WInnComm 2017

The Wireless Innovation Forum Conference on Wireless Communications Technologies:
Connecting technical, business and regulatory leaders ~ Defining the future of radio communications

*Conference: 15-16 November * Technical Exchange Meetings: 13-17 November*
Qualcomm Institute, UCSD, San Diego, California

Proceedings of
WInnComm 2017
Wireless Innovation Conference on
Wireless Communications Technologies and Software Defined Radio
*15-16 November 2017 * San Diego, California*

Editors: John Glossner, Lee Pucker, Stephanie Hamill

Platinum Sponsors:

Google

LEONARDO

MOTOROLA SOLUTIONS

Silver Sponsors:

Innovative Integration
a molex company

PENTEK

Media Sponsor:

THALES

Copyright Information

WInnComm 2017 Organization

Conference Organizers:

Claude Belisle, NordiaSoft

Andrew Clegg, Google

Ken Dingman, Harris

John Glossner, Optimum Semiconductor Technologies

Manuel Uhm, Ettus Research

Session Chairs:

Claude Belisle, NordiaSoft

Andrew Clegg, Google

Nina Figueira, Brazilian Army & NIPCAD Company

Tim Fountain, National Instruments & National Instruments

frederic j harris (San Diego State University

Daniel S Iancu, Optimum Semiconductor Technologies & Tampere University of Technology

Heikki Kokkinen, Fairspectrum

Arjuna Madanayake, University of Akron

Preston Marshall, Google

Ajay Kumar Poddar, Synergy Microwave Corporation

Table of Contents

MESHAPP: AN ARCHITECTURE TO SHIFT APPLICATIONS FROM SCA-BASED RADIOS TO COTS SMART DEVICES

Li Zhou (zhouli2035@nudt.edu.cn)[1,2], Qi Tang[1,2], Shan Wang[1,2], Fanglin Gu[1,2], Haitao Zhao[1,2], and Jibo Wei[1,2]

[1]College of Electronic Science, National University of Defense Technology, Changsha, China
[2]Hunan Engineering Research Center of Software Radio, Changsha, China

ABSTRACT

In recent years, LTE and 5G standardizations have made a huge success and wireless innovations are increasingly driven by the mobile industry. There are numerous and abundant applications designed for commercial-off-the-shelf (COTS) smart devices, such as smartphones, tablets and smart watches. Meanwhile, Software Communication Architecture (SCA) has also achieved a great success which enhances portability of waveforms and interoperability among diverse software defined radio (SDR) platforms. However, most SCA-based radios still inherit the traditional style which is not open for application developers, making their capabilities far from well utilized. Therefore, It is becoming very important to study how to leverage the advantages of mobile industry to boost the applications and enhance user experience of SCA-based radios. In this demonstration, we propose MeshAPP, an architecture to shift applications from SCA-based radios to COTS smart devices. Two applications, Waveform Manager and MeshChat, are developed based on MeshAPP. We would like to present our SDR education platform, developing tools and applications in this demonstration.

1. INTRODUCTION

Software Communication Architecture (SCA) can enhance portability and interoperability of software defined radio (SDR) platforms potentially [1]. To develop higher applications for SCA-based platforms, three system architectures are considered initially.

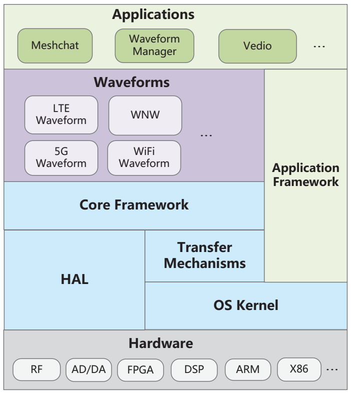


Figure 1: The first system design.

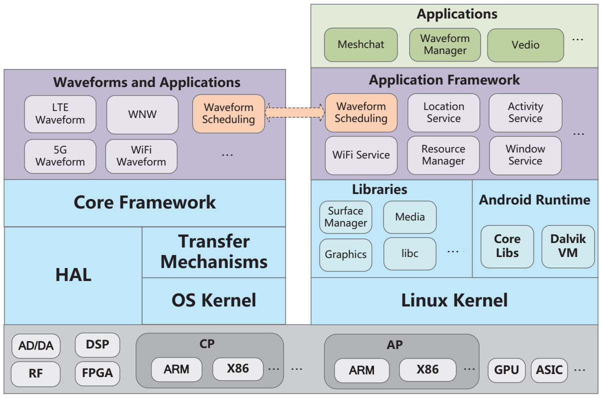


Figure 2: The second system design.

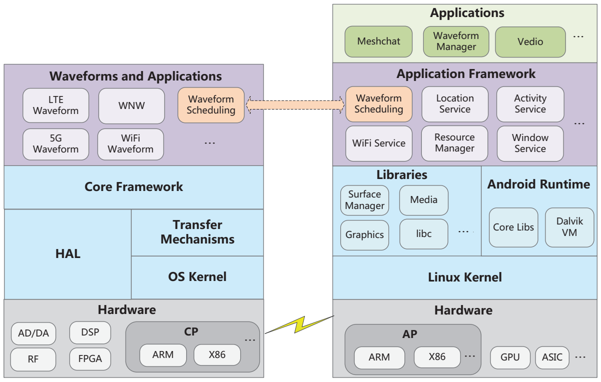


Figure 3: The third system design: MeshApp.

In the first system architecture, higher applications and waveforms share an operating system (OS) on the same platform, as depicted in Fig. 1. An OS-coupled application framework is designed to support higher applications. The main drawback of this architecture lies in two parts. The first drawback is that SCA can be deployed in diverse OSs, e.g., Linux, VxWorks. Some OSs are lack of basic drivers, such as display driver and audio driver, which is not friendly for application developing. The second drawback is that the overhead of applications may worsen the performance of waveforms and shorten the battery lifetime of platforms.

In the second system architecture, the SDR module and application module are deployed on different OSs so that waveforms and higher applications would not affect each other, as shown in Fig. 2. However, application modules are still specially designed within the platform. Although developers could update application modules with the development of System on Chip (SoC) technologies, the cost is relatively high and the developing time is relatively long.

The third system architecture is illustrated in Fig. 3. We simply replace the application module with a commercial-off-the-shelf (COTS) smart device so that we can obtain the best COTS devices with the lowest cost at any time. In this way, more computing demanding applications could be deployed on platforms and user interfaces could be much more friendly. We adopt this architecture in our demonstration.

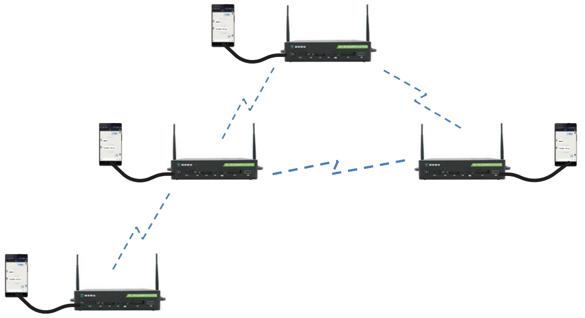


Figure 4: Wired configuration.

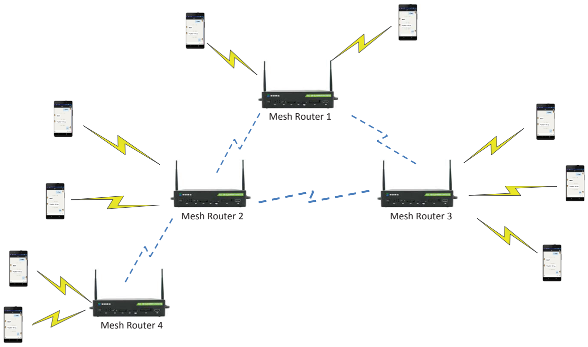


Figure 5: Wireless configuration.

The connection between application module and communication module could be either wired or wireless with various interface technologies. In our demonstration, we adopt USB for wired configuration and WiFi for wireless connection. Wired connections are usually limited due to the hardware design while wireless configuration could provide much more connections. Wired configuration and wireless configuration are shown in Fig. 4 and Fig. 5 respectively. From the figures we can observe that four SDRs form an ad hoc network. With the integration of smart devices, the network turns into a mesh network and the SDRs turns into mesh routers. We name the system achitecture as MeshAPP, which enables to shift applications from SCA-based radios to COTS smart devices.

2. DEMONSTRATION

Our demonstration setup is shown in Fig. 6. We use the educational SDR platform KDZL-SDR-EL01 and software suite produced by HNWiseCom. KDZL-SDR-EL01 is a high performance SDR platform, which is designed for rapid prototyping of waveforms. The baseband chip is ZYNQ 7030 which includes a dual-core Cortex-A9 ARM and a Kintex-7 FPGA. The DDR2 memory size is 1GB and the OS running on the platform is Linux 3.17. WiFi module and USB interfaces are integrated in the platform which supports connections between the platform and smart devices. The software suite includes the core framework KDZL-SCA-CF [2], the middleware KD-RPC [3] and the integrated development environment (IDE) KDZL-SCA-IDE. The products KDZL-SCA-CF and KDZL-SCA-IDE follow the SCA 4.1 standards [1]. A wideband network waveform (WNW) developed by the KDZL-SCA-IDE, as shown in Fig. 7, is launched on each SDR platform. The smart devices that we adopt in the demonstrations are ordinary Android smartphones.

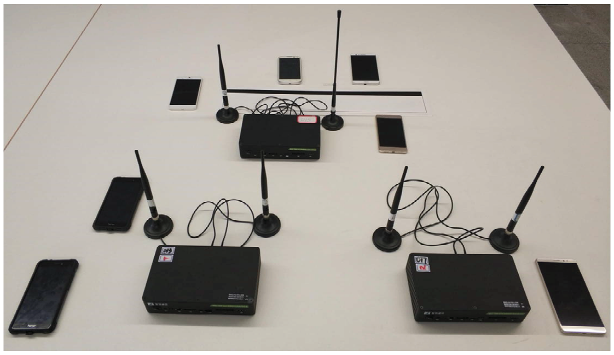
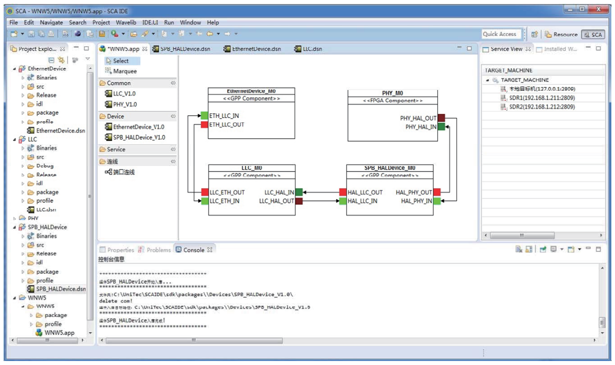
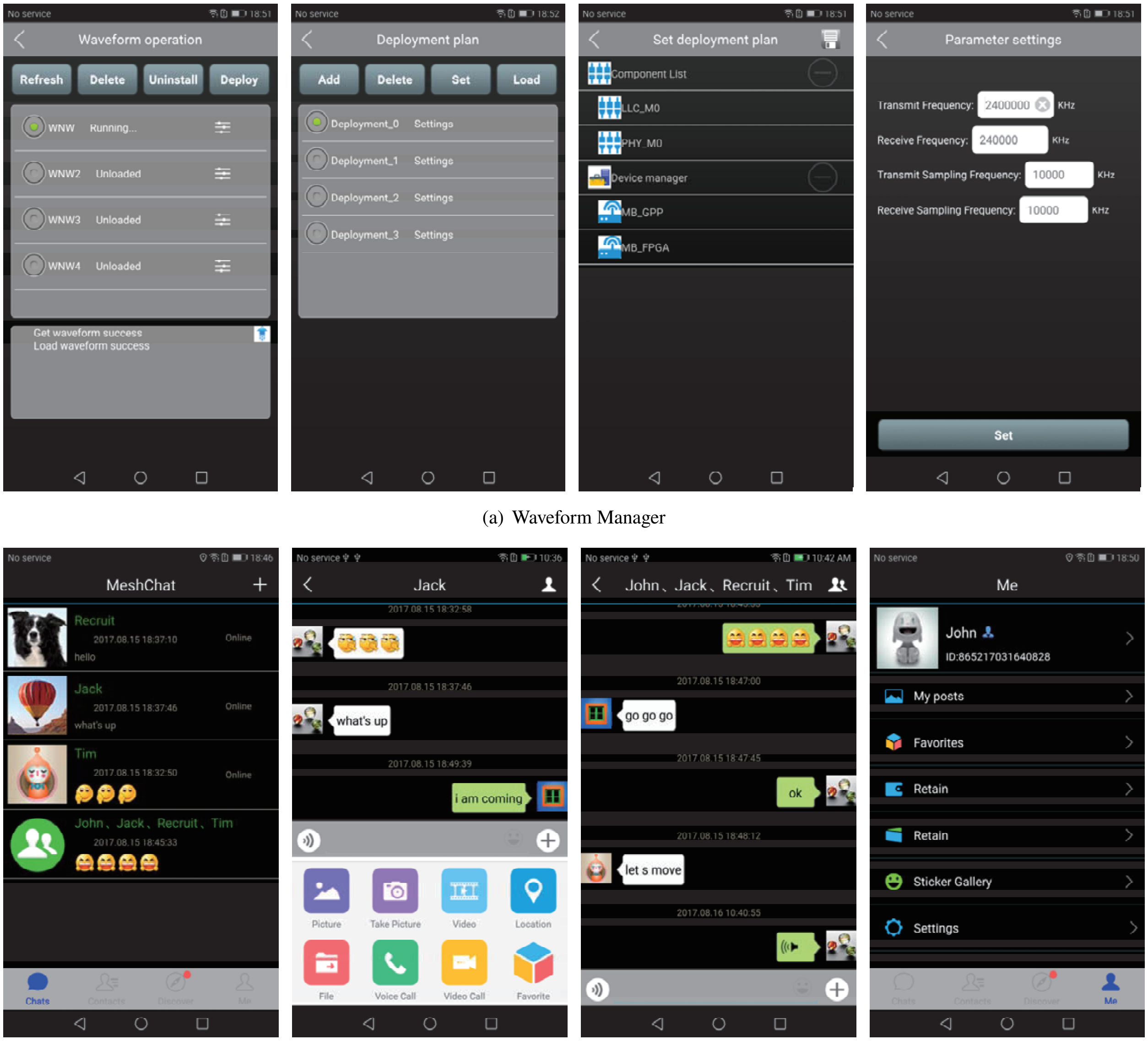


Figure 6: Demonstration setup.



```
************************************
组件SPB_HALDevice开始入库...
************************************
文件夹:C:\UniTec\SCAIDE\sdk\packages\\Devices\SPB_HALDevice_V1.0\
delete com!
组件入库目标路径: C:\UniTec\SCAIDE\sdk\packages\\Devices\SPB_HALDevice_V1.0
************************************
组件SPB_HALDevice入库完成!
************************************
```

Figure 7: KDZL-SCA-IDE.



Get waveform success
Load waveform success

Transmit Frequency: 2400000 KHz
Receive Frequency: 240000 KHz
Transmit Sampling Frequency: 10000 KHz
Receive Sampling Frequency: 10000 KHz

Recruit 2017.08.15 18:37:10 Online hello
Jack 2017.08.15 18:37:46 Online what's up
Tim 2017.08.15 18:32:50 Online
John、Jack、Recruit、Tim 2017.08.15 18:45:33

2017.08.15 18:32:58
2017.08.15 18:37:46
what's up
2017.08.15 18:49:39
i am coming

2017.08.15 18:47:00
go go go
2017.08.15 18:47:45
ok
2017.08.15 18:48:12
let s move
2017.08.16 10:40:55

ID:865217031640828

(a) Waveform Manager

(b) MeshChat

Figure 8: Our Android applications.

The first application we introduce here is called Waveform Manager. The interfacial designs are illustrated in Fig 8(a). Waveform Manager is designed for waveform operation. The actions include deploy, load, uninstall and delete the waveforms. It also allows to configure and query parameters of a certain waveform. Without computers, the operation of SDR platforms would be much easier with this application.

The second application we introduce is called MeshChat. MeshChat provide the abilities to communicate with or without a cloud server. As illustrated in Fig. 8(b), our demonstration is based on a non-cloud configuration. The services include transmission and reception of messages, voices, videos, files and locations. It supports both single chat and group chat.

REFERENCES

[1] J. T. N. Center, “SCA specification version 4.1,” 2016.

[2] R. Cheng, L. Zhou, Q. Tang, D. Ma, H. Zhao, S. Wang, and J. Wei, “Can SCA 4.1 replace STRS in space applications?,” in *WInnComm-Europe 2017*, 2017.

[3] Y. Zhao, L. Zhou, Q. Tang, D. Ma, H. Zhao, S. Wang, and J. Wei, “Investigation of high-efficient transfer mechanisms for SCA 4.1,” in *WInnComm-Europe 2017*, 2017.

DEMO: IMPLEMENTATION OF AN OPEN SOURCE SPECTRUM ACCESS SYSTEM

Shem Kikamaze shemk@vt.edu
Vuk Marojevic maroje@vt.edu
Carl Dietrich cdietric@vt.edu
Jeffrey H. Reed reedjh@vt.edu
Bradley Dept. Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA

1. ABSTRACT

It is not a matter of debate that dynamic spectrum sharing leads to better spectrum utilization than the current auctioned licensed allocations by the Federal Communication Commission (FCC). The debate ensues on the choice between the different methods used for dynamic spectrum sharing, and their performance compared to the traditional static spectrum allocation. To lead the move towards dynamic spectrum allocation, the FCC proposed commercial utilization of the 3.5 GHz band, which it termed as the “Innovation Band”. In the proposal, a specific band of the spectrum would be made available for secondary use by new users as long as the primary or incumbent user is not using that band. This was to provide a platform which would enable researchers to find a way of maximizing the use of spectrum resources while ensuring harmonious spectrum coexistence between Primary Users (PUs) and Secondary Users (SUs).

The FCC proposal involves a central entity known as the Spectrum Access System (SAS) which coordinates spectrum allocations for the different SUs. The SAS collects spectrum information for a specific geographical area from its sensors or SUs to create a Radio Environment Map (REM). The information from the REM is then utilized by the SAS to make decisions on spectrum allocation for the different SUs.

Recent research on the performance of spectrum coordination by the SAS has been theoretical at best, based on various PU and SU interaction models. There have also been applications of isolated Cognitive Radios (CRs) that are able to coexist with a primary user, but very few based on a networked cluster of SUs whose spectrum allocation decisions are controlled by a central entity.

We are creating a framework that can be utilized to perform practical analyses of the performance of the SAS under varying experimental scenarios in which PUs and SUs need to coexist in the available spectrum channels. Initially, we are focusing on the 3.5 GHz band.

In this demo, we demonstrate the functioning of an Open-Source Spectrum Access System, that is able to leverage a collection of CRs to effectively coexist in shared spectrum. This is achievable through a collection of Software Defined Radios (SDRs) connected to a centralized node that is able to dynamically assign channels to them.

2. SYSTEM OVERVIEW

Physical Setup: The Open Source SAS is being built on top of the CORNET testbed at Virginia Tech (http://www.cornet.wireless.vt.edu/). This testbed consists of 48 interconnected SDRs with additional being deployed.

All nodes on the CORNET testbed are remotely accessible.

Software Setup: There are different functionalities that need to be exercised by the CRs to create a stable cluster of nodes that can enable an effective SAS. The different functionalities include:

a) Environment Sensing Capability (ESC): The CRs are able to sense the different spectrum bands to make a decision on the channel availability. For an SU, this information is sent to the SAS which makes the final decision. For a cluster of CRs, all relevant information from the SUs is aggregated at the SAS. To improve on sensing, some of the nodes on CORNET are assigned as sensing nodes for the SAS. The only function of the sensing nodes is to provide spectrum information to the SAS in a specific area. In summary, the overall sensing capability of the system is based on spectrum information from both the SUs and dedicated sensing nodes. This crowdsourced information collection is enabled by CORNET. All sensing is currently carried out with GNU Radio.

b) Radio Environment Map (REM): To construct a REM, each CR in the network has an internal database in which all radio-related information available to that node is stored. This ensures that decisions for the availability of a spectrum channel are based on both current and past events to predict future events. All CRs have local databases, but the SAS has the global database that encompasses information from all other nodes.

c) Spectrum Access System (SAS): This is the centralized node with access to the Global REM. It makes decisions on whether the spectrum channel is occupied by the PUs or not. It also assigns different channels to the SUs when the PUs are not transmitting. Furthermore, if the PUs start transmitting, the SAS reassigns the SUs to a different available channel to prevent interference to the PUs. The PUs have the highest priority for a given channel.

These are the main functionalities needed by the SAS to be able to protect the PUs while allocating channels to the SUs that might need them and resolve any conflicting requests among them.

2. DEMONSTRATION

Each of the CRs on CORNET has the above system installed. The system is setup in such a way that one of the nodes takes the role of the SAS, and some of the nodes are the sensing nodes that gather spectrum information of the varying radio environment around them. In the vicinity of a sensing node, another node is assigned as an SU that needs access to spectrum. The SU provides some sensing spectrum information for the REM to the SAS. If an SU needs access to an RF channel, it requests one from the SAS which uses the information in the REM to create a response to the SU. Different SUs are scattered all over the testbed. Finally, some of the nodes are assigned as PUs that randomly start transmitting since they have sole rights to the shared band and do not need to coordinate its use with the SAS or other users.

The role of the SAS is to maintain a harmonious relationship between the SUs and the PUs while making sure that the SUs maintain a certain Quality of Service with little to no interference towards the PUs. This is all done with the SAS having no prior information about when the PUs are transmitting.

The SDR system is fully automated and can be accessed from anywhere through the Internet. The demo will showcase different scenarios that run remotely on the CORNET testbed, located in Blacksburg, VA. Remote login from local laptops will run the experiments and visualize the system operation and performance in life spectrum.

Map-Reduce Based Hybrid Beamforming:Trade-Off between Complexity and Cost

Ture Peken Ravi Tandon Tamal Bose
Department of Electrical and Computer Engineering
University of Arizona
Tucson, Arizona 85719
Email: {*turepeken*, *tandonr*, *tbose*}@email.arizona.edu

***Abstract*—High data rates up to 10 Gbps can be achieved for next generation wireless communication by using the millimeter wave (mmWs) bands. Hybrid beamforming, which combines analog beamformers in the RF domain and digital beamformers in the baseband domain, allows the reduction of RF chains while achieving high performance gains in mmWs. Therefore, wireless systems operating at mmWs are expected to use hybrid beamforming. Analog and digital beamformers, which achieve the maximum mutual information over the channel, should be designed in hybrid beamforming. The computational complexity of finding optimal beamformers is significantly high since it grows exponentially with the number of subarrays at the transmitter and the receiver. MapReduce is a framework which can be used to process large data sets in a parallel fashion. Optimization of precoders in hybrid beamforming is actually a distributed sorting problem. The MapReduce framework can be used to increase the speed of the optimum precoder design in hybrid beamforming by executing the algorithm distributively among the multiple cores. We want to show the optimum number of cores to run the MapReduce based hybrid beamforming algorithm based on the trade-off between the cost and complexity. The optimum number of cores to use in the MapReduce based hybrid beamforming has not been studied to the best of our knowledge. In this paper, we analyze the optimum number of cores in terms of the computational complexity and the cost due to the communication load.**

I. Introduction

The millimeter-waves (mmWs) is a promising technology which is expected to play a critical role towards the 5G systems. The available mmW spectrum is 200 times larger than the spectrum below 3 GHz, in which today‘s cellular systems operate [1]. The main drawback of using higher frequencies is the increase in path loss due to Friis' Law [2]. However, the performance degradation due to the path loss can be compensated by using appropriate beamforming in mmWs [3]. Beamforming can boost the signal-to-noise ratio (SNR) at the receiver and decrease the co-channel interference when there are multiple users [4].

Conventionally, beamforming is implemented in analog or digital domain. In analog beamforming, time delaying or phase shifting can be used to apply antenna weights [5]. The data stream is split among array elements and the signal in each substream is processed by a time delay element or a phase shifter, is amplified and fed into the array element. Even though it is the most cost-effective way of building beamforming, one data stream can be handled with a single analog beamformer. In order to form multiple beams, multiple analog beamformers must be used. Digital beamforming can handle multiple data streams. By feeding each array element with a separate transceiver and data converter, multiple beams can be generated simultaneously. Since each array element requires a complete dedicated RF chain, it is less cost-effective than analog beamforming [6]. Hybrid beamforming, introduced in [7], [8], has been proposed to strike a balance between the system performance and cost objectives. Hybrid beamforming is the combination of analog and digital beamformers. The advantage of hybrid beamforming over conventional methods is due to the fact that number of RF chains can be lower-bounded by the number of data streams while the beamforming gain can be still set as high as the number of array elements. Hybrid beamforming can be implemented by combining multiple array elements into subarray modules.

The hybrid beamforming architecture consist of RF and baseband precoders at the transmitter and the receiver [9], [10]. In order to obtain the highest data rates in these systems, optimum precoders need to be designed based on the channel conditions and available beams. In particular, the optimum RF and baseband precoders achieve the maximum mutual information over the channel. Computational complexity of finding the optimum precoders by searching all different combinations of the precoders, which are constructed by using available beams, grows exponentially with the number of subarrays at the transmitter and the receiver [11]. With even a modest number of subarrays, a great number of computations is required to calculate the optimum precoders. For instance, the number of computations for 6 transmitter and receiver subarrays with 3 available beams at the each transmitter and receiver subarray would be $3^6 \times 3^6 = 531441$.

MapReduce is a framework that allows to process the large

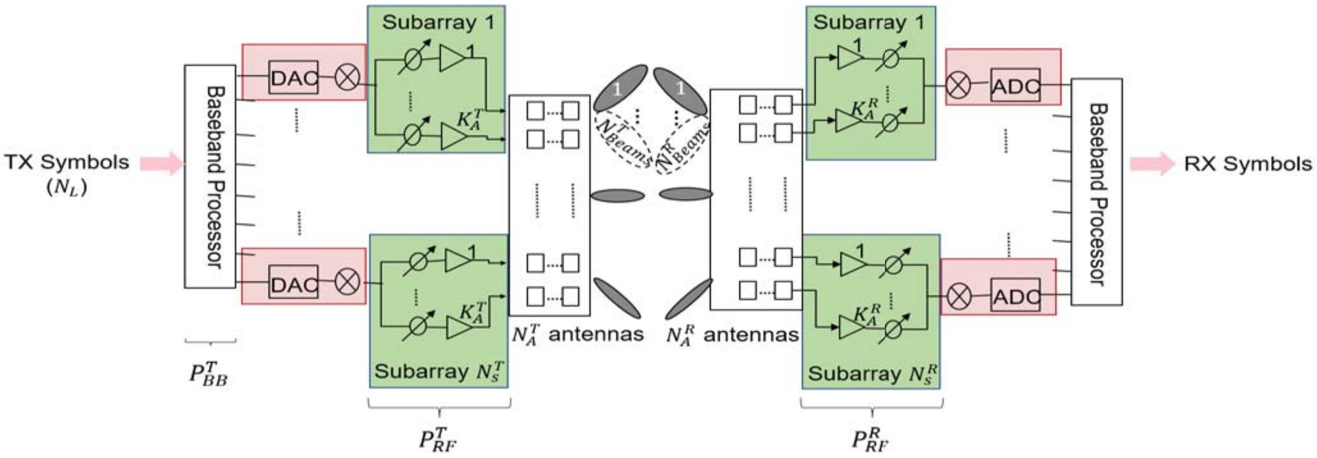


Fig. 1: Above the hybrid beamforming architecture is shown with RF and baseband blocks at the transmitter and the receiver. N_L transmitter (TX) symbols are processed by the baseband precoder P_{BB}^T. Then, each baseband signal is processed by the RF precoder P_{RF}^T and fed to the one of the N_S^T subarrays with K_A^T antennas each. K_A^T antennas in a particular subarray at the transmitter can form a RF signal with a direction towards one of the N_{Beams}^T beams. The reverse of this operation is performed at the receiver.

datasets with a distributed manner on a cluster of servers [12]. It is commonly used to execute data-intensive tasks such as data sorting, which is a key step in most of the machine learning algorithms [13]. Therefore, this framework would be very suitable to find the optimum RF and baseband precoders at the transmitter and the receiver, which is actually a distributed data sorting problem. In our recent paper, we propose a MapReduce based hybrid beamforming to reduce the computational complexity of finding the optimum RF and baseband precoders by solving this problem in a parallel fashion [14]. At the beginning of this algorithm, all possible combinations of RF and baseband precoders are divided into multiple cores. Map or reduce tasks are assigned to each core. The core which executes map and reduce task is called as mapper and reducer, respectively. Each mapper computes mutual information obtained by the assigned precoders and generates intermediate key/value pairs which denote precoder indexes/mutual information. The intermediate key/value pairs are passed to the reducer which sorts the mutual information and finds precoders that achieve the maximum mutual information. The reducer returns the indexes of the optimum precoders as the output. [14] shows that a linear relationship is obtained between speed-up in hybrid beamforming algorithm and the cores. In this paper, they also propose an optimized MapReduce based hybrid beamforming algorithm which partitions the precoder matrices into submatrices and runs the conventional algorithm on each submatrix separately. They show that the nearly same performance is achieved with the optimized algorithm in terms of bit error rate (BER).

In this paper, we focus on deeply analyzing the optimized and the conventional MapReduce based hybrid beamforming algorithms. This paper's main contributions are:

1) We give an analysis for the computational complexity and the communication load of the optimized and the conventional MapReduce based hybrid beamforming algorithms.
2) We show the optimum number of cores to run the optimized and the conventional algorithms in terms of the computational complexity and the cost due to the communication load.

