BICM-ID in each layer. In future, we plan a novel receiver using more powerful codes, e.g., turbo codes. Thus, an unequal FEC for each layer shall help to balance performance between layers.

## 6. REFERENCES

- [1] X. Li, J.A. Ritcey, "Bit Interleaved Coded Modulation with Iterative Decoding", *IEEE Communications Letters*, vol. 1, no. 6, pp. 169-171, November 1997.

- [2] X. Li, A. Chindapol, J.A. Ritcey, "Bit-Interleaved Coded Modulation with Iterative Decoding and 8-PSK Signaling", *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1250-1257, August 2002.
- [3] G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 56-67, Januar 1982.
- [4] J. Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, March 1996.
- [5] X. Li, J.A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback", *IEEE Electronics Letters*, vol. 34, no. 10, May 1998.
- [6] A. Seeger, "A new Signal Constellation for the Hierarchical Transmission of Two equally sized data stream", *Proc. of IEEE International Symposium on Information Theory (ISIT)*, p. 169, Ulm, Germany, June 1997.
- [7] H. Jiang, P.A. Wilford, "A Hierarchical Modulation for Upgrading Digital Broadcast Systems", *IEEE Transactions on Broadcasting*, vol. 51, no. 2, pp. 223-229, June 2005.
- [8] ETSI EN 302 307 V1.3.1 (2013-03), "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)"
- [9] ETSI EN 302 755 V1.3.1 (2012-04), "Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)"
- [10] X. Zhe, W.Y. Sheng, F. Alberge, P. Duhamel, "A Turbo Iteration Algorithm in 16QAM Hierarchical Modulation", *Proc. of IEEE Wireless Communications, Networking and Information Security (WCNIS)*, pp. 9-12, Beijing (China), June 2010.
- [11] Q. Li, J. Zhang, L. Bai, J. Choi, "Performance Analysis and System Design for Hierarchical Modulated BICM-ID", *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3056-3069, June 2014.
- [12] M. Adrat, M.F.T. Oshim, M. Tschauer, M. Antweiler, B. Eschbach, P.Vary, "On Hierarchical Modulated BICM-ID for Receivers with Different Combinations of Code Rate and Modulation Order", *SDR15-WInnComm*, pp. 129-134, San Diego, USA, March 2015.
- [13] S. Kallel, "Complementary punctured convolutional (CPC) codes and their applications", *IEEE Transactions on Communications*, vol. 43, no. 6, pp. 2005-2009, June 1995.
- [14] C.F. Ball, K. Ivanov, P. Stockl, C. Masseroni, S. Parolari, R. Trivisonno, "Link quality control benefits from a combined incremental redundancy and link adaptation in EDGE networks", *Proc. of IEEE 59th Vehicular Technology Conference (VTC)*, vol. 2, pp. 1004-1008, Milan (Italy), May 2004.
- [15] D.G. Daut, J.W. Modestino, L. Wismer, "New Short Constraint Length Convolutional Code Constructions for Selected Rational Rates (Corresp.)", *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 794-800, 1982.
- [16] P. Frenger, P. Orten, T. Ottosson, "Convolutional Codes with Optimum Distance Spectrum", *IEEE Communication Letters*, vol. 3, no. 11, pp. 317-319, Nov. 1999.
- [17] F. Schreckenbach, N. Görtz, J. Hagenauer, G. Bauch, "Optimization of Symbol Mappings for Bit-Interleaved Coded Modulation With Iterative Decoding", *IEEE Communications Letters*, vol. 7, no. 12, pp. 593-595, December 2003.
- [18] T. Clevorn, S. Godtmann, P. Vary. "EXIT Chart Analysis of Non-Regular Signal Constellation Sets for BICM-ID", *International Symposium on Information Theory and its Applications (ISITA)*, Parma, Italy, October 2004.
- [19] T. Clevorn, P. Vary. "Iterative Decoding of BICM with Non-Regular Signal Constellation Sets", *5th International ITG Conf. on Source and Channel Coding (SCC)*, Erlangen, Germany, January 2004.

## ADVANCED LOW POWER, HIGH SPEED NONLINEAR SIGNAL PROCESSING: AN ANALOG VLSI EXAMPLE

Giuseppe Oliveri (Ulm University, Ulm, Germany; giuseppe.oliveri@uni-ulm.de);  
 Mohamad Mostafa (German Aerospace Center, Wessling, Germany);  
 Werner G. Teich (Ulm University, Ulm, Germany);  
 Jürgen Lindner (Ulm University, Ulm, Germany);  
 Hermann Schumacher (Ulm University, Ulm, Germany).

### ABSTRACT

We revisit the topic of signal processing with analog circuits and its potential to increase the energy efficiency. A vector equalizer based on a recurrent neural network structure is taken as an example to demonstrate what can be achieved with state of the art in VLSI design. First measurements of our analog VLSI circuit confirm the possibility to achieve an energy efficiency of about 36 TFlops/Watt, which is an improvement factor of three to four orders of magnitude compared with today's most energy efficient digital circuits.

### 1. INTRODUCTION

Energy efficiency became an increasingly important topic in the past years, especially for mobile devices. The latest Green500 [1] and Top500 [2] ranking show that the most efficient heterogeneous supercomputer can reach an energy efficiency of about five GFlops/Watt, and there is a lot of effort to increase this value. We address here alternatives offered by analog circuits. Some authors of earlier work in the field of advanced analog signal processing concluded that analog systems have the potential to improve the efficiency substantially [3]. Moreover, depending on the application, there might be no need for additional A/D conversion [4].

For sophisticated algorithms, nonlinear processing is needed. The nonlinearity offers the chance to use analog circuits out of their conventional field of linear signal processing, with all its disadvantages, like accumulation of noise and inaccuracies of circuits elements. The corresponding algorithms result as robust as their digital counterparts. This is no surprise, since "digital circuits" are in essence analog circuits with strong nonlinearities. To demonstrate how far the energy efficiency can be increased with state of the art VLSI design, in this paper we take a so-called vector equalizer as an example.

Related to our work are activities in building circuits emulating functions of natural neural networks, e.g. the European "Human Brain" project [5]. In this context it is

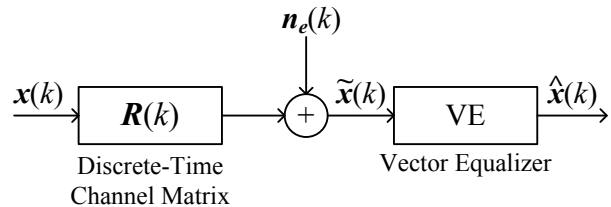


Fig. 1. Discrete-time model on symbol basis for an uncoded transmission with linear modulation over MIMO channels.

common to use the term "neuromorphic computing", see also the early work by Mead [6]. In contrast to neuromorphic computing, our focus is more on common signal processing algorithms, usually realized today with digital circuits or digital processors.

The paper is organized as follows: In Section 2 we explain the background of our example and the structure of the algorithm, and in Section 3 we describe the circuit. Section 4 discusses simulation results, while Section 5 shows measurements on a real chip. Conclusions in Section 6 close the paper.

## 2. STRUCTURE OF THE ALGORITHM

### 2.1. Background

The background for our example is an uncoded digital transmission over radio channels with multiple antennas (Multiple-Input-Multiple-Output, MIMO), and we assume a linear modulation scheme. Fig. 1 shows a model for such a transmission, which is a discrete-time model on symbol basis. More about this model and its relation to the continuous-time (physical) transmission model can be found in [7]. The quantities in the figure are:

- $k$  is the discrete-time symbol interval variable;
- $\mathbf{x}(k)$  is the transmit symbol vector of length  $N$  at symbol interval  $k$ . We assume binary phase shift keying (BPSK), i.e.  $x_i(k) \in \{-1, +1\}$ , so the transmit symbol alphabet  $A_x$  contains  $2^N$  possible transmit vectors;

- $\mathbf{R}(k)$  is the discrete-time channel matrix on symbol basis. Its size is  $(N \times N)$  and it is hermitian and positive semidefinite;
- $\mathbf{n}_e(k)$  is a sample function of an additive Gaussian noise vector process.  $\Phi_{\mathbf{n}_e \mathbf{n}_e}(k) = \frac{N_0}{2} \cdot \mathbf{R}(k)$  is the covariance matrix, with  $N_0$  the single-sided noise power spectral density;
- $\tilde{\mathbf{x}}(k) = \mathbf{R}(k) * \mathbf{x}(k) + \mathbf{n}_e(k)$  is the received symbol vector.  $*$  means matrix-vector convolution;
- $\hat{\mathbf{x}}(k) \in A_x$  is the decided vector at the output of the vector equalizer (VE).

$\mathbf{R}(k)$  includes the antennas at transmit and receive sides, the transmit impulses and the multipath propagation on the radio channel as well. In general it is a sequence of matrices with respect to the symbol interval variable  $k$ . Because we assume here no interblock interference,  $k$  can be omitted and it is sufficient to consider a transmission of isolated vectors (or “blocks”). The model in Fig. 1 can then be described mathematically as follows:

$$\tilde{\mathbf{x}} = \mathbf{R} \cdot \mathbf{x} + \mathbf{n}_e,$$

$$\tilde{\mathbf{x}} = \underbrace{\mathbf{R}_d}_{\text{signal}} \cdot \mathbf{x} + \underbrace{\mathbf{R}_{\setminus d}}_{\text{interference}} \cdot \mathbf{x} + \underbrace{\mathbf{n}_e}_{\text{additive noise}},$$

$$\mathbf{R} = \underbrace{\mathbf{R}_d}_{\text{diagonal elements}} + \underbrace{\mathbf{R}_{\setminus d}}_{\text{non-diagonal elements}}. \quad (1)$$

We notice that the non-diagonal elements of  $\mathbf{R}$  lead to interference between the components of the transmitted vectors at the receive side. For more details, see [7].

The computational complexity of the optimum VE (i.e. maximum likelihood, ML) grows exponentially with  $N$ . Because this can result in an unrealistic number of operations per symbol vector, suboptimum schemes are commonly used. Our approach is to use a recurrent neural network (RNN).

The application of the RNN as vector equalizer has been discussed first in the context of multiuser detection for code division multiple access (CDMA) transmission systems [8], [9], [10], see also [11], [12]. It can be shown that this RNN tries to maximize the likelihood function of the optimum VE. In general it converges to a local maximum, but in many cases this local maximum turns out to be close to or identical with the global maximum, see e.g. [13]. The VE-RNN does not need a general training algorithm like backpropagation, the entries of  $\mathbf{R}$  can be measured and taken directly as weights of the RNN.

## 2.2. Continuous-Time RNN

The RNN discussed before is a discrete-time RNN, but for an analog circuit design we need a continuous-time RNN as basis. Also this type of RNN has been known for long. Its dynamical behavior can be described by a set of first order nonlinear differential equations as follows:

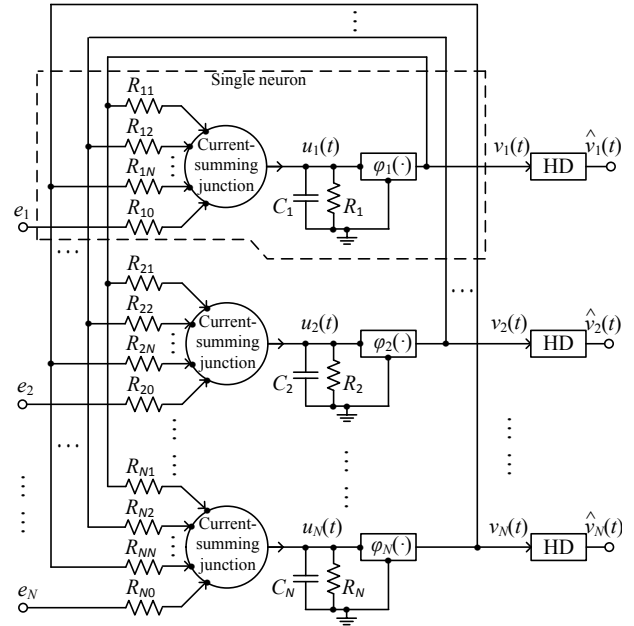


Fig. 2. Resistance-capacitance structure model of a real-valued, continuous-time recurrent neural network.

$$\mathbf{Y} \cdot \frac{d\mathbf{u}(t)}{dt} = -\mathbf{u}(t) + \mathbf{W} \cdot \mathbf{v}(t) + \mathbf{W}_0 \cdot \mathbf{e},$$

$$\mathbf{v}(t) = \boldsymbol{\varphi}[\mathbf{u}(t)]$$

$$= [\varphi_1[u_1(t)], \varphi_2[u_2(t)], \dots, \varphi_N[u_N(t)]]^T,$$

$$\hat{\mathbf{v}}(t) = \mathbf{HD}[\mathbf{v}(t)]. \quad (2)$$

The quantities are:

- $t$  is the continuous-time evolution time variable.
- $N$  is the number of neurons and also the length of all vectors.
- $\mathbf{Y}$  is a diagonal matrix with time constants  $\tau_j$  on its main diagonal.
- $\mathbf{W}$  is a  $(N \times N)$  weight matrix with entries  $w_{jj'}$ .
- $\mathbf{W}_0$  is a diagonal matrix with input weights on its main diagonal.
- $\mathbf{u}(t)$  is the state vector of length  $N$ .
- $\mathbf{v}(t)$  is the corresponding output vector.
- $\hat{\mathbf{v}}(t) = \mathbf{HD}[\mathbf{v}(t)]$  is the corresponding hard decision (HD) output vector.
- $\mathbf{e}$  is the external input vector.
- $\varphi_j(\cdot)$  is the  $j$ -th activation function.