The paper is organized as follows. In Section II, the system and the channel models that we use in this paper are summarized. In Section III, we explain MapReduce based hybrid beamforming algorithm. In Section IV, we analyze the computational complexity and the communication load of the optimized MapReduce based hybrid beamforming algorithm. We also give the optimum number of cores in terms of computational complexity and the communication load to run the conventional and the optimized algorithms. In Section V, we present our simulation results. We conclude our work in Section VI.

II. System and Channel Models

In this paper, we consider a hybrid beamforming architecture in which an array of antennas is divided into multiple subarrays at the transmitter and the receiver. The hybrid beamforming architecture is shown in Figure 1. Each RF chain at the transmitter and the receiver feeds one of the subarrays in this architecture. The transmitter and the receiver antenna arrays consist of N_A^T and N_A^R antennas, respectively. Antenna array at the transmitter (receiver) is divided into N_S^T (N_S^R) subarrays with K_A^T (K_A^R) antennas, respectively. We consider N_{Beams}^T and N_{Beams}^R number of beams at the transmitter and the receiver subarray, respectively. We denote MIMO channel in this architecture with a complex matrix $H \in \mathbb{C}^{N_A^T \times N_A^R}$. Transmitter baseband and RF precoders are denoted by $P_{BB}^T \in \mathbb{C}^{N_S^T \times N_L}$ and $P_{RF}^T \in \mathbb{C}^{N_A^T \times N_S^T}$, respectively. The receiver RF precoder is shown with $P_{RF}^R \in \mathbb{C}^{N_A^R \times N_S^R}$. N_L denotes the number of layers.

The received symbols vector of dimension $N_S^R \times 1$ is given as:

$$y = {P_{RF}^R}^* H P_{RF}^T P_{BB}^T x + n, \tag{1}$$

where $(.)^*$ denotes the conjugate transpose operation, x is the transmitted symbols vector $N_L \times 1$ and n is the noise vector of dimension $N_S^R \times 1$ with i.i.d. $CN(0, \sigma^2)$ entries.

In order to model mmW channel, various measurements have been carried on [3], [15], [16]. Ray-cluster based channel models are well-suited for mmW channel which is characterized by a finite number of scatterers. In ray-cluster based channel model, each scatterer produces a cluster of channel rays. The authors of [11] give a ray-cluster based spatial channel model for mmWs. We use the model which is given in [11] in this paper. The channel representation based on this model is shown as:

$$H = \sqrt{N_A^T N_A^R} \sum_{i=0}^{C-1} \sum_{j=0}^{r-1} G_{i,j} \mathbf{a}^R(\phi_{i,j}^{AoA}, \theta_{i,j}^{AoA}) \mathbf{a}^T(\phi_{i,j}^{AoD}, \theta_{i,j}^{AoD}), \tag{2}$$

where $G_{i,j}$, $\phi_{i,j}^{AoA}$, $\phi_{i,j}^{AoD}$, $\theta_{i,j}^{AoA}$, and $\theta_{i,j}^{AoD}$ denote the complex gain, azimuthal AoA, azimuthal AoD, elevation AoA, and elevation AoD of ray j in cluster i, respectively. We consider there are C clusters and each of the cluster consists of r rays. The array response vectors of the receiver and the transmitter arrays are defined as $\mathbf{a}^R(.)$ and $\mathbf{a}^T(.)$.

III. MapReduce based Hybrid Beamforming

The general problem of hybrid beamformer design is to jointly optimize the transmitter/receiver RF precoders and the transmitter baseband precoder based on channel measurements. Channel measurements can be done by using training sequences, which are transmitted from a particular beam of each transmitter subarray to a particular beam of each receiver subarray. By using the measured channel, the achieved mutual information with all possible RF precoders at the transmitter and the receiver and the baseband preocder at the transmitter is calculated. The solution to the optimization problem given in (3) are the optimum precoders which maximize the mutual information.

$$\underset{P_{BB}^T, P_{RF}^T, P_{RF}^R}{\text{argmax}} \quad \log_2 det\left(I + \frac{1}{\sigma^2}\widetilde{H}^* \widetilde{H}\right), \tag{3}$$

where $\widetilde{H} = {P_{RF}^R}^* \hat{H} P_{RF}^T P_{BB}^T$ and $\hat{H}$ is the estimated channel based on the measurements obtained with training sequences. Least-squares (LS) channel estimation method is used in this paper. P_{BB}^T, P_{RF}^T, and P_{RF}^R are selected from codebook C_{BB}^T, C_{RF}^T, and C_{RF}^R, respectively.

The number of all different precoder combinations is given below:

$$N = (N_{Beams}^R)^{N_S^R} \times (N_{Beams}^T)^{N_S^T} \times N_B^T, \tag{4}$$

where $(N_{Beams}^R)^{N_S^R}$, $(N_{Beams}^T)^{N_S^T}$, and N_B^T are the number of different precoder matrices to possibly build for the RF receiver, the RF transmitter, and the baseband transmitter, respectively. The mutual information, which involves four matrix multiplications and one determinant operation, needs to be calculated for each precoder combinations. Therefore, we should execute $4 \times N$ number of matrix multiplications and N number of determinant operations to solve the hybrid optimization problem, given in (3). As it is explained in Section I, the number of computations for finding the solution to (3) grows rapidly.

One can use MapReduce framework to easily implement the parallel version of an algorithm which involves data-intensive tasks. In order to design the optimum precoders in hybrid beamforming, the mutual information obtained with all different combination of precoder matrices need to be sorted. The precoder matrices, which achieve the maximum mutual information, are the optimum among all. Since this is simply a data sorting problem, hybrid beamforming algorithm can be implemented by using MapReduce framework. We propose a MapReduce based hybrid beamforming algorithm, which parallelizes the computations required to find the optimum precoders, in our recent paper [14]. The MapReduce based hybrid beamforming algorithm is summarized in Figure 2.

There are N possible combinations for the precoder matrices P_{RF}^T, P_{RF}^R, and P_{BB}^T to design the hybrid beamformer. Let $p_i = 1, ..., (N_{Beams}^R)^{N_S^R}$, $p_j = 1, ..., (N_{Beams}^T)^{N_S^T}$, and $p_k = 1, ..., N_B^T$ denote the indexes of P_{RF}^R, P_{RF}^T, and P_{BB}^T, respectively. First, a master controller divides the input data into multiple subsets, then it assigns map and reduce tasks to the idle cores. Each subset consists of N/L different combinations of matrices. Each mapper calculates the mutual information for N/L different matrix combinations as given below:

$$I = \log_2 det(I + \frac{1}{\sigma^2}\widetilde{H}^* \widetilde{H}). \tag{5}$$

The channel frequency response $\hat{H} \in \mathbb{C}^{N_A^T \times N_A^R}$ can be estimated with a pilot-based channel estimation method such as LS. We assume that each mapper knows the channel frequency response. Mutual information of the assigned precoder matrices, and the corresponding indexes of the precoders p_i, p_j, and p_k are generated as intermediate key/value pairs by each mapper. These intermediate key/value pairs are stored in local disks whose locations are known by the master. Then, the master sends these locations to the reducer. Reducer sorts the mutual information obtained by different precoder matrix selections and finds the maximum mutual information. The indexes of the precoder matrices that achieve the highest mutual information are returned as output values by the reducer.

IV. Analysis for Computational Complexity and Communication Load in MapReduce based Hybrid Beamforming

The computational complexity of MapReduce based hybrid beamforming has been studied in [14]. For N number of precoder combinations, the equation given in (5) needs to be computed. $M_1 = {P_{RF}^R}^* \hat{H}$ has complexity $O(N_S^R \times N_A^R \times$

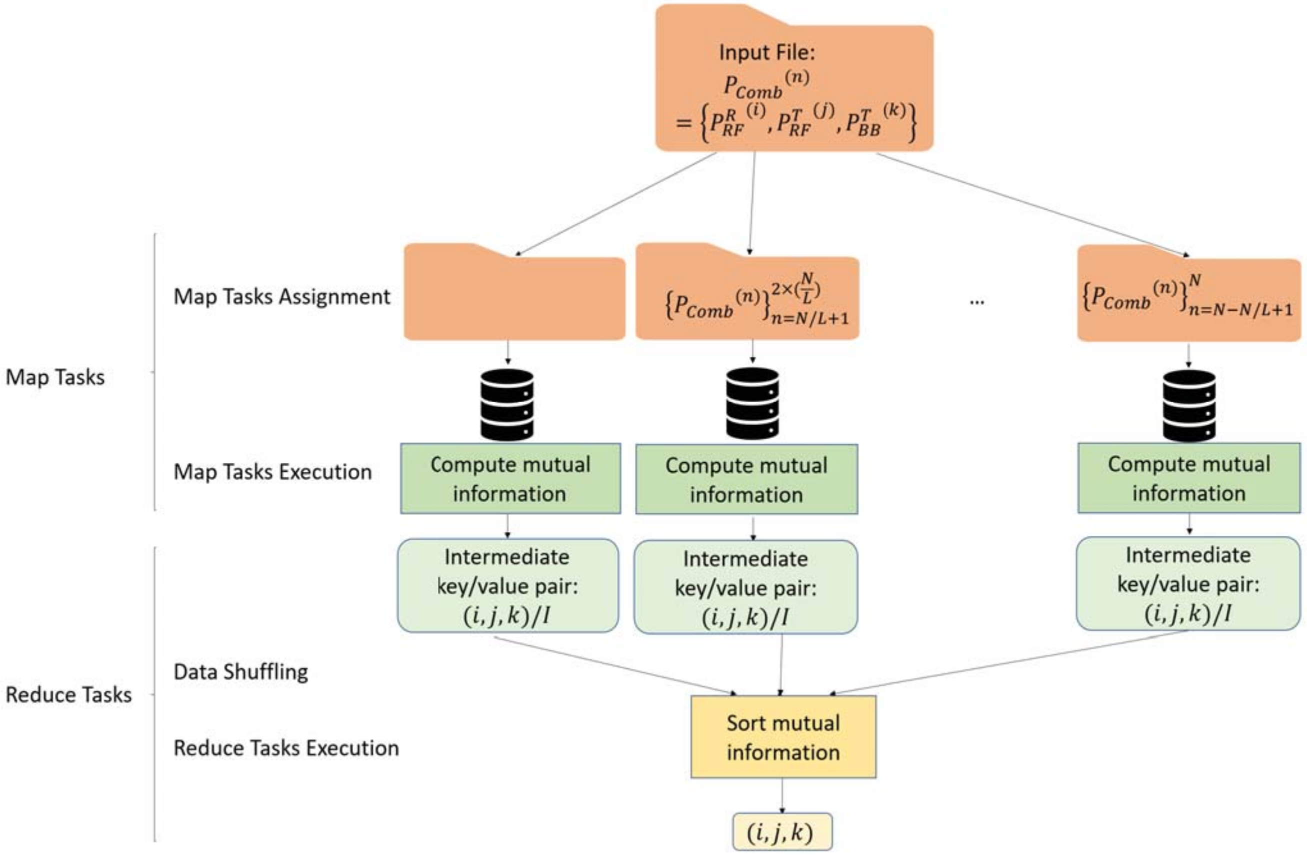

Input File:
$P_{Comb}{}^{(n)}$
$= \{P_{RF}^{R}{}^{(i)}, P_{RF}^{T}{}^{(j)}, P_{BB}^{T}{}^{(k)}\}$
Map Tasks Assignment
$\{P_{Comb}{}^{(n)}\}_{n=N/L+1}^{2\times(\frac{N}{L})}$
...
$\{P_{Comb}{}^{(n)}\}_{n=N-N/L+1}^{N}$
Map Tasks
Map Tasks Execution
Compute mutual information
Compute mutual information
Compute mutual information
Intermediate key/value pair: $(i,j,k)/I$
Intermediate key/value pair: $(i,j,k)/I$
Intermediate key/value pair: $(i,j,k)/I$
Data Shuffling
Reduce Tasks
Reduce Tasks Execution
Sort mutual information
(i,j,k)

Fig. 2: MapReduce based hybrid beamforming algorithm structure is shown. $i = 1, ..., (N_{Beams}^R)^{N_S^R}$, $j = 1, ..., (N_{Beams}^T)^{N_S^T}$, and $k = 1, ..., N_B^T$ denote indexes of P_{RF}^R, P_{RF}^T, and P_{BB}^T, respectively. There are N possible combinations of the precoder matrices and n^{th} combination is shown by $P_{Comb}^{(n)}$. N/L number of precoder combinations are assigned to one of L cores. I denotes the mutual information obtained by the selected precoder matrices.

N_A^T). The complexity of $M_2 = M_1 P_{RF}^T$ is $O(N_S^R \times N_A^T \times N_S^T)$. $\widetilde{H} = M_2 P_{BB}^T$ has complexity $O(N_S^R \times N_S^T \times N_L)$. The complexity of $\widetilde{H}^* \widetilde{H}$ is $O(N_L \times N_S^R \times N_L)$. The determinant operation has complexity $O(N_L^3)$. The total computational complexity for calculating (5) is $O(N_S^R \times N_A^R \times N_A^T + N_S^R \times N_A^T \times N_S^T + N_S^R \times N_S^T \times N_L + N_L^2 \times N_S^R + N_L^3)$. The computational complexity of hybrid beamforming, which calculates (5) for N different combinations of precoders, is $O(N \times (N_S^R \times N_A^R \times N_A^T + N_S^R \times N_A^T \times N_S^T + N_S^R \times N_S^T \times N_L + N_L^2 \times N_S^R + N_L^3))$. The computational complexity of MapReduce based hybrid beamforming is $O(N \times (\frac{N_S^R \times N_A^R \times N_A^T}{L^{'}} + \frac{N_S^R \times N_A^T \times N_S^T}{L^{'}} + \frac{N_S^R \times N_S^T \times N_L}{L^{'}} + \frac{N_L^2 \times N_S^R}{L^{'}} + \frac{N_L^3}{L^{'}}))$. Here $L^{'} \approx L$ and L is the number of cores. We achieve $L^{'}$, which approximately equals to the number of cores L, speed-up gain in hybrid beamforming algorithm.

We also show in [14] that the communication load of MapReduce based hybrid beamforming due to the data shuffling between different cores is given as:

$$CommLoad = QN\left(1 - \frac{1}{L}\right). \tag{6}$$

We assume that each core maps N/L subfiles and reduces Q/L keys. Q is the number of total intermediate values. Since the communication load increases linearly with N, the speed-up gain decreases with the large values of N. In this case, there is a trade-off between the computational complexity and the communication load. The communication load is the bottleneck of MapReduce based hybrid beamforming algorithm.

In [14], we also show that we can further reduce the computational complexity of MapReduce based hybrid beamforming algorithm by dividing the precoder matrices into submatrices. Let us analyze the computational complexity of the optimized MapReduce based hybrid beamforming algorithm which partitions precoder matrices into multiple submatrices. We assume that RF precoder matrix P_{RF}^T at the transmitter is divided into dimension of $K_A^T \times (N_S^T)^{'}$ submatrices and RF precoder matrix P_{RF}^R at the receiver is divided into dimension of $K_A^R \times (N_S^R)^{'}$ submatrices. $(N_S^T)^{'}((N_S^R)^{'})$ can be 1 and $N_A^T(N_A^R)$ as minimum and maximum, respectively. We set $(N_S^T)^{'} = (N_S^R)^{'} = 1$ and solve the optimization problem given in (3) separately for each submatrix. Optimization problem given in (3) is solved for each submatrix in the optimized MapReduce based hybrid beamforming algorithm. In this case, the total number of precoder combinations is given as:

$$N_{Opt} = N_S^R \times N_S^T \times N_{Beams}^R \times N_{Beams}^T \times N_B^T. \quad (7)$$

In each computation, the mutual information is calculated for the selected precoder matrices. This calculation consists of four matrix multiplications and one determinant operation. Let us calculate the computational complexity of calculating mutual information for the selected precoder matrices. $P_{RF}^{T}{}' \in \mathbb{C}^{K_A^T \times 1}$ and $P_{RF}^{R}{}' \in \mathbb{C}^{K_A^R \times 1}$ denote one submatrix of RF precoders at the transmitter and the receiver, respectively. $P_{BB}^{T}{}' \in \mathbb{C}^{1 \times N_L}$ and $\hat{H}' \in \mathbb{C}^{K_A^T \times K_A^R}$ denote one submatrix of the baseband precoder at the transmitter and the estimated channel coefficients matrix, respectively. $M_1 = P_{RF}^{R}{}'^{*} \hat{H}'$ has complexity $O(K_A^R \times K_A^T)$. The complexity of $M_2 = M_1 P_{RF}^{T}{}'$ is $O(K_A^T)$. $\widetilde{H}' = M_2 P_{BB}^{T}{}'$ has complexity $O(N_L)$. The complexity of $(\widetilde{H}')^* \widetilde{H}'$ is $O(N_L^2)$. The determinant operation in (3) has complexity $O(N_L^3)$. In this case, the total number of computations that occurs in calculation of mutual information is $O(K_A^R \times K_A^T + K_A^T + N_L + N_L^2 + N_L^3)$. These computations occur for N_{Opt} number of precoder combinations, so the computational complexity of the optimized hybrid beamforming is $O(N_{Opt} \times (K_A^R \times K_A^T + K_A^T + N_L + N_L^2 + N_L^3))$. These computations can be sped up by a factor of $L' \approx L$ on L cores by using MapReduce framework. Therefore, the computational complexity of optimized MapReduce based hybrid beamforming algorithm becomes $O(N_{Opt} \times (\frac{K_A^R \times K_A^T}{L'} + \frac{K_A^T}{L'} + \frac{N_L}{L'} + \frac{N_L^2}{L'} + \frac{N_L^3}{L'}))$.

The communication load can be decreased with the optimized MapReduce based hybrid beamforming algorithm since N_{Opt} is less than N. The communication load of the optimized MapReduce based hybrid beamforming can be calculated as:

$$OptCommLoad = QN_{Opt}\left(1 - \frac{1}{L}\right). \quad (8)$$

V. Simulation Results

We obtained the results of our analysis in Section IV by using MATLAB. We assume a MIMO system with 16 and 8 antennas at the transmitter and the receiver, respectively. There are 3 available beams in the codebooks of the RF precoders at the transmitter and the receiver. The baseband precoder is chosen from 2×2 codebook which is defined in [17].

In Figure 3, we show number of operations required to run the MapReduce based hybrid beamforming and the optimized MapReduce based hybrid beamforming algorithms while number of cores (L) increases from 1 to 16. We obtain these results when N_S^T and N_S^R are chosen as 2 and 4. We observe that the number of operations decreases for both of the algorithms while L increases. The computational complexity of the optimized algorithm is less than the computational complexity of the conventional algorithm. For example, the number of operations is decreased by a factor of 14 when $L = 16$ and $N_S^T = N_S^R = 2$. According to the results in Figure 3, the computational complexity of the conventional algorithm increases more than the computational complexity of the optimized algorithm when the number of subarrays is doubled at the transmitter and the receiver. Therefore, the improvement in the reduction of the computational complexity increases as the number of subarrays increases.

In Figure 4, we observe the communication load, which denotes the number of data shuffling occurs between any two cores, while L increases from 1 to 16. N_S^T and N_S^R are set as 2 and 4. It can be seen in Figure 4 that the communication load of both of the algorithms increases with the number of cores. The communication load is decreased with the optimized MapReduce based hybrid beamforming algorithm. For instance, the communication load of the optimized algorithm is 2 times of the communication load of the conventional algorithm when $L = 16$ and $N_S^T = N_S^R = 2$. The communication load of both algorithms also increases significantly for the larger number of subarrays. The gain in decrease in the communication load also further improves with the increasing number of subarrays at the transmitter and the receiver. For example, the communication load is decreased by almost a factor of 46 with the optimized algorithm when number of cores is 16 and $N_S^T = N_S^R = 4$.

There is a trade-off between the computational complexity and the communication load for both of the algorithms when the number of cores increases. According to the results in Figure 3 and Figure 4, the computational complexity significantly decreases while the communication load increases with the number of cores for both algorithms. Our aim is to show the optimum number of cores based on this trade-off. The optimum number of cores depends on the upper bound on the number of operations and the communication load that is allowed in the system. For example, when the maximum number of operations and the communication load are restricted to 2000 and the number of subarrays is 2 at the transmitter and the receiver, the optimum number of cores is 4 and 8 for the optimized and the conventional algorithms, respectively. In general, the optimum number of cores required for the optimized algorithm is less than the conventional algorithm.

VI. Conclusion

In this paper, we studied MapReduce based hybrid beamforming algorithms, and investigated thetrade-off between computational complexity and communication load. We analyze these tradeoffs for two algorithmic variations, namely, conventional and optimized versions of hybrid beamforming. Our results show that both computational complexity and communication load can be reduced at the expense of increasing the number of cores. Furthermore, we also analyze the optimum number of cores for both algorithmic variations as a function of the operating point on the trade-off between the computational complexity and the communication load.

VII. Acknowledgements

This project was partially supported by the Broadband Wireless Access and Applications Center (BWAC); NSF

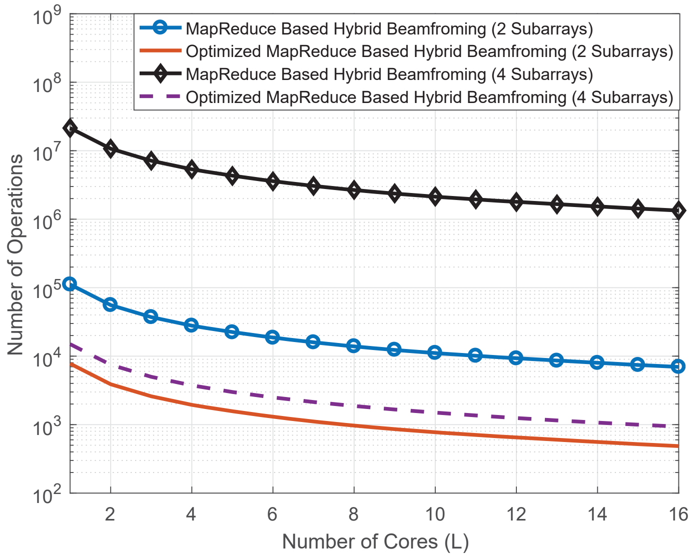


Fig. 3: Computational complexity comparison of the optimized and the conventional MapReduce based hybrid beamforming algorithms when the number of subarrays at the transmitter and the receiver is 2 and 4

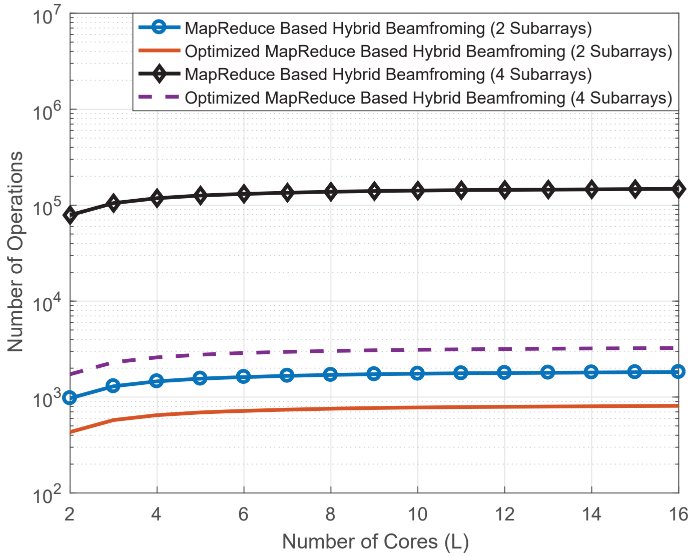


Fig. 4: Communication load comparison of the optimized and the conventional MapReduce based hybrid beamforming algorithms when the number of subarrays at the transmitter and the receiver is 2 and 4

Award No. 1265960.

REFERENCES

[1] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 101–107, June 2011.

[2] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[3] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, June 2014.

[4] J. Mietzner, R. Schober, L. Lampe, W. H. Gerstacker, and P. A. Hoeher, "Multiple-antenna techniques for wireless communications - a comprehensive literature survey," *IEEE Communications Surveys Tutorials*, vol. 11, no. 2, pp. 87–105, Second 2009.

[5] W. Liu and S. Weiss, *Wideband Beamforming: Concepts and Techniques*. Wiley Publishing, 2010.

[6] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Simon & Schuster, 1992.

[7] P. Sudarshan, N. B. Mehta, A. F. Molisch, and J. Zhang, "Channel Statistics-Based RF Pre-Processing with Antenna Selection," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3501–3511, December 2006.

[8] X. Zhang, A. F. Molisch, and S.-Y. Kung, "Variable-phase-shift-based RF-baseband codesign for MIMO antenna selection," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4091–4103, Nov 2005.

[9] T. Kim, J. Park, J.-Y. Seol, S. Jeong, J. Cho, and W. Roh, "Tens of Gbps support with mmWave beamforming systems for next generation communications," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 3685–3690.

[10] S. Han, C. l. I, Z. Xu, and C. Rowell, "Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5g," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 186–194, January 2015.

[11] J. Singh and S. Ramakrishna, "On the Feasibility of Codebook-Based Beamforming in Millimeter Wave Systems With Multiple Antenna Arrays," *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2670–2683, May 2015.

[12] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, ser. OSDI'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 10–10.

[13] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, "Parallel data processing with mapreduce: A survey," *SIGMOD Rec.*, vol. 40, no. 4, pp. 11–20, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2094114.2094118

[14] T. Peken, R. Tandon, and T. Bose, "On the Efficient Hybrid Beamforming in Millimeter Wave Systems," *IEEE Transactions on Wireless Communications*, submitted for publication **.

[15] H. Xu, V. Kukshya, and T. S. Rappaport, "Spatial and temporal characteristics of 60-ghz indoor channels," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 3, pp. 620–630, Apr 2002.

[16] E. Ben-Dor, T. S. Rappaport, Y. Qiao, and S. J. Lauffenburger, "Millimeter-wave 60 ghz outdoor and vehicle aoa propagation measurements using a broadband channel sounder," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Dec 2011, pp. 1–6.

[17] "3rd Generation Partnership Project, Technical specification group RAN;Technical specification group RAN; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation," 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36211.htm

Implementation of VPN-SSL on OpenBTS as Data Security Transmitted between BTS and VoIP

Rino[1], Irfan Setiadi[2]
[1]Binus University, anakrangunpalsan@gmail.com
[2]Open University, irfaaniumuchem@gmail.com

ABSTRACT

OpenBTS is an open source software that utilizes a hardware device called USRP (Universal Software Radio Peripheral) as a transmitter and receiver of frequencies. OpenBTS also uses Asterisk's open source software for interconnection with other telephone networks such as PSTN (Public Switched Telephone Network) or other telecommunication operators using VoIP (Voice over IP). However in this communication there is a vulnerability to data transmitted between BTS via Ethernet.

The main objective of this research is to build GSM cellular networks using OpenBTS and USRP as a medium of information exchange in areas not yet covered by cellular services. In addition to provide communication security services, data transmitted will be secured using a VPN - SSL with SEED algorithm based on symmetric cryptography.

Keywords: Universal Software Radio Peripheral (USRP), SEED symmetric cryptographic algorithm, Global System for Mobile Communication (GSM), Base Transceiver Station (BTS), VPN-SSL.

1. Preliminary

1.1 Background

Mobile communication network is one of the services those are widely used for the exchange information on the mobile communication network mainly based on Global System for Mobile Communication (GSM). Almost all mobile communication developed at this time using GSM-based cellular communications network, because the network is a GSM mobile telecommunications network that has a standard architecture of the European Telecommunications Standards Institute (ETSI) [1]. Currently, GSM has become the global standard for communication, both in Indonesia and in the world [2].

In the exchange information is not easy when you're in an area that not yet covered by the BTS signal or can be called by blankspot area. To overcome these problems, one way to do that is, build Open Base Transceiver Station (OpenBTS) as a GSM-based mobile communication network to exchange information on blankspot area. In addition, this technology is so useful for building telecommunication networks in remote, rural, rural and disaster areas. Because if conventional BTS is built, the probability of tower success in these areas is so small and the required cost is quite large.

In the Fuadi's research in 2012, OpenBTS designed by implementing an asterisk in Ubuntu 10.10, you can use open source software that runs on Linux platform [11]. This OpenBTS Software utilizes Universal Software Radio Peripheral (USRP) as a medium to send and receive data via the GSM cellular. While OpenBTS will be investigated on this occasion there is a difference, namely the implementation of VPN-SSL using SEED algorithm on OpenBTS to secure data transmitted between BTS and VoIP. VPN-SSL implementation using SEED algorithm is aimed as security needs that can enable to obtain high level of strength and adequate speed in a network. [12]

1.2 Problem Formulation

The problem formulation in this research are:

1. Is the GSM-based mobile communication network using OpenBTS software can provide solution to blankspot area?

2. How to implement VPN-SSL on OpenBTS for communication between BTS and VoIP?
3. Is the VPN-SSL using SEED algorithm based symmetric cryptography can provide security service data transmitted between BTS and VoIP?

1.3 Objectives and Benefits of Research

In this study built GSM cellular communication system with OpenBTS as a solution to the blankspot area and carried VPN-SSL implementation on OpenBTS as a medium of exchange secure information on blankspot area as well as added security services on data transmission that is sent between BTS and VoIP. It is expected that this research can also enrich the library of education literature, as well as a reference in the field of security applications based on OpenBTS.

2. Theoritical Basis

2.1 GSM (*Global System for Mobile Communication*)

GSM (Global System for Mobile communication) is a technology used in communication with the digital systems and networks that has been already global. As a technology that can be said to be quite revolutionary because it successfully shifted the popular analog mobile telecommunication system technology in the decade of the 80s, GSM has provided a new communication alternative for the world of better telecommunication. By using a digital signal system in data transmission, the quality of data produced is better than analog system. In daily life we are more familiar with Handphone (HP) as the application of the most popular GSM technology. Since the first GSM implementation until now has been developed into three groups: the GSM 900, 1800 and 1900. The third difference is located based on group of frequency bands used. GSM 900 uses 900 MHz frequency as its transmission channel. GSM 1800 and 1900 each use the 1800 and 1900 MHz frequencies. Picture of GSM network architecture shown in Figure 1. [11]

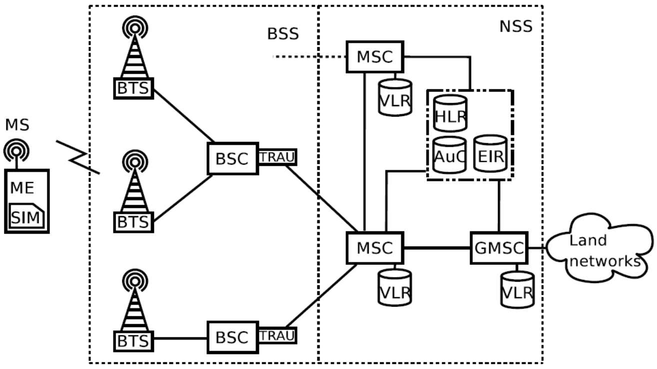


Figure 1. GSM network architecture

A GSM network is built of several functional components which have special function and interface. In general, GSM network divided into three main parts, namely:

1) *Mobile Station* (MS)

MS is a device used by customers to communicate. MS consists of Mobile Equipment (ME) and a Subscriber Identity Module (SIM).

- *Mobile Equipment* (ME) is a radio transmission terminal equipped with International Mobile Equipment Identity (IMEI), while the SIM provides the customer identification number to enter the GSM operator's network. ME serves as a transceiver (transmitter and receiver), as a transducer that converts sound signals into electric signals, monitor the condition of power and signal quality of the surrounding cells, and has a memory for storing user data.
- *Subscriber Identity Module* (SIM) has a microprocessor and a memory to store some users' data. Inside the SIM there are some important information that is used in communication such as the International Mobile Subscriber

Identity (IMSI), Authentication Key (consisting of algorithms A3 and A8) were used during the authentication process and it contains Personal Identification Number (PIN) and PIN Unblocking Key (PUK).

2) Base Station System (BSS)

BSS consists of three devices:

- *Base Transceiver Station (BTS)* is a transmitter and receiver device that handle radio access and interacts directly with MS through the air interface. BTS also sets the handover process that happens inside BTS and monitored by BSC.
- *Base Station Controller (BSC)* is the interface between the BTS to Mobile Switching Center (MSC) and Operation And Maintenance Center (OMC). BSC controls several BTSs and arranges traffic in and out from BSC to MSC or BTS. BSC sets of radio resource in the provision of frequency for each BTS and arranges handover when the MS crosses the line between cells.
- *Transcoder (XCDR)* serves to compress data or voice output from the MSC (64 Kbps) to 16 Kbps in the direction of the BSC and vice versa for the efficiency of the transmission channels.

3) *Network Switching System* (NSS)

NSS works as switch in a GSM network, sets the network, and becomes a media interface between the GSM network with other networks. NSS component on GSM network consists of:

- *Mobile Switching Center* (MSC), a network elements central in a GSM network. MSC as the core of the cellular network, where the MSC plays a role for interconnection of speech, either between cellular or with PSTN cable network, or with data network. MSC is responsible for managing communication between customers and other telecommunication network users.
- Home Location Register (HLR) is a database containing customer data remains an area of coverage. These data include customer services, additional service and information about the most recent customer location.
- Visitor Location Register (VLR) is a database that contains temporary information about subscribers who are roaming from another coveraged area.
- Authentication Center (AuC) contains confidential database that is stored in the form of a code format for securing and controlling the legal-based use of mobile systems and prevent customer fraud.
- Equipment Identity Register (EIR) is a centralized database that the function is to validate the International Mobile (IM).
- Inter Working Function (IWF) serves as an interface between a specific GSM network and other GSM networks.
- Echo Canceller (EC) is used as PSTN connector for reducing the echo and delay.

2.2 OpenBTS Architecture

OpenBTS is a prototype of GSM-based mobile communications network consisting of software and hardware. The software used can be downloaded for free like asterisk, GNURadio, etc. While the hardware used is the Universal Software Radio Peripheral (USRP). Both devices are what makes OpenBTS like a standard GSM mobile phone network. A software like asterisk is used to interconnect with other phone networks such as the Public Switched Telephone Network (PSTN) or other telecommunication operators by using Voice over IP (VoIP).[3] The OpenBTS architecture image is shown as in Figure 2. [1]

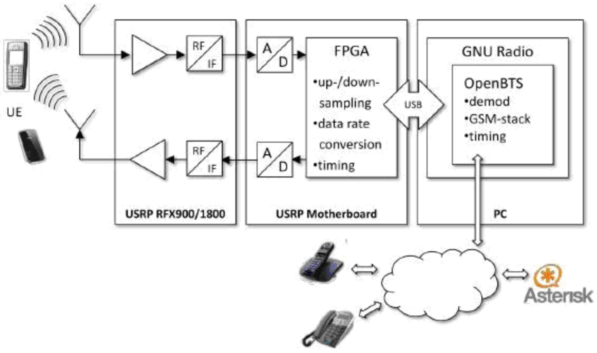


Figure 2. OpenBTS Architecture

1) Asterisk

Asterisk is an open source software that is usually used to build a system of communication services and provide more convenience for users to develop their own phone services with the broadest customization given to the user. From the definition of open source itself means that any developer can view and modify the source code available, so that existing applications can be added easily by any developer. Asterisk can also be regarded as a complete PBX in the form of software, and provides all the features like PBX. PBX (private branch exchange) is a telephone service provider that seves a telephone exchange with the center. [9]

2) Universal Software Radio Peripheral (USRP)

Universal Software Radio Peripheral (USRP) is a radio software based highly speed Digital Signal Processing (DSP). USRP is hardware and currently has several versions. The most recent version is using Gigabit Ethernet that can be stored above the tower easily so it can cover a wide area. [5]

3) GNU Radio

GNU Radio is one of the software that will be used in operating OpenBTS. GNU Radio is a set of device that provide signal processing. One of the advantage of GNU Radio is a software with open source code and free software. [5]

2.3 VPN (Virtual Private Network)

VPN or Virtual Private Network is a private connection via a public network or the Internet. Virtual network means the network must be characterized as virtual. Private mens nobody can access the network. The sent data will be encrypted so it remains confidential even though over a public network. Using VPN likes making a network inside network called by tunnel. VPN uses one of the three existing tunneling technology are: PPTP, L2TPand latest standards, Internet Protocol Security (commonly abbreviated as IPSec). VPN is a combination of tunneling and encryption technology.

How VPN works VPN (with the PPTP protocol) are:

- VPN requires a server to connect the PC, thisVPN server can be a computer with VPN application server or a router
- To start a connection, a computer with Client VPN application contacts the VPN server, VPN server then verifies the username and password, and if successful then the VPN Server delivers a new IP address on the client computer and then a connection / tunnel will be formed.
- Furthermore, the client computer can be used to access various resources (computer or LAN) located behind the VPN Server such as data transfer, document printing, browsing with the gateway provided from the VPN Server, remote desktop and so forth.

2.3.1 VPN Function

VPN technology provides three main functions for its users. The main functions are as follows:

- **Confidentiality**
  VPN technology has a working system to encrypt all the data passing through it. With this encryption technology, then your secrecy becomes more awake. Even if there are parties who can tap your data passing by, but not necessarily they can read it easily because it is already randomized. By implementing this encryption system, no one can access and read the contents of your data network easily.
- **Data Integrity**
  As it passes through the Internet network, your data has actually gone so far further across countries. In the middle of his journey, anything can happen to its content. Whether it is lost, damaged, even manipulated by bad person. VPN has technology that can keep the integrity of the data you send in order to arrive at its destination without being flawed, lost, corrupted, or manipulated by others.
- **Orign Authentication**
  VPN technology has the ability to authenticate the sources of data senders that will be received. VPN will check all incoming data and retrieve its data source information. Then the source address of this data will be approved if the authentication process succeeds. Thus, the VPN guarantees that all data sent and received by you, comes from the appropriate source. No data is forged or transmitted by other parties.

2.4 SSL (Secure Socket Layer)

According Stalling (2005), SSL is a tunneling layer transport protocol that is often used. SSL has several applications and can easily be used to build a tunnel layer transport. The protocol uses three cryptographic functions, namely:

1. **Key exchange**
   Both parties need a way to exchange keys. Part of the key exchange can provide the authentication process on the server.
2. **Data encryption**
   In this SSL there is a method for encrypting the data executed in this protocol. It uses symmetric algorithm, both stream and block ciphers.
3. **Authentication**
   In each transmitted record, it must be authenticated first. This can be done by adding a secret message of digest Hash Message Aunthentication Code (HMAC) for each record.

3. Research methodology

In this study the method used is qualitative. In qualitative research, the basic theory is used as a foothold or reference. This is done by means of techniques of data collecting, data analysis, system design, implementation, and testing.

The data used are descriptive, where the data can be photographs, documents, and field notes at the time of research is in progress. The technique used in qualitative method is observation technique, where the researcher is directly involved with the research. With the technique of document review, researchers will conduct a review of the documents.

3.1 Data Collection Technique

Data collection in this research is done by literature and interview technique. Literature technique is done by studying the theories related to research, such as SEED Algorithm-based symmetrical, Global System for Mobile Communication (GSM), Open Base Transceiver Station (OpenBTS), Virtual

Private Network (VPN), Secure Socket Layer (SSL), Wireshark, and others. In addition to studying the theory, literature technique is also done by studying the research or system ever made. The interview technique is done to the resource person, either directly or indirectly. By doing interview technique, it will get a lot of precise and accurate data.

3.2 Needs Analysis

After collecting the data, the results are used to determine what kind of system will be generated. In the need analysis phase, four objectives are achieved: to explain the complete system, to describe the system ideally, to bring the ideal system into the current state by paying attention to resource constraints, and confidence in the client about the system to be created. There are two kinds of needs found in this study, namely: functional and non-functional needs. The functional analysis is to define what should and can be done by the system ie, the system provides GSM-based communication services safely on the transmission conducted between BTS. Security used is with VPN-SSL. While the non-functional need analysis does not define what the system should do, but it is how the system does what it should and can do. For example, system performance, process or cryptographic scheme used, operational, security, and policies by IEEE, Cisco, and ITU.

3.3 System Planning

The system design will be made using the method of flowchart design. It is a method that is structured in the making. If the previous stage has not been completed, then the next stage can not be done, and the system built is not running. System design image created by using a flowchart shown in Figure 3.

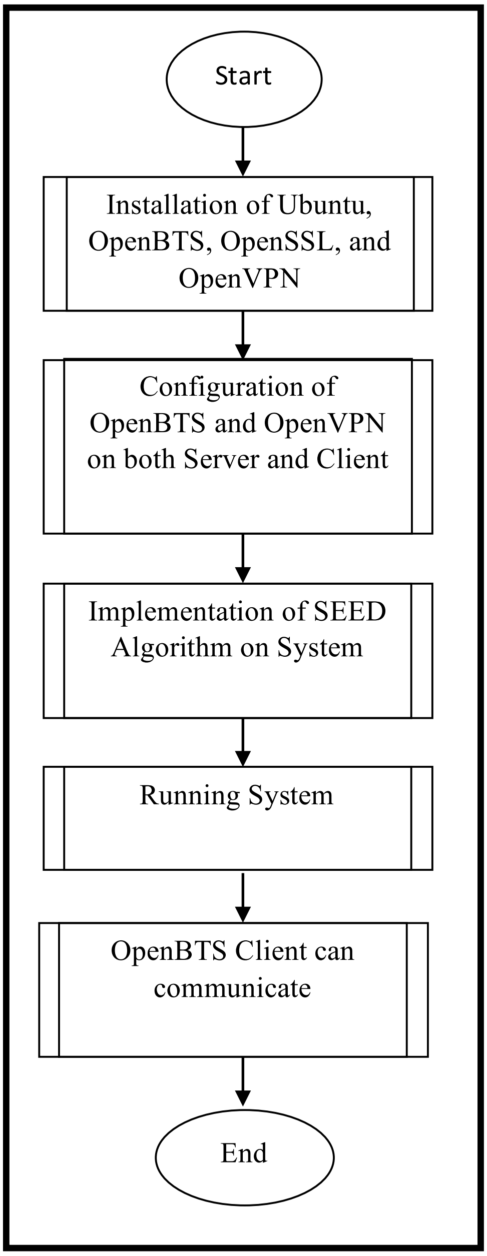

Start
Installation of Ubuntu, OpenBTS, OpenSSL, and OpenVPN
Configuration of OpenBTS and OpenVPN on both Server and Client
Implementation of SEED Algorithm on System
Running System
OpenBTS Client can communicate
End

Figure 3. General System Design

3.4 Implementation

Implementation of this system begins with installing the required software such as, Ubuntu operating system, Asterisk, GNU Radio, OpenBTS, OpenVPN, and OpenSSL. Then continued with the configuration of the system that has been installed. After that compiling between OpenVPN and OpenSSL (VPN-SSL). Then the SEED algorithm in OpenSSL is implemented on this system with the aim of securing data on the transmission between BTS and VoIP that exist in this system.

3.5 Testing

In this research, the system has been made was tested. The tests performed are as follows:

1. OpenBTS Parameter Measurement

   The purpose of this measurement is to determine the transmission quality of Mobile Station or GSM Handphome served OpenBTS when making phone calls. The transmission quality parameter analyzed are Rx level, Rx Quality, and SQI. The size of the transmission quality will provide information on whether the OpenBTS network implemented is feasible and meets the value of KPI (Key Performance Index) or not yet. While other parameters are BCCH and ARFCN to know the frequency used by OpenBTS. BCCH (broadcast control channel) itself is part of GSM channel control as the name implies is doing broadcasting of cell network data where the user is located and what the neighbour cells. While ARFCN is the code that defines a pair of radio and channel carriers that are used for transmitters and receivers in the Um interface, one for uplink signals and one for downlink signals.
2. Measurement of QoS (Quality of Service)

   Measurement of Quality of Service (QoS) aims to measure parameters that support the quality of OpenBTS and VoIP. The measured parameters include delay, jitter, throughput, and packetloss. The parameter values obtained are compared with the existing QoS standard values, whether they are within predetermined standards or not. Meanwhile, to get the values of the QoS parameters itself used Wireshark as a network protocol analyzer. As a value reference of QoS parameters obtained, then the standards of some institutions served as a reference, among others:

   a. Jitter is considered good-value <50 ms (ITU-T G.1010), and is worth <30 ms (Cisco)
   b. The best delay value between 0-150 ms (ITU-T standard)
   c. Packetloss considered good if the value of <1% (ITU-T standard)
3. CPU Usage Measurements

   Measuring performance of Asterisk server by paying attention on the CPU Usage at Asterisk server when communication occurs on the client asterisk. There are three scenarios of communication made at the time of measurement, namely, VoIP to VoIP, OpenBTS to VoIP, and OpenBTS to OpenBTS, so we can determine how much CPU Usage amount used in these three processes. Going forward, there will be an estimated number of users that can be handled by the server at a time.
4. Testing Data Traffic Security

   Knowing and proving the performance of VPN-SSL by capturing data packet transmitted to the server when communication occurs on the client. There are three scenarios of communication made at the time of measurement, namely, VoIP to VoIP, OpenBTS to VoIP, and OpenBTS to OpenBTS, so it can be known whether data packet transmitted over VPN-SSL tunnel was encrypted or not when these three processes were done. Going forward, it can be predicted whether securing using VPN-SSL with symmetric-based SEED algorithms is effective and feasible to use or not.

3.6 Analysis

Analysis was based on the analysis from the results of the implementation of a mobile communication system that is built, the analysis of data security, performance analysis and analysis of system deficiencies. In the data security analysis, analysis is conducted based on the results of the testing process. While the analysis of system deficiencies is an analysis from the lack of systems that still need improvement or further research. Moreover, there was also conducted analysis of performance testing results where the system is built to know how strong the performance is owned.

4. Design and Implementation

Here is the OpenBTS system design in this research.

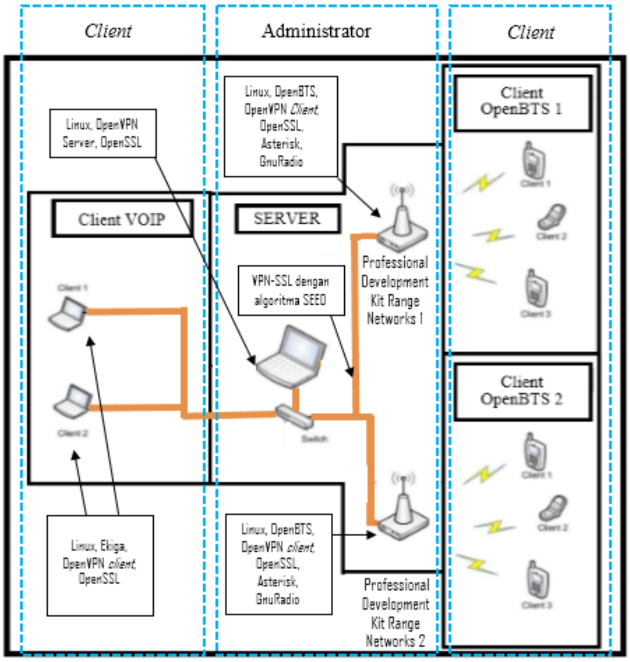

Client
Administrator
Client
Linux, OpenVPN Server, OpenSSL
Linux, OpenBTS, OpenVPN Client, OpenSSL, Asterisk, GnuRadio
Client OpenBTS 1
Client VOIP
SERVER
VPN-SSL dengan algoritma SEED
Professional Development Kit Range Networks I
Client 1
Client 2
Client 3
Switch
Client OpenBTS 2
Linux, Ekiga, OpenVPN client, OpenSSL
Linux, OpenBTS, OpenVPN client, OpenSSL, Asterisk, GnuRadio
Professional Development Kit Range Networks 2

Figure 4. System Design

From the picture above, system design is divided into several parts, namely:

- OpenBTS Client
  OpenBTS Client is a form of mobile device that is used as the MS to connect the communication between Professional Development Kit Range Networks with the server and serves as a transceiver terminal (transmitter and receiver signal) to communicate. While simcard in MS will serve to provide information on the number of International Mobile Subscriber Identity (IMSI) which can be recognized by the OpenBTS network.
- VOIP client
  VOIP client is a laptop device with software telephone in this case using Ekiga application used as a client that can be connected to the OpenBTS network using or past an IP network. This is done because they want to connect/interconnect to external network and this client can be regarded as a gateway.
- Server
  Server consists of a laptop and a Professional Development Kit Range Networks. Laptop are only used to control whether Professional Development Kit Range Networks 2 as the management of information data center information or as the Home Location Register, the running system control and user registration management well. Professional Develop-ment Kit Range Networks is a replacement from the Base Transceiver Station (BTS) which is served as a sender and recipient from emitted network signal. There are several programs and supporting applications that needed to be installed as well, such as: Asterisk, GNU Radio, OpenBTS, MySQL Server, Smqueue.

5. Testing and Analysis Results

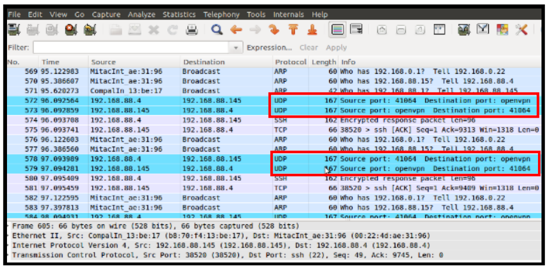
To verify the VPN capabilities which is implemented in this OpenBTS system, some tests are performed. Tests were done is Quality of Service (QoS) and data traffic security testing. From the test results obtained, are presented in the table below:

TABLE 1. OF QUALITY OF SERVICE RESULTS

No	Parameter	Result
1	Delay	• OpenBTS to VoIP = 19.98ms • VoIP to Voip = 19.99ms • Delay < 150ms (Good by ITU)
2	Jitter	• OpenBTS to VoIP = 0.82ms • VoIP to Voip = 0.92ms • Jitter is said to be good, if < 50ms by cisco and < 30ms by ITU

3	Packetloss	• OpenBTS to VoIP = 0% • VoIP to Voip = 0% • Package Loss is said to be well interrupted < 1 % by ITU
4	Throughput	The more dense the background traffic on the network will be, the less successful number of bits sent, measured in bytes per second

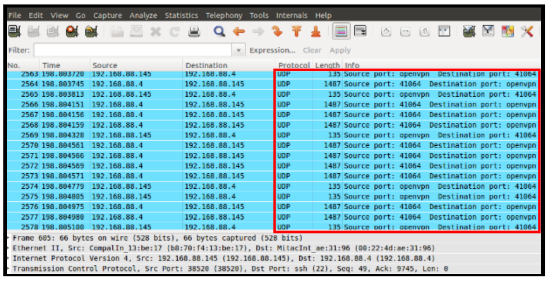
From the results captured using Wireshark, it can be shown that the application used is the UDP protocol with OpenVPN port. Figure 5 shown the results of SMS communication, the use of OpenVPN port is indicated by red striped box.


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.  Time       Source              Destination      Protocol Length Info
 569 95.122983  MitacInt_ae:31:96   Broadcast        ARP       60 Who has 192.168.0.1?  Tell 192.168.0.22
 570 95.386607  MitacInt_ae:31:96   Broadcast        ARP       60 Who has 192.168.88.15?  Tell 192.168.88.4
 571 95.620273  CompalIn_13:be:17   Broadcast        ARP       42 Who has 192.168.88.1?  Tell 192.168.88.145
 572 96.092564  192.168.88.4        192.168.88.145   UDP      167 Source port: 41064  Destination port: openvpn
 573 96.092859  192.168.88.145      192.168.88.4     UDP      167 Source port: openvpn  Destination port: 41064
 574 96.093708  192.168.88.4        192.168.88.145   SSH      162 Encrypted response packet len=96
 575 96.093741  192.168.88.145      192.168.88.4     TCP       66 38520 > ssh [ACK] Seq=1 Ack=9313 Win=1318 Len=0
 576 96.122603  MitacInt_ae:31:96   Broadcast        ARP       60 Who has 192.168.0.1?  Tell 192.168.0.22
 577 96.386560  MitacInt_ae:31:96   Broadcast        ARP       60 Who has 192.168.88.15?  Tell 192.168.88.4
 578 97.093989  192.168.88.4        192.168.88.145   UDP      167 Source port: 41064  Destination port: openvpn
 579 97.094281  192.168.88.145      192.168.88.4     UDP      167 Source port: openvpn  Destination port: 41064
 580 97.095409  192.168.88.4        192.168.88.145   SSH      162 Encrypted response packet len=96
 581 97.095459  192.168.88.145      192.168.88.4     TCP       66 38520 > ssh [ACK] Seq=1 Ack=9409 Win=1318 Len=0
 582 97.122595  MitacInt_ae:31:96   Broadcast        ARP       60 Who has 192.168.0.1?  Tell 192.168.0.22
 583 97.397813  MitacInt_ae:31:96   Broadcast        ARP       60 Who has 192.168.88.15?  Tell 192.168.88.4
 584 98.094931  192.168.88.4        192.168.88.145   UDP      167 Source port: 41064  Destination port: openvpn
▸ Frame 605: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: CompalIn_13:be:17 (b8:70:f4:13:be:17), Dst: MitacInt_ae:31:96 (00:22:4d:ae:31:96)
▸ Internet Protocol Version 4, Src: 192.168.88.145 (192.168.88.145), Dst: 192.168.88.4 (192.168.88.4)
▸ Transmission Control Protocol, Src Port: 38520 (38520), Dst Port: ssh (22), Seq: 49, Ack: 9745, Len: 0
```

Figure 5. Results Captured from SMS Communication

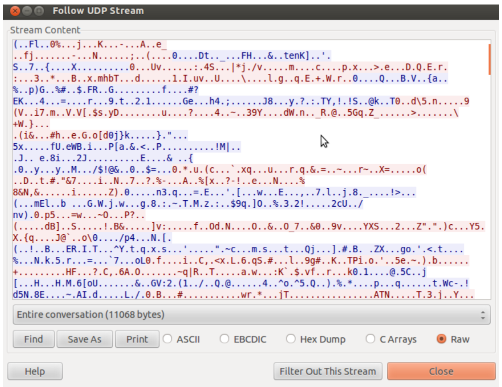
Otherwise, when communicating by phone, the use of OpenVPN port is shown in picture 6 premises red striped box as well.


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

```
No.   Time        Source           Destination      Protocol Length Info
 2563 198.803720  192.168.88.145   192.168.88.4     UDP       135 Source port: openvpn  Destination port: 41064
 2564 198.803745  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2565 198.803813  192.168.88.145   192.168.88.4     UDP       135 Source port: openvpn  Destination port: 41064
 2566 198.804151  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2567 198.804156  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2568 198.804159  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2569 198.804328  192.168.88.145   192.168.88.4     UDP       135 Source port: openvpn  Destination port: 41064
 2570 198.804561  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2571 198.804566  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2572 198.804569  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2573 198.804571  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2574 198.804779  192.168.88.145   192.168.88.4     UDP       135 Source port: openvpn  Destination port: 41064
 2575 198.804805  192.168.88.145   192.168.88.4     UDP       135 Source port: openvpn  Destination port: 41064
 2576 198.804975  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2577 198.804980  192.168.88.4     192.168.88.145   UDP      1487 Source port: 41064  Destination port: openvpn
 2578 198.805100  192.168.88.145   192.168.88.4     UDP       135 Source port: openvpn  Destination port: 41064
▸ Frame 605: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: CompalIn_13:be:17 (b8:70:f4:13:be:17), Dst: MitacInt_ae:31:96 (00:22:4d:ae:31:96)
▸ Internet Protocol Version 4, Src: 192.168.88.145 (192.168.88.145), Dst: 192.168.88.4 (192.168.88.4)
▸ Transmission Control Protocol, Src Port: 38520 (38520), Dst Port: ssh (22), Seq: 49, Ack: 9745, Len: 0
```

Figure 6. Results Captured from Communication

To see more details, then seen the results of encryption that has been done with OpenVPN, in Figure 7 is a packet of data encrypted with SEED algorithm, the encrypted message is meaningless.