Fig. 2 shows a resistance-capacitance structure model for a real-valued continuous-time RNN [1]. The stability of this RNN in the sense of Lyapunov has been intensively investigated, e.g. in [2]. For this case the weights in the equation above are related to the resistors in Fig. 2 by normalization as follows:  $w_{jj'} = \frac{R_j}{R_{jj'}}$  and  $w_{j0} = \frac{R_j}{R_{j0}}$ .

$\tau_j = R_j \cdot C_j$  is the time constant of the  $j$ -th neuron. To distinguish between resistors and channel matrix, the symbol for the channel matrix is bold.

### 2.3. Vector Equalization based on Continuous-Time RNN

The discrete-time RNN discussed in Section 2.1 for the application as VE is on a symbol basis. This means that the clock for the VE is  $kT_S$ , with  $k$  being the discrete time variable and  $T_S$  the symbol interval. To use a continuous-time RNN and the corresponding analog circuit as VE, we have to connect both types of RNNs. It can be shown that the following conditions must be fulfilled, see Eqs. (1) and (2):

$$\begin{aligned} \mathbf{e} = \tilde{\mathbf{x}}, \quad \hat{\mathbf{v}}(t_{equ}) = \tilde{\mathbf{x}}, \quad \varphi(u(t)) = \alpha \cdot \tanh(\beta \cdot u(t)), \\ \mathbf{W}_0 = \mathbf{R}_d^{-1}, \quad \mathbf{W} = \mathbf{I} - \mathbf{R}_d^{-1} \cdot \mathbf{R}, \quad \text{HD}(\cdot) = \text{sign}(\cdot). \end{aligned} \quad (3)$$

$\mathbf{I}$  is the identity matrix and  $t_{equ}$  the evolution time of the RNN, i.e. the time slot the RNN is granted to reach its equilibrium state. Given the symbol interval  $T_S$  for the digital transmission, the equalization must finish within  $T_S$ , i.e.  $t_{equ} \leq T_S$ .  $\alpha = 1$  V gives the dimension of Volts to the hyperbolic tangent, while  $\beta$  [1/V] is a positive variable which must be optimized for achieving best performance. Eq. (3) is for BPSK, but can be generalized, by combining the results of [2], [3], [4].

With (3) and the assumption that all time constants have the same values, i.e.  $\tau_1, \tau_2, \dots, \tau_N = \tau$ , Eq. (2) can be simulated on a digital computer by applying the first Euler method's update rule:

$$\begin{aligned} \mathbf{u}(l+1) &= \left\{1 - \frac{1}{\tau/\Delta t}\right\} \mathbf{u}(l) + \frac{1}{\tau/\Delta t} \{\mathbf{W} \cdot \mathbf{v}(l) + \mathbf{W}_0 \cdot \mathbf{e}\}, \\ \mathbf{v}(l) &= \varphi[\mathbf{u}(l)]. \end{aligned} \quad (4)$$

$l$  is now a discrete time variable and  $\Delta t$  the sampling step, which should be as small as possible. For our simulations we assume  $\tau/\Delta t = 10$ . Since the RNN is Lyapunov stable,  $\hat{\mathbf{v}}(t)$  reaches the equilibrium state after the evolution time, i.e. for  $t = t_{equ}$ .

## 3. CIRCUIT DESIGN

Potential implementations of an RNN cover a wide variety of solutions, from a discrete-time RNN implemented with field programmable gate arrays (FPGA) – as in [11] – to continuous-time analog hardware – as in [18] and [19]. Since here we focus on speed of operation and power efficiency, analog VLSI design and the continuous-time RNN will be the topic. The RNN is implemented with  $N = 4$  neurons, and realized in IHP 0.25  $\mu\text{m}$  SiGe BiCMOS technology (SG25H3).

The dynamic system of Eqs. (2) and (3) must fit the limited voltage swings that the analog circuit can handle. It is thus convenient to introduce a dimensionless scaling factor  $S$  and operate the following positions:

$$\mathbf{u}'(t) = S \cdot \mathbf{u}(t), \quad \mathbf{v}'(t) = S \cdot \mathbf{v}(t), \quad \mathbf{e}'(t) = S \cdot \mathbf{e}(t) \quad (5)$$

The scaled set of equations, describing the dynamical behavior of the analog RNN, can be written as:

$$\begin{aligned} \mathbf{Y} \cdot \frac{d\mathbf{u}'(t)}{dt} &= -\mathbf{u}'(t) + \mathbf{W} \cdot \mathbf{v}'(t) + \mathbf{W}_0 \cdot \mathbf{e}', \\ \mathbf{v}'(t) &= S \cdot \alpha \cdot \tanh\left(\frac{\beta \cdot \mathbf{u}'(t)}{S}\right) \end{aligned} \quad (6)$$

A simplified schematic of a neuron is shown in Fig. 3. For clarity, shaded boxes are detailed separately in Fig. 4. Bipolar junction transistors (BJTs) are assumed ideally matched, and base currents are neglected. The circuit is provided with an integrated MOSFET switch – Sequencer (Seq) in Fig. 3 – whose fundamental function is clarified in Sec. 3.2. The circuit is fully differential, and the following notation is used to denote differential currents and voltages:

$$\begin{aligned} I_{ji} &= I_{ji}^+ - I_{ji}^-, & I_{ji,w} &= I_{ji,w}^+ - I_{ji,w}^-, \\ I_j &= I_j^+ - I_j^-, & I_o &= I_o^+ - I_o^-, \\ u_j' &= u_j'^+ - u_j'^-, & e_j' &= e_j'^+ - e_j'^-. \end{aligned} \quad (7)$$

An additional remark is necessary: the time constant  $\tau$  (given by  $R_i$  and  $C_i$  in Fig. 2) is the basis for scaling of the evolution time  $t$ . To fully exploit the speed of the BJTs, a novel architecture with no external lumped capacitors is used. To provide a correspondence between the RNN circuit model and the resistance-capacitance structure model of Fig. 2, we make the assumption that the frequency response of the neuron can be described in terms of the resistances  $R$  and a fictitious capacitance  $C$  between nodes  $u_j^+$  and  $u_j^-$ . The validation of this hypothesis, both in a simulation environment and in a measurement setup on the real chip, is presented in Sec. 5.

### 3.1. RNN Behavioral model

Each neuron receives as inputs the feedback currents from all other neurons in the RNN.  $I_{ji}$  ( $i \in [1, \dots, N], i \neq j$ ) denotes the input feedback current reaching the  $j^{\text{th}}$  neuron, and coming from the  $i^{\text{th}}$  neuron. Using a four quadrant analog multiplier (Gilbert cell), each feedback current is multiplied by a weight  $w_{ji}$  in the range  $[-1, +1]$ . The value of  $w_{ji}$  is set to the corresponding entry of the channel matrix. The Gilbert cell is controlled by the voltage  $G_{ji}$  and a constant reference voltage  $V_{ref}$ . An attenuator – in the form of a common emitter amplifier with gain lower than

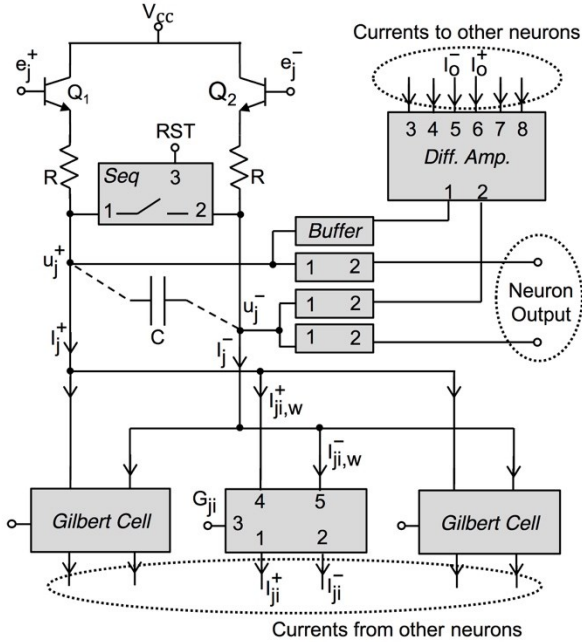


Fig. 3. Simplified schematic of a single neuron as part of a  $N = 4$  analog vector equalizer:  $u_j^+$  is the inner state,  $e_j^+$  the external input,  $G_{ji}$  the voltage for the weights' choice from the output of the  $i^{\text{th}}$  neuron to the input of the  $j^{\text{th}}$  neuron.

unity – allows each individual feedback current  $I_{ji}$  to be tuned with fine resolution:

$$\begin{aligned} \mathbf{w}_{ji} &= \mathbf{f}[\mathbf{G}_{ji}] \in [-1, +1], \\ \mathbf{I}_{j,w}(t) &= \mathbf{w}_{ji} \cdot \mathbf{I}_{ji}(t). \end{aligned}$$

Connecting the output branches of the Gilbert cells, the total weighted feedback current  $I_j$  for the  $j^{\text{th}}$  neuron is obtained by applying Kirchoff's current law:

$$I_j(t) = \sum_{\substack{i=1 \\ i \neq j}}^N w_{ji} \cdot I_{ji}(t) \quad (8)$$

Two voltage followers ( $Q_1$ - $Q_2$ ), biased by the same current  $I_j$  used for the summation of the feedback currents, create an additional differential voltage drop on nodes  $u_j^+$ - $u_j^-$ , proportional to the correspondent external input  $e_j^+$ . Considering the MOSFET switch in off state, defining  $\tau = R \cdot C$ , and using Eq. (8), the nodal analysis on nodes  $u_j^+$  and  $u_j^-$  gives:

$$\tau \cdot \frac{du_j^+(t)}{dt} = -u_j^+(t) - R \cdot \sum_{\substack{i=1 \\ i \neq j}}^N w_{ji} \cdot I_{ji}(t) + e_j^+ \quad (9)$$

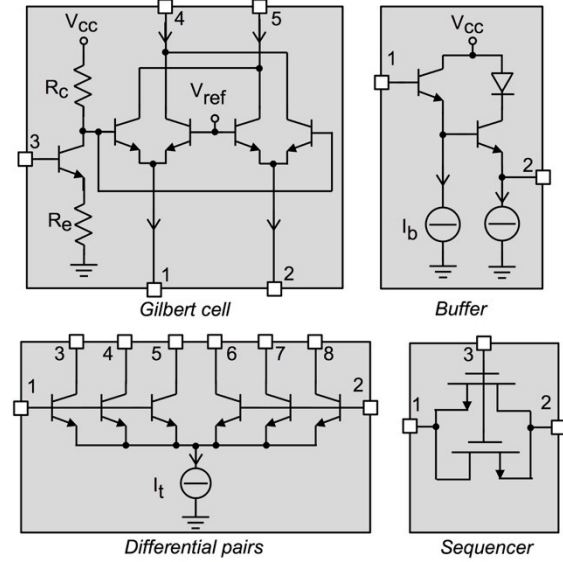


Fig. 4. Details of the circuit building blocks: (i) Gilbert cell used as four quadrant analog multiplier; (ii) Buffer stages; (iii) BJT differential pairs for the generation of the activation function  $\varphi(\cdot)$ ; (iv) MOSFET switch used as a sequencer.

The activation function  $\varphi(\cdot)$  is realized with  $(N - 1)$  replicas of differential transistor pairs, in which the tail current  $I_t$  is generated through a current mirror and is shared among the replicas. The output current  $I_o$  represents one copy out of the  $(N - 1)$  feedback currents that the  $j^{\text{th}}$  neuron will distribute to the other neurons in the network:

$$I_o(t) = -\frac{I_t}{(N - 1)} \cdot \tanh\left(\frac{u_j^+(t)}{2 \cdot V_t}\right) \quad (10)$$

Generalizing, a feedback current  $I_{ji}$  can be written as:

$$I_{ji}(t) = -\frac{I_t}{(N - 1)} \cdot \tanh\left(\frac{u_i^+(t)}{2 \cdot V_t}\right) \quad (11)$$

Substituting (11) in (9), the dynamics of the analog neural network can be finally written in vector form:

$$\begin{aligned} \mathbf{Y} \cdot \frac{d\mathbf{u}'(t)}{dt} &= -\mathbf{u}'(t) + \mathbf{W} \cdot \mathbf{v}'(t) + \mathbf{W}_0 \cdot \mathbf{e}', \\ \mathbf{v}'(t) &= \frac{R \cdot I_t}{N - 1} \cdot \tanh\left(\frac{\mathbf{u}'(t)}{2 \cdot V_t}\right) \end{aligned} \quad (12)$$

Two additional buffer stages output the differential voltage  $u_j'$ . According to Eq. (12), the sign of  $u_j'$  coincides with the sign of  $v_j'$ , and can thus be used to perform a hard decision at the end of an equalization.