```
(..Fl..0%...j...K...-...A..e_
..fj.......-...N......;..(....0....Dt.._...FH...&..tenK]..'.
S..7..{....X..........0...Uv......:.4S...|*j./v.....m....c....p.x...>.e...D.Q.E.r.
:...3..*...B..x.mhbT...d......1.I.uv..U....\....l.g..q.E.+.W.r..0....Q...B.V..{a..
%..p)G..%#..$.FR..G.........f....#?
EK...4...=....r...9.t..2.1......Ge...h4.;......J8...y.?.:.TY,!.!S..@k..T0..d\5.n.....9
(V..i7.m..V.V[.$s.yD........u....?....4..~..39Y....dW.n.._R.@..5Gq.Z_......>.......\
+W.}...
.(i&...#h..e.G.o[d0j}k.....}."...
5x.....fU.eWB.i...P[a.&.<..P..........!M|..
.J.. e.8i...2J..........E....& ..{
.0..y...y..M.../$!@&..0..$=...0.*.u.(c...`.xq...u...r.q.&.=..~...r~..X=.....o(
..D..t.#."&7....i..N..7..?.%-...A..%[x..?-!..e...N....%
8&N,&......i......Z).0.....n3.q...=.E...'.[...w...E...,..7.l..j.8._.....!>...
(...mEl..b ...G.W.j.w...g.8.:.~.T.M.z.:..$9q.]O..%.3.2!.....2cU../
nv).0.p5...=w...~O...P?..
(.....dB]..S.....!.B&.....]v:.....f..Od.N....0..&..0_7..&0..9v....YXS...2...Z".".)c...Y5.
X.{q....J@`..o\0..../p4...N.[.
(..!..B...ER.I.T...^Y.t.q.x.s...'....."~c...m.s...t...Qj...].#.B. .ZX...go.'.<.t....
%...N.k.5.r...=...`7...oL0.f....i..C,.<x.L.6.qS.#...l..9g#..K..TPi.o.'..5e.~.).b......
+.........HF...?.C,.6A.0.......~q|R..T.....a.w...:K`.$.vf..r...k0.1....@.5C..j
[...H...H.M.6[oU.......&..GV:2.(1../..Q.@......4..^o.^5.Q..).%.*....p...q......t.Wc-.!
d5N.8E....~.AI.d.....L./.0.B...#..........wr.*...jT...............ATN.....T.3.j..Y...
```

Entire conversation (11068 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Figure 7. Encrypted Data Packet Content

From the three images above, it appears that the path used is a preconfigured OpenVPN port. The path uses symmetric-based SEED algorithm encryption for communication of the three determined scenarios, the client OpenBTS to OpenBTS, OpenBTS to VoIP and VoIP to VoIP.

6. Cover

6.1 Conclusion

Based on the results tests of the implementation, process, and analysis, it can be concluded as follows:

1. GSM-based mobile communication network by using OpenBTS software can be used on a local blankspot area.
2. Safeguarding the transmission of data via Ethernet using VPN-SSL with symmetric-based SEED algorithm can be implemented well. All of the data transmitted via VPN-SSL tunnel are encrypted first. To sum up VPN-SSL can be implemented as additional security during data transmission over Ethernet.
3. parameters testing associated with OpenBTS transmission analysis by using walktest test through Terms Investigation Software conducted by two scenarios shows that the OpenBTS network communication link is feasible despite the reach is small. The resulting value is Rx average level of -55 dBm, Rx Quality average of 1.17, and the average SQI is 13:59.
4. QoS measurement result indicates that the VoIP network connected with OpenBTS meets VoIP QoS standard

(Delay = <50ms, Jitter = <15ms and Packet Loss = <1%).

5. CPU Usage measured on the server generates 1-2% increase for VoIP to VoIP, for VoIP communication to OpenBTS increase as much as 4-5%, while for OpenBTS to OpenBTS increase 6-7%. To achieve more CPU Usage (95-100%) the estimated number of OpenBTS client to OpenBTS that can be served is approximately 12 clients that communicate at the same time.

Reference

[1] Azad, Abul. (2013). OpenBTS Implementation With Universal Software Radio Peripheral.

[2] Kemetmuller, C. (2010). Installation Guide for OpenBTS. Darmstadt: CASED

[3] David A. Burgess, Harvind S. Samra. 2008. The OpenBTS Project. Kestrel Signal Processing, Inc. Fairfield, California.

[4] Kanaiya Kanzaria, sanjay s.c. (2014). Active GSM Monitoring.

[5] Desai Karan, Ravikiran Dinakar. (2010). Licensing and Security Issues in the Implementation of OpenBTS Base GSM System.

[6] Fabian V.D.B. (2010). Catching and Understanding GSM-Signal.

[7] KISA. SEED Algorithm Specification.

[8] NIST SP.800-113. (2008). Guide to SSL VPNs.

[9] ITU-T P.800. (1996). Methods for Subjective determination of transmission quality.

[10] RFC5669 (2010). The SEED Cipher Algorithm and Its Use with the Secure Real-Time Transport Protocol.

[11] Fuadi, H., 2012. *Perancangan dan Implementasi OpenBTS dengan Menggunakan Asterisk di Ubuntu 10.10.* s.l.:s.n

[12] Hosner, C., 2004. OpenVPN and the SSL VPN Revolution. Volume GSEC v.1.4b.

The Impact of Delay on the Decisions of a Cognitive Radio Engine

[1]Hamed Asadi, [2]Haris Volos, [1]Michael M. Marefat, and [1]Tamal Bose
[1]Dept. of Electrical and Computer Engr. The University of Arizona, Tucson, AZ 85721-0104
{hasadi, marefat, tbose}@arizona.edu
[2]DENSO International America Inc., San Jose, CA 95110-1342
volos@ieee.org

Abstract—**In this paper, we investigate the effect of delay in the decision-making and operation of a cognitive radio engine. In particular applications, such as deep space communications, communicating with space exploration equipment throughout the solar system the roundtrip delay can be minutes to hours. The CE is faced with the task of making decisions that it will not know of their outcome after a considerable delay. In this paper, we provide the system model, and we evaluate various decision strategies taking into account the expected channel states during the transmission period. Our results show that the expected and variance of the amount of delayed feedback have a significant impact on the decision-making and performance of the system.**

I. Introduction

In all types of the communications, it takes some time that a message travels from a transmitter to a receiver. In addition, it takes some time that the receiver process the data and notifies the transmitter that the message is received correctly or not. In result, whenever a transmitter set a new configuration (modulation type, coding, MIMO techniques, power, ...) and start to transmit, there will be some delay, until it can have an evaluation about the quality of its decision. Depending on the distance and protocols of a communication system, the amount of delay varies significantly. Nevertheless, in almost all of the designed CEs for wireless communications [1]–[10], there is a strong assumption which assumes the CE will see the result of its decision immediately and perfectly represent the actual conditions. However, in an actual implementation, the observed data, i.e., the received feedback of CE's decisions or the estimated channel conditions, are most likely to arrive with differnet amount of delays. Therefore, it is of paramount importance to estimate the effect of these degradations on the CE's performance. For example, in link adaptation, when the CE transmits a packet by specific configuration, it's assumed that the CE will receive an acknowledgment (ACK/NACK) immediately; however, even in the LTE-Advanced (3GPP Release 10) [11]–[13] it's assumed that the ACK/NACK will be sent after 4 resource blocks. Each resource block is assumed to take 0.5 milliseconds. Therefore, the CE needs to wait at least for 2 milliseconds to be able to take advantages of the previous decision's result. For other communication systems, the problem is even more serious. For instance, in long-range HF communications, the CE needs to make multiple decisions without receiving any feedback on previous decisions. Furthermore, the behavior of the delay can be different for various actions. It's possible that the CE receives the ACK/NACK of a packet which is sent after the other packet in advance.

In this paper, we studied the effect of delay on wireless communication systems by evaluating the relationship between the amount of delay and CE's performance. Then, we analyzed the impact of delay on the performance of various CE algorithms.

The first contribution of this paper lies in fully modeling the delayed feedback scenario in wireless communication systems. More specifically, we propose a general stochastic model for the CE's decision-making when it's operating in a delayed feedback environment.

And the second contribution of this work is analyzing the delays' effects on some of the proposed CE algorithms in the literature, and finding the most effective parameters on their performance.

This paper is organized as follows: Section II provides an overview of the delayed feedback model for the CE algorithms, which we are going to use in the paper. Section III analyzes the effect of delay in different wireless communication protocols. Finally, Section IV provides concluding remarks.

II. Problem Formulation

To provide a model for delayed feedback scenarios, we consider a general reinforcement learning model with delayed rewards [14]. This model is pretty similar to the CE model which we used in [1], [6], [15] without delay. The general CE model needs to make sequential decisions based on information about the channel scenarios, conditions of the radio (power level, capabilities, etc.), and its own experience which will specifically be exploited from its experience database. Formally, given a current channel scenario x_t, which is a vector of all features of channel scenario at time step *t*, a set of possible communication configurations *A*, which is a complete set of adjustable communication parameters (i.e. modulation type, coding rate, antenna technique, etc.), we will have a set of reward functions $\varPhi \subset \{\varphi : X \times A \to R\}$, and possible reward values *r*, for different time steps *t*.

The CE senses the environment and receives x_t, then chooses an action a_t from the list of possible actions A while the environment picks a reward function $\varphi_t \in \varPhi$ at the same time. Finally, the CE receives a reward value r_t from the reward function $\varphi_t(x_t, a_t)$. The CE algorithm aims to maximize the expected reward $\sum_{t=1}^{n} \varphi_t(x_t, a_t)$ which in the link adaptation problem definition [1] is equal to maximizing $\sum_{t=1}^{n} r_t(n \geq 1)$. To compare the performance of different CE algorithms, we are going to use the concept of *regret*. The *regret* of a CE is equal to the difference between the maximum possible reward in each time step with the reward of an action which is taken by the operated CE algorithm.

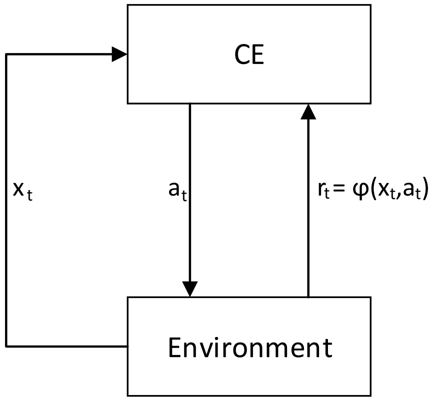
$$\Delta_n = sup_{a\in A}\sum_{t=1}^{n}\varphi_t(x_t, a(x_t)) - \sum_{t=1}^{n}\varphi_t(x_t, a_t) \quad (1)$$

A CE is consistent if it achieves the average reward of the best possible action. To compare the performance of different CE algorithms, we are not only interested in how fast the average regret can be made to converge to 0, $E[\Delta_n]/n \to 0$, but we are also interested in minimizing the amount of Δ_n that will result in higher throughput, in the case of maximizing throughput objective. Figure 1 shows the CE operation in the delayed case.

In this work, we assume that the delay at t_{th} time step (τ_t) is a constant value and is equal to τ_0 which depends on the communication's protocol.

III. Effects of Delayed Feedback

In wireless communication protocols such as 3G [13], 4G [13], LTE [12], and LTE-Advanced [11] a specific time-frame for the ACK/NACK is considered. If a radio doesn't hear back from the receiver after a certain amount of time, it will assume that the packet is dropped. Therefore, in many of the wireless protocols, we can assume a constant delay for the CE operations where the CE needs to skip a particular number of decisions, based on the amount of delay.



Parameters: CE action set $\boldsymbol{A}$, possible channel scenarios $\boldsymbol{X}$, reward function $\boldsymbol{\phi}$: $\boldsymbol{X} \times \boldsymbol{A} \rightarrow \boldsymbol{\Phi}$,
At each time step *t=1, 2, ..., n:*

1. The radio senses the environment condition (channel scenario) $\boldsymbol{x_t} \in \boldsymbol{X}$
2. The CE will take an action $\boldsymbol{a_t} \in \boldsymbol{A}$, based on environment condition $\boldsymbol{x_t}$
3. The reward $\boldsymbol{r_t = \varphi(x_t, a_t)}$ is scheduled to be revealed after $\boldsymbol{\tau_t}$ time steps
4. The CE observes $\boldsymbol{R_t = \{(t', r_{t'}) : t' \leq t, t' + \tau_{t'} = t\}}$, i.e., all the reward values scheduled to be revealed at time step t, together with their timestamps.

Fig. 1. CE operation under delayed modified from [14], [16], [17]

The proposed non-delayed CE algorithms in the literature need to observe the feedback of the previous decision to be able to decide about the next configuration. Therefore, the CE process will be as follows:

- The CE observes the channel conditions at time t which is x_t.
- The CE takes an action a_t based on the current channel condition x_t.
- The CE waits until receives the feedback of transmitted packet at $t + \tau_t$.
- If the channel condition at $t + \tau_t$ is same as the channel condition at time t, the CE takes the next action $a_{t+\tau_t}$ based on the observed data.

The result of this approach will be lots of idle status of transmitter on the times that the CE is waiting to hear back from receiver about the status of transmitted data packets.

Weinberger & Ordentlich [18] proposed a delay handling strategy for the constant delay (τ_0) problem. In their approach, they assumed a non-delayed CE algorithm with τ_0 independent instants which are operating in sequences. They showed that the regret bound of the new algorithm is $(\tau_0)f(n/\tau_0)$ [19]. In this method, the regret bound of the delayed algorithm has a multiplicative effect on the regret bound of the individual non-delayed CE algorithm.

To be able to use Weinberger's strategy, during the operation of CEs, the channel conditions need to be constant. In wireless communications, we can assume that the channel impulse response is essentially invariant over the time frame, also known as coherence time. The coherence time is proportional to the Doppler spread, and the popular rule of thumb for calculating coherence time in modern digital communication is: [20]

$$T_C = \sqrt{\frac{9}{16\pi f_m^2}} = \frac{0.423}{f_m} \tag{2}$$

where f_m is the maximum doppler frequency and $f_m = \nu f_c/c$, where ν is the speed of the receiving or transmitting radio (assuming the other one is stationary), f_c is the radio carrier frequency, and $c = 3 \times 10^8 m/s$ is the speed of light.

To have a better understanding of the above formula, assuming the classic Jake's channel model, the envelope autocorrelation $R(\tau)$ of the channel is given by [21]:

$$R(\tau) = J_0(2\pi f_m \tau) \tag{3}$$

"The coherence time is the time duration over which two received signal have a strong potential for amplitude correlation" [22]. If we consider coherence time as the time over which the correlation coefficient is greater than 0.5 then the coherence time will be approximately [23] $T_c \approx \dfrac{9}{16\pi f_m}$, however this formula is too restrictive and 2 equation is more popular. Figure 2 depicts equation 3 versus delay time for $\nu \in [0,\ 1,\ 3,\ 10,\ 60]\ mph$ and $f_c = 2.4\,GHz$. Assuming correlation coefficient greater than 0.5, from Figure 2, it can be noticed that at walking speed ($3\ mph$), the channel condition can be considered stationary for slightly more than 20 ms. However, for the vehicle speed of 60 mph, this time will be just 3 ms.

Since the channel scenario can be assumed to be constant, in our CE model, the x_t will be constant for $t \leq T_C$. Therefore, the reward function $\varphi_t(x_t, a_t)$ will be based on the taken action a_t. As result, if the amount of delay will be less than $t \leq T_C$, we will be able to use Weinberger model for running the non-delay CE algorithms in delayed feedback scenarios.

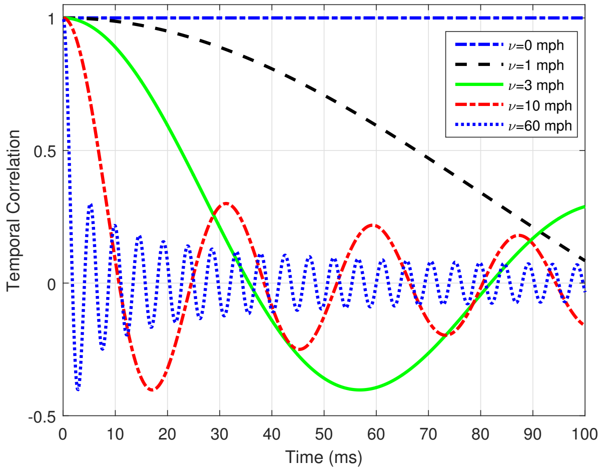


Fig. 2. Temporal correlation vs. delay time, $f_c = 2.4GHz$

The Weinberger's algorithm operates as follows. First, let's assume that the constant delay is equal to the time of the transmission of 4 packets or $\tau_0 = 4$ time steps. At the start of operation, an instance of a non-delayed CE algorithm (I_1) makes a decision. Then the status of I_1 will change to waiting until receiving the feedback of its decision. While I_1 instance is waiting to receive the feedback, another instance of non-delayed CE algorithm starts to operate. Clearly, since our delay is equal to four time steps, we need four different instances of non-delayed CE algorithm. As soon as the first feedback arrives, the waiting instance will start to operate. Figure 3 illustrates the time sequence of the Weinberger's algorithm operation with the constant delay of $\tau_0 = 4$.

In addition to Weinberger's algorithm, to handle the delayed feedback problem in non-delayed CE algorithms, we can formulate CE in the form which ignores the delayed feedback until they arrived. In this form, the CE avoids the idle status in the times that it's waiting to hear back from the receiver. The non-delayed CE algorithm needs to pretend that no decisions are made up to the current time. Then, it will make a decision based on the currently available information. The operation of these CEs will be implemented with two independent threads. While the main Thread is making decisions based on the observed channel conditions at time t, the second thread is waiting to receive feedback from the previously made decisions to update the observation database of the operating CE. We are going to call this type of CEs as Not Waiting CE (NW-CE) algorithms.

To evaluate the performance of the Weinberger and NW-CE algorithms and analyze the effect of delay on a communication system, we use two non-delayed CE algorithms: Gittins strategy and ϵ-Greedy. We also use a 4×4 MIMO system with QPSK, 8PSK, 16, 32, 64, 128 and 256 QAM as a modulation type with eight

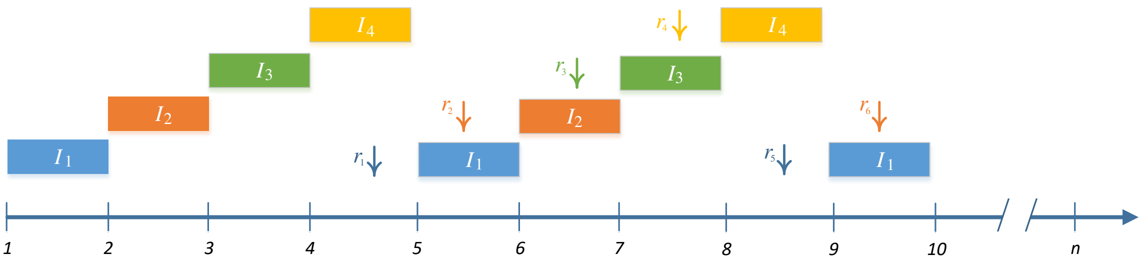


Fig. 3. Time sequence of Weinberger's algorithm operation.

error correction rates: 1, $\frac{7}{8}$, $\frac{3}{4}$, $\frac{2}{3}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{6}$ and $\frac{1}{8}$ and antenna techniques: VBLAST, STBC and MRC. For our channel scenarios, we consider an SNR in the range of 0-50 dB and the log10 of the eigen spread (κ) of the channel matrix in the range of 0-12. The CR also has 12 channels available with different SNRs and bandwidths (either 1.25 or 2.5 MHz).

In the first experiment, to compare the effect of delayed feedback, we consider two scenarios. First, we assume that there is no delay, and that the CE algorithms will receive the feedback immediately. Second, we assume a constant delay based on the LTE protocol to be $\tau_0 = 4$. Figure 4 illustrates the obtained throughput, with their confidence bound by the CE algorithms when there is no delay $\tau_0 = 0$. The results are the mean of 1000 independent experiments. The performance of the CEs represents that after almost 500 time steps, both algorithms will converge to the optimal performance with a tight confidence bound.

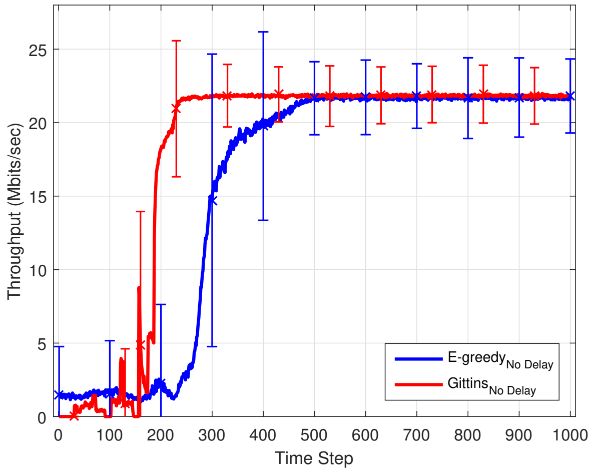


Fig. 4. ϵ-Greedy and Gitting strategy CE algorithms without delay

Figure 5 represents the effect of the constant delay. As we discussed, delay has a multiplicative effect on the regret bound of a CE based on the amount of the delay. The plot shows that none of the CE algorithms are able to find the optimal option in 1000 time steps In addition, their confidence bound illustrates the high level of uncertainty on the obtained performance over 1000 independent experiments.

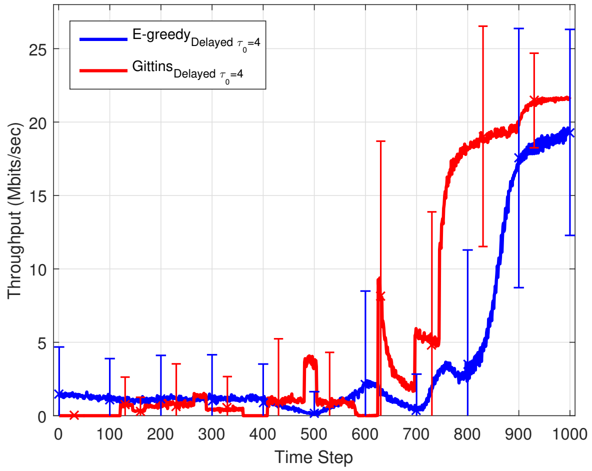


Fig. 5. ϵ-Greedy and Gitting strategy CE algorithms with the constant delay $\tau_0 = 4$

Figure 6 shows the total amount of transferred data with the presence of delay $\tau_0 = 4$ and when the communication system receives the feedback immediately. Clearly, we can see the multiplicative effect of the delay on the total transferred data. This is more clear when we compare the performance of annealing ϵ-greedy [24] algorithm in both cases.

Figure 7 illustrates the results of the NW-CE algorithm in the presence of delay. To generate this result we used the same communication system used for previous experiments and we assumed the constant amount of delay $\tau_0 = 4$. Figure 7(b) shows the effect of the constant delay on the performance of CE algorithms when we use NW-CEs.

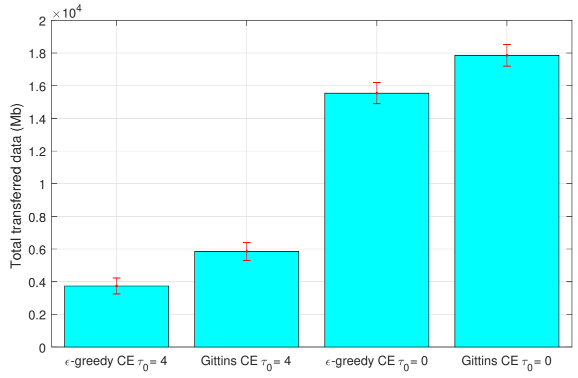
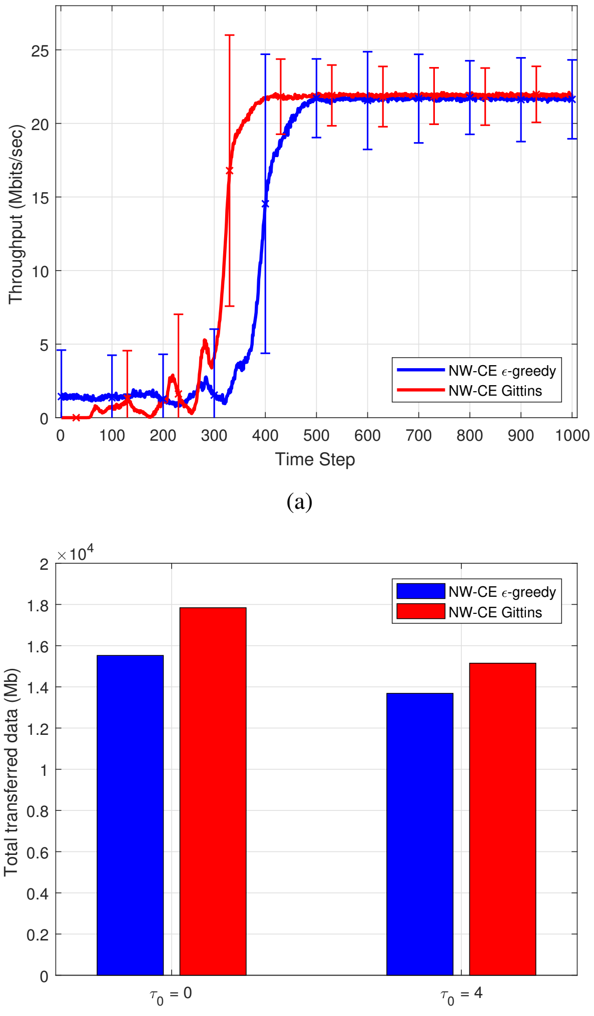


Fig. 6. Total amount of transfered data by using CE algorithms in the presence of delay and without delay



(a)

(b)

Fig. 7. Performance of the NW-CE ϵ-greedy and gittins strategy in the presence of delay with ϵ-greedy and gittins CEs. Part (a) represents the obtained throughput as the objective of NW-CE when $\tau_0 = 4$. (b) shows the total transferred data when NW-CE faces different amount of delays. The figure clears the additive effect of the delay on the regret bound.

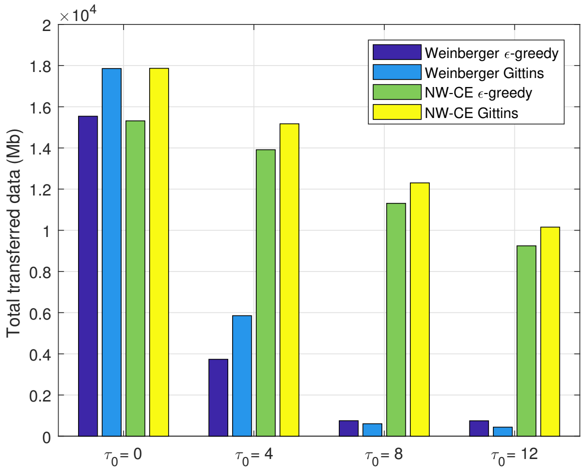


Fig. 8. Comparison between the effects of delay on Weinberger and NW-CE algorithms.

To compare the impacts of the delay on CE algorithms, we considered three different amount of delays and compared the performance of CEs with the non-delayed environment. Figure 8 shows the total data transferred with two CE algorithms: ϵ-greedy and Gittins. We used both Weinberger and NW strategies to handle the delayed feedback with each of CE algorithms. The results indicate that the delay handler strategy plays more important role than the CE algorithm's type. For example, in the Figure 8, the performance of both Gittins and ϵ-greedy CEs exponentially decreased with respect to the amount of delay. However, by using NW strategy, their performance decrease linearly.

IV. Conclusions

In this paper, we studied the effects of delayed feedback on the performance of cognitive radios engines and compared the performance of two different CE algorithms and two different delay handling strategies together. To this end, we first formalized the delayed feedback scenario in wireless communication systems and proposed a stochastic model for the CE's decision-making process in delayed feedback environments. Secondly, we analyzed the effects of delay on CE algorithm's performance. Our results indicate that the delay handling strategies are more effective than the CE algorithms to deal with delayed feedback problem.

Acknowledgements

This project was partially supported by the Broadband Wireless Access and Applications Center (BWAC); NSF Award No. 1265960.

References

[1] H. I. Volos and R. M. Buehrer, "Cognitive Engine Design for Link Adaptation: An Application to Multi-Antenna Systems," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2902–2913, Sept. 2010.

[2] H. I. Volos, C. I. Phelps, and R. M. Buehrer, "Initial Design of a Cognitive Engine for MIMO Systems," in *SDR Forum Technical Conference*, Nov 2007.

[3] ——, "Physical Layer Cognitive Engine for Multi-Antenna Systems," in *Proceedings of the IEEE Military Communications Conference*, Nov. 2008, pp. 1–7.

[4] H. Asadi, H. Volos, M. Marefat, and T. Bose, "Metacognitive radio engine design and standardization," *Selected Areas in Communications, IEEE Journal on*, 2015.

[5] H. Asadi, H. Volos, M. M. Marefat, and T. Bose, "Metacognition and the next generation of cognitive radio engines," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 76–82, January 2016.

[6] ——, "Enhancing robustness and perturbation tolerance of cognitive radio engines with metacognition," *Analog Integrated Circuits and Signal Processing*, vol. 91, no. 2, pp. 173–185, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10470-017-0930-6

[7] H. I. Volos and R. M. Buehrer, "Cognitive Radio Engine Training," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 11, pp. 3878–3889, 2012.

[8] T. W. Rondeau, "Application of Artificial Intelligence to Wireless Communications," Ph.D. dissertation, Virginia Tech, 2007.

[9] A. He, K. K. Bae, T. R. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. J. Neel, Y. Zhao, J. H. Reed, and W. H. Tranter, "A Survey of Artificial Intelligence for Cognitive Radios," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1578–1592, May 2010.

[10] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans, and G. J. Minden, "Cognitive Engine Implementation for Wireless Multicarrier Transceivers," *Wiley Journal on Wireless Communications and Mobile Computing*, vol. 7, no. 9, pp. 1129–1142, 2007.

[11] 3rd Generation Partnership Project, "LTE-Advanced." [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced

[12] European Telecommunications Standards Institute, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," Tech. Rep., 2010.

[13] 3rd Generation Partnership Project, "LTE." [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/98-lte

[14] N. J. Nilsson, "Introduction to machine learning: An early draft of a proposed textbook. pages 175-188. http://robotics.stanford.edu/people/nilsson/mlbook.ht ml," 1996.

[15] H. Asadi, H. I. Volos, M. Marefat, and T. Bose, "Learning Characterization Framework and Analysis for a Meta-Cognitive Radio Engine," in *Proceedings of SDRWInnComm 2014 Wireless Innovation Conference on Wireless Communications Technologies and Software Defined Radio*, Mar. 2014, pp. 132–139.

[16] P. Joulani, A. György, and C. Szepesvári, "Online learning under delayed feedback," in *International Conference on Machine Learning (ICML), Atlanta, Georgia,*, June 2013.

[17] S. Guha, K. Munagala, and M. Pal, "Iterated allocations with delayed feedback," *CoRR*, vol. abs/1011.1161, 2010. [Online]. Available: http://arxiv.org/abs/1011.1161

[18] M. J. Weinberger and E. Ordentlich, "On delayed prediction of individual sequences," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1959–1976, Jul 2002.

[19] ——, "On delayed prediction of individual sequences," *IEEE Trans. Inf. Theor.*, vol. 48, no. 7, pp. 1959–1976, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1109/TIT.2002.1013136

[20] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[21] J. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[22] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2002.

[23] R. Steele, *Mobile Radio Communications*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, March 1998.

STOCHASTIC MODELS FOR OPTIMIZATION OF SOFTWARE-DEFINED RADIO OPERATION

Shuvra Bhattacharyya (Department of ECE, University of Maryland and Dept. of Pervasive Computing, Tampere University of T echnology); Marilyn Wolf School of ECE, Georgia Institute of Technology)

ABSTRACT

Stochastic models can be used to capture uncertainty and variation in communication and computation systems as well as in communication channels. Our factored Markov decision process models capture characteristics of a communication system and its operating environment, allowing more robust optimization of its operation.

1. INTRODUCTION

Software-defined radios are complex systems that operate in complex environments. System operation must be optimized for communication parameters as well as operational parameters such as power consumption. We use factored Markov decision processes to efficiently represent both the communication channel and the characteristics of the communication system platform. Our approach allows us to more robustly optimize system operation compared to previous methods.

2. MARKOV DECISION PROCESSES

A Markov decision process (MDP) is an extension of a Markov model that provides for external input. In each state, the model accepts an input; the transitions out of each input are assigned probabilities. Each transition is associated with a reward.

Since the transitions out of a state depend on the input, the choice of inputs affects the possible rewards available as the model moves from one state to the next. A policy is a set of inputs to be applied to the machine. We can solve for an optimal policy that maximizes the reward on an infinite horizon; rewards are discounted over time using a discount factor. A number of algorithms have been developed to solve for optimal discounted rewards of an MDP.

Markov decision processes have been widely used to model communication channels [1]. They have also been used to optimize power management policies in embedded computing systems [2].

3. FACTORED MDP MODELS

We use factored Markov decision processes to model communication systems. Our models capture both the communication system as well as the channel and environment. Our approach exposes a larger design space because we consider the effects of the communication system as well as the channel. Our approach also allows us to optimize system operation for a wider range of objectives, including power consumption and thermal behavior.

The two major components of the system model are the channel state model and the control state machine. The channel state model is a hidden Markov model; it can be designed using simulation methods and updated using online training. The control state machine is a finite-state machine that describes the communication platform’s behaviors; its states describe the performance and power consumption of the platform in various modes. The product of these two models produces a Markov decision process known as the configuration control machine (CCM). The CCM describes the channel, the communication platform, and their potential interactions.

We can solve the configuration control machine at design time to find a policy used to operate the machine. That policy allows the optimization framework to adapt to the environment in addition to the system being optimized.

We can also update and re-solve the model at run time. Dynamic updating of the model and policy allows the system to adapt to the environment.

4. COMMUNICATION SYSTEM MODELING

We have applied our factored MDP approach to the optimization of a channelization system [3]. The channelizer has three top-level processing states: an FIR downconverter with eight subconfigurations; a DFT filter bank; and a sleep mode. A two-frame delay was incurred to switch between algorithms. Our power cost estimates were based on measurements of an ARM Cortex-M3. We analyzed the system’s behavior for two different use cases. Compared to the Highly Adaptive Reconfiguration Platform (HARP), our

MDP-based approach requires no a priori tuning and greater robustness.

ACKNOWLEDGMENTS

This work was supported by NSF grants CNS151304 and CNS1514425.

REFERENCES

[1] H.-P. Shiang and M. van der Schaar, "Online learning in autonomic multi-hop wireless networks for transmitting mission-critical applications," Selected Areas in Communications, IEEE Journal on, vol. 28, no. 5, pp. 728–741, June 2010.

[2] Zhiyuan Ren, Bruce H. Krogh, and Radu Marculescu. 2004. Hierarchical Adaptive Dynamic Power Management. In *Proceedings of the conference on Design, automation and test in Europe - Volume 1* (DATE '04), Vol. 1. IEEE Computer Society, Washington, DC, USA, 10136-.

[3] Adrian Sapio, Marilyn Wolf, and Shuvra Bhattacharyya, "Compact modeling and management of reconfiguration in digital channelizer implementation," in *Proceedings, GlobalSIP 2016*, IEEE, 2016.

DEEP LEARNING-BASED INTELLIGENT METHOD FOR AUTOMATIC MODULATION CLASSIFICATION IN COGNITIVE RADIOS

Gihan J. Mendis (ijm11@zips.uakron.edu), Jin Wei (jwei1@uakron.edu), and Arjuna Madanayake
(arjuna@uakron.edu)

The University of Akron, Akron, Ohio, USA

ABSTRACT

This paper develop an automatic modulation classification (AMC) method for cognitive radio (CR). The proposed method employs the spectral correlation function (SCF) to generate the unique pattern signatures for each modulation scheme. Furthermore, a low-complexity binarized convolutional neural network (CNN) is designed to classify modulation schemes by recognizing the SCF-based pattern signatures. By employing the low-complexity CNN, the computationally costly 617632 floating point multiplication operations required in the conventional CNN are represented by zero computational cost no connections, simple connections, negation operations, bit-shifting operations, and bit-shifting with negation operations. This work utilizes simulated modulated signals that employed BPSK, QPSK, 2-FSK, 4-FSK, and OFDM with QPSK sub-carrier modulation schemes. The performance of the proposed method is evaluted for modulated signals with different SNRs. Furthermore, the modulation classification accuracy achieved by the proposed low-complexity CNN is compared with that achieved by the conventional CNN. The simulations section evaluates the performance of the proposed method. As shown in simulations, the accuracy of the proposed automatic modulation classification remains higher than 91% even when the SNR of the measurement environment is as low as 0 dB.

1. INTRODUCTION

With growing increase of demand frequency bands of electromagnetic spectrum using for radio frequency (RF) communication becoming a scared natural resource. Cognitive radio (CR) has attracted a lot of interest as a technique for efficiently utilizing the scarce spectrum resources [1]. In order to achieve an efficient transmission and to address the challenges of data security such as jamming, interference, and blocking, modulation schemes are being used in RF communication. BPSK, QPSK, 2FSK, 4FSK, and OFDM are some of the widely used modulation schemes that are used for encoding data on multiple carrier frequencies. Receivers of the CR systems should have the capability to automatically identify the modulation schemes used, it is an intermediate step for signal demodulation of the intelligent reciever [2]. Furthermore, detecting available modulation scheme on a carrier channel is a crucial step for spectrum sensing which is an essential function of CR systems [3, 4].

Likelihood-Based (LB) and Feature-Based (FB) approaches are the two main commonly established approaches for AMC. LB approaches provide optimal performance but require perfect knowledge of receive signals. FB approaches are sub-optimal approaches but require less prior knowledge about the received signal [2]. Most modern approaches of AMC are analytical feature-based approaches that use advanced signal processing methods to analytically derive known features. In [5], Deng *et al.* used template machine-based approach to classify multi-level amplitude phase shift keying (MAPSK) signals. In [6], Wu *et al.* suggest using analysis of higher orders statistics to engineered features for modulation detection.

Deep learning methods are artificial neural network (ANN) based machine learning techniques that have multiple layer hierarchies of ANNs. These techniques are more effective in extracting hierarchical features from raw data [7]. Deep learning methods have been used for pattern recognition in various application areas [8–12]. In our previous work, we proposed an automated modulation classification (AMC) method with a signal processing mechanism that uses spectral correlation function (SCF) to generate noise-resilient and distinguishable 2-D signature patterns and deep belief network (DBN) based classifier for classification of modulation schemes [13]. Convolutional neural networks (CNNs) are one of the most successful deep learning techniques inspired by the neuron arrangement of the visual cortex of mammals [14]. CNN-based methods are widely used for image classification tasks including radar signature analysis [15–18], In this paper, we leverage CNN as the deep learning-based classification method for the AMC.

The main challenge of implementing deep learning methods is the high computation-complexity that increases the power and area cost of digital implementations for deep learning based classifiers. High computation-complexity is a result of the high number of floating-point multiplications operations. In our previous work, we proposed a multiplierless low-complexity DBN with direct mapping to binary logic circuits to use as the classification technique for AMC [19]. In this work, we design a multiplierless CNN-based classifier for classification of SCF patterns of modulation methods. In [20], Lin *et al.* proposed a binarization method for backpropagation algorithm which produces binary weights $\{-1, 0, 1\}$. In this paper, we exploit this method to realize the multiplierless low-complexity CNN.

In the following section, we provide an overview of the proposed system. Sections 3 and 4 briefly discuss SCF pattern generation and the low-complexity CNN, respectively. The simulation of modulated signals, implementation of proposed low-complexity CNN for modulation classification, and the summary results obtained are shown in Section 6. In Section 7, the conclusions and future work are presented.

2. OVERVIEW

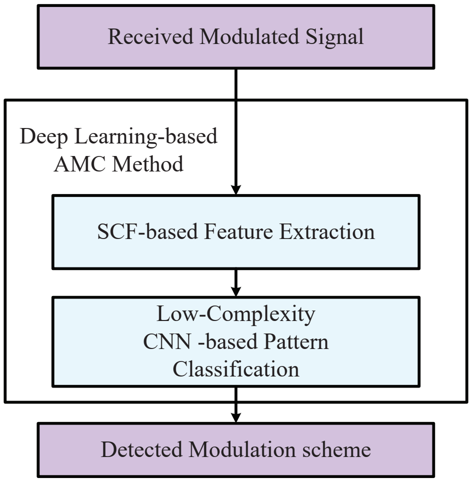

Received Modulated Signal
Deep Learning-based AMC Method
SCF-based Feature Extraction
Low-Complexity CNN -based Pattern Classification
Detected Modulation scheme

Figure 1: Overview of the proposed system.

As shown in Fig. 1, the proposed deep learning-based AMC method contains a spectral correlation function (SCF)-based feature extraction method followed by the low-complexity CNN-based pattern classifier, which classifies the generated SCF patterns to identify the embodied modulation scheme of the received signal.

3. SCF SIGNATURE PATTERNS

Cyclic Autocorrelation Function (CAF) is defined to quantize the amount of correlation between different frequency shifted versions of a given signal and represent the fundamental parameters of their second order periodicity [21]. CAF is calculated as follows:

$$R_x^\alpha[l] = \left[\lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]x^*[n-l]e^{-j2\pi\alpha n} \right] e^{-j\pi\alpha l} \tag{1}$$

Where $x[\cdot]$ is the given signal and $\alpha = m/T_0$ is the cyclic frequency, T_0 is the process period, and m is an integer. Spectral correlation function (SCF) is the Fourier transform of CAF, which is described in the following equation.

$$S_x^\alpha[f] = \sum_{l=-\infty}^{\infty} R_x^\alpha[l]e^{-j2\pi fl} \tag{2}$$

Table 1: The floating-point multiplication operations required in CNN.

Layer	Floating point Multiplications
Convolution layer 1	$24 \times 24 \times 5 \times 5 \times 32 = 460800$
Convolution layer 2	$12 \times 12 \times 5 \times 5 \times 32 = 115200$
Convolution layer 3	$6 \times 6 \times 2 \times 2 \times 32 = 4608$
Fully connected ReLU	$1152 \times 32 = 36864$
Fully connected softmax	$32 \times 5 = 160$
The Number of Required Multiplications	617632

Where f the temporal frequency of the given signal.

Modulated signals contain 2nd order periodic statistical features associated with the corresponding modulation scheme. 2-D order features unique to each modulation scheme can be extracted from the SCF of the modulated signal [21]. In this work, we use SCF pattern classify modulation schemes such as FSK, BPSK, QPSK, and OFDM. However, to identify higher order modulations, such as 16QAM and 64QAM, higher order methods need to be used [22]. Another advantage of using the SCF patterns is the resilience to stationary impairments such as additive white Gaussian noise (AWGN) because the SCF suppresses stationary features [21].

4. LOW-COMPLEXITY CONVOLUTIONAL NEURAL NETWORK

As shown in the Fig. 2, the CNN designed in our work consists of 3 convolution layers, 2 pooling layers, a fully connected layer with rectifier linear units (ReLU), and a softmax-based output layer. The inputs to the CNN are 2D images having the size of 24×24. The first convolution layer evaluates 32 features with 5×5 kernel size. Maximum pooling is performed after the first convolution layer with the kernel size of 2×2, which reduces the image size to 12×12. The second convolution layer evaluates 32 features with 5×5 kernel size along with 2 maximum pooling, which reduces the size of the image to 6×6. The third convolution layer evaluates 32 features with 2×2 kernel size. The outputs of the 32 kernels of the third convolution layer are reshaped and combined to form a vector of the size $6 \times 6 \times 32 = 1152$. A fully connected layer with 1024 ReLU units is added on top along with a softmax layer for classification.

If the weights of the convolution layers and fully connected layers remain as floating-point numbers, the total number of floating point multiplication operations required to perform in a single iteration of testing is shown in Table 1, where we assume the number of class labels as 5. Since floating point multiplication is computationally expensive in digital logic and the number of the total multiplications required for the CNN is very high, the deployments of the above CNN becomes a hardware-expensive task. By modifying the backpropagation algorithm of the CNN as shown in Table 2, we replace the floating-point weights of the CNN by using five possible values $-2^p, -1, 0, 1, 2^p$, where

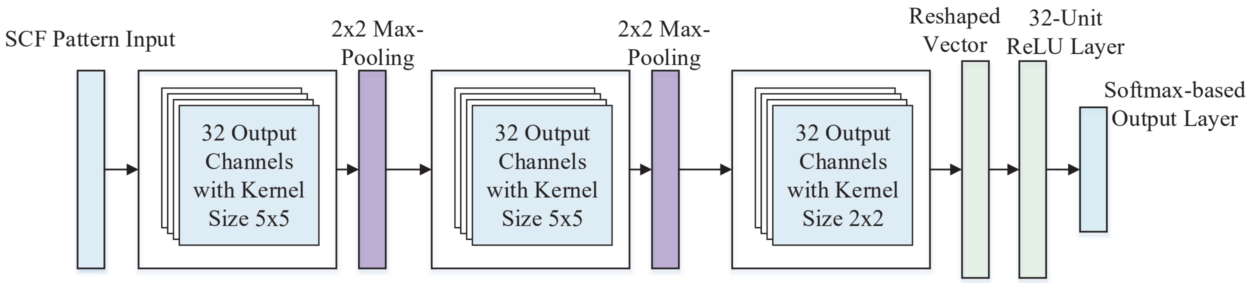

SCF Pattern Input
2x2 Max-Pooling
2x2 Max-Pooling
Reshaped Vector
32-Unit ReLU Layer
Softmax-based Output Layer
32 Output Channels with Kernel Size 5x5
32 Output Channels with Kernel Size 5x5
32 Output Channels with Kernel Size 2x2

Figure 2: The structure of CNN.

p is a positive integer. By doing so, we reduce the hardware-expensive floating-point multiplications to the operations that are much less costly in the digital hardware as shown in Table 3.

5. SIMULATION RESULTS

In this section, we evaluate our proposed method. Furthermore, we also compare the accuracy of our proposed low-complexity CNN, that is the essential component of our method, with that achieved by the conventional CNN.

Simulation of Modulated Signals

We simulate the modulated signals with different modulations schemes using MATLAB/Simulink software. The modulation schemes used in this work are ASK, 2FSK, 4FSK, BPSK, QPSK, and OFDM with BPSK modulated sub-carriers. For all simulated signals, the carrier frequency is selected as 1 kHz and the symbol rate is chosen to be 100 Hz. The amplitudes of the signals are normalized to the range [0, 1]. For simulation of BPSK and 2-FSK modulated signal, a data stream of 256 symbols with binary symbols in random order is used. In 2-FSK modulation scheme, the two frequencies used for modulation are 100 Hz and 160 Hz. For simulation of QPSK and 4-FSK modulated signal, a data stream of 256 symbols with 4 symbols in random order is used. In 4-FSK modulation scheme, the four frequencies used for modulation are 100 Hz, 120 Hz, 140 Hz, and 160 Hz. For the simulation of OFDM, 256 data random data stream is used with 4 symbols. OFDM system simulated contains 128 QPSK modulated sub-carriers.

In order to study the resilience of proposed method in fading channels, we added additive white Gaussian noise (AWGN) to simulated modulated signals. Therefore, the signals are simulated with a range of SNR 0 - 5 dB. SCF patterns of simulated modulated signals are generated using a MATLAB Communications System Toolbox functions [23]. SCF patterns generated for simulated BPSK, QPSK, 2-FSK, 4-FSK, and OFDM with QPSK sub-carriers modulation schemes with SNR 5 dB are shown in Figs. 3, 4, 5, 6, and 7, respectively. Figure 8 shows the 2D projection of the SCF patterns for the simulated modulation schemes when SNR is 5 dB.

Table 2: The training algorithm for updating our low-complexity CNN.

Operators and functions:

$.\geq$: the elementwise more than or equal comparison of two matrices.

$.\times$: the elementwise multiplication of two matrices.

$y = sign(x)$: if $x < 0$, $y = -1$, else $y = 1$.

$y = absolute(x)$: if $x < 0$, $y = -x$, else $y = x$.

$\mathbf{Y} = rand(\mathbf{X})$: randomly assigns $y_{ij} \in [0, 1]$ and $dim(\mathbf{Y}) = dim(\mathbf{X})$.

$y = cast(x)$: if $x = true$, $y = 1$, else $y = 0$.

$\mathbf{W} = backprop(\mathbf{W}, f)$: applies the gradient descent based backpropagation algorithm to fine-tune the weight matrix $\mathbf{W}$, where f is a batch of training data.

$f = nextbatch(\mathbf{F}, batchsize)$: returns the next batch of training data according to batch size, where $\mathbf{F}$ is the training data set.

$\mathbf{W_c} = clipping(\mathbf{W}, L)$: clips the element values of weight matrix $\mathbf{W}$ to be in the range $[-L, L]$ where L is a predetermined scalar.

Inputs: L-clipping level, $\mathbf{W}$-initial weight matrix , $\mathbf{F}$-training data, $\mathbf{T}$-labels corresponding to training data, N-number of training iterations.

Output: $\mathbf{W_b}$-binarized weight matrix ($w_{ij} \in \{-1, 0, 1\}$)

Steps:

```
For $epoch \leqslant N$
    $f = nextbatch(\mathbf{F}, 50)$
    $\mathbf{W} = backprop(\mathbf{W}, f)$
    If $(mode(epoch, 100) = 0)$
        $\mathbf{W_c} = clipping(\mathbf{W}, L)$
        $\mathbf{S} = sign(\mathbf{W_c})$
        $\mathbf{P} = absolute(\mathbf{W_c})/L$
        $\mathbf{T} = \mathbf{P}. \geq rand(\mathbf{P})$
        $\mathbf{W_b} = cast(\mathbf{T}). \times \mathbf{S}$
    End
End
```

Table 3: Digital logic mapping of multiplications with low-complexity weights.

Weight Value	Mapping
0	No connection
1	Connection
-1	Negation
2^p	Right shift by p bits
-2^p	Right shift by p bits and negation

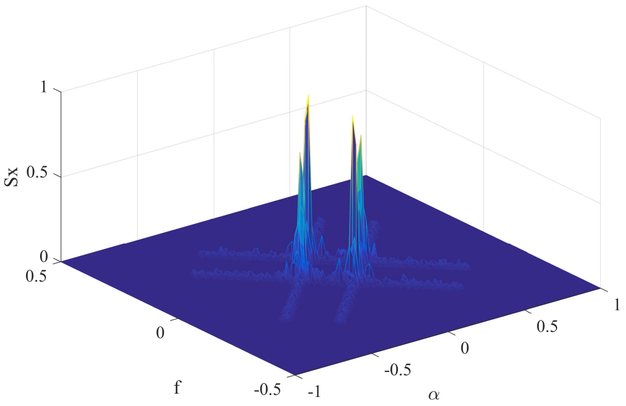


Figure 3: SCF pattern of simulated BPSK modulated signal: SNR of the simulated signal is 5 dB.

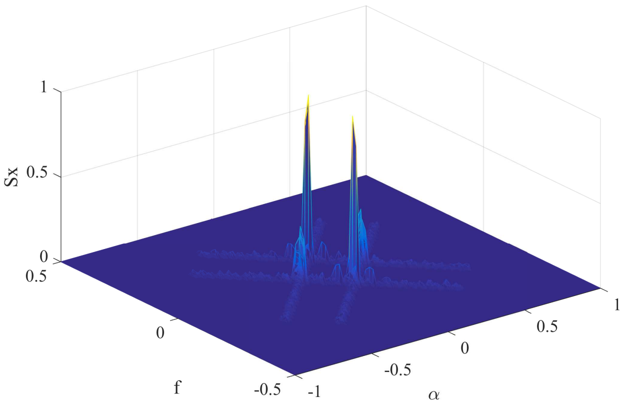


Figure 4: SCF pattern of simulated QPSK modulated signal: SNR of the simulated signal is 5 dB.

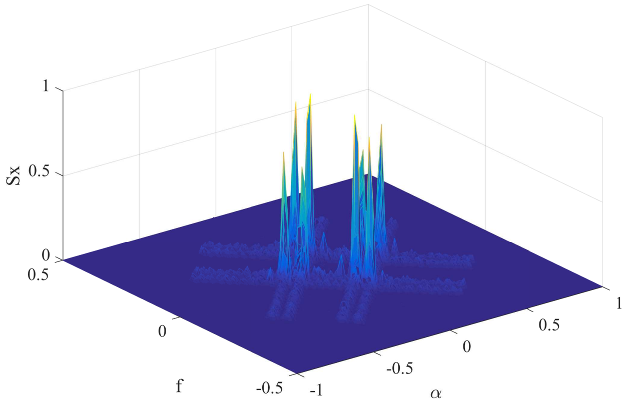


Figure 5: SCF pattern of simulated 2-FSK modulated signal: SNR of the simulated signal is 5 dB.

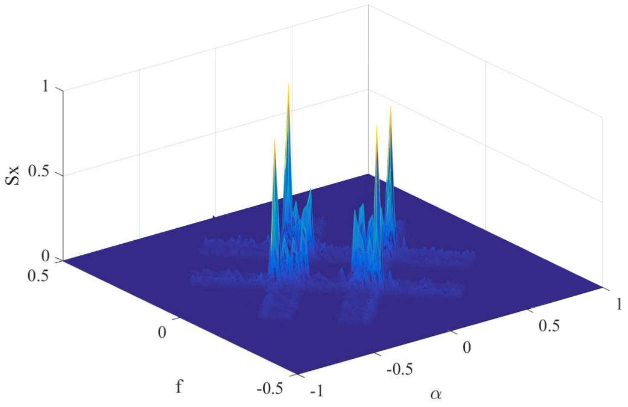


Figure 6: SCF pattern of simulated 4-FSK modulated signal: SNR of the simulated signal is 5 dB.

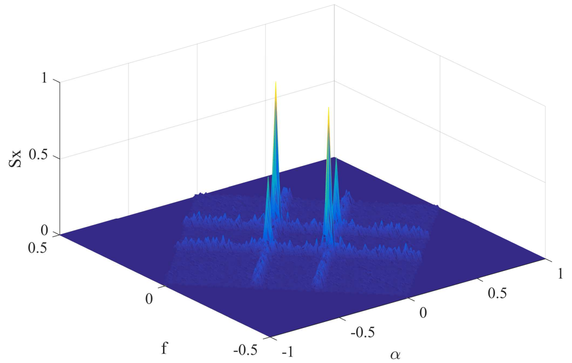


Figure 7: SCF pattern of simulated OFDM modulated signal: SNR of the simulated signal is 5 dB.

Pre-Processing and Training

The gray-scale images of the SCF patterns are resized to be 48×48 images. Fast Fourier transform (FFT) based method is used for image scaling. A 2-dimensional (2D) FFT operation is implemented on the original grayscale images and a 48×48 pixel square is selected from the center of the FFT transformed image. Then inverse 2D FFT is performed to achieve the scaled down image. By doing so, high-frequency components of the original image are filtered out, and thus high-frequency noise is removed from the scaled down image. Considering the symmetry of the patterns, a quarter of the pixels from the resized images is used as the input for the low-complexity CNN classifier. Therefore, the input size of the low-complexity CNN is 24×24.

The low-complexity CNN is trained using data that includes 400 of 2000 patterns corresponding to each modulation scheme which contained SCF patterns generated for signals with different SNR levels. As a comparison, a conventional CNN, which has the same structure but uses floating-point accurate weights, are also trained with the same 2000 training data. Another data set containing 250 patterns from each SNR for each modulation scheme is used to evaluate the performances of the CNNs. The classifiers based on low-complexity and conventional CNNs are implemented using TensorFlow APIs [24]. In the simulation, we set the number of iterations as 2000 and the batch size to be 20. At each 100th iteration of backpropagation training, binarization is performed for the low-complexity CNN. The simulation results are illustrated in the following subsection.

Results

The classification accuracy low-complexity CNNs and conventional CNN as the SNR of the modulated signal changes from 0 to 5 dB are shown in Figs. 9 and 10, respectively.

From Figs. 9 and 10, we can observe that the classification accuracy reduces as the SNR decreases from 5 to 0 dB. For conventional CNN, classification accuracy is above 98% for all modulation schemes except BPSK. For BPSK modulation scheme, classification accuracy observed from conventional CNN is 92%. For our low-complexity CNN, the classification accuracy is above 97% for all modulation schemes except BPSK. For BPSK modulation scheme, the classification accuracy observed for low-complexity CNN is 91.2%. Therefore, based on the simulation results, we can observe that our low-complexity CNN achieves comparable accuracy in classification of modulation schemes compared with that achieved by conventional CNN. Furthermore, our proposed CNN outperforms the conventional CNN in low computational complexity. Overall, our proposed CNN achieves a good tradeoff between the performance and the computational complexity.

6. CONCLUSION

This paper proposes a deep learning-based cognitive radar system for automatic modulation classification in cognitive radio by using SCF function and low-complexity CNN method. The noise resilient SCF generate unique signature patterns for each modulation scheme and the low-complexity CNN is designed to classify the patterns. Our proposed low-complexity CNN has the advantage of containing no multipliers while a conventional CNN with the same structure requires performing 617632 floating-point multiplication operations. As illustrated in the simulation results, our proposed low-complexity CNN method achieve comparable accuracy compared with conventional CNN. As future work, we plan to study the possibility of SCF base feature extraction for higher order modulation schemes and apply the proposed method for real-time detection of modulations in experimentally captured signals.

REFERENCES

[1] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, Feb 2005.

[2] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[3] W.-Y. Lee and I. F. Akyildiz, "Optimal spectrum sensing framework for cognitive radio networks," *IEEE Transactions on wireless communications*, vol. 7, no. 10, 2008.

[4] C.-T. Chou, H. Kim, K. G. Shin *et al.*, "What and how much to gain by spectrum agility?" *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, 2007.

[5] Y. c. Deng and Z. g. Wang, "Modulation recognition of mapsk signals using template matching," *Electronics Letters*, vol. 50, no. 25, pp. 1986–1988, 2014.

[6] H.-C. Wu, M. Saquib, and Z. Yun, "Novel automatic modulation classification using cumulant features for communications via multipath channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 8, 2008.

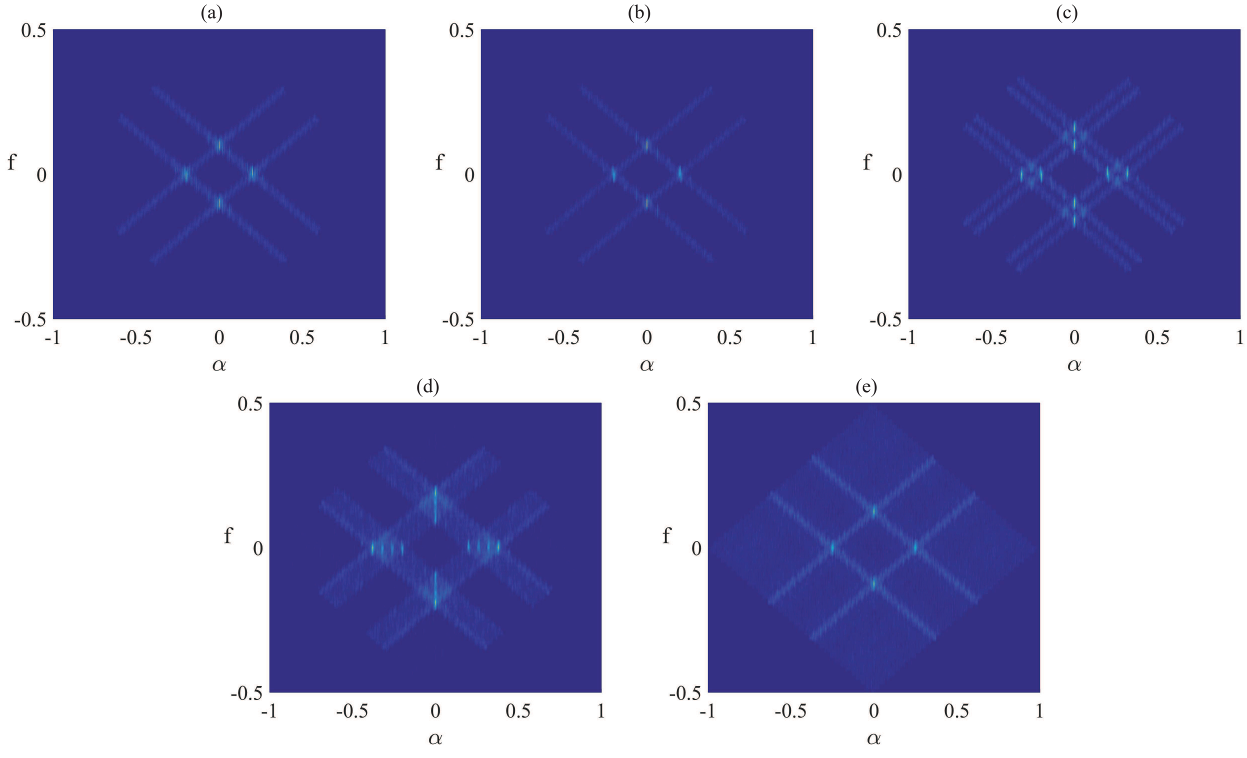


Figure 8: 2D SCF patterns of simulated modulated signal (a) BPSK; (b) QPSK; (c) 2-FSK; (d) 4-FSK; (e) OFDM with BPSK sub-carrier modulation: SNR of the simulated signals is 5 dB.

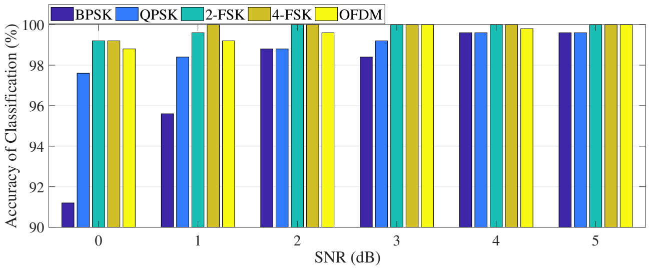


Figure 9: Accuracy of classification of modulation schemes when using low-complexity CNN as the SNR of modulated signal varies from 0 to 5 dB.

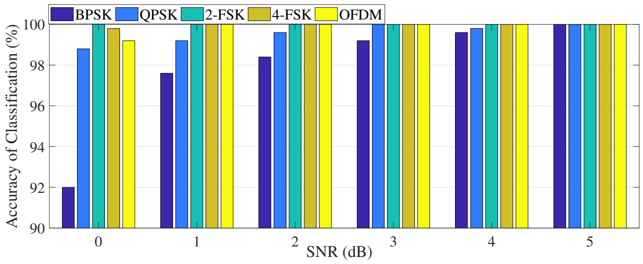


Figure 10: Accuracy of classification of modulation schemes when using conventional CNN as the SNR of modulated signal varies from 0 to 5 dB.

[7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[10] A. Venkataraman, “Deep learning algorithms based text classifier,” in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, July 2016, pp. 220–224.

[11] Q. Weng, Z. Mao, J. Lin, and W. Guo, “Land-use classification via extreme learning classifier based on deep convolutional features,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 704–708, May 2017.

[12] K. Matsumoto, Y. Tajima, R. Saito, M. Nakata, H. Sato, T. Kovacs, and K. Takadama, “Learning classifier system with deep autoencoder,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 4739–4746.

[13] G. J. Mendis, J. Wei, and A. Madanayake, “Deep learning-based automated modulation classification for cognitive radio,” in *Communication Systems (ICCS), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[14] D. D. Cox and T. Dean, “Neural networks and neuroscience-inspired computer vision,” *Current Biology*, vol. 24, no. 18, pp. R921–R929, 2014.

[15] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: a convolutional neural-network approach,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, Jan 1997.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[17] J. Yang, Y. Zhao, J. C. W. Chan, and C. Yi, “Hyperspectral image classification using two-channel deep convolutional neural network,” in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2016, pp. 5079–5082.

[18] Y. Kim and T. Moon, “Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, Jan 2016.

[19] G. J. Mendis, J. Wei, and A. Madanayake, “Deep belief network for automated modulation classification in cognitive radio,” in *Cognitive Communications for Aerospace Applications Workshop (CCAA), 2017*. IEEE, 2017, pp. 1–5.

[20] Z. Lin, M. Courbariaux, R. Memisevic, and Y. Bengio, “Neural networks with few multiplications,” *arXiv preprint arXiv:1510.03009*, 2015.

[21] W. A. Gardner, A. Napolitano, and L. Paura, “Cyclostationarity: Half a century of research,” *Signal processing*, vol. 86, no. 4, pp. 639–697, 2006.

[22] A. Fehske, J. Gaeddert, and J. H. Reed, “A new approach to signal classification using spectral correlation and neural networks,” in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*. IEEE, 2005, pp. 144–150.

[23] “P25 spectrum sensing with synthesized and captured data,” https://www.mathworks.com/help/comm/examples/p25-spectrum-sensing-with-synthesized-and-captured-data.html.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

GWN: A FRAMEWORK FOR PACKET RADIO AND MEDIUM ACCESS CONTROL IN GNU RADIO

Víctor Gonzalez-Barbone (vagonbar@fing.edu.uy)[1], Pablo Belzarena (belza@fing.edu.uy)[1], Federico Larroca (flarroca@fing.edu.uy)[1], Martín Randall (mrandall@fing.edu.uy)[1], Paola Romero (paolar@fing.edu.uy)[1], and Mariana Gelós (mariana.gelos@fing.edu.uy)[1]

[1]Facultad de Ingeniería, Universidad de la República, Montevideo , Uruguay

ABSTRACT

Software Defined Radio, and GNU Radio in particular, were conceived for communication systems such as radio and TV, where information is conveyed in a continuous flow, called a stream. Data networks by contrast use small portions of information, called messages, frames or packets according to the context. A further important difference is that in data networks several nodes share the same medium. In order to operate properly, a so-called medium access control must be enforced.

GNU Radio was originally stream oriented, but more recently added support for message communications. A block may thus comprise two different types of inputs and outputs: stream ports for continuous flows of data, and message ports for discrete portions of bytes. As a consequence, some projects that strive at implementing data network standards in GNU Radio have emerged, but they are oriented towards specific communication protocols.

In this paper we present GWN (GNU Radio Wireless Network), an open and free extension to GNU Radio specifically oriented to data networks, but not tied to any specific protocol. Its aim is to provide a framework for experimentation and development, working either on existing protocols or devising entirely new ones. To this purpose, GWN provides a new generic block (`gwnblock`) which adds the tools necessary for data network designs, and at the same time decouples all GWN data network blocks from the GNU Radio generic basic-block. This means a new GWN block only needs to inherit from `gwnblock` and follow GWN design rules to build a data network application, while keeping full compatibility and access to GNU Radio standard blocks.

The GWN generic block adds the following facilities to GNU Radio: Message orientation, Events, Handling of time, and Finite State Machines. As a proof of concept, and to illustrate its usage, we briefly present two examples: an ARQ (Automatic Repeat reQuest) protocol with its different flavors and a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol.

1. INTRODUCTION

The ultimate objective of Software Defined Radio (SDR) is to implement a complete communication system in software. Current SDR implementations allow to sample a portion of several megahertz of the spectrum and feed them into a programmable device (such as a personal computer), where a software suite processes them as required. Transmission is analogous. This allows to implement a cellular base station [1], an FM radio [2], or a Digital TV receiver [3] by running different programs on the computer, just to name a few examples.

Instead of using expensive proprietary hardware implementations, SDR allows to design and implement most of the communication system in software, which may be tailored specifically for the given application. Several SDR software suites are free and open, the most popular being GNU Radio [4]. Moreover, SDR hardware implementations also enforce some level of openness. Open-source technologies transcend geographical and cultural boundaries, as anyone in any place can contribute and collaborate. This new paradigm offers new opportunities for research and teaching in telecommunications, providing a freely available design framework apt for technical education, research, development, and deployment, through a cost-effective, easily approachable tool with modest infrastructure requirements.

GNU Radio was originally designed to support the processing of continuous data streams from a source to a sink passing through different blocks, each performing a specific task (e.g. filtering, decision). The stream is a flow of basic types like bytes, integers or complexes. Each GNU Radio block defines input and output signatures which specify the number of input and output streams and their respective type. The designer can choose which blocks are needed and how they are connected to build a flow graph for a particular physical layer implementation. GNU Radio has an internal scheduler which invokes sequentially each block and communication between blocks is performed through shared memory.

This data stream model works well for samples, bits, etc., but it is not appropriate to handle control data, metadata, and packet structures. GNU Radio partially alleviates this problem by introducing two new communication mechanisms. The *tag* system is a stream parallel to the data stream that holds metadata and control information. This mechanism allows to add additional information to a particular sample or flow, but the paradigm is still the same. A *message passing* system was also added, with two main goals: to allow downstream blocks to communicate back to upstream blocks, and to provide an easier way to communicate between external applications and GNU Radio.

As we discuss with greater detail in the following section, further extensions are necessary for GNU Radio to support packet communication, in particular handling of time and support for finite state machines (FSM), a preferred way to implement data network protocols. Thus, the main goal of our research is to provide a fully functional SDR-based wireless network, developed on top and integrated with GNU Radio. To date, we have defined a general framework which allows the implementation and use of different data link protocols (and as a consequence wireless networks) integrated to the GNU Radio project, which we have thus named *GNU Radio Wireless Networks* (GWN). This framework is being developed as a free, open source software project under the GNU license, with the explicit intent of disseminating these ideas, contribute our present achievements, and allow interested researchers and developers to contribute.

This article is structured as follows. First, we analyze the Data Link Layer protocols requirements. Based on these requirements we describe the architecture of GWN and we explain the methodology used to build the GWN framework. Next, we analyze how to use and how to extend GWN. Finally, we explain how GWN has been tested and we state the results and conclusions of this work.

2. DATA LINK LAYER REQUIREMENTS

GNU Radio and more generally SDR stem from radio frequency communications such as radio and TV. In these fields, information is conveyed in a continuous flow, called a stream. On the other hand, data networks use messages, frames or packets according to the context. A file transfer is thus carried out by dividing the mass of bytes in small packets which travel through the air on their own. Several distortions may affect their travel: their delay may be different and arrive out of order, they may be altered in their content, they may be lost and require retransmission. At the receiving end, these packets must be validated for errors and either corrected or asked for retransmission, ensure all of them have arrived, restore the correct sequencing, and then be aggregated to rebuild the file exactly as the original.

Moreover, in radio and TV the electromagnetic spectrum is divided into frequency bands, called channels, to avoid mutual interference. In bilateral radio communications either transmission and reception must happen in different bands, or the parts must take turns to speak. In data networks many actors share the same communication channel, called a shared medium. Some discipline must be imposed to avoid “all speaking at the same time". This discipline is called a *channel access method*.

All in all, data networks involve multiple users on a shared medium, use packets, these packets may suffer errors, variable delays, losses, and sequence alterations. These and other problems are addressed in a number of standard which regulate network communication, from small local area networks to the Internet.

At data link layer, each protocol control logic is typically based on a state machine that implements a set of service primitives. State machines are very effective in modeling the behavior of sequential control operations, and most MAC protocols and other link layer protocols are formally described in terms of state machines [5].

The state machine reacts to different actions performing state changes and/or generating a new set of actions. These actions may handle control or management data units, but also modify or reconfigure this block or other blocks. These actions and service primitives can be implemented as asynchronous events. This is another main requirement of data link layer protocols: the communications between blocks must be driven by events. A block may also require a service primitive from another network layer. For example a CSMA/CA block will need to ask the physical layer for the channel state (if it is idle or not).

In addition to the above, there is another important type of event that must be handled at data link layer: timer events. For example in CSMA/CA or in ARQ protocols each time a packet is sent, the control logic of the protocol must start a timer. When the timer ends, it generates an event informing the control logic of its expiration. If a packet with an acknowledgment arrives after the timer expiration the packet must be resent, and if it arrives before the timer must be stopped. Therefore, data link layer protocols require to handle a set of asynchronous timers.

GNU Radio was originally stream oriented, but as we explained in Sec. 1., it recently added support for message communications. A block may thus comprise two different types of inputs and outputs: stream ports for continuous flows of data, and message ports for discrete portions of bytes. This is an important step to allow data link layer protocols implementation in GNU Radio. However, as discussed before, it is also necessary to associate a finite state machine to each control block, as well as the capacity of handling events and (several) timers also associated to each control block.

Some projects exist to implement data network standards in GNU Radio (for instance [6]), but they are oriented towards partial implementations of specific communication protocols. We developed GWN as an extension of the GNU Radio toolkit specifically oriented to data networks, but not tied to any specific protocol; its aim is to provide a framework with the tools for experimentation and development, working either on existing protocols or devising entirely new ones.

Summarizing, a framework for implementing data link layer protocols must be modular, flexible and adaptable, not only to allow modification or replacement of a certain protocol but also to seamlessly include new and future network architectures with a moderate effort. This framework must include the capabilities to easily implement finite state machines and timers, as well as the ability to handle different types of events.

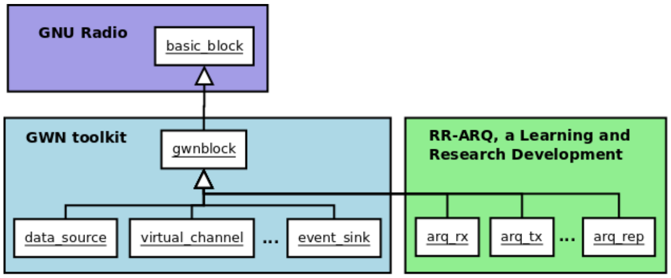


Figure 1: GWN blocks.The GWN toolkit provides a generic `gwnblock` which inherits from GNU Radio *basic_block*. All GWN toolkit blocks and blocks developed by students and researchers inherit from `gwnblock`. RR-ARQ is an example implementation of Automatic Repeat reQuest (ARQ), an error control protocol for data transmission.

In view of the success achieved by GNU Radio, our approach was to follow its design model and to add functionalities to GNU Radio blocks. This is discussed in the following section together with an introductory description of GWN.

3. GWN ARCHITECTURE

GWN extends GNU Radio towards data networks in a toolkit with its own features. Blocks in GWN and GNU Radio can be mixed in the same flowgraph. New GWN blocks can be built alongside the GWN design in the certainty of their compatibility with both GWN and GNU Radio. To this purpose, GWN provides a generic `gwnblock` which adds the tools necessary for data network designs, and at the same time decouples all GWN data network blocks from the GNU Radio generic `basic-block`. This means a new GWN block only needs to inherit from `gwnblock` and follow GWN design rules, shielding from users most of the complexity of GNU Radio.

In addition to the GWN generic block, the GWN toolkit includes some common function blocks such as message sources and sinks, a channel emulator, message converters, and framers. This allows for the demonstration of basic network data communications just by interconnecting GWN blocks in a flowgraph. Figure 1 shows how learning and research developments need only interact with GWN in their construction. This architectural scheme simplifies access of students to development, both in coding and in documentation. In this sense, special care has been taken to provide complete, readable documentation on all GWN blocks and tutorial material on new block development.

In particular, the GWN generic block adds the following facilities to GNU Radio.

1. Message orientation. GNU Radio is mainly stream oriented, GWN is message oriented; items interchanged among GWN blocks are discrete groups of bytes. GWN makes use of the message mechanism of GNU Radio, but provides some blocks to interact with stream GR blocks when necessary, thus relieving users of stream oriented worries.
2. Events. GWN elaborates on the message interchange mechanism of GR into a more structured item of interchange called an *event*. GWN blocks interchange events. The event inner structure reflects the needs of network data protocols and is closer to their design conception.
3. Handling of time. This is a feature absent in GNU Radio, and essential in data networking. Answers are waited for a certain time, keep-alive signals are emitted at regular intervals; timing pervades data communications. GWN provides two forms of handling time: timeouts and timers. A timeout just waits for some time and emits a timeout event; it is a one-shot gun. A timer emits timing events regularly.
4. Finite State Machines. Most data communication protocols involve a complex logic usually described in a mathematical model of computation called a Finite State Machine (FSM). An FSM comprises *states* and *transitions*, and reacts to events: when the machine is in a certain state and receives an event, a transition to another state is performed, optionally with some parallel task. FSMs are a very powerful tool, and the complexity of some protocols makes it almost impossible to implement them otherwise.

All GWN blocks communicate among themselves by the interchange of events. An event is an instantiation of the GWN Event class, described by a nickname, and including a type, a subtype, and optionally other items in dictionary form. The GWN Event class is intended to be subclassed in a hierarchy of different event types. This allows to define event classes which closely reflect the contents of the different types of packets used in data networks, were they control, management or data. Besides, events can also be used to interchange information among blocks, such as the timer events used to start a timer, signal a timeout, or perform some action at regular periods for a number of times.

The inner structure of a typical GWN block is shown in Figure 2. Messages are received and sent as PMTs (Polymorphic Data Type), the standard data type in GNU Radio. From these messages the encapsulated events are recovered and passed to the `process_data` function, the only place where a programmer must code the functions of the block. This function also receives events from timers and timeouts, and from the FSM. According to the events received, and the logics of the function to be performed, a new event is generated and sent to the output ports. Encapsulation and recovery of events from PMT messages is done in the input and output ports in a way transparent to the programmer, who may just think in "events".

To create a new block, the user indicates the number of input and output ports, and codes the logic in `process_data`. Optionally, the number of timers and timeouts may be indicated, and an FSM may be loaded with transition rules. This is a quite straightforward process: inheritance from `gwnblock` ensures ports, timers, timeouts and FSM work as expected. The use of the PMT data type to interchange among blocks makes GWN blocks fully compatible with GNU Radio blocks.

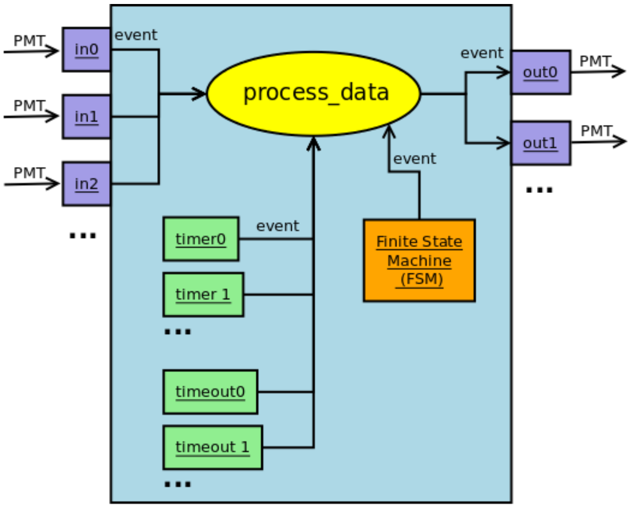


Figure 2: A typical GWN block. Messages (PMT) received on input ports are decoded into events and served to the `process_data` function, which also receives events from timers, timeouts and the FSM. The `process_data` function generates and emits events through the output ports as messages (PMT).

GWN includes a simplified version of an eXtended Finite State Machine (XFSM) which has been used to implement complex packet processing tasks inside network switches, and is considered a powerful enough tool to implement any protocol for data networks [5].

The framework is built on a modular architecture of building blocks, where each block performs some specific functions. These blocks communicate with each other according to protocols implemented as asynchronous message events. This concept allows for the integration of several blocks to form a new block capable or performing a set of related tasks. Thus, it becomes relatively simple to integrate functionalities implemented in different blocks to cover the different requirements of a wireless network, such as medium access, neighbors discovering, etc. The GWN code and documentation may be found in [7].

The next section describes the methodology of research and design used in this project, where one of the most difficult challenges is to evolve from sheer prototyping into sound architectural design.

4. METHODOLOGY

In the area of telecommunications, as well as in many other technical areas, research is based on extensive prototyping, followed by testing, correcting, and further prototyping. In this way, many proposals in the area end up with a weak design in software architecture. Since considerable work has been done, and positive results have been obtained, architectural design naturally falls to a second plane. In many cases, once the desired results have been obtained, and the final prototype is working as desired, there is not much motivation to worry about software architecture, so much so that it would imply to almost start all over again. This is not the case with this project: a far reaching enterprise as this calls for a reasonable architectural design, well modularized, with clear interfaces, consistent patterns of development, and quality documentation, so as to make it apt for reuse and extension.

The first stage of this project was mainly prototyping, with a rather flat package and module organization; its purpose was the development and testing of software modules to implement a wireless network link in software over a generic piece of hardware (e.g. USRP) using GNU Radio for the physical layer. Once this goal was achieved, some architectural requirements specific of the project started to emerge. An analysis and evaluation of existing architectural designs and patterns was carried out, looking for the best approach towards an architectural design of the framework. The GNU Radio model was also studied and evaluated.

The convergence of prototyping and architectural design evaluations led to a first approach in the architectural design of the GWN framework. Needless to say, the first draft of a software architecture, in a far reaching enterprise as this, is most critical. At the same time, it is very difficult to envision all possible architectural requirements at an early stage. For this reason, our methodological approach was conceived as a balance between prototyping and design, in which design tries to apply good software engineering practices to the needs discovered by prototype work, and prototypes test the usability of the design, show its shortcomings, and help improve design.

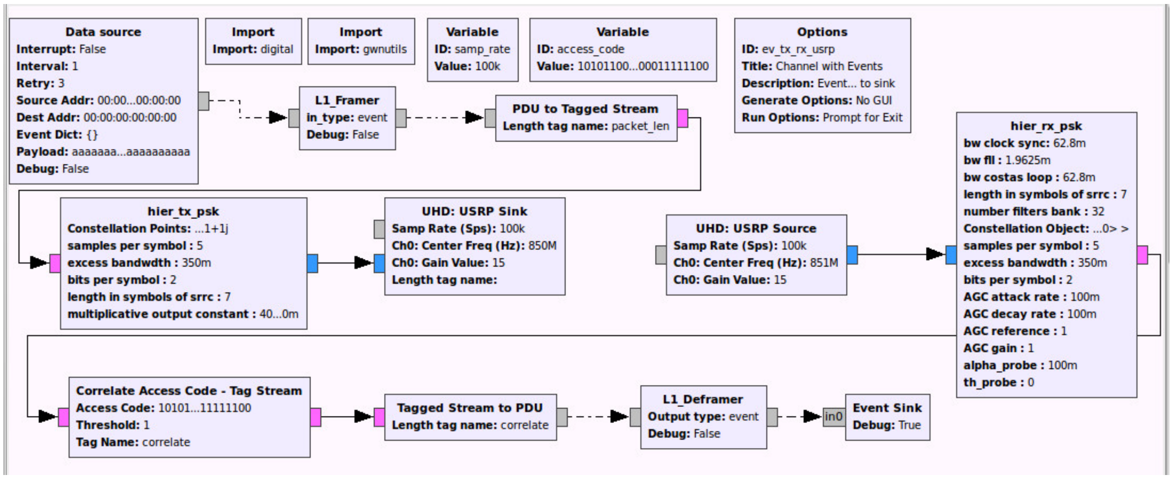
5. FIRST STEPS USING GWN

5.1. A working example

GWN allows a step by step construction, ideal for showing and experimenting how each block performs its duty. The simplest flowgraph is an event source connected to an event sink: events produced in the first block are displayed by the second. From this on, gradual addition of blocks may lead to the simulation of a transmission over air, using a channel emulator block in place of hardware and air.

Experimenting a communication link with emitter and sender in the same personal computer, as described above, is a must task for beginners. Once the simulation works, substituting the channel emulator by the SDR hardware, one for emission and the other for reception, should render a working, real communications link. In this way two computers can be interconnected, and start a chat session, a file transfer, or a graphical application in the remote machine. Figure 3 shows the flowgraph to establish a bilateral link with another node.

Along the former lines, data network protocols can be implemented, tested, and improved, starting from simulation and ending in real world communications. This brings a hands-on experience on the many difficulties data network communications face, and the effectiveness of protocols to achieve reliable



Data source
Interrupt: False
Interval: 1
Retry: 3
Source Addr: 00:00...00:00:00
Dest Addr: 00:00:00:00:00:00
Event Dict: {}
Payload: aaaaaaa...aaaaaaaaaa
Debug: False

Import
Import: digital

Import
Import: gwnutils

Variable
ID: samp_rate
Value: 100k

Variable
ID: access_code
Value: 10101100...00011111100

Options
ID: ev_tx_rx_usrp
Title: Channel with Events
Description: Event... to sink
Generate Options: No GUI
Run Options: Prompt for Exit

L1_Framer
in_type: event
Debug: False

PDU to Tagged Stream
Length tag name: packet_len

hier_rx_psk
bw clock sync: 62.8m
bw fll : 1.9625m
bw costas loop : 62.8m
length in symbols of srrc : 7
number filters bank : 32
Constellation Object: ...0> >
samples per symbol : 5
excess bandwdth : 350m
bits per symbol : 2
AGC attack rate : 100m
AGC decay rate : 100m
AGC reference : 1
AGC gain : 1
alpha_probe : 100m
th_probe : 0

hier_tx_psk
Constellation Points: ...1+1j
samples per symbol : 5
excess bandwdth : 350m
bits per symbol : 2
length in symbols of srrc : 7
multiplicative output constant : 40...0m

UHD: USRP Sink
Samp Rate (Sps): 100k
Ch0: Center Freq (Hz): 850M
Ch0: Gain Value: 15
Length tag name:

UHD: USRP Source
Samp Rate (Sps): 100k
Ch0: Center Freq (Hz): 851M
Ch0: Gain Value: 15

Correlate Access Code - Tag Stream
Access Code: 10101...11111100
Threshold: 1
Tag Name: correlate

Tagged Stream to PDU
Length tag name: correlate

L1_Deframer
Output type: event
Debug: False

Event Sink
Debug: True

Figure 3: A flowgraph to establish a bilateral link with another node. The `Data source` block produces events at regular intervals. Event objects are serialized and packed as PDUs (Protocol Data Units) in our block `L1_Framer`, which are transformed into tagged streams by GNU Radio's `PDU to Tagged Stream` block. These are modulated into a complex signal and sent to the `UHD: USRP Sink`; this block interacts with the RF device, which sends the signal over the air. At the same time, this same node is receiving messages through the `UHD: USRP Source`, which captures the signal from the air. The received stream of bytes is correlated to detect an access code in `Correlate Access Code - Tag Stream`, and the resulting tagged stream is fed into `Tagged Stream to PDU`, and `L1_Deframer` recovers the original event produced by the sender. Please note that all blocks present in the flowgraph are included in GNU Radio or GWN.

data transfer among a group of nodes, i.e. computers with RF peripherals.

5.2. Building a new block

GWN can be extended by creating new blocks to implement new functionalities. GWN has been coded in Python, and can be extended using the same language. Though in the future some GWN blocks are expected to be rewritten in C++ for performance reasons, extensions in Python will always be possible, as in GNU Radio. This section describes how to create a new GWN block by means of a simple example. In what follows, we create a block called `Virtual Channel`, a block we mentioned at the beginning of this section. In particular, this block receives an item, generates a uniform random number in the interval $[0, 1]$, and writes out the item received only if the random number is greater than a probability of loss set as a parameter of the block. The block has one input, and one output.

After a new block is created (for which GNU Radio's `gr_modtool` may be used) we first add the following imports to the code of the block:

```
from gwnblock import gwnblock
import pmt
```

Next, we modify the class definition to inherit from `gwnblock`, the GWN generic block including implementations for message ports and internal timers, and write its constructor according to the functionality described before. That is to say:

```
class virtual_channel(gwnblock):
```

```
def __init__(self, blkname='virtual_chanel
    ', blkid='id_virtual_channel',
    prob_loss=0):
        gwnblock.__init__(self, blkname=
            blkname, blkid=blkid,
            number_in=1, number_out=1,
            number_timers=0)
                self.prob_loss = prob_loss
                return
```

Now we write the processing function (i.e. `process_data`), which in this case will generate a random number and write out the received message only if it is greater than the parameter `prob_loss`:

```
def process_data(self, ev):
        rand_nr = random.random()
        if rand_nr <= self.prob_loss:
                pass
        else:
                self.write_out(ev)
        return
```

GNU Radio provides a graphical interface to build flowgraphs interconnecting blocks; this is called the GNU Radio Companion, or GRC for short. To make the new block available in GRC, an XML file describing the block's ports and parameters is required. A template version for the XML file is created by the `gr_modtool` script, and must be updated as usual in GNU Radio.

6. GWN IMPLEMENTATION OF A FSM

This section describes the GWN implementation of a Finite State Machine (FSM). An instance of the GWN FSM can be associated to a GWN block to implement its logic.

In addition to the usual states and transitions, GWN's FSM includes *actions*, *memory*, and *conditions*:

- An action is a user-written function executed on a transition, before setting the machine to the next state.
- Memory may be any object capable of recording and retrieving information, in whatever access mode the application may need (LIFO, FIFO, etc). The memory facility is not part of the FSM machine, but an independent object. Memory may be handled in the action functions.
- A condition is another user-written function or expression which produces True or False when executed or evaluated. The action function and the transition are only executed if the condition evaluates to True. If the condition evaluates to False, no action is executed and the machine remains in its current state.

The FSM is defined through tables of transitions. For a given input (termed *symbol* in this context) the `process()` method of the FSM uses these tables to decide which action to call and which the next state will be, if and only if the condition evaluates to True, otherwise nothing happens.

The table of transitions defines the associations `(input_symbol, current_state)` –> `(action, next_state, condition)`, where `action` is a function, `symbol` and `state` can be any object, and `condition` is a function or an expression which returns a boolean. This table is maintained through the FSM methods `add_transition()` and `add_transition_list()`.

In the following example we show the definition in the code of a transition:

```
fsm.add_transition ('goA', 'INIT', fn_goA,
    'State A', "self.where=='A'")
```

This code adds a transition to the FSM `fsm` where if the input symbol is `goA`, the FSM is in state `INIT` and the variable `self.where` is equal to `'A'`, then the function `fn_goA` is invoked and the FSM moves to the state `State A`.

Transitions valid for any input symbol may also be defined. That is to say, associations of the kind `(current_state)` –> `(action, next_state, condition)`. This table is maintained through the FSM method `add_transition_any()`.

Finally, the FSM may also have a default transition not associated with any specific input symbol or state. The default transition matches any symbol on any state, and may be used as a catch-all transition. The default transition is set through the `set_default_transition()` method. There can be only one default transition.

Thus, upon receiving a symbol, the FSM will look in the transition tables in the following order: 1. The transitions table for `(input_symbol, current_state)`. 2. The transitions table for `(current_state)`, valid for and any input symbol. 3. The default transition. 4. If no valid transition is found, the FSM will raise an exception.

Matched transitions with the former criteria may produce a list of `(action, next_state, condition)`. The condition is evaluated for each tuple in the list, and the first tuple on which the condition is found True is executed. That is to say, the corresponding action function is called, and the next state is set as the current state. If no transition is defined for an input symbol, the FSM will raise an exception. This can be prevented by defining a default transition. The action function receives a reference to the FSM as a parameter, hence the action function has access to all attributes in the FSM, such as `current_state`, `input_symbol` or `memory`. The GWN Finite State Machine implementation is an extension of Noah Spurrier's FSM [8].

7. GWN TIMERS

This section describes how to add timing to GWN blocks. GWN internal timers are objects that can be attached to any block, in the same way as input or output ports. The number of timers to create in a block can be indicated as a parameter in the invocation to the `gwnblock` constructor as shown in Sec. 5.2. with the Virtual Channel example. Let us recall that all GWN blocks inherit from GWN's basic block `gwnblock`.

In its present version, GWN provides two different mechanisms for timing: GWN Timers and GWN Timeouts. Both act sending messages to the block to which they are attached. Messages from timing blocks are made available in the block's `process_data()` function, as if they had been received at an input port. Timing messages can be recognized by their type, i.e. a timing message must be in some way different from messages received at any input port.

7.1. Using GWN Timers

A GWN Timer sends a first type of message to the block to which it is attached to at regular intervals, for a given number of times. Then it sends a single second type of message to indicate the first series has exhausted. A GWN Timer accepts the following parameters:

- `nickname1`: message to send at regular intervals for `retry` times.
- `interval`: time between messages.
- `retry`: how many times to send message 1, then send message 2 once.
- `nickname2`: message to send when retries have exhausted.
- `interrupt`: if True, the timer is interrupted, i.e. it does not send any messages, but keeps alive, and the `retry` counter

keeps counting; when set to False, sending of messages is restored.

A GWN Timer can be controlled through the following functions:

- `set_interrupt(True | False)`: sets interrupt state.
- `stop()`: stops the timer and no more messages will be sent.
- `reset()`: sets the counter to 0, interrupt to False, and starts counting again and sending messages.

The following excerpts of code show the use of internal timers in a block called `Timer Source`. This is a simple block which produces messages regularly as 'timer events'. These messages can be used by other blocks to trigger some action.

This block uses an internal timer to produce a certain type of event regularly with a given interval (the `interval` parameter), for a specified number of times (the `retry` parameter). Events produced are of the type defined by the parameter `nickname1`. Once the retry number has exhausted, a final event of the type defined by the parameter `nickname2` is written out.

```
class timer_source(gwnblock):
'''Timer events source, sends Events
    produced by an internal timer.

def __init__(self, blkname='timer_source
    ', blkid='timer_source_id', interrupt=
    False, interval=1.0, retry=5,
    nickname1='TimerTOR1', nickname2='
    TimerTOR2'):
     # invocation of ancestor constructor
         gwnblock.__init__(self, blkname,
             blkid,number_in=0, number_out
             =1, number_timers=1)
         self.counter = 1  # counts until
             retry
         self.time_init = time.time()  #
             retuns current time
         self.set_timer(0, interrupt=
             interrupt, interval=interval,
             retry=retry, nickname1=
             nickname1, nickname2=nickname2
             )
         self.start_timers()
         return
```

The timer events produced by this block are received by this same block as any received event, hence the function `process_data()` should be written to handle this type of event, as any other. The type of event can be determined within the function to act accordingly.

```
def process_data(self, ev):
         '''Sends timer events produced by
             the internal timer.
         @param ev: an Event object.
         '''
         ev.frmpkt = 'Timer Event ' + str(
             self.counter)
         self.counter += 1
         self.write_out(ev, port_nr=0)
         return
```

7.2. Using GWN Timeout

A GWN Timeout object sends a message after some specified interval has elapsed. It can be interrupted before its action starts, and hence no message will be received. It can also be restarted in a new cycle. A GWN Timeout accepts the following parameters:

- `timeout`: the time before message is sent.
- `nickname`: message to send.

A GWN Timeout can be controlled through the following functions:

- `start(timeout=None, nickname=None)`: starts timeout counting.
- `cancel()`: stops counting. If `timeout` has not been reached no message will be sent.

A note on efficiency. A GWN Timeout may be more computer efficient than a GWN Timer, since a GWN Timer runs in its own thread, which cannot be destroyed without destroying the GWN Timer object. On the other hand, the GWN Timeout creates a Python `threading.Timer` object, which runs in its own thread, and is destroyed on invocation of the GWN Timeot `cancel()` function. This is more accurate, since the `threading.Timer` object is immediately canceled, without waiting for object destruction (creation and destruction of objects may be demanding in process time).

8. GWN AS TEACHING TOOLS

Our goal in the development of GWN is to build a framework that can be usable for education, experimentation, and research in wireless networks. In this section we explain how we have used GWN as teaching tool. Our integration of GWN into Education was carried out along these trends. First, use of GWN as such, in demonstration of data oriented communication and the problems involved, e.g. losses and corruption of messages, both in simulation and real word communications, by implementing a data link between two personal computers. Second, extension of GWN, adding blocks for new functionalities. The goal here was twofold: the implementation per se and the early training of students in research.

Moreover, documentation of blocks is quite complete, tutorial information is given in the project's wiki [9], example flowgraphs can be found in the examples subdirectory after installation, and also on the project's homepage [7]. In this version

GWN is coded in Python, and extensions can be also written in Python. This makes code more accessible to students working on extension projects. A migration to C++ is expected to occur in the near future, for performance reasons, but it will only affect `gwnblock` and associate classes, which are maintained by the GWN team; student developments will not be affected. As to the fast renewal of versions in GNU Radio, extensions of GWN are shielded from them by `gwnblock`, because all GWN blocks inherit from it.

Though GWN is used for demonstrations in the classroom, it excels in the lab, where students can interact freely with it. Educational purposes pursued include demonstration and experimentation in data networks, where students can see "in action" what they learned in introductory courses on data networks. Furthermore, we make use of the toolkit for class assignments and small projects, using the available blocks or simple adaptations. The latter requires some basic knowledge of Python, but the structure of GWN blocks allows to include code only in the `process_data` function; the task is easily achieved with some guidance. Finally, we have used the framework for end of course projects, graduation projects in telecommunications engineering, research, and research training, both in undergraduate and graduate levels.

In the following section we describe two application projects: students implemented ARQ and a CSMA/CA protocol, as an extension to GWN. This projects required the use of timing and the implementation of Finite State Machines. The following section illustrates the use of these two features of GWN.

9. TESTING THE FRAMEWORK

Automatic Repeat reQuest protocols are layer 2 control protocols for ensuring packet exchange. Even though there are different versions ranging from simplicity to efficiency, they all share some common premises: packets are to be acknowledged (and otherwise considered lost), and packets are to be delivered in proper order to their destination.

In order to test GWN as a simple but complete framework for communications protocols, the task of developing the ARQ protocols was conveyed to undergraduate students. With no prior knowledge of GNU Radio programming, and using the tutorial provided by the GWN team, three variants were implemented and tested (Stop and Wait, Go Back N and Selective Repeat), both in GRC simulation flowgraphs and real wireless communication. A second programming of the protocols was done using FSM, with their respective tests.

Finally, as part of a graduate project in telecommunications engineering, a CSMA/CA protocol was implemented using FSM. It was also tested in both simulation and real life communications, implementing a simple chat application.

These protocols were implemented in two GWN blocks: one in the transmission node with the logic of the protocol, and the other in reception mainly for acknowledgement. In the next sections we discus the Selective Repeat and CSMA/CA protocols as we believe them to be the most illustrative.

9.1. ARQ: Selective Repeat in FSM

The Selective Repeat protocol is the most complex ARQ protocol as it implies some data processing at reception. The transmitter uses a sliding window for sending packets and each packet has a timeout of its own, which starts when the packet is sent. When an acknowledge (Ack) for this packet is received, the packet is removed from the window, freeing a slot for a new packet. If the timeout of a sent packet expires and no Ack has been received, the packet is retransmitted. If the transmission window is full, new packets to send are stored in a buffer until a free slot appears in the transmission window. In the reception node, received packets are placed in order by their sequence number in a reception window, while their respective Acks are sent back to the transmitter. Would there be a missing packet in the reception window, a "not acknowledged" message (Nak) is sent back to the transmitter. Fig. 4 shows the implemented FSM diagram.

For this implementation, the following considerations were made:

- Packets to be transmitted turn up in sequence number order; reception of these packets in their correct order is ensured by the protocol.
- There is one transmitter and one receiver at each node, and communication is limited to two nodes.

The assignment of timeouts to packets were implemented as dictionaries. The event types used were Timeout, Ack, Nak or Data. Simulation tests were carried out using a `Virtual Channel` block. `Data source` and `Event Sink` blocks were used at the ends. Real wireless tests were carried out using the same physical layer as in in Fig. 3. This is basically a discrete digital communication system using QPSK at 850 MHz, including CRC checking, with gain, frequency, phase, and time corrections in reception. For this experiment, two USRPs B100 were used. Both simulated tests and real communication using ARQ proved successful as all data sent was received in the correct order.

Besides the specific implementation of the protocol, this instance proved the GWN framework to be within easy reach of new users, who successfully added new functionalities by creating new blocks, in a seamless extension to GWN.

9.2. CSMA/CA

The Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) is a data communications protocol used as a part of the Medium Access Control sublayer (MAC). As a part of a graduation project in telecommunications engineering, a simple CSMA/CA protocol was implemented using GWN.

The proposal was to implement CSMA/CA over a Stop And Wait ARQ protocol. Before sending a packet, the channel is

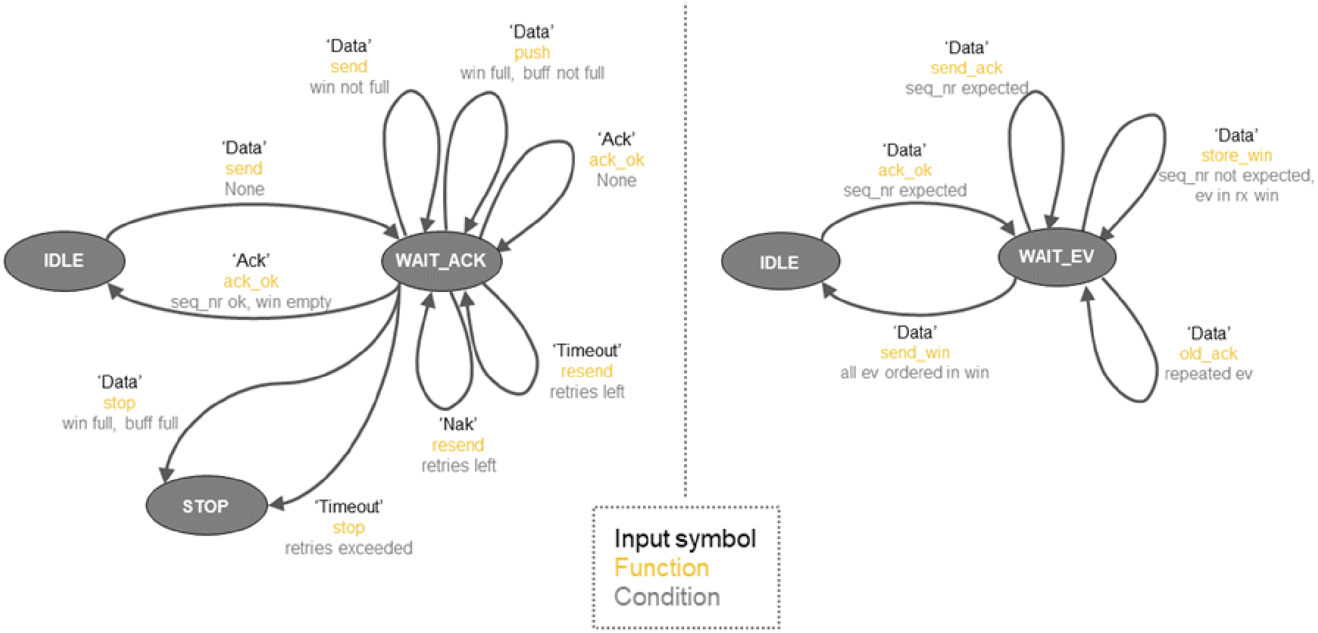


Figure 4: FSM implementations of transmitter (left) and receiver (right) for selective repeat protocol.

measured for activity. If the medium is busy, a exponential backoff algorithm starts a (random) timer. If the medium is free, the packet is sent. When the channel is busy and a timeout arrives, a new measure of activity in the medium is made. In case of activity, the process is restarted (random timer, with times possibly incremented). There is no RTS (Request To Send) nor CTS (Clear To Send) exchange in this simple version of a CSMA/CA, and the times of real wireless communications are not considered, as the goal of the project was not to fulfill the many requirements and efficiency of established protocols (such as IEEE 802.11) but to build a simple MAC with shared medium access control using the GWN and GNU Radio toolkit.

Activity in the medium is determined by a measurement of signal power at the USRP source block, by means of GNU Radio block `Probe Avg Mag^2`. The CSMA block included an FSM, importing the value of the Probe block as a condition to be read at transitions. The import was done through the usage of cheetah embedded in the XML code of the CSMA block, leading to a pretty straight way to read data into the CSMA's code.

The first test was a simulation with a self-generated noisy signal. The protocol was tested in real wireless communication by carrying out a three-people chat between USRPs. This implementation of the CSMA protocol was then used to communicate three autonomous mobile robots in a graduation project; to this purpose, it can optionally send debugging information to the chat client for measuring QoS (Quality of Service). Modulation was the same shown in 3 although adjustments had to be made for different USRP models, distances, and personal computers. Tests used combinations of USRP B100, B200, B200mini, and different kinds of personal computers.

This experience involved mainly by two undergraduate students, one of them with some prior work related to GWN, and the other without any experience in GWN / GNU Radio programming. The time and effort dedicated to the project was quite reasonable in view of the results obtained. This can be traced to the modularity and extensibility of design, as well as to the care given to documentation and tutorial material in GWN.

10. RESULTS AND CONCLUSIONS

The initial motivation for this project sprang from two different areas: research and education. While addressing questions on wireless data networks, such as cognitive networks and protocol efficiency, we felt the need of experimenting some ideas in real communication. In education, we carry on several courses on data networks and wireless communications, and were in search of a hands-on way for the students to appreciate the internals of data networking protocols, experiment their pros and cons, and attempt some improvements on their own ideas.

GNU Radio and the GNU Radio Companion seemed the ideal tool to face both needs, for several reasons: separation of functions in specific blocks, a graphical presentation, a practicable extension mechanism, simulation, open and free availability, low cost wireless hardware, and accessibility on a low budget. Hence, the question became: how can GNU Radio be extended so as to allow real world communications demonstration and experimentation, while at the same time follow the design patterns of data networks protocols as closely as possible? The new framework should be friendly to data network specialists, or at least close to their expectations. After trying several approaches, the notion of an Event object for inter block communication, their conversion to and from GNU Radio PMTs, and the integration of Event inputs and outputs in a generic GWN block, proved to bridge the gap between wireless communication and data networks quite successfully. Handling of time and FSM machines were added and made accessible through the GWN block, thus completing the picture.

After some years of development, the GWN framework is now considered to be mature enough to be used both in research and education. It has been used in several editions of data network courses, both for demonstration and experimentation, and some undergraduate and graduate projects have used it for validation and measurement. Though the set of data network oriented blocks provided by GWN is still modest, it has proved to be enough for simple projects, and easily enhanced to provide new functions. Extensions are expected to come in a natural way as the use of GWN becomes more extended. To date, its main limitation is performance. Though this is a drawback of all SDR based projects, GWN blocks are nowadays coded in Python; rewriting `gwnblock` and some other essential GWN blocks in C++ is expected to improve on performance, even if other new blocks are coded in Python, a must for fast prototyping and wide accessibility.

In our experience, GWN has proved a valuable tool in education and research, and its use in industrial prototyping is not to be discarded. It has proved to be accessible to students and to network specialists as well, and extensions through the creation of new GWN blocks came to be quite straightforward. All these tools are open and free; they can not only be obtained immediately and at no cost, but can also be explored in its internals, modified, extended, or applied to new developments with no limitations. This opened a universe of potential realizations never seen before, and within reach of even very small budgets. This is very good news for underdeveloped countries.

REFERENCES

[1] Range Networks, “OpenBTS.” [Online]. Available: http://openbts.org/

[2] Ettus Research, “How to build an FM receiver with the USRP in less than 10 minutes.” [Online]. Available: https://kb.ettus.com/Implementation_of_a_Simple_FM_Receiver_in_GNU_Radio

[3] F. Larroca, P. Flores Guridi, G. Gómez Sena, V. González Barbone, and P. Belzarena, “An open and free ISDB-T full_seg receiver implemented in GNU Radio,” in *Wireless Innovation Forum Conference on Wireless Communications Technologies and Software Defined Radio (WInnComm ’16)*, 2016.

[4] GNU Radio, “GNU Radio webpage.” [Online]. Available: https://www.gnuradio.org/

[5] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, “Wireless mac processors: Programming mac protocols on commodity hardware,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1269–1277.

[6] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, “An IEEE 802.11a/g/p OFDM receiver for GNU Radio,” in *Proceedings of the Second Workshop on Software Radio Implementation Forum*, ser. SRIF ’13. New York, NY, USA: ACM, 2013, pp. 9–16.

[7] ARTES, “GWN, the GNU Radio Wireless Network project, homepage.” [Online]. Available: https://github.com/vagonbar/gr-gwn/

[8] N. Spurrier, “Noah Spurrier’s FSM.” [Online]. Available: http://www.noah.org/python/FSM

[9] ARTES, “GWN, the GNU Radio Wireless Network project wiki.” [Online]. Available: https://github.com/vagonbar/gr-gwn/wiki

SOFTWARE DEFINED RADIO IMPLEMENTATION OF A DVB-S TRANSCEIVER

Ashwin Amanna, James Bohl, Zachary Goldsmith (ANDRO Computational Solutions, LLC, Rome, NY, USA; aamanna@androcs.com, jbohl@androcs.com, zgoldsmith@androcs.com); Michael Gudaitis, Benjamin Kraines, Robert DiMeo, William Lipe, Richard Butler II (Air Force Research Lab RIT, Rome, NY, USA; michael.gudaitis@us.af.mil, benjamin.kraines.1@us.af.mil, robert.dimeo@us.af.mil, william.lipe@us.af.mil, richard.butler.10@us.af.mil)

ABSTRACT

Most waveforms operating on Software Defined Radios (SDR) are actually ‘firmware’ defined implementations with significant elements operating in the FPGA on closed platforms. This diminishes the full potential of SDR in terms of rapid development time cycles, accessibility to waveform code, and adaptable rapid reconfiguration. We address the challenges of rapid waveform development on SDR using the Digital Video Broadcast Satellite (DVB-S) standard as a case study with a true software defined implementation where all I/Q processing is performed on an Intel processor in conjunction with low-cost Ettus B205mini and Nuand bladeRF SDRs. We distill the waveform to core functional components and sequence the implementation into “sprints” while maintaining end-to-end functionality at each iteration. Innovations include strategic use of machine code for computational intensive operations and efficient multi-thread management. Our approach yielded a full transceiver implementation within 2 man-weeks using only the published specification for reference. The transmitter was tested for interoperability with a consumer satellite receiver. CPU usage on an i7 processor was approximately 22%, with a memory usage of 16MB out of the 8GB available. The mean latency of the system is approximately 50 milliseconds. A similar test showed that the system latency is affected by symbol rate and decreases as the rate is increased. When the symbol rate is set to 15 Msymbols/sec, the latency drops to 17 milliseconds.

1. INTRODUCTION

Early promises of software defined radios (SDR) included accelerated development timelines, greater accessibility to waveforms, and ease of modification. However, the general-purpose processors of the 1990’s were not powerful enough to support complex waveforms that have demanding digital signal processing (DSP) algorithms. This led to implementations with firmware emphasis where most complex processing was performed in a FPGA. A drawback of this FPGA trend was lack of portability across different FPGA products, which results in closed systems, longer development times, and inability to adapt waveform code on-the-fly. Recent improvements in general purpose processors (GPP) to accommodate graphics processing allow GPPs to handle more DSP functions and thus enabling the original potential of SDR. As a case study for this GPP-based software approach, we present a 100% software implementation of a Digital Video Broadcast Satellite (DVB-S) standard transceiver on low-cost hardware with all I/Q processing performed in a general-purpose processor [1].

A practical challenge for affordability was to limit the total hardware costs to less than $3000 for a laboratory demonstration. This affordability constraint encouraged developers to seek innovative technical optimizations while working within the constraints of the low-cost hardware and processing-intensive operations of forward error correction (FEC) with the GPP. Government waveform efforts typically follow a traditional requirements-based acquisition cycle. We overcame this slower approach by adopting an agile-based development approach.

Our methodology to waveform development starts by distilling the waveform into its core functional components. We then simplify or eliminate blocks to implement an initial minimal functional prototype. From here, we incrementally add components and adapt existing elements to match the configurations defined in the waveform specification.

To achieve efficient operations on a GPP platform, we implement targeted functions in assembly code. Similarly, we divided the waveform code into multiple threads to equalize multi-core utilization. The Viterbi decoder, filtering and carrier recovery proved to be the most difficult to optimize. Interoperability was demonstrated using our transmitter with an off-the-shelf Coolsat DVB-S satellite receiver. We quantified bit error rate performance, CPU usage of transmitter and receiver signal flows, and measured latency.

We have shown that the GPP platform can achieve performance previously reserved for firmware implementations. Our agile-based development process yields waveforms significantly faster than a traditional requirements-based approach and proven through interoperability with off-the-shelf devices. The structure of this paper is as follows: we summarize existing software implementations of the DVB family and present our DVB-S implementation. Interoperability validation is described followed by benchmarking performance tests.

2. BACKGROUND

A literature search indicates several SDR implementations of DVB-related waveforms [2-4]. Most of the papers focus on DVB-Terrestrial (DVB-T). DVB-T is like DVB-S with similar data framing and error correction. Modulation is more complex with orthogonal frequency division multiplexing (OFDM) in DVB-T compared to QPSK in DVB-S.

Our implementation shares several similarities with [2] including 100% software implementation, leveraging multiple threads to optimize performance, and use of SIMD code to parallelize some functions. A key difference is in the multi-threading model. We are breaking up the processing into separate threads that pass data between each thread. They are using a thread pool with a main thread directing the individual threads to awaken when there is input available to process. In our approach, there is no main thread. Instead, the inter-thread buffers manage blocking/unblocking thread execution when an input/output buffer is available. Our approach should simplify waveform development as there is no need to set up the main thread and implement the logic for determining when data is ready for the threads. This logic is effectively implemented locally by the inter-thread buffers. The thread-management code is contained in a library that never needs to be modified by the waveform developer.

Their use of an older processor in [2] most likely limits SIMD to SSE while our more modern processor enables Advanced Vector Extensions (AVX). They are using multi-threading functions. We have only employed this technique for more complicated waveforms and found it unnecessary for DVB-S. Finally, we implemented a complete real-time receiver, while they created an offline receiver.

3. IMPLEMENTATION

The transceiver is based on DVB-S standard *EN 300 421 V1.1.2 (1997-08).* The transmitter block diagram is shown in Figure 1. A video file is encoded by VLC producing MPEG2 transport stream frames. Null frames are inserted in this stream as needed to adapt the bit rate of the video stream to the bit rate of the DVB-S transmitter. The randomizer operates on 8 MPEG2 frames at a time. The first byte in an MPEG2 frame is a synchronization marker. These synchronization markers are used directly by DVB-S for its own synchronization purposes. The synchronization byte in the first MPEG2 frame of each randomizer frame is inverted to indicate the start of the randomizer frame.

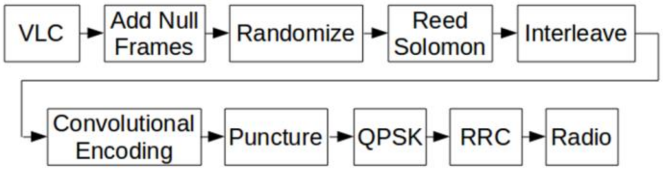


Figure 1. DVB-S Transmitter Block Diagram

Each 188-byte MPEG2 frame is Reed-Solomon encoded producing 204-byte frames. Frames are interleaved using a convolutional interleaver. The byte stream output from the interleaver is converted to a bit stream, most significant bit (MSB) first, and convolutionally encoded. The output of the convolutional encoder is punctured by the selected rate, either 1/2, 2/3, 4/5, 5/6, or 7/8. The encoded bit streams are QPSK modulated, RRC filtered, and transmitted by the radio.

The DVB-S receiver is shown in Figure 2. The system first removes DC offset introduced by the transmitter and receiver. This is performed twice, before the carrier synchronization to remove the receiver induced offset, and again after carrier synchronization to remove the transmitter induced component. The process is performed on blocks of 4096 consecutive samples. The average is calculated and then subtracted from each sample.