Using the values provided in Table 1, Eq. (12) is linked to Eq. (5), with scaling factor  $S = 0.2$  and slope of the hyperbolic tangent  $\beta = 3.87$  1/V at the origin ( $\mathbf{u}' = \mathbf{0}$ ).

Table 1  
SUMMARY OF MAIN CIRCUIT PARAMETERS

Parameter	Value	Unit	Note
$R$	900	$\Omega$	Load resistor
$I_t$	665	$\mu\text{A}$	Tail current
$S$	0.2		Scaling factor
$\beta$	3.87	$\text{V}^{-1}$	Hyperbolic tangent's slope
$\tau$	42	ps	Equivalent time constant
$A_{chip}$	0.68	$\text{mm}^2$	Chip area
$A_{act}$	0.087	$\text{mm}^2$	Active area
$Cnt$	171		Transistor count ( $\propto N^2$ )
$W_{st,1}$	35	mW	VE-RNN power consumption

### 3.2. Time Evolution

The VE-RNN is a dynamical system, where the network evolves from an initial state (a saddle equilibrium point, represented by a null vector) to a stable state, following a non-monotonic trajectory in the state-space according to the set of first-order nonlinear differential equations in Eq. (12). As previously mentioned, at the output of the equalizer a hard decision is taken on the state vector  $\mathbf{u}'$ , which in the BPSK case is formulated as  $\text{sign}(\mathbf{u}')$ .

Given a sequence of input vectors  $\mathbf{e}'$ , Fig. 5 details how the VE reaches stability (and consequently when the output vector can be considered “valid”), and how it is possible to discard the memory of a previous equalization.

The evolution time  $t_e$  is defined as the time slot granted to the circuit, in order to reach a stable point. External inputs are applied only during this time slot. Before the next input is applied, it is crucial that the network returns – and stays pinned – to a predefined initial state.

A reset time  $t_{RST}$  can be defined as the time granted to the circuit to return to the initial state after a vector equalization. In our implementation the inner state  $\mathbf{u}'$  is forced to return to zero, since a different starting point would represent a biased vector, which is not equidistant from the  $2^N$  possible output vectors. From a circuitual point of view, this effect can be compared to a capacitor, which is not fully discharged at the beginning of the equalization, thus maintaining a “memory” of the previous equalization.

For best performance, i.e. highest throughput, both  $t_e$  and  $t_{RST}$  can be adjusted and minimized for each channel matrix. This is translated in the statistical optimization of the equalization time  $t_{e,min}$  and of the reset time  $t_{RST,min}$ , as shown in Sec. 4.

$RST$  is the reset signal, indicating if an equalization is running or if the circuit is resetting.  $RST$  acts on the gate

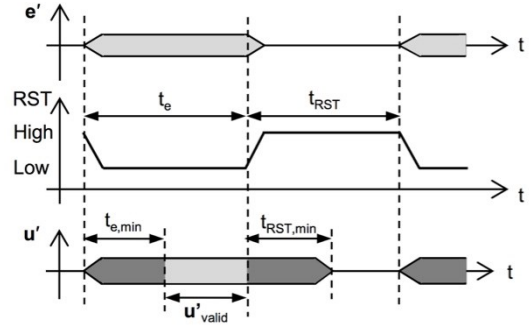


Fig. 5. Time domain evolution of an equalization.  $\mathbf{e}'$  is the scaled input vector,  $\mathbf{u}'$  the scaled state vector,  $RST$  the reset signal. Because of the iterative nature of the algorithm, the outputs are “valid” after a minimum equalization time. A minimum reset time is also necessary before a new equalization.

port of a MOSFET switch (Seq or Sequencer, in Fig. 3 and Fig. 4 respectively). When high,  $RST$  switches the two NMOS FETs into a low channel resistance state, short circuiting the differential internal state  $\mathbf{u}'$ . The width of the MOSFETs is chosen as a trade-off between the parasitic capacitance seen with the switch in off-state (to be minimized, since it strongly contributes to the increase of the equivalent  $\tau$ ), and the equivalent resistance seen in on-state (to be minimized, since it represents the “goodness” of the short circuit).

### 4. SIMULATION RESULTS

In this section two types of simulations of the continuous-time RNN equalizer are compared and shortly discussed: one represents Eq. (4) simulated in Matlab, and labeled in the following as “algorithm”. The second is a circuit-based simulation, performed in Keysight ADS, and labeled as “circuit”. Both are on general-purpose computers, the modulation is BPSK, and the number of neurons is four. Here results are presented for two channel matrices:

$$\mathbf{R}_m = \begin{bmatrix} 1 & +0.60 & +0.60 & +0.60 \\ +0.60 & 1 & +0.60 & +0.60 \\ +0.60 & +0.60 & 1 & +0.60 \\ +0.60 & +0.60 & +0.60 & 1 \end{bmatrix}$$

$$\mathbf{R}_h = \begin{bmatrix} 1 & +0.85 & +0.66 & -0.67 \\ +0.85 & 1 & +0.85 & -0.79 \\ +0.66 & +0.85 & 1 & -0.89 \\ -0.67 & -0.79 & -0.89 & 1 \end{bmatrix}$$

They are representative for channels with moderate ( $\mathbf{R}_m$ ) and high ( $\mathbf{R}_h$ ) crosstalk (interference between vector components), respectively. A pseudo-random sequence of symbol vectors was generated, multiplied with one of these matrices, and Gaussian noise vectors according to the  $E_b/N_0$  signal-to-noise ratio were added.  $E_b$  is the average

energy per bit. For the circuit simulations, all the applied signals have a rise/fall time of  $t_{r/f} = \tau/3$ .

Fig. 6 shows the good agreement of the bit error rate (BER) curves between the algorithm and the circuit simulation. Because the vector equalization based on RNNs is a suboptimum scheme, the Maximum Likelihood curves are also shown for reference.

Because of the iterative nature of the RNN algorithm, the BER curves are – additionally to  $E_b/N_0$  – functions of the evolution ( $t_e$ ) and reset ( $t_{RST}$ ) time. Given a channel matrix, a BER surface is obtained by sweeping the evolution and reset time, and keeping the signal-to-noise ratio constant, as shown in Fig. 7 for  $\mathbf{R}_h$  and  $E_b/N_0 = 18$  dB. Following this optimization procedure, and considering the region in which the BER performance becomes flat, values for the minimum equalization ( $t_{e,min}$ ) and reset ( $t_{RST,min}$ ) time can be found:

$$\begin{aligned} \mathbf{R}_m & : [t_{e,min}, t_{RST,min}] = [3.67, 1.33]\tau, \\ \mathbf{R}_h & : [t_{e,min}, t_{RST,min}] = [4, 2]\tau. \end{aligned}$$

$t_{equ} = t_{e,min} + t_{RST,min}$  is the total equalization time, i.e. the minimum relative time between two successive symbol vectors.  $t_{equ}$  must be equal or larger than the symbol interval  $T_S$  of the digital transmission. With the numbers from before and  $\tau = 42$  ps (see Sec. 5) we get  $T_S$  for the worst case channel  $\mathbf{R}_h$ :

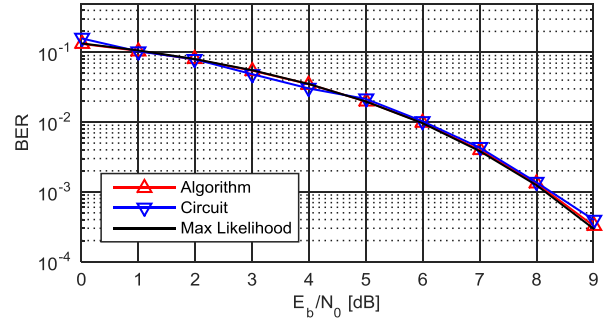
$$T_S = 42 \cdot (4 + 2) \text{ ps} = 252 \text{ ps},$$

corresponding to a throughput of four GSymbol/s (16 Gbit/s). For the BER simulations of Fig. 6, the minimum values for  $T_S$  were taken.

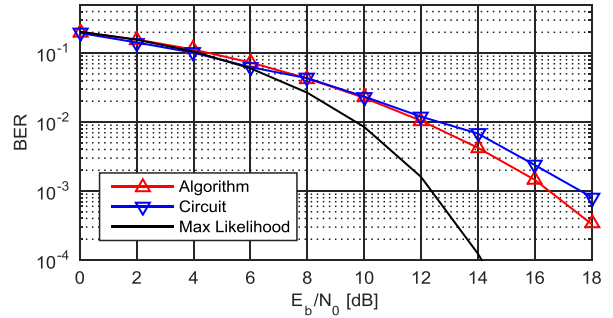
To compare the energy efficiency of the analog circuit with the one achievable with digital signal processing, we assume that one iteration of the algorithm includes eight floating point operations (three multiplications, four sums and one hyperbolic tangent computation). Relying on our Matlab simulations, ten iterations/neuron are sufficient to equalize a vector, resulting in an algorithmic complexity of 320 floating point operations per symbol. Given the power consumption of the analog vector equalizer (35 mW, measured on a real chip, see Fig. 11 in Sec. 5), the energy efficiency can be computed:

$$\text{Energy efficiency} = \frac{320 \frac{\text{Fl. Op.}}{\text{Symbol}} \cdot 4 \frac{\text{GSymbol}}{\text{s}}}{35 \text{ mW}} \approx 36 \frac{\text{TFlops}}{\text{W}}$$

With a value of five GFlops/W for today's most efficient heterogeneous supercomputer, we can conclude that our dedicated hardware shows an efficiency



(a) BER vs.  $E_b/N_0$  for moderate crosstalk  $\mathbf{R}_m$



(b) BER vs.  $E_b/N_0$  for high crosstalk  $\mathbf{R}_h$

Fig. 6. BER evaluation of the continuous-time RNN equalizer, including circuit and algorithm simulations. Maximum likelihood algorithm showed for reference.

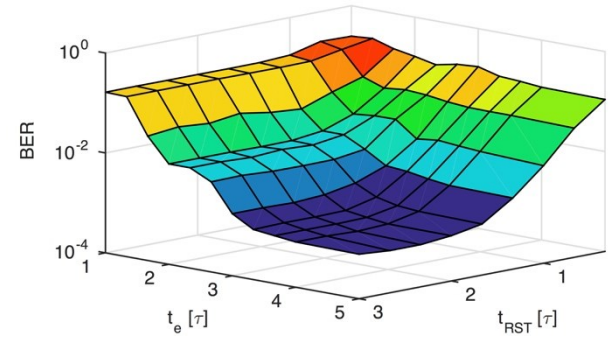


Fig. 7.  $\mathbf{R}_h$  BER surface for different evaluation and reset times, and constant signal-to-noise ratio ( $E_b/N_0 = 18$  dB). Flat performance indicates that (i) the circuit reaches a proper stable equilibrium point, and (ii) does not possess memory of previous equalization.  $[t_{e,min}, t_{RST,min}] = [4, 2]\tau$ .

improvement between three and four orders of magnitude over digital implementations.

## 5. MEASUREMENT RESULTS

Our first measurements focused on the functional validation of the single neuron: weighted multiplication,  $\beta$  of the activation function, and cutoff frequency, i.e. the equivalent time constant  $\tau$ . For this purpose the circuit of Fig. 8 was



realized and measured. The test structure was bonded and mounted on a Rogers RO4003 printed circuit board (PCB).

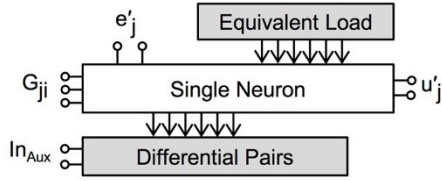


Fig. 8. Single neuron characterization: realized test structure, including the single neuron (cf. Fig. 3), together with additional circuitry (shaded blocks) to facilitate the measurements.  $e'_j$  is the external input,  $G_{ji}$  the voltage for the weights' choice,  $u'_j$  the inner state, and  $In_{aux}$  and auxiliary differential input to generate the feedback currents to the neuron.

The feedback currents coming from other neurons  $I_{ji}$  ( $i \in [1, \dots, N], i \neq j$ ) are here generated by an additional block of differential pairs – identical to Fig. 4 – controlled by the auxiliary differential input  $In_{aux}$ . Provided that the neuron under test also drives an identical load as in the full vector equalizer, the characterization of this elementary cell remains valid at system level.

Fig. 9 shows the gain variation  $w_{ji}$  as a function of the voltage  $G_{ji}$  to apply to the Gilbert cell, as measured at a frequency  $f = 0.1$  GHz. The attenuator – cf. Fig. 4 – allows the weights to be fine-tuned, within a span of 1.2 V. The measured curve presents a shift  $\Delta G_{ji} \approx 0.1$  V with respect to simulations. This shift can however be easily calibrated in the measurement setup and has not any impact on neuron's performance.