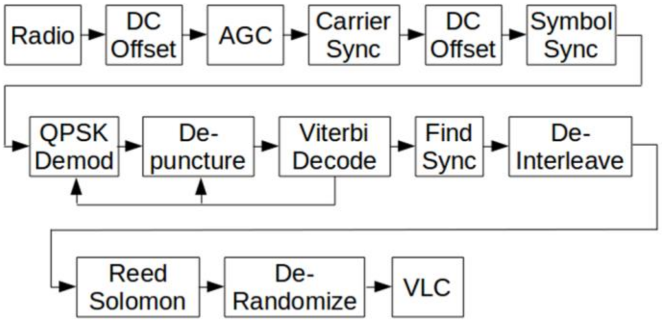


Figure 2. DVB-S Receiver Block Diagram

Automatic gain control (AGC) is similarly performed on blocks of 4096 samples to normalize the amplitude of the received signal. This is required to prevent a variation in carrier synchronization and symbol synchronization response time with the received signal level. Each sample is divided by the root mean squared (RMS) amplitude of the entire 4096 sample block.

Next, a phase-locked loop (PLL) removes the phase/frequency offset from the signal. The phase/frequency error is calculated by taking the 4th power of each input sample. This produces a strong impulse at 4x the frequency offset. This step initially showed high computation cost. Originally, this PLL was calculating the phase error and updating the feedback loop on every sample coming in. This resulted in an update rate of 25 million samples/second. Since the frequency offset is minor compared to the sample rate, it is not necessary to update the PLL feedback loop for every sample. The PLL was redesigned to calculate the phase error on a block of samples and then update the feedback loop for each block. The block size is currently set to 8 samples.

The same technique was applied to the symbol timing PLL. In this case, the maximum sample rate offset is smaller than the maximum frequency offset. Therefore, the block size can be made larger resulting in a greater reduction in computations. The block size is currently set to 1024 samples.

A similar approach is used for symbol synchronization. By squaring each sample, a strong frequency component is identified at the symbol rate. A PLL locks onto this frequency and optimal symbol timing is determined based on the phase of the PLL numerically controlled oscillator (NCO). A Root-Raised-Cosine (RRC) filter is again used to interpolate at the optimal time. Here, the RRC filter dot product operation is implemented in assembly code.

At this point, the symbols are QPSK demodulated producing two bit streams. These bits have the values +1/-1. The two bit streams are de-punctured which inserts 0 wherever a bit was punctured. The de-punctured bit streams are Viterbi decoded producing a single output bit stream. Since a QPSK constellation is symmetric over a 90-degree rotation, there may be a 90-degree phase ambiguity in the received symbols. To correct for this, the constellation is rotated periodically until synchronization is detected by the Viterbi decoder. Also, the de-puncturing must be correctly aligned with the incoming bit streams. To obtain the proper alignment, the de-puncturing is periodically shifted with respect to the incoming bit stream until the Viterbi decoder detects synchronization. The output bit stream from the Viterbi decoder is searched to locate the MPEG2 sync bytes. Once found, the bit stream is converted to bytes and passed into the de-interleaver. De-interleaved code-words are decoded by the Reed-Solomon decoder and de-randomized. The de-randomized code-words are sent to VLC to display the video.

4. LIMITATIONS

This DVB-S implementation has several limitations related to synchronization stability. Synchronization loss occurs intermittently with data transfers exceeding approximately 900,000 MPEG2 frames. When synchronization loss occurs, frames are dropped and bit errors are incurred in some received frames as synchronization stabilizes. Furthermore, these low-cost SDR platforms have limited hardware automatic gain control (AGC) functionality. To compensate for the limited AGC, calibration of transmitter and receiver gain is consequently sensitive and requires frequent tuning.

5. TEST PLATFORM

We have tested the DVB-S transceiver on software defined radios in cabled RF and over-the-air (OTA) wireless configurations. To demonstrate interoperability, we transmitted a video file to a commercial off the shelf (COTS) Coolsat receiver connected to a television. Table 1 lists configuration parameters necessary to recreate the system model.

Note that the bladeRF [5] is operating at 1GHz which is the intermediate frequency (IF) of the Satellite TV receiver. A label on the Coolsat receiver states that 950-2150MHz is the IF range. In a real system, there would be an upconverter to the KU band (to 11.7 to 12.7GHZ) at the transmitter. At the receiver (Coolsat), there would be a Ku downconverter. In this case we are only operating at 1GHz which is directly received by the Coolsat.

The low noise block (LNB) downconverter input also has 18 volts DC component. This DC voltage powers the downconverter when it is used. It is important to disable this DC voltage or use a DC blocking capacitor when directly connect this to the bladeRF SDR. At one point, we directly connected an attenuator to the LNB input and then connected it to the SDR which led to the attenuator getting hot to the touch. There is an option to disable the 18V DC which would allow direct connection.

The RF out of the satellite receiver connects directly to an old television. The LNB Input port of the receiver is connected to a coaxial cable. Soldering was required to connect the coaxial cable's center wire and ground to the antenna connector. We have found that the receiver sensitivity is good enough that the system receives video without the antenna on the Coolsat receiver.

Table 1. Components Used in Demonstration System

Item	Description
DVBS receiver	Coolsat 5000 Platinum
DVBS specification	EN 300 421 V1.1.2 (1997-08)
SDR platforms	Tested with BladeRF [5] and USRP B205MINI [6]
Linux distribution	Ubuntu 14.04.3
Linux kernel version	3.13.0
Software dependencies	• VLC 2.1.6-0-gea01d28 • libbladerf • libuhd
Antenna	OmniLOG 70600 Antenna
Receiver gain	Between 20dB and 40dB
Transmitter gain	Between 20dB and 35dB
Samples/symbol	2.25M
Sample rate	33.75M
Frame size	188 Bytes

6. RESULTS

Benchmarking tests included CPU usage, bit error rate, and latency. The Linux utility, *htop*, was used to monitor CPU usage of the transmitter software alone, the receiver alone, and both the transmitter and receiver running simultaneously. CPU usage was measured on an Intel Core i7 and i3, as indicated in Table 2.

Table 2. *htop* DVBS CPU Usage Results

Laptop	Features	HTop Reported CPU Usage out of 100%		
		Tx/Rx	Tx	Rx
Dell Precision M2800	Intel® Core™ i7-4610M CPU@3GHz 8GB RAM, 4CPUs	22%	8%	16%
Lenovo L530	Intel® Core™ i3-2348M CPU@2.3GHz 4GB RAM, 4CPUs	50%	24%	35%

Note that for quad-core processors, *htop* typically reports results out of a maximum of 400% based on the utilization of four cores. Here, results are normalized to 100% maximum. On average, the overall CPU usage for the entire system is relatively low. On a Dell Precision M2800 with an Intel i7 processor with 4 CPUs, the lowest usage percentage we achieved was 22% for the transceiver, 8% for the transmitter only, and 14% for the receiver only. Additionally, the RAM usage on the same platform remained constant at 0.2%, which is a total usage of 16MB out of the remaining 8GB.

Processor usage increases with symbol rate as shown in Figure 3. Normalized CPU usage increases as the symbol rate is increased from 5.5 to 15.5 symbols. Note that the transmitter usage is shown in blue and the receiver usage is shown in orange.

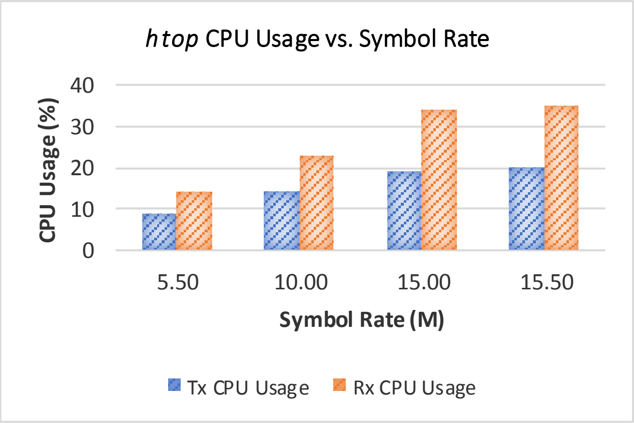


Figure 3. DVB-S *Htop* CPU Usage vs Symbol Rate

BER tests were conducted by measuring bits in error from fully synchronized MPEG2 frames at the receiver. Mock MPEG frames 188 bytes in length were transmitted in 2ms bursts with 10 frames/burst over a wired and wireless link. Under stable synchronization, 13 repetitions of 90,000 bursts (900,000 total frames) were received with no measured BER. In tests where synchronization was unstable, measured BER was on the order of 4.5E-5. Commercial DVB-S strives for BER on the order of 1E-10. To statistically measure levels that low, approximately 1E12 bits needs to be transmitted continuously. We were unable to transmit that much data at one time and maintain synchronization.

System latency is defined as the time between generating a MPEG2 frame and decoding a received MPEG2 frame, as shown in Figure 4. Table 3 shows the mean latency in the DVB-S transceiver in a wired environment (coaxial cable connecting the Tx and Rx RF ports) with increasing test burst sizes and a constant period.

Table 3. DVBS Transceiver Latency (Wired)

Bursts	Period	Mean Latency (µs)	Standard Deviation (µs)
100	100	49172	284
1000		49117	196
10000		49242	188

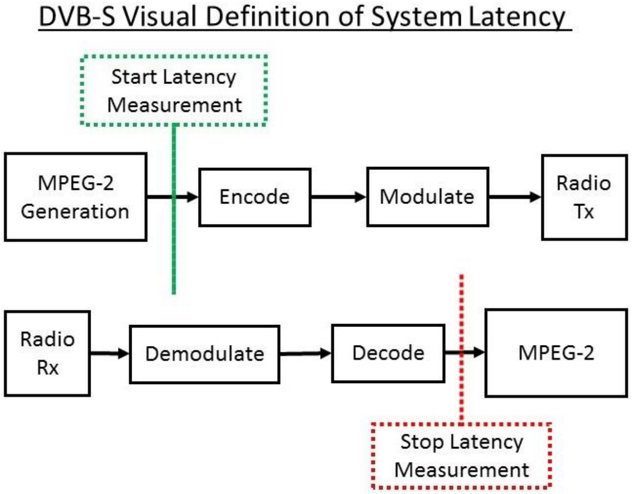


Figure 4. Latency Measurement Endpoints

The mean latency of our system hovers around 50 milliseconds despite the increase in the number of bursts that are sent. Although increasing burst size has no effect on the mean latency, it does reduce standard deviation of reported latency as more are sent. We further compare the mean latency of the DVB-S system to the set symbol rate shown in Figure 5 below.

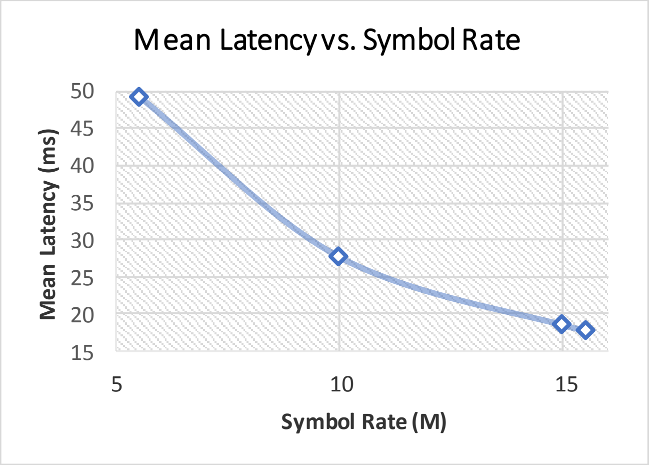


Figure 5. DVB-S Latency vs. Symbol Rate

The set symbol rate directly effects the mean latency of the system. As symbol rate is increased from 5.5 to 15.5 Msymbols, the mean latency decreases from 50ms to approximately 17ms.

7. CONCLUSION

We presented software implementation of a DVB-S transmitter and receiver where all I/Q processing is performed in GPP. Key elements include taking advantage of multiple processors through efficient threading and optimal usage of AVX instructions. Our agile approach to waveform development increases productivity and lends itself to faster timelines and real-time troubleshooting due to the flexibility of an all software implementation. Our approach has been successfully applied across multiple commercial and military waveforms.

We successfully showed that real-time operation of both transmitter and receiver I/Q digital processing is possible on an i3 or better processor. Benchmark results indicate peak normalized CPU usage at 50% for second generation i3 mobile processor, and 22% for a fourth generation i7 mobile processor. Bit error performance was comparable to the DVB-S waveform specification requirements when stable synchronization is achieved, with system latencies measured between 50ms and 17ms depending on the symbol rate.

8. ACKNOWLEDGEMENT

Approved for Public Release; Distribution Unlimited: 88ABW-2017-4648, 22 Sep 2017.This material is based upon work supported by Air Force Research Laboratory (AFRL/RITE) under Contract No. FA8750-15-C-0250. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author (s) and do not necessarily reflect the views of AFRL.

9. REFERENCES

[1] DVB-S Specification, EN 300 421 V1.1.2 (1997-08), European Telecommunications Standards Institute, www.etsi.org/deliver/etsi_en/300400_300499/300421/01.01.02_60/en_300421v010102p.pdf

[2] G. Baruffa, L. Rugini and P. Banelli, "Design and Validation of a Software Defined Radio Testbed for DVB-T Transmissions," Radioengineering, vol. 23, pp. 387-398, 2014.

[3] Y. Jiang, W. Xu and C. Grassman, "Implementing a DVB-T/H Receiver on a Software-Defined Radio Platform," International Journal of Digital Multimedia Broadcasting, vol. 009, p. 7, 2009.

[4] C. Fantozzi, L. Vangelista, D. Vorig and O. Campana, "SDR Implementation of a DVB-T2 transmitter: The core building blocks," IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, 2011.

[5] bladeRF Software Defined Radio (SDR), www.nuand.com

[6] USRP B205mini, Ettus Research, A National Instruments Company, www.ettus.com/product/details/USRP-B205mini-i

Improved Physical Layer Implementation of a MIL-STD-188 CPM Modem

fred harris (fred.harris@sdsu.edu)[1], Richard Bell (richbell@spawar.navy.mil)[2],
[1]San Diego State University, San Diego, CA, USA
[2]Space and Naval Warfare Systems Center Pacific (SSC Pacific)

ABSTRACT

Continuous Phase Modulation (CPM) maintains a constant amplitude and exhibits reasonable spectral confinement. Modern implementations perform direct phase modulation of Direct Digital Synthesizers. The direct phase modulation is understood to be a non-linear modulation process [1]. In spite of this fact most receiver structures demodulate the CPM signal by treating it as an O-QPSK signal. For small phase modulation indices, the errors are small but are not zero. We examine and present CPM receiver structures that perform phase demodulation with time reversed and conjugate phase matched filter aligned with the phase profiles of the modulation processes. We also examine innovative processing techniques that extract Doppler offsets, and perform carrier phase recovery and symbol timing recovery from traditional structured preambles.

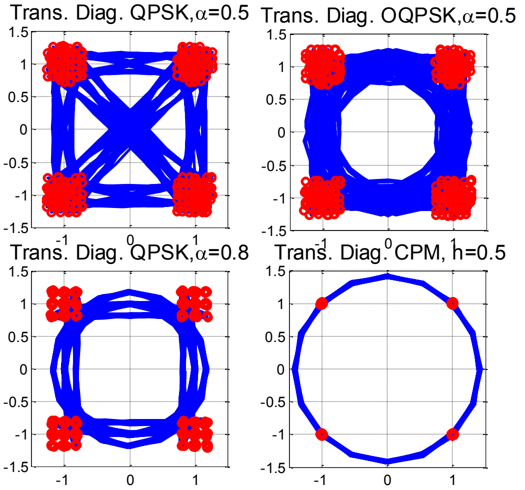


Fig. 1 Constellations and State Transition Profiles for SQRT Nyquist filtered QPSK α=0.5, O-QPSK for α=0.5, and O-QPSK for α=0.8, and CPM Half Sinewave Shaping Filter with Mod. Index h=0.25.

I. INTRODUCTION

The digital modulation process alters selected parameters of a sinewave in response to states of a finite state machine. We assign values selected from a limited set of amplitudes, frequencies or phases of a sinewave carrier to form an amplitude shift keying (ASK), a frequency shift keying (FSK) and a phase shift keying (PSK) modulation signal. Specifically, we examine 4-PSK (QPSK, quadrature phase shift keying) and identify 4 equally spaced constellation points on a circle corresponding to the 4 phases. Figure 1 shows the 4-constellation points for a QPSK, two O-QPSK, and a CPM modulator. The signal levels have been scaled for the constellation points to reside on the corners of the unit square which scales the circumscribed circle radius to sqrt(2). The phases can be identified by their polar coordinates as well as by their Cartesian coordinates. Traditionally receivers are designed to determine the Cartesian coordinates of successive symbols by processing and extracting their in-phase and quadrature components. In ordinary QPSK modulation, transitions between constellation points are not specifically controlled but are affected by shaping filters which restrict the modulation bandwidth of the process. The unrestricted transitions result in incidental amplitude modulation. There are a number of reasons to restrict or to control the state trajectories transitioning between states. A common desired attribute avoids trajectories that go through the origin, (to avoid zero amplitude envelope) while another avoids trajectories between repeated states (to ensure state transitions for the synchronizers). One way to avoid trajectories through the origin is to offset the in-phase and quadrature signal components so they cannot simultaneously change states. This modification, called Offset-QPSK (O-QPSK), results in state trajectories that do not transition through the origin but also exhibit reduced amplitude variations. Receivers designed to demodulate O-QPSK are variations of the QPSK modems that accommodate the time offset between the In-Phase and Quadrature signal paths [2]. There is a reason to further control the state transitions between states so that the trajectories between states do not leave the unit circle. We can incorporate this constraint by selecting the shaping filters for the In-Phase and Quadrature components of the O-QPSK to be half sine-waves. An equivalent option is to phase modulate the signal with linear phase profiles in each symbol interval. The constant amplitude envelope permits the final stage power amplifiers in the transmitter to operate at the edge of saturation and thus exhibit maximum power efficiency while preserving signal and spectral fidelity by avoiding non-linear distortion of the output amplifier stage. The half cycle offset sinewave shaping

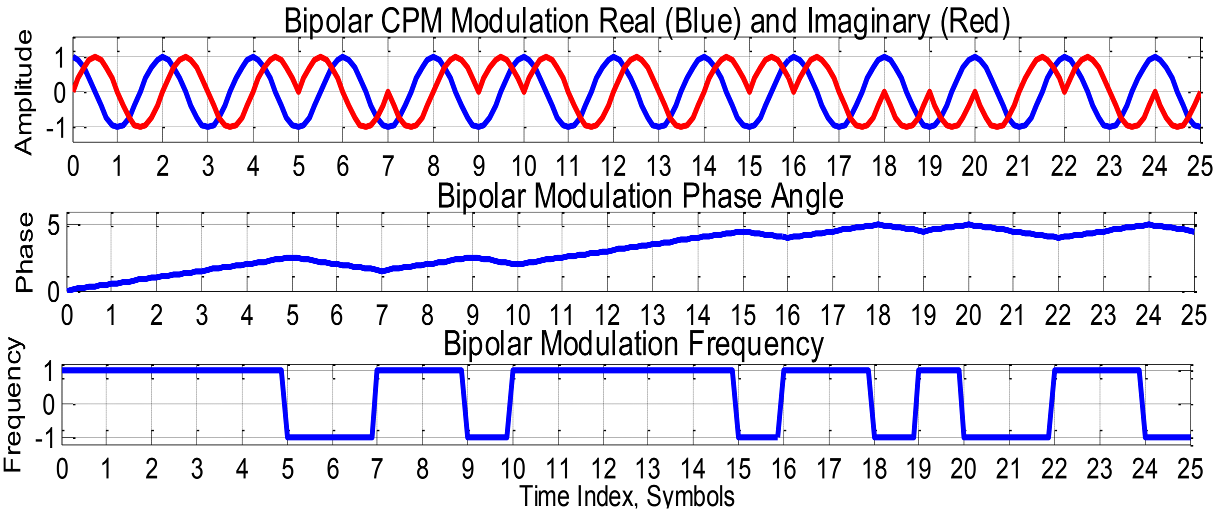


Fig. 2. In-Phase and Quadrature Components of CPM O-QPSK Signal Along with Its Phase and Frequency Profiles.

filter forms a signal with phase continuity across symbol boundaries. The class of signal so formed are described as CPM modulation. An example of the I-Q components of a binary CPM signal is shown in Fig. 2 which also shows its baseband phase and frequency profiles.

Because the CPM signal arrived at its final configuration by transitioning from QPSK through the O-QPSK process we persist in thinking of the CPM signal as an O-QPSK signal. In a true CPM modem the phase is directly modulated and the in-phase and quadrature components of the modulator are correlated. The O-QPSK receiver does not use this correlation while a true phase modulator and demodulator would. We might ask "Why is this a problem?" The answer is, other CPM signals formed by direct modulation of the phase function, such as is done in GSM, the Gaussian filtered Minimum phase Shift Keying (GMSK), is not optimally demodulated by an O-QPSK receiver [3]. In a phase modulated system phase noise and amplitude noise are orthogonal. In an O-QPSK modem, the in-phase and quadrature phase are projected on the two quadrature components and the receiver is unaware that the two signal components are correlated.

Since phase modulation requires a phase reference we find that phase progression due to a frequency error has a severe effect on the demodulation process. Frequency offsets due to Doppler shifts must be removed for successful demodulation. This is true in both the O-QPSK and the true phase demodulator. In the modulation forms described in MIL-STD 188 preambles are used to resolve Doppler frequency offsets and timing clock alignment prior to payload processing. The standard proposes a discrete Fourier transform based technique to resolve Doppler offsets that the standard suggests to be required when the offset is greater than the modulation bandwidth. The standard also references papers supporting the suggested technique [4].

In this paper we present a new technique to extract the Doppler offset information from the preamble which differs significantly from the technique proposed in the standard. We also introduce a demodulation scheme that performs phase matched filtering using the conjugate phase profile of the modulation process and introduce maximum likelihood timing recovery and frequency tracking methods to phase demodulate the CPM signal [5].

II. PREAMBLE PROCESSING

A common preamble for a number of CPM signals is a sequence of repeated phase shifts corresponding to 1 1 0 0 for phase shifts with modulation index 0.5. The waveform formed by this phase profile is one cycle of a complex sinewave corresponding to phasor rotation in the positive direction followed by one complex cycle of a complex sinewave corresponding to phasor rotation in the positive direction followed by one complex cycle of a sinewave with phasor rotation in the negative direction. Fig. 3 shows the Doppler free time series of the preamble along with the preamble distorted by a 9-kHz Doppler shift and then the spectrum of the Doppler shifted preamble. The suggested method of determining the Doppler offset is centered about examination of the preamble's Discrete Fourier Transform (DFT).

An alternate method of determining the Doppler offset frequency is illustrated in Fig. 4. We note that the tones in successive Doppler free preamble intervals are conjugates with frequencies $-\theta_P$ and $+\theta_P$ radians per sample respectively. We then see that when Doppler shifted, the tones in successive intervals are distorted and have new frequencies $+\theta_D-\theta_P$ and $+\theta_D+\theta_P$. The product of the distorted tones from the two intervals forms a third non-distorted tone with frequency $+2\theta_{D.}$

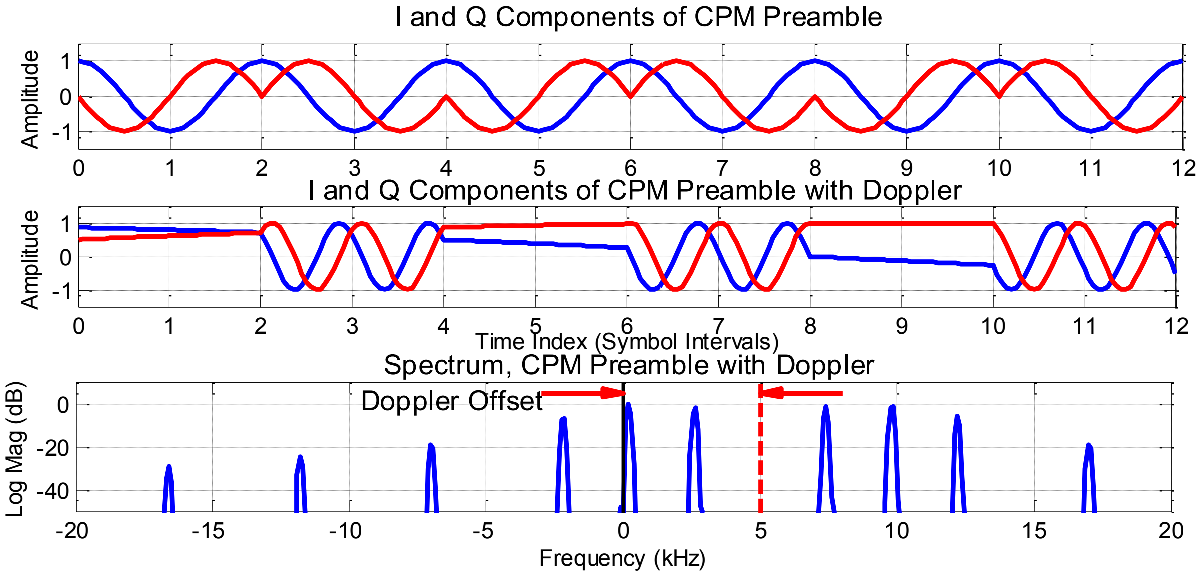


Figure 3. Preamble Pattern Multiple Repetitions of 1 1 0 0, (One Cycle Cos–j Sin & One Cycle (Cos + j Sin) With no Doppler, Same with 5 kHz of Doppler and Spectrum of Dopplered Periodic Preamble Signal.

$$
\begin{aligned}
s(\mathrm{n}) &= \mathrm{e}^{j\theta_{prmbl}n}\,\mathrm{e}^{j\theta_{dplr}n} \;:\; \mathrm{n}=0:39\\
s(\mathrm{n}+40) &= \mathrm{e}^{-j\theta_{prmbl}n}\,\mathrm{e}^{j\theta_{dplr}n} \;:\; \mathrm{n}=0:39\\
p(n) &= s(n)\cdot s(n+40) \qquad\qquad (1)\\
&= \mathrm{e}^{+j\theta_{prmbl}n}\,\mathrm{e}^{j\theta_{dplr}n}\,\mathrm{e}^{-j\theta_{prmbl}n}\,\mathrm{e}^{j\theta_{dplr}n}\\
&= \mathrm{e}^{\mathrm{j}2\theta_{dplr}n}
\end{aligned}
$$

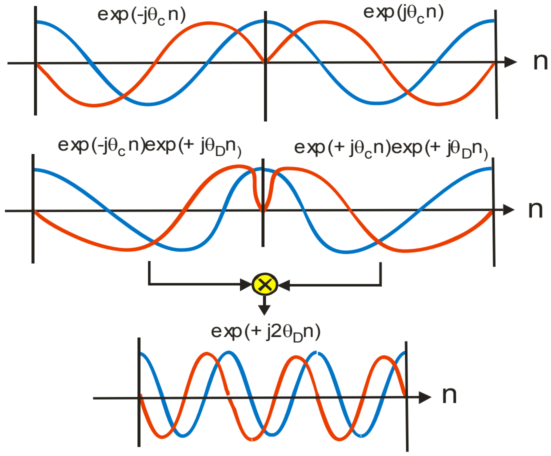


Figure 4. Top Subplot, Doppler Free 4-Symbols of Preamble, Center Subplot, and Same Symbols Doppler Distorted. Bottom Subplot Product of Successive Intervals with Canceled Preamble Modulation Components.

This relationship is shown in eq(1). Figure 5 illustrates formation of the double frequency Doppler tone $+2\theta_D$ formed by the product of the preamble adjacent delayed intervals as well the spectrum of the complex tone formed by that product. The frequency of Doppler tone is obtained from the ATAN of the average conjugate product of successive complex samples of this tone. The Doppler offset can be removed by a complex down conversion formed by a DDS set to the half frequency of the measured tone. This removal process is simple to implement and performs its task remarkably well.

In a similar fashion of delayed product, the conjugate delayed product spanning 4-symbol durations (80 samples) correlates the current sample values with the previous 80 offset samples. Since the data repeats in 4-symbol duration with or without the Doppler offset, the running average of this product forms the cross correlation, which when normalized by the delayed auto correlation forms a very effective squelch signal detector as well as the signal strength estimate to operate an AGC loop. Figure 6 shows the auto and cross product terms, the correlation running average of the two terms, and the normalized correlation ratio of the two averages which crosses the threshold and declares signal present within a few samples of signal start.

For completeness Figure 7 shows the spectrum of the CPM periodic preamble and for the modulated random data. In each figure we overlaid the spectrum of the half sinewave shaping filter and a box indicating the CPM symbol rate of 9.6 kHz. Note the preamble fundamental frequency is 2.4 kHz and the tone in the random modulation is at 4.8 kHz.

-

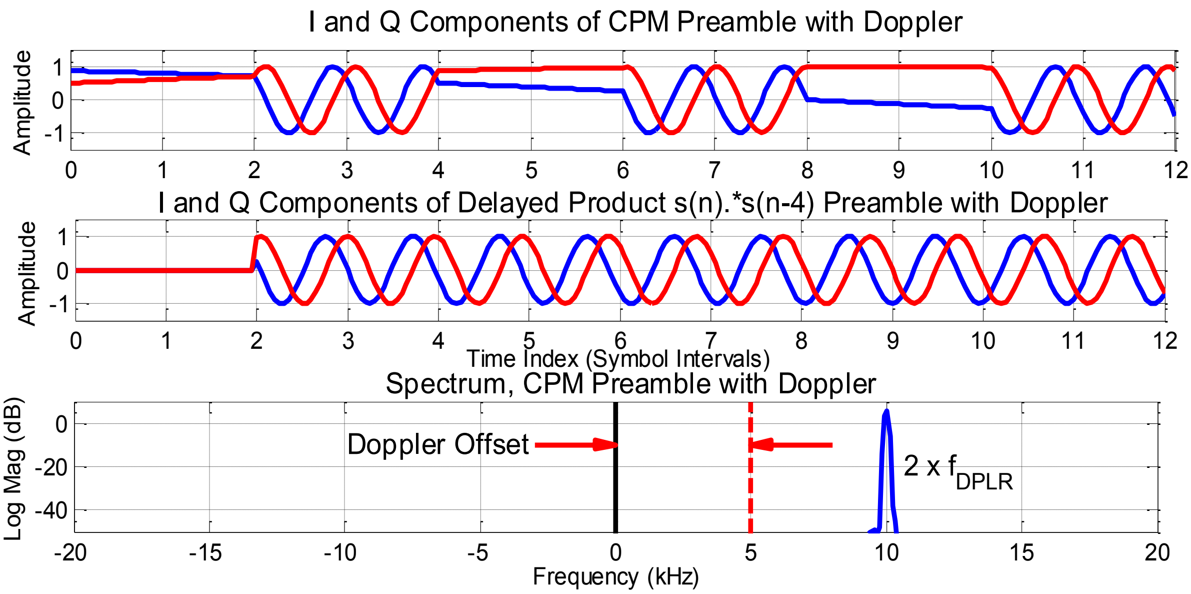


Figure 5. Top Subplot. Doppler Distorted Tones in Successive Preamble Intervals Center Subplot, Product of Successive Conjugate Symbol Intervals Forms Complex Tone at Twice Doppler Frequency. Third Subplot Spectrum of Doppler Tone Formed.

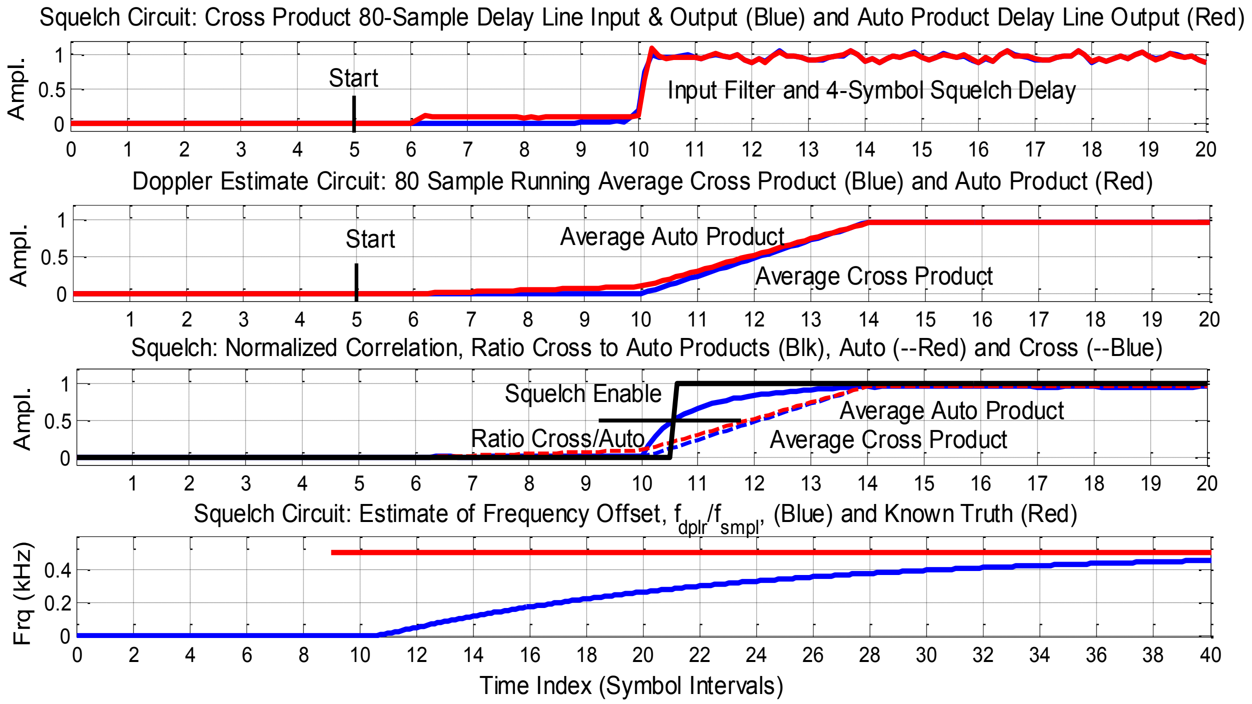


Figure 6. Squelch Network, Auto and 4-Symbol Delayed Conjugate Cross Product and Auto and Cross Correlation Normalized Correlation and Threshold Crossing, 2-Symbol Delayed Product, and 0.3 kHz Doppler Frequency Estimate.

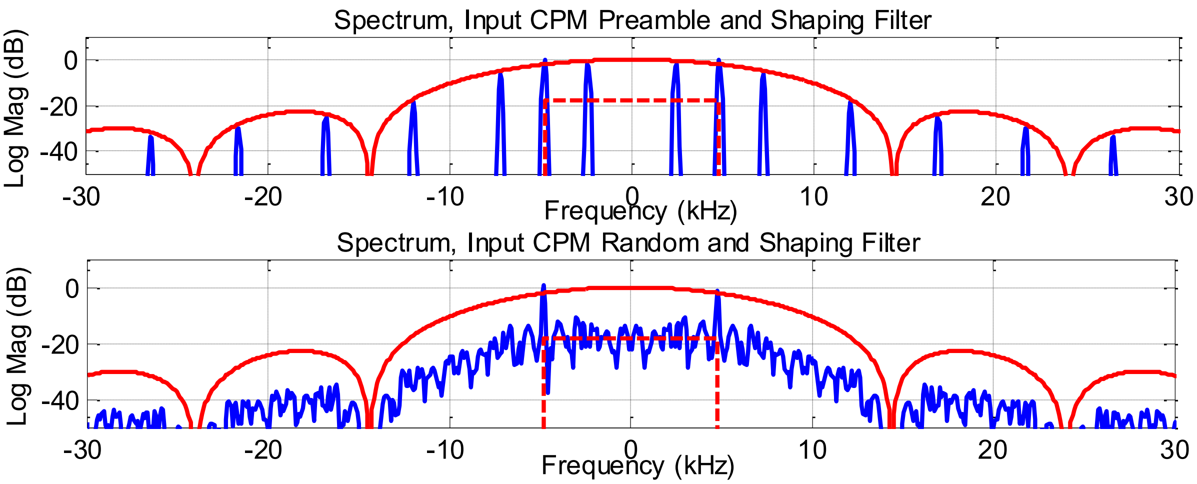


Figure 7. Upper Subplot: Spectra of Periodic Preamble and of Shaping Filter
Lower Subplot: Spectra of Random CPM Data and Shaping Filter.

III. OPTIMUM RECEIVER FOR CPM SIGNALS

The waveform used by the radio is the conventional continuous phase modulation formed by linear phase transitions around the unit circle in the positive and negative directions for logical 0‘s and 1’s respectively. Demodulation of this signal can be performed by an O QPSK down conversion as is done in most CPM receivers. Here we form matched filters for the I and Q components separately as the sum of products with the two weight sets cos(θ n) and –sin(θ n). The CPM phase matched filter forms the sum of products of the conjugate phase trajectory exp(j (θ n). Contrary to first impression, these are not the same. Figure 8 shows the I and Q components formed by the two processes. The O-QPSK peak I and Q components are orthogonal with magnitude 10 sqrt(2) while the CPM peak I component is 20. The noise component responses of the complex O-QPSK and real CPM signal are the same at 3.16.The output SNR of the two options are 13 dB and 16 dB respectively. Two effects are at play here; the real CPM matched filter forms x·c + y·s as opposed to x·c + j·y·s and the CPM filter also rejects half the noise of the O-QPSK process. An interesting property of the CPM process is that the imaginary part forms the derivative matched filter which the O-QPSK system must also form in two additional