The slope of the activation function  $\beta$  at the origin – cf. Eqs. (3) and (6) – is a free parameter that can be optimized. From our simulations, the condition to fulfill for best performance is  $\beta \geq 3 \text{ V}^{-1}$ . Measurements performed at  $f = 0.1$  GHz resulted in a value of 3.47, slightly smaller than the simulated one  $\beta = 3.87 \text{ V}^{-1}$ . Reasons can probably be imputed to small losses in the measurement setup.

The equivalent  $\tau$  for time scaling can be measured by applying a sinusoidal excitation to the external input  $e'_j$  and measuring the frequency response at the neuron output  $u'_j$ . Fig. 10 (a) shows the simulated transfer function  $|u'_j/e'_j|$  and a comparison with an ideal RC low pass filter with cutoff frequency  $f = 3.79$  GHz ( $\tau = 42$  ps). The hypothesis of a frequency response which resembles an ideal RC behavior is confirmed by Fig. 10 (b), showing the single-input single-output  $|u'_j/e'_j|$  measurement and the comparison with the expected curve.

Having the single neuron been validated by measurement data, a full vector equalizer has been fabricated (Fig. 11). The chip area of  $0.68 \text{ mm}^2$  is dominated by the several pads needed for measurements (four

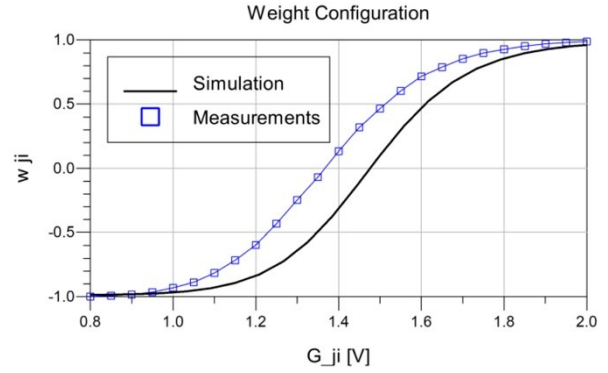
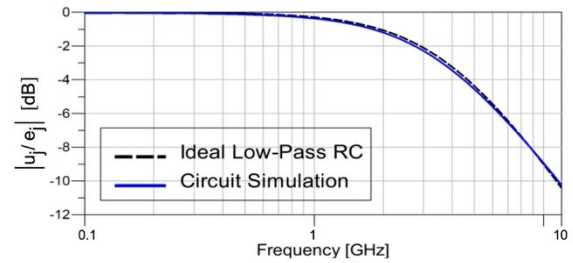
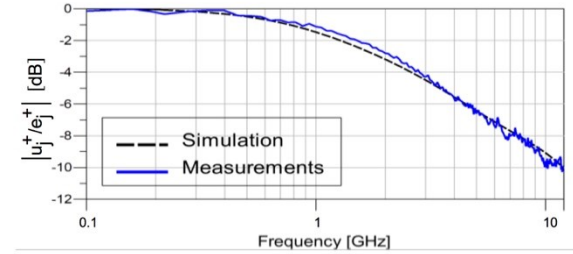


Fig. 9. Voltage mapping  $w_{ji} = f[G_{ji}] \in [-1, 1]$



(a)  $|u'_j/e'_j|$  frequency response simulation, and comparison with an ideal low pass RC filter with  $\tau = 42$  ps.



(b) Single input, single output  $|u'_j/e'_j|$  frequency response measurement, and comparison with simulation results.

Fig. 10. Equivalent  $\tau$  of a single neuron. Measurements confirm the hypothesis of a first-order low-pass filter comparable to an ideal low-pass RC filter, lumped between  $u'_j$  and  $u'_j$ .

differential inputs, four differential outputs, six weight controls, supplies and grounds).

The active area is approximately  $0.09 \text{ mm}^2$ , with a transistor count  $C = 171$  for four neurons. The power consumption of 35 mW was measured, confirming simulation results. The equalizer was also tested with a predefined set of input-output vectors, and always provided the correct steady-state solution. The current research aims at developing a setup environment to perform a full-speed vector equalization, while observing the evolution of the outputs at real time.

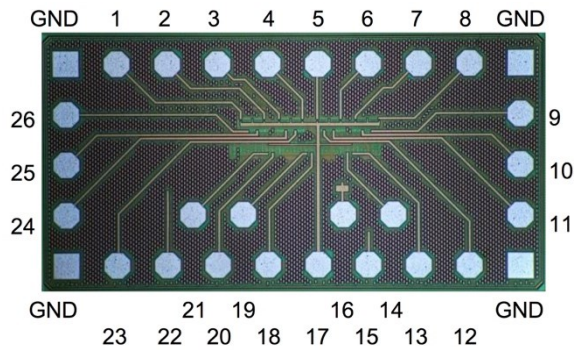


Fig. 11. Chip of the vector equalizer. Pin configuration: differential external inputs [1,2,3,4,5,6,7,8]; differential outputs [9,10,11,12,23,24,25,26]; weights configuration [13,14,18,19,20,21]; reset [15]; voltage supplies [16,17,22].

## 6. CONCLUSIONS

We presented an example for nonlinear signal processing with analog circuits: a four-neuron RNN vector equalizer for MIMO transmissions, realized in SiGe BiCMOS technology.

Bit error rate performance comparisons showed virtually the same or similar behavior for the common digital signal processing and the analog VLSI circuit version. The reason for the comparable robustness – the input is noisy – is that both types of processing use equilibrium states of nonlinear dynamical systems to get the outputs, rather than simple amplitude levels.

The throughput of the vector equalizer is influenced by the evolution time the analog RNN needs to reach the equilibrium state. This time in its turn depends on the equivalent time constant  $\tau$ , which in our circuit design was minimized by taking advantage of the parasitic capacitances of bipolar transistors and MOSFETs. Furthermore, an on-chip switch gives the possibility to reset the internal states of the equalizer – a fundamental prerequisite to handle sequences of vectors.

Our intention is to revitalize the topic of “analog-assisted digital”, i.e. implementing algorithms with dedicated analog circuits. In comparison with common digital signal processing, we could show that the energy efficiency is improved by three to four orders of magnitude. This confirms earlier conjectures stating a huge potential for nonlinear signal processing with analog circuits.

## 7. ACKNOWLEDGMENTS

Financial support by the Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged. A note of thanks goes also to IHP GmbH for the Si/SiGe foundry processes needed to realize the circuit.

## 8. REFERENCES

- [1] CompuGreen, LLC. (2014) The Green500 List. [Online]. [www.green500.org](http://www.green500.org).
- [2] Prometheus GmbH. (2014) The Top500 List. [Online]. [www.top500.org](http://www.top500.org).
- [3] H.-A. Loeliger, "Decoding in analog VLSI," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 99-101, April 1999.
- [4] S. Draghici, "Neural Networks in Analog Hardware - Design and Implementation Issues," *International Journal of Neural Systems*, vol. 10, no. 1, pp. 19-42, February 2000.
- [5] H. Markram, "The Human Brain Project - A Report to the European Commission," The HBP-PS Consortium, 2012.
- [6] C. Mead, *Analog VLSI and neural systems*.: Addison-Wesley, 1989.
- [7] J. Lindner, "MC-CDMA in the context of general multiuser/multisubchannel transmission methods," *European Transactions on Telecommunications*, vol. 10, no. 4, pp. 351-367, July/August 1999.
- [8] G. I. Kechriotis and E. S. Manolakos, "Hopfield neural network implementation for optimum CDMA multiuser detector," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 131-141, January 1996.
- [9] W. G. Teich and M. Seidl, "Code division multiple access communications: multiuser detection based on a recurrent neural network structure," *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications*, vol. 3, pp. 979-984, 1996.
- [10] T. Miyajima, T. Hasegawa, and M. Haneishi, "On the multiuser detection using a neural network in code-division multiple-access communications," *IEICE Trans. on Communications*, vol. E76, no. B, pp. 961-968, 1993.
- [11] W. G. Teich, A. Engelhart, W. Schlecker, R. Gessler, and H. J. Pfliederer, "Towards an efficient hardware implementation of recurrent neural network based multiuser detection," in *IEEE 6th International Symposium on Spread Spectrum Techniques and Applications*, NJIT, New Jersey, USA, 2000, pp. 662-665.
- [12] A. Engelhart, "Vector detection techniques with moderate complexity," Ulm University, Institute of Information Technology, PhD Thesis 2003.
- [13] A. Engelhart et al., "A Survey of Multiuser/Multisubchannel Detection Schemes Based on Recurrent Neural Network," *Wireless Communications and Mobile Computing, Special Issue on Advances in 3G Wireless Networks*, vol. 2, no. 3, pp. 269-284, May 2002.
- [14] S. Haykin, *Neural networks: A comprehensive foundation*. USA: Macmillan college publishing company, Inc., 1994.
- [15] Y. Kuroe, N. Hashimoto, and T. Mori, "On Energy Function for Complex-Valued Neural Networks and its Applications," in *Proc. of the 9th international conference on neural information processing ICONIP'02*, vol. 13, 2002, pp. 1079-1083.
- [16] M. Mostafa, W. G. Teich, and J. Lindner, "Vector equalization based on continuous-time recurrent neural networks," in *6th IEEE International Conference on Signal Processing and Communication Systems*, Gold Coast, Australia, December 2012, pp. 1-7.
- [17] M. Mostafa, W. G. Teich, and J. Lindner, "Approximation of activation functions for vector equalization based on recurrent neural networks," in *6th International Symposium on Turbo*

*Codes and Iterative Information Processing*, Bremen, Germany, 2014.

- [18] G. Cauwenberghs, "An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory," *IEEE Transactions on Neural Network*, vol. 2, pp. 346-361, March 1996.

- [19] G. Kothapalli, "An analogue recurrent neural network for trajectory learning and other industrial applications," in *3rd IEEE international Conference on Industrial Informatics (INDIN)*, Perth, Western Australia, 2005, pp. 462-466.

## ADOPTING WINNF TRANSCEIVER FACILITY FOR SPECTRUM SENSORS

Tomaž Šolc (“Jožef Stefan” Institute, Ljubljana, Slovenia; tomaz.solc@ijs.si)

### ABSTRACT

An implementation of the Wireless Innovation Forum Transceiver Facility was developed for VESNA SNE-ESHTER, a specialized spectrum sensor deployed in wireless testbeds within the FP7 CREW project. The goal was to simplify experimentation and portability of experiments. A C++ library was developed that runs on the host PC and implements an event-based scheduler and an asynchronous interface conforming to the Transceiver Facility specification. The library communicates with the sensor over a serial line and does not require modification of the spectrum sensor’s firmware. We present the challenges encountered and show results of some latency benchmarks of this Transceiver Facility implementation. Finally, we provide some suggestions on how the Transceiver Facility could be improved in future versions to better support such hardware.

### 1. INTRODUCTION

#### 1.1. Transceiver Facility

In a software defined radio architecture, all waveform processing tasks are implemented in software. To make software portable between different transceiver hardware, a standardized programming interface is desired. The Wireless Innovation Forum Transceiver Facility [1] is an effort to develop such a standardized programming interface. The version 1.0 of the specification is available on-line.

Transceiver Facility describes in detail a modular interface that allows the software to control the radio hardware, the hardware to describe itself to the software and a streaming interface for passing digital baseband data between hardware and software. Hardware control functions of the Transceiver Facility include radio-frequency (RF) front-end control (for example, central frequency for up- and down- conversion, gain, channel filter, etc.), analog-to-digital and digital-to-analog conversion details (for example, sampling rate, etc.). An event-based mechanism is provided for accurate time synchronization between hardware and software. While the Transceiver Facility specification is language-agnostic, it includes reference examples of the interface in C++ and VHDL.

#### 1.2. FP7 CREW project

The FP7 CREW project [2] developed a federation of wireless testbeds. CREW testbeds allow experimentation in diverse environments, technologies and frequency bands. A testbed consists of a number of remotely controlled computing nodes with attached radio hardware. Environments range from RF shielded rooms to out-door deployments. For example, nodes in the LOG-a-TEC testbed [3] are mounted on street lights in several urban areas. An experimenter develops their application, uploads it to one or more nodes in a testbed and performs measurements using testbed instrumentation.

Radio hardware in CREW testbeds can be roughly grouped into: a) SDR front-ends like the Ettus Research USRP in combination with high-performance general-purpose computers, b) narrow-band radios mounted on low-power wireless sensor nodes and c) specialized spectrum sensing devices like the VESNA SNE-ESHTER [4] and Imec Sensing Engine [5].

Each of these devices typically provides its own programming interface. One of the goals of the FP7 CREW project was to develop a common programming interface across the federation. A common interface to testbed hardware simplifies application development for experimenters and enables easy portability of experiments from one testbed to another. Early in the project, it has been decided to adopt the Transceiver Facility as the common interface to SDR nodes in testbeds [6].

To develop further the concept of a unified interface, we have decided to also adopt the Transceiver Facility for other categories of radio hardware in our testbeds. Adopting the Transceiver Facility for sensor node radios seemed impractical. On the other hand, adopting the Transceiver Facility for our specialized spectrum sensing hardware seemed feasible. Still, this posed several challenges. Spectrum sensors are specialized devices that differ from general-purpose SDR frontends in some significant ways. For example, they are receive-only devices with on-board signal processing. They are optimized for continuous scanning of a radio-frequency band and typically report only statistical data to the host PC. They are typically incapable of providing an uninterrupted stream of unprocessed baseband samples due to bitrate restrictions in various parts of the system.

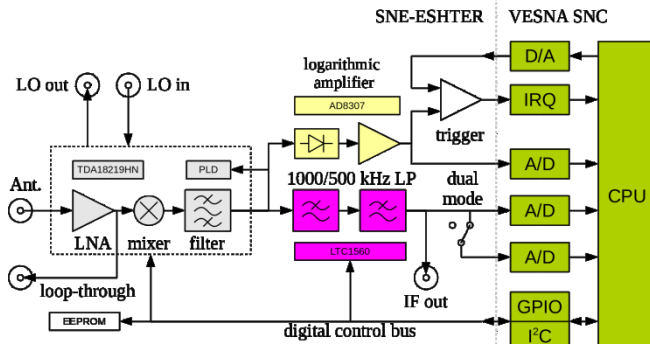


Figure 1: Block diagram of the SNE-ESHTER receiver, showing analog front-end and the interface to the VESNA sensor node core.

The rest of this paper is structured as follows: In section 2 we introduce VESNA SNE-ESHTER spectrum sensor we targeted in this extension of the Transceiver Facility support in the FP7 CREW project. In section 3 we describe the software architecture behind the current implementation of the Transceiver Facility. In section 4 we show some latency benchmarks of the implementation. In section 5 we comment on the possible future improvements. Finally, we conclude the paper in section 6.

## 2. VESNA SNE-ESHTER

VESNA SNE-ESHTER is a low-cost, compact spectrum sensor for the VHF and UHF frequency range that was developed at the Jožef Stefan Institute. It is based on VESNA [7], a low-power sensor node core. These devices are deployed in the LOG-a-TEC testbed. A VESNA SNE-ESHTER setup typically consists of three parts: the SNE-ESHTER analog front-end, the VESNA sensor node core (SNC) and a host PC running a GNU/Linux operating system.

### 2.1. Analog Front-End

The analog front-end contains the radio frequency analog electronic circuit that performs the frequency down-conversion and signal conditioning before analog-to-digital conversion. A simplified block diagram is presented in Figure 1. The front-end is a custom designed single-conversion, low-IF receiver based on the NXP TDA18219HN integrated circuit. The receiver has a specified input frequency range between 42 MHz and 870 MHz. The local oscillator (LO) signal is generated by a FRAC-N phase-locked loop (PLL) and has a typical settling time of 5 ms on channel change.

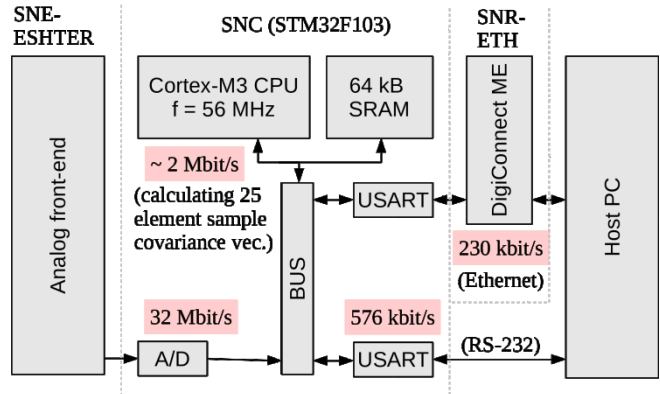


Figure 2: Block diagram showing different bottlenecks in passing the data from the front-end to the host PC

The RF signal from the antenna is amplified in a low-noise amplifier (LNA) and mixed with the LO in an image-rejection mixer to produce a signal at an intermediate frequency (IF). Several stages of automatic gain control are used to minimize non-linear distortion and maximize signal-to-noise ratio of the signal. The signal passes through one tracking RF and two IF band-pass filters with software-selectable bandwidth. The final stage is a 10th order elliptic anti-aliasing filter with two settings: 500 kHz and 1000 kHz, corresponding to 1 Msample/s and 2 Msample/s sampling rates. After the anti-aliasing filter, the signal is routed to the VESNA sensor node core to be sampled by an analog-to-digital converter (ADC).

In addition to the main signal path, the SNE-ESHTER front-end board also includes additional analog energy detection blocks. Two logarithmic signal level detectors can be calibrated for accurate measurement of absolute signal power. An analog trigger circuit can also provide an interrupt to the CPU when the signal level in the tuned channel reaches a defined threshold. These functionalities are currently unused when SNE-ESHTER is used through the Transceiver Facility.

The SNE-ESHTER design allows for hosting two analog front-end boards on a single sensor node core. This allows simultaneous sensing of two different frequency channels or reception on a single channel using two antennas. Using multiple front-ends is not supported in the current implementation of the Transceiver Facility.

### 2.2. VESNA Sensor Node Core

VESNA SNC contains an integrated microcontroller with an ARM Cortex M3 CPU core with a 56 MHz clock and 64 KB of SRAM. The SNC also contains a RS-232 interface to the host PC with a 576 kbit/s maximum bitrate. An optional Ethernet interface can be installed in case the sensor is remotely installed.

1 -> select channel 650000:1:650001 config 0,2	# Host PC instructs the sensor to tune to # 700 MHz and hardware configuration 2 # (defining filter bandwidth and sampling rate)
2 <- ok	# Sensor confirms the command
3 -> samples 1024	# Host PC sets sampling buffer length to 1024
4 <- ok	# Sensor confirms the command
5 -> sample on	# Host PC instructs the sensor to start sensing
6 <- TS 0.001 CH 650000 DS 2042 2045 2053 ...	# Sensor sends first full sampling buffer # containing 1024 samples and a timestamp.
7 <- TS 0.136 CH 650000 DS 2053 2056 2042 ...	# Sensor continues to send reports until
8 <- ...	# commanded to stop
9 -> sample off	# Host PC instructs the sensor to stop sensing
10 <- ok	# Sensor confirms the command

Figure 3: Example conversation between the SNE-ESHTER spectrum sensor and the host PC using the native serial protocol.

The SNC includes three 12-bit successive approximation ADCs with up to 2 Msample/s sample rates. ADCs are driven by a DMA controller and store samples directly into a sample buffer in SRAM without any intervention from the CPU. The sample buffer has space for up to 25000 samples (up to 12.5 ms at 2 Msample/s sample rate). Collected signal samples are read by the CPU. They can be either processed on-board or sent in a raw form over the RS-232 or Ethernet interface to the host PC. For example, the firmware currently implements calculating a sample covariance vector on-board. This significantly reduces the amount of data that needs to be sent from the sensor when sensing algorithms like covariance or Eigenvalue detection are employed.

The CPU is unable to either process or forward the samples to the host PC at the rates provided by the ADC. Different bottlenecks preventing this are shown in Figure 2. Hence the device typically operates in a sample-process-report cycle which includes significant blind time. For simple operations, like the covariance vector calculation, the signal processing capability of the CPU exceeds the bandwidth of interfaces to the host PC. Therefore, using on-board processing typically reduces the blind time.

The native serial interface between the firmware running on the VESNA SNE-ESHTER CPU and the host PC is an ASCII based protocol. In this interface, details of ADC and most analog front-end settings, like filter and AGC settings, are abstracted in the form of a low number of discrete hardware configurations identified by numerical identifiers. Configurations 2 (2 MHz sampling frequency/ 1 MHz anti-aliasing filter bandwidth) and 3 (1 MHz sampling frequency/500 kHz anti-aliasing filter bandwidth) allow for sampling of the IF signal and are the only two hardware configurations currently used by the Transceiver Facility implementation.

Figure 3 shows an example of a serial line session that includes all native commands, relevant for the current implementation of the Transceiver Facility interface. It can

be seen that this interface itself does not allow for accurate scheduling of receive start and stop time or synchronization of the signal samples. It does however provide information on relative timing of individual samplings based on the sensing start time, which is derived from the quartz oscillator on the SNC. This provides information on how much of the signal has been lost. For example, the interval between two sample buffers sent to the host PC on lines 6 and 7 in Figure 3 is 135.0 ms, while a buffer with its 1024 samples sampled at 2 Msample/s only covers 0.5 ms of that time.

### 3. IMPLEMENTATION

We implemented the Receive Channel of the Transceiver Facility for SNE-ESHTER in the form of a C++ library, targeting the GNU/Linux operating system. The library exports the same user-facing interface as other Transceiver Facility implementations used in FP7 CREW (similar libraries exist for USRP devices and the Imec Sensing Engine). An experimenter that wants to use a receiver for spectrum sensing through the Transceiver Facility writes a C++ program and links it against one of these libraries, depending on which receiver they want to use. Except for some initialization parameters, the experimenter's code does not need any adaptations when switching from one receiver hardware to the other.

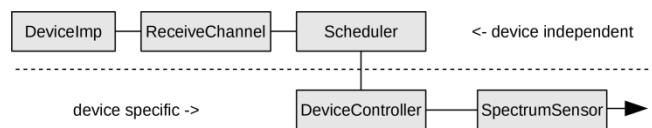


Figure 4: Main classes in SNE-ESHTER Transceiver Facility implementation

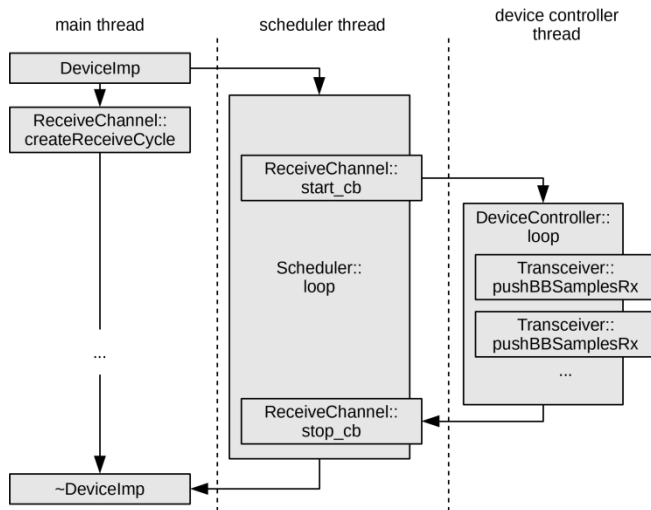


Figure 5: A diagram of threads of execution and their most important method calls in the SNE-ESHTER Transceiver Facility.

Experimenter’s code using the Transceiver Facility interface runs on the host PC (and not on the CPU in the SNE-ESHTER device itself). This provides the benefit of not needing to adapt the device’s firmware for each experiment, while on the other hand bounds the Transceiver Facility to the limitations of the serial interface. This same approach has been used in all other Transceiver Facility implementations in the CREW project.

Our library consists of 5 main object classes that are shown in Figure 4 and can be divided into two parts: a) a hardware independent event scheduler and the user-facing Transceiver Facility interface and b) the adaptor for the serial interface to the device. The library uses three threads of execution, illustrated in Figure 5. The source code is available on-line at <https://github.com/sensorlab/xcvr-eshter>.

### 3.1. Scheduler

The main task of the event scheduler is to translate between the asynchronous, event-based interface specified by the Transceiver Facility and the synchronous native serial interface to the spectrum sensor.

The user controls the Transceiver through the *DeviceImp* (“device implementation”) class. This class contains the *ReceiveChannel* object that conforms to the Transceiver Facility specification and forms the user-facing part of the library. *TransmitChannel* is unimplemented, as the sensor is a receive-only device.

Construction of a *DeviceImp* object instance is the only step that is device specific. The user must supply the constructor with an instance of the *SpectrumSensor* object. *SpectrumSensor* constructor necessarily requires knowledge of the underlying hardware. For VESNA SNE-ESHTER

device, the constructor requires a path to the Unix device file (e.g. “/dev/ttyUSB0”) that is used to communicate with the sensor.

Most of the user’s interaction with the transceiver happens through the *createReceiveCycleProfile()* method of the *ReceiveChannel* object. It allows the user to specify when the sensor starts and stops recording signal samples and other details of reception. *createReceiveCycleProfile()* schedules *ReceiveStartTime* and *ReceiveStopTime* events with the scheduler. In case *undefinedDiscriminator* has been used for *requestedReceiveStopTime*, the *ReceiveStopTime* event remains unscheduled. This allows for signal reception of undefined length. In that case, the *setReceiveStopTime()* method can be used to schedule the *ReceiveStopTime* event at a later time.

In the current implementation, *setReceiveStopTime()* cannot be used once a *ReceiveStopTime* event has been scheduled, as that would require cancelling an existing event. This operation is not currently supported by the underlying scheduler. For a similar reason, *configureReceiveCycle()* method has not been implemented.

The *Scheduler* class is hidden from the user of the library. It performs all asynchronous event scheduling using an event loop in a separate thread. The event loop uses the Boost.Asio library [8] using the system clock of the host PC as the reference. All discriminators, including the *eventBased* discriminator, have been implemented.

The *eventBased* discriminator supports selection of event count origin, event count and time offset. However, only up to one event in the past can be used as the origin. This for example, allows the use of *eventCountOrigin=Previous*, *eventCount=0* setting, which is a common pattern. On the other hand, the Transceiver Facility specification appears to allow for selecting an arbitrary past event as the reference for the *eventBased* discriminator. Implementing this functionality would require the scheduler to keep a log of timestamps for all past events. This was considered an unnecessary complication considering the limited use of such discriminators.

### 3.2. Device controller

The only hardware specific parts of the code are the *DeviceController* and *SpectrumSensor* classes.

*DeviceController* implements a thin asynchronous wrapper around the device-specific *SpectrumSensor* class. It provides only two methods: *start()* and *stop()*. These two methods start and stop *DeviceController*’s thread which runs its own event loop. *DeviceController* configures the hardware through the *SpectrumSensor* class before starting the reception. *DeviceController*’s event loop calls back to the user’s code through the *pushBBSamplesRx()* method every time the sensor sends a packet of signal samples to the host PC. Hence the *pushBBSamplesRx()* is called

asynchronously from the perspective of the library user's code. *start()* and *stop()* methods are called from *ReceiveStart* and *ReceiveStop* event callbacks that were scheduled by the user's initial call to *createReceiveCycle()*.

Transceiver facility specifies that complex signal samples are pushed by the transceiver to the user code in the *BBPacket.packet* structure. However SNE-ESHTER is using low-IF sampling and provides only real-valued samples. To accommodate for that, *DeviceController* writes the actual signal samples in I field and fills Q field in *BBPacket.packet* structure with zeros.

The *SpectrumSensor* class provides a synchronous interface to the native serial ASCII protocol. This class is a straightforward C++ port of the Python *SpectrumSensor* class that is usually used to control SNE-ESHTER from a host PC [9]. C++ Serial library [10] was used, since it provides a similar interface to the Python Serial library and simplified the porting procedure. Numerical tuning profile identifiers from the Transceiver Facility are directly translated into SNE-ESHTER hardware configurations that are passed to the device over the native serial protocol. Hence only tuning profiles 2 and 3 can currently be used. *PacketSize* parameter on the Transceiver Facility side is directly used as the sample buffer length in the native serial protocol.

### 3.3. Test driven development

A test-driven development methodology [11] was employed when developing the Transceiver Facility implementation. Individual components were designed with minimal implicit dependencies to enable testing of each component separately. Whenever possible, dependencies between classes are injected explicitly through constructor parameters. This is, for example, the reason why *DeviceImp* constructor requires the user to provide a *SpectrumSensor* instance instead of the *SpectrumSensor* object being instantiated by the constructor implicitly. Tests were developed using the *cpputest* framework [12].

The component that benefited most from test driven development was the *Scheduler* class. Transceiver Facility specifies a relatively broad range of possibilities for event scheduling. Test driven development proved to be a very efficient way of deriving a reliable implementation.

Where dependency on other classes could not be avoided in tests, mock classes were created and used instead of real classes via dependency injection. For example, several tests of the *DeviceImp* class use a *SpectrumSensor* implementation that does not communicate with hardware. This approach simplified development, since it was possible to develop software without having a sensing device constantly connected. It excluded the possibility that a failed test would be caused by a malfunctioning device or a bug in

the device's firmware. Tests with a mocked sensor were also faster to execute.

After individual components were developed, a suite of system tests was also created that tested the whole Transceiver Facility implementation. This suite uses the same *cpputest* framework, but is compiled separately. This allows the developer to choose between running tests that do not require a connected sensor and tests that do.

## 4. BENCHMARKS

The foremost concern with our implementation of the Transceiver Facility was the latency between the event scheduler and the sensing device. The latency is the time difference between when the user scheduled the signal reception to start and when the signal reception actually occurred. The Transceiver Facility specifies nanosecond precision. However such level of precision is optimistic even for USRP devices, which are much lower latency devices than our sensors.

To measure the latency when using Transceiver Facility with SNE-ESHTER, an instrumented version of the Transceiver Facility library was developed that logged function calls and serial line traffic together with current host PC system time. Additionally, a digital storage oscilloscope was setup to capture the waveform on the RS-232 line connecting the host PC and the SNE-ESHTER sensor. This setup was then used to observe the series of events that happen when a user of the Transceiver Facility calls the *createReceiveCycle()* method on the *ReceiveChain* object. *requestedReceiveStartTime* was set to immediate. SNE-ESHTER was configured to use 2 Msample/s sample rate and 2048 samples per packet.

A measurement result is shown in Figure 6 in form of an oscilloscope screenshot. Data from the instrumented C++ library is shown overlaid in form of text annotations. It can be seen that the first *pushBBPacketRx()* callback from the Transceiver Facility to the user's code happens 570 ms after the *createReceiveCycle()* call and the *ReceiveStart* scheduler event. Further *pushBBPacketRX()* calls follow each 270 ms. For each *pushBBPacketRX()* call, 10264 bytes of data have been transferred over the serial line.

Configuration parameters are sent to the device immediately after the *createReceiveCycle()* call. This involves two round-trips on the serial line and concludes with the »sample on« command. This measurement shows that this contributes a negligible amount to the overall latency.

Channel sampling time also takes a very small part of the interval between two *pushBBPacketRX()* calls:

$$t_{\text{sampling}} = \frac{2048 \text{ samples}}{2 \frac{\text{Msample}}{\text{s}}} \cong 1 \text{ ms}$$



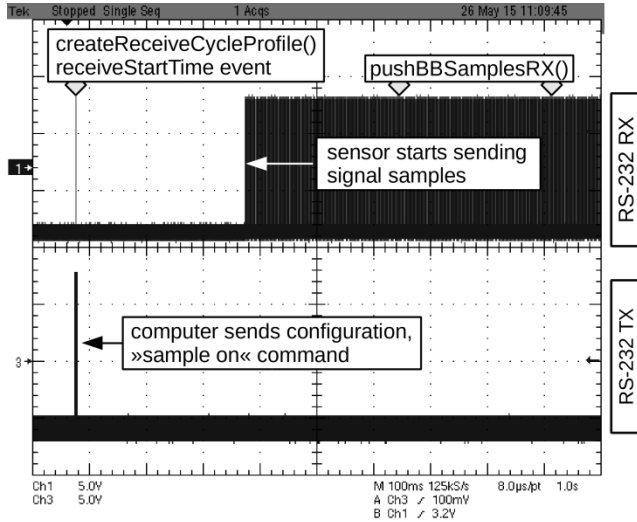


Figure 6: Timeline of events following a `createReceiveCycle()` call

Our experiment didn't provide information on when exactly the signal sampling occurred on the SNE-ESHTER device. From the firmware running on the device, we can infer that first sampling happened immediately before the sensor started sending signal samples over the serial line.

Approximately half of the time interval (270 ms) before the first `pushBBPacketRX()` call is taken by the transmission of the packet of signal samples over the serial interface ( $t_{transmission}$ ). The throughput is less than the theoretical 576 kbps of the serial line (RS-232 line uses 1 start bit and 1 stop bit, hence 10 transferred bits per byte):

$$B = \frac{10264 \text{ bytes} \cdot 10 \frac{\text{bits}}{\text{byte}}}{270 \text{ ms}} \cong 380 \text{ kbps}$$

This result suggests that even the throughput of unprocessed signal samples is in part limited by the CPU on the sensor. It also shows that the latency is heavily dependent on the size of the sampling buffer. Doubling the size of the sample buffer would double the time spent in transmission.

The rest of the time between the “sample on” command and first signal samples appearing on the serial line is spent in power up sequence for the analog front-end ( $t_{start}$ ).

Since the information on the actual instant of signal sampling is not available to the user of Transceiver Facility, we can define worst-case latency for the purpose of this benchmark as the time between the `createReceiveCycle()` method call and the `pushBBPacketRX()` method call. A histogram of 100 such latency measurements is shown in Figure 7. Since the host PC system time was used to perform these measurements, the variance in measurements is most likely caused by the granularity of task switching in the Linux kernel on the host PC rather than by the device itself.

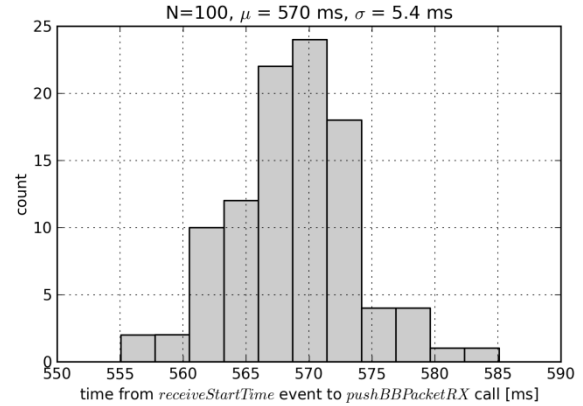


Figure 7: A histogram of latency measurements

## 5. FUTURE IMPROVEMENTS

### 5.1. Improving latency

The simplest way to improve the latency would be to adapt the native serial protocol to use binary instead of ASCII representation when sending signal samples to the host PC. This would significantly reduce the amount of data sent over the serial line per each sampling. Our benchmark has shown that current ASCII encoding requires 10264 bytes to transmit 2048 signal samples. With 12 significant ADC bits per sample, that is 3072 bytes of information and encoding overhead of approximately 300%. Overhead of an efficient binary encoding would likely be significantly lower. It would also likely reduce the CPU load on the sensor core, since sample data would no longer need to pass through an ASCII formatter.

The best theoretical transmission time achievable with this setup would be:

$$t_{transmission} = \frac{3072 \text{ bytes} \cdot 10 \frac{\text{bits}}{\text{byte}}}{576 \text{ kbps}} \cong 53 \text{ ms}$$

Another relatively simple way of significantly improving the latency would be to optimize  $t_{start}$ . The current SNE-ESHTER firmware puts the analog hardware into a complete power down mode between receive cycles. Power up requires a lengthy setup and calibration sequence. A modified firmware that would leave the receiver turned on would hence reduce the latency by approximately 300 ms.

A more complex approach would be to implement the Transceiver Facility on the VESNA SNE-ESHTER CPU. In theory such an approach would enable the Transceiver Facility to operate with minimum latency, since then it

would no longer be acting merely as a façade over the existing firmware. This would however require users to reprogram the device firmware, which would significantly raise the barrier to entry for testbed users. Running the experiment on the sensor node would also put a much stricter limit to the complexity of the software due to the limited amount of RAM and ROM storage on the sensor node. Due to the asynchronous nature of the Transceiver Facility interface, a multi-tasking real-time operating system would be required on the device (current firmware does not use an operating system), further reducing storage available for experimenter's code.

### 5.2. Missing timing information

Transceiver Facility is specified for devices capable of supplying a continuous stream of signal samples to the application. While the Facility enables the operation of a transceiver in burst mode, it assumes that the burst length and timing is defined by software (by scheduling multiple receive cycles) and that hardware can accommodate arbitrary timing of bursts. As has been discussed in previous sections, SNE-ESHTER can only provide a packet of continuous samples up to maximum sampling buffer size of 25000 samples with some mandatory amount of blind time between packets. This creates two problems: a) the Transceiver Facility provides no means of communicating such a hardware limitation to the software and b) there is no means of communicating the length of time or number of dropped samples between two consecutive calls of *pushBBSamplesRx()*. The latter problem could be solved in a backwards compatible way by adding a timestamp field to the *BBPacket* structure.

### 5.3. Exploiting other capabilities of the hardware

As discussed in Section 2, our spectrum sensors are optimized for the use case where some signal processing occurs on the device itself. At the moment, all this functionality is disabled when the sensor is used through the Transceiver Facility interface. The Transceiver Facility could be expanded in a way to allow the user to specify certain pre-processing functions to be applied in the hardware before data is passed to the waveform application.

For instance, an expanded *createReceiveCycle()* method could take an additional parameter specifying such a function. The transceiver could define a list of supported preprocessing functions. For example, specifying a constant *nullFunction* would pass unprocessed signal samples to the waveform application, as per existing specification. Specifying *covarianceFunction* would pass elements of the covariance vector to the waveform application. Specifying *fftFunction* would pass the result of the discrete Fourier transform and so on. This would make it simple for Facility

implementations to off-load such functions to hardware, if the hardware supports it. In CREW testbeds where portability of experiments between testbeds is important, those Transceiver Facility implementations that do not have hardware support for a certain function could implement it in the C++ library on the host PC.

## 6. CONCLUSIONS

We have described an implementation of the Wireless Innovation Forum Transceiver Facility for VESNA SNE-ESHTER spectrum sensor devices deployed in wireless testbeds of the FP7 CREW project. While this usage of the Transceiver Facility is likely not "in the spirit" of the specification, it currently appears useful in our case for providing a unified interface to a broader set of testbed hardware. It enables for example a spectrum sensing algorithm to be developed once and used on USRP-equipped testbed as well as on SNE-ESHTER and Imec Sensing Engine devices with minimal adaptations.

Our measurements show that typical latency that can be expected using the SNE-ESHTER device is 570 ms for a packet length of 2048 samples. Latency increases with higher packet lengths. Most of the latency is due to the delay in communication between the host PC and the device and due to analog front-end setup. We provided some suggestions how the latency could be reduced by improving spectrum sensor's firmware. The high latency currently makes this implementation unsuitable for use cases requiring fast or well-synchronized changes to radio configuration.

Current implementation leaves several functions of the SNE-ESHTER device inaccessible due to the limitations of the current Transceiver Facility specification. The specific limitations on the size and timing of sample packets also remains undescribed within the framework of the specification. We provided some suggestions on how these drawbacks could be mitigated through changes to the Transceiver Facility specification.

Whether the high latency and limitations regarding the particular properties of spectrum sensors will make this Transceiver Facility implementation usable in practical experiments on CREW testbeds remains to be seen.

## 7. ACKNOWLEDGEMENTS

This work has been partially funded by the European Community through the 7th Framework Programme project CREW (FP7-ICT-2009-258301).

## 8. REFERENCES

- [1] E. Nicollet, S. Pothin and A. Sanchez, *Transceiver Facility Specification*, Wireless Innovation Forum, 28 January 2009. Available from: <http://groups.winnforum.org/p/cm/ld/fid=85>
- [2] *CREW project – project overview*, CREW consortium, 2015 [viewed 3 June 2015]. Available from: <http://www.crew-project.eu/>
- [3] M. Mohorčič, M. Smolnikar and T. Javornik, “Wireless Sensor Network Based Infrastructure for Experimentally Driven Research,” *The Tenth International Symposium on Wireless Communication Systems (ISWCS)*, Ilmenau, Germany, August 2013.
- [4] T. Šolc, “SNE-ESHTER: A low-cost, compact receiver for advanced spectrum sensing in TV White Spaces,” *ETSI workshop on Reconfigurable Radio Systems*, Sophia Antipolis, France, December 3-4 2014.
- [5] S. Pollin, et al, “An integrated reconfigurable engine for multi-purpose sensing up to 6 GHz,” *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pp. 656-657, IEEE, May 2011.
- [6] S. Thao, “Transceiver Facility / API - SDR applications Fast-prototyping on Ettus Research USRP platform,” *CREW Training Days*, Brussels, Belgium, February 19-20 2013.
- [7] M. Smolnikar, et al, “Wireless Sensor Network Testbed on Public Lighting Infrastructure,” *The Second International Workshop on Sensing Technologies in Agriculture, Forestry and Environment*, pp. 6-7, 2011.
- [8] C. Kohlhoff, *Boost.Asio*, 2008 [viewed 3 June 2015]. Available from: [http://www.boost.org/doc/libs/1\\_37\\_0/doc/html/boost\\_asio.html](http://www.boost.org/doc/libs/1_37_0/doc/html/boost_asio.html)
- [9] *Spectrum sensing application for the VESNA platform*, 2015 [viewed 3 June 2015]. Available from: <https://github.com/sensorlab/vesna-spectrum-sensor>
- [10] W. Woodall, *Serial Communication Library*, 2015 [viewed 3 June 2015]. Available from: <https://github.com/wjwwood/serial>
- [11] J. W. Grenning, *Test Driven Development for Embedded C*, Pragmatic Bookshelf, 2011.
- [12] *Cpputest* [viewed 3 June 2015]. Available from: <http://cpputest.github.io>

## USING OPENCL TO INCREASE SCA APPLICATION PORTABILITY

Steve Bernier (Nordiasoft, Gatineau, Québec, Canada; Steve.Bernier@Nordiasoft.com);  
 François Lévesque (Nordiasoft, Gatineau, Québec, Canada;  
 Francois.Levesque@Nordiasoft.com);  
 Martin Phisel (Nordiasoft, Gatineau, Québec, Canada; Martin.Phisel@Nordiasoft.com);  
 David Hagood (Aeroflex, Wichita, Kansas, USA; David.Hagood@Aeroflex.com);

### ABSTRACT

The Software Communications Architecture (SCA) is the defacto standard to build Software Defined Radio (SDR) radios. Over one hundred thousand SCA military radios have been deployed worldwide by several nations. The SCA offers a component-based operating environment for heterogeneous embedded system that ensures applications are portable across platforms made of General Purpose Processors (GPPs) and Digital Signal Processors (DSPs).

The SCA offers a high level of portability for applications have been implemented for GPPs and DSPs. SCA components can easily be ported across different processors using different operating systems and communication buses. However, the level of portability is reduced when source code is tuned for specific instructions sets. Furthermore, using Field Programmable Gate Arrays (FPGAs) drastically reduces the level of portability for SCA components.

Specialized instruction sets are very widely used for high performance military radio platforms. Consequently, finding a solution to increase portability of components that run on such processing elements could provide significant cost reductions when an application is ported. In fact, application portability is the number one innovation on the top ten list of most wanted innovations compiled by the Wireless Innovation Forum (WInnF).

This paper describes how the Open Computing Language (OpenCL) can be used in conjunction with the SCA to build more portable applications. OpenCL is a framework for writing programs that execute across heterogeneous platforms consisting of GPPs, DSPs, FPGAs, and graphics processing units (GPUs). The paper starts with an overview of OpenCL, describes how SCA components can be built using OpenCL, provides performance metrics, and concludes on how the SCA could be improved to offer better support for OpenCL.

### 1. INTRODUCTION

The SCA was created to standardize how real-time embedded applications are implemented, packaged, installed, deployed, and controlled. The main goal of the SCA is to make applications very portable across different heterogeneous systems. It was created for the Joint Tactical Radio System (JTRS) program, a US DoD program that funded the development of a new kind of military radios: Software-Defined Radios (SDRs). The JTRS program started by funding the definition of a new standard called SCA and ended with the acquisition of SCA-compliant SDR military radios.

Software-Defined Radios are embedded systems that process a very large quantity of data in real-time. As such, in addition to embedded GPPs, SDR platforms often use DSPs and FPGAs as well. Thanks to the SCA, software can be made very portable even for embedded GPPs and DSPs. SCA components are typically made of control source code and signal processing source code. Portability of SCA components can be affected when the signal processing part is optimized for special instructions sets such as the Streaming SIMD Extensions (SSE) for Pentium processors, the AltiVec instructions for PowerPC processors, or the NEON instructions for ARM processors.

Furthermore, portability is very limited when FPGA firmware is used for signal processing. Different FPGAs offer different resources. Often firmware is designed to use specific resources (e.g. block RAMs, FIFOs, DSP blocks, multipliers) that vary from one FPGA manufacturer to another. Besides, the FPGAs of a single manufacturer can vary significantly from one model to another in terms of such resources. As such, portability has been the holy grail of FPGA firmware designers. It is a research topic that has received a lot of attention over the years. Thus far, no one solution have prevailed over the others. Over time, the SCA has improved some aspects of portability for applications that use FPGAs. It did so by standardizing how software components running on DSPs and GPPs can interact with

components that run on FPGAs. With that approach, firmware can be adapted or rewritten for new FPGAs without having a serious impact on the software it interacts with. Nevertheless, the SCA does not improve the portability of the actual FPGA firmware.

One of the popular approaches to improve portability of high performance signal processing source code is to use domain-specific accelerators. The approach consists in writing source code for widely available libraries of domain-specific APIs that execute fast thanks to co-processors. Microsoft uses this approach with DirectX which offers a large number of functions that can be optimized to run on GPUs [1]. The same approach has also been used with FPGAs as co-processors [2, 3].

While the concept of accelerators can increase portability, different APIs must be used for different type of processing elements (GPPs, DSPs, GPUs, FPGAs). Relying on different APIs adds complexity for designers of applications for heterogeneous embedded systems. It also prevents portability across different processing elements.

Open Computing Language (OpenCL) is a framework for implementing software components that can execute across different processing elements [4]. It allows a developer to implement a function in source code that can be compiled for GPPs, DSPs, GPUs, and FPGAs. The following sections of this paper provide an overview of OpenCL, describe how SCA components can be built using OpenCL, and provide performance metrics. The paper concludes on how the SCA could be improved to offer better support for OpenCL.

## 2. THE OPEN COMPUTING LANGUAGE

OpenCL is an open and royalty-free standard maintained by a non-profit technology consortium called the Khronos Group [5]. It has been created to allow high-performance applications to execute on various devices of different architectures implemented by different vendors.

OpenCL greatly improves performances for a wide range of applications by allowing task-based and data-based parallel programming. With OpenCL, a computing system is made of a number of compute devices connected to a host processor. Compute devices are GPPs, GPUs, DSPs, or FPGAs. The host processor is a GPP.

An OpenCL application is made of two parts: kernels and a host program. OpenCL kernels are routines (algorithms) performing the data processing. Kernels are implemented in a C-like language and executed on the compute devices. A single compute device typically consists

of many individual processing elements (PEs) and a kernel can run on all or many of the PEs in parallel. The host program runs on the host processor and is implemented using an application programming interface (API) to launch kernels on the compute devices and manage device memory. The OpenCL standard defines host APIs for C and C++; third-party APIs also exist for other programming languages [6, 7, 8]. An OpenCL framework consists of a library that implements the host APIs, and an OpenCL compiler for the target compute device(s).

### 2.1. Portability

The goal of OpenCL is to allow high-performance applications to run on any hardware. It provides portability by allowing the same source code to be compiled for different target compute devices. Host programs are compiled using the C/C++ compiler and the appropriate library for host APIs. Kernel programs can be pre-compiled for specific target compute devices before run-time. They can also be compiled on-the-fly at run-time for the required target devices.

OpenCL also extends C/C++ by providing standardized vector processing instructions and data types to exploit vector engines of the modern processors [9].

## 3. USING OPENCL TO INCREASE PORTABILITY OF SCA APPLICATIONS

SCA applications are made of one or many components that perform data processing. Every SCA component is made of configuration properties and ports to process data. Components also contain several implementations; one for each processing element it supports. For portable components, the different implementations are produced by building the same source code for the different processing elements.

For components that need to be optimized, the data processing source code needs to change significantly to exploit processor-specific instructions. The software part of a component that deals with control does not need to change much from one implementation to another.

However, using OpenCL, the data processing source code does not require any change to exploit the different processor architectures. In fact, OpenCL code can also be executed on FPGAs [10, 11]. OpenCL effectively reduces the development time required for a component to run on multiple processing elements including FPGAs. FPGA firmware is built using platform-specific features and requires very long development cycles.

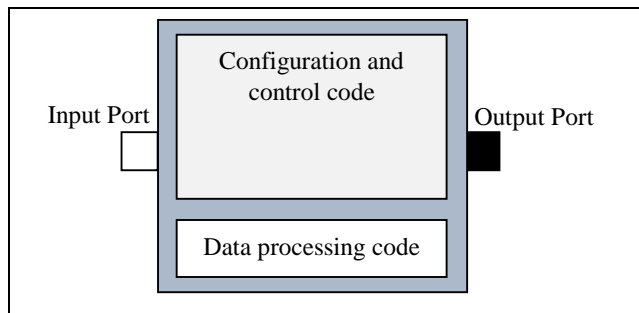
### 3.1 A Simple Approach to using OpenCL with the SCA

SCA applications are deployed on SCA platforms via the execution of their components. The SCA Core Framework chooses an implementation for each component and executes it using an SCA device. The choice of the SCA device is made by matching the requirements of the component implementations with the capabilities being advertised by the SCA device. For instance, a component that only has one implementation that requires an x86 processor can only be executed by an SCA Device that advertises being capable of running x86 implementations.

Deploying a component implementation that uses OpenCL to perform signal processing works the same way as deploying a component that requires SSE or AltiVec instructions. The application component that is implemented using OpenCL simply needs to specify a requirement to be deployed on an SCA device that represents an OpenCL-capable processing element. Such an SCA device must advertise capabilities that identify its capability to host OpenCL programs.

## 4. CREATING AN OPENCL SCA COMPONENT

Figure 1 shows the structure of a typical SCA component where data to be processed is received via an input port and sent, after the processing is performed, to another component via an output port.



**Figure 1. Structure of a typical SCA component.**

The figure shows the distinction between the configuration and control code, and the data processing code. For an OpenCL SCA component, the host program is part of the configuration and control code, and the kernels are part of the data processing code. The kernels can potentially be executed on different compute devices (i.e. OpenCL-capable processing elements) when many compute devices are connected to the GPP where the host program runs. OpenCL provides APIs to list platform and compute device information, to obtain the identifiers for compute devices, and to specify which device should be used to execute kernels. One single SCA device can therefore load

and execute kernels on any compute device connected to the GPP where the SCA device runs.

Typically, the source code for a kernel is located in a separate file from the OpenCL program source file. The OpenCL API offers several ways to create a program from which kernels are instantiated. A program can be created from a buffer containing program source code, from a buffer containing the program binaries, either in binary format specific to a device or in an intermediate representation that will be converted to the device-specific code format. The appropriate format is selected based on the level of portability and performance needed for an application.

For the SCA, this means the kernel files are not embedded in the source file for the SCA component implementation itself. The way to model this with the SCA is to define a software dependency between the SCA component implementation and the OpenCL kernel files it uses. Doing so will cause the SCA Core Framework to load the kernel files on the same SCA device used to execute the SCA component implementation.

### 4.1. Loading the kernels

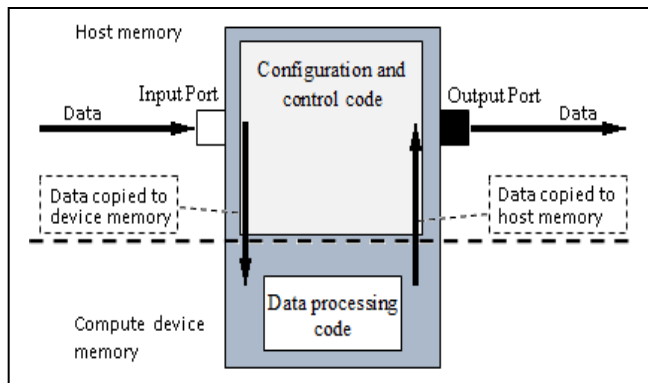
Once an SCA component start running, it must load the kernels and instantiate them before the data processing starts. In our experiments, the kernel creation was done from during the initialization of the SCA application component (i.e. `LifeCycle::initialize()`). Kernel creation involves initializing OpenCL, listing and selecting compute devices, loading kernel files, and creating the kernels. This is all done using OpenCL APIs which makes calls to device drivers.

To be more portable, it is forbidden for SCA application components to make calls to native device drivers. However, just like applications are allowed to use several POSIX APIs, the SCA specification should allow OpenCL APIs since this standard is broadly supported across different types of processing elements. Alternatively, it would be possible to create an SCA-level API that SCA devices could implement for application components to use. This would prevent implementations of application component from being compiled and linked against native device drivers.

### 4.2. The Data Flow

OpenCL kernels use compute device memory to get input data and provide output data. The host program is responsible for creating compute device memory to be used by the kernels. The host is also responsible for copying data from its memory to the compute device memory and vice-versa if it is required.

SCA components usually receive and send data through ports. This means the data is in the memory of the host processor. Therefore, the input data received by an input port must be copied into the OpenCL compute device memory (H2D) before executing a kernel, and the output data produced by a kernel must be copied from the compute device memory to the host memory (D2H), after a kernel has executed. Figure 2 shows the data flow for every sequence of data being processed by an OpenCL SCA component.



**Figure 2. Data flow of data processed by an OpenCL SCA component.**

Copying data between different memories affect the overall data processing performance. Copy of data can be avoided when the compute device is a CPU since the memory of the device is the same as the host. But, when the compute device is not a CPU then data must be copied. We have collected some metrics regarding this topic that will be presented in the next section.

## 5. METRICS

In this section, we discuss some metrics that can impact the performance of data processing using OpenCL. We also suggest solutions or research areas to address the issue we identify. To perform our experimentation, we used a desktop computer with an Intel i7-4770 CPU with 8 cores clocked at 3.40 GHz, 4GB of memory. We used the 64 bits version of Fedora 20 with the Linux kernel version 3.11.10-301. As for OpenCL, we used two compute devices. The first one was the CPU device of the Intel OpenCL platform with OpenCL 1.2. The second OpenCL device was PCI-E 3.0 NVIDIA GeForce GT 635 GPU using the NVIDIA OpenCL CUDA 7.0.41 platform with OpenCL1.1.

### 5.1 OpenCL Program Format

In section 2.1, we described that OpenCL brings portability by allowing the same source code to be compiled and

executed for various compute devices with different hardware architecture. Building every kernel a head of time and packaging the binaries with the application components is in line with the common SCA. Each SCA application component contains several implementations of the component. Using OpenCL means each SCA component implementation will come with kernel binaries targeting a specific compute device. The deployment of an SCA application lead to the choosing of the right implementations of each component and each kernels based on the hardware available in the SCA platform.

However, with the proper driver support, kernels can be built on the fly at the moment the SCA application gets deployed. In such a case, the application is packaged with the kernels either in source code format or in an intermediate binary format which is portable across different compute devices. Indeed, OpenCL supports a format called Standard Portable Intermediate Representation (SPIR) for kernel binaries. SPIR is cross-platform and designed for heterogeneous parallel computing. It is based on LLVM IR [12].

Using this approach reduces the requirement for having several implementations of an SCA component and OpenCL kernels. If the SCA platform contains one GPP and several OpenCL compute devices, there is no need to prebuild all the kernels. The kernels can be built on the fly based on the selected compute devices. This approach also future-proves the SCA application since it supports any compute device that might be integrated in the future. In short, it makes the SCA application more portable to different SCA platforms that use the same GPP but different OpenCL compute devices. However, using this approach incurs a runtime cost during the deployment of applications since the OpenCL builder is invoked on the fly.

To evaluate the impact of selecting an approach over another, measurements have been made regarding the time it takes to create a kernel from source code, SPIR format, and from native binaries prebuilt for specific compute devices. The tests have been executed ten times for each file format and file size (i.e. small vs large) of the source code. To represent a small source file, we used a kernel routine implemented in 16 lines of code (LOC). We used a routine implemented with 398 LOC to represent a large source file. The SPIR binaries were created using the options “-x spir -spir-std=1.2” with the OpenCL compiler. Table 1 shows the average times it takes to create a kernel that is ready to be executed starting with above-mentioned 3 types of kernel files.

**Table 1. Average time in  $\mu$ s to create a kernel based on source code file size.**

Format in kernel file	Small		Large	
	CPU	GPU	CPU	GPU
Source code	13149	391	142089	447
Native binary	968	378	4381	396
Binary in SPIR	923	--	4187	--

As it can be seen from Table 1, creating a kernel from source code is surprisingly fast. Creating a kernel involves compiling and linking the kernel source code for different compute devices. For a CPU compute device, it takes approximately 13 to 142ms to create a kernel from source code. Doing the same for the GPU compute device only takes 0.3 to 0.5ms. Note that creating kernels only happens once each time an application is launched, no matter how long the application runs for. The reason it takes a different amount of time to create kernels for different compute devices is that different tool chains are used. Another surprising result is that creating a kernel for a GPU compute device takes about the same time whether from source code or from native binary. For a CPU compute device, creating a kernel from binary SPIR format takes about the same time as creating from native binary, even slightly faster. Since SPIR binaries are portable, this format represent the best solution for use with the SCA. The SPIR format also offers the side benefit of not exposing the kernel source code on the deployment platform.

## 5.2 Buffer Size

As mentioned before, the input data must be moved from the host memory to the target compute device memory on which a kernel will be executed. Similarly, the output data must be moved back to the host memory after the execution of the kernel. The time spent copying data affects the overall time

required for OpenCL kernels to process data. Experiments have been conducted to measure the impact of copying of data across the bus that connects the host and the target devices.

The experiments used various buffer sizes, from 4KB for the size of small buffers to 3.125 MB for the size of large buffers (800 times the size of the small buffers). The measurements were averaged over twenty tests in each direction. Table 2 provides the averages in microseconds and illustrates the difference in performance between different types of compute devices. It also quantifies that the cumulative cost of copying data across memory types can be significant. Figure 3 shows the plotting of these numbers. NordiaSoft is currently investigating, with good success, different approaches to reduce the costs of moving data. Results to be published in a follow up paper.

**Table 2. Average time to copy buffers.**

Buffer size (KB)	CPU		GPU	
	H2D ( $\mu$ s)	D2H ( $\mu$ s)	H2D ( $\mu$ s)	D2H ( $\mu$ s)
4	5	9	10	12
32	7	12	19	19
320	32	42	101	104
640	67	75	191	312
960	112	105	406	464
1280	155	153	468	614
1600	193	161	520	694
1920	247	186	577	814
2240	274	209	653	903
2560	333	234	706	1020
2880	608	296	746	1194
3200	694	372	794	1307



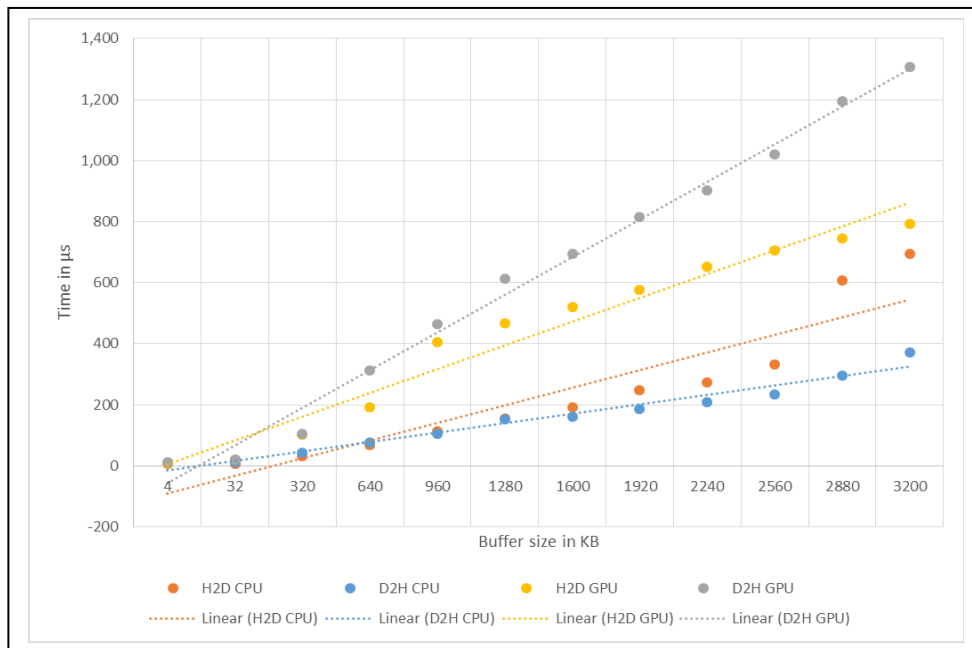


Figure 3. Average time to copy buffers from H2D and D2H.

## 6. CONCLUSION

OpenCL is effective to increase the portability of SCA applications across heterogeneous platforms. It allows application components to be portable between GPPs, DSPs, GPUs, and FPGAs. In short, OpenCL addresses directly the number one innovation from the top 10 most wanted innovations as defined the Wireless Innovation Forum. The paper describes how the SCA can benefit from OpenCL. It explains how OpenCL SCA components can support multiple compute devices with a single implementation of the signal processing source code.

The paper underlined the fact that portability for signal processing functions can be achieved at the source code level and at the binary level which offers more protection for intellectual property. Metrics have been presented to illustrate how fast it is to instantiate OpenCL kernels. The paper also provided metrics that show the performances associated with moving data across different types of memory.

A simple approach to support OpenCL with SCA has been presented. It described how an SCA Device must advertise its capabilities to execute OpenCL kernels. It also explained how SCA application components can integrate OpenCL kernels. We have identified some areas of potential improvement for the SCA specification to better support OpenCL.

Finally, the paper showed how the copy of data between the OpenCL host processor and a target compute

device can potentially affect real-time performances. More research can be performed on this topic to alleviate the issue.

## 7. REFERENCES

- [1] F. D. Luna, Introduction to 3D Game Programming with DirectX 10, WordWare Publishing Inc., Sudbury, MA, USA, 2008.
- [2] W. Zhang, V. Betz, and J. Rose, Portable and Scalable FPGA-Based Acceleration of a Direct Linear System Solver, ACM Transactions on Reconfigurable Technology and Systems, Vol. 5, No. 1, Article 6, March 2012.
- [3] G. C. T. Chow, K. Eguro, W. Luk, and P. Leong, A Karatsuba-based Montgomery Multiplier. FPL '10 Proceedings of the 2010 International Conference on Field Programmable Logic and Applications. 2010.
- [4] <http://en.wikipedia.org/wiki/OpenCL>.
- [5] The Khronos OpenCL Working Group, The OpenCL Specification version 2.0, 2014, <https://www.khronos.org/opencl/>.
- [6] <http://mathematician.de/software/pyopencl/>
- [7] <https://code.google.com/p/javacl/>
- [8] <https://github.com/Nanosim-LIG/opencl-ruby>
- [9] M. Scarpino, *OpenCL in Action*, Manning Publications Co., Shelter Island, 2012.
- [10] R. Brueckner, How OpenCL Could Open the Gates for FPGAs, 2015, <http://insidehpc.com/2015/02/how-opencl-could-open-the-gates-for-fpgas/>.
- [11] Implementing FPGA Design with the OpenCL Standard, November 2013, [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/wp/wp-01173-opencl.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01173-opencl.pdf).
- [12] The Khronos Group Inc., The SPIR™ Specification version 1.2, 2014, <https://www.khronos.org/registry/spir/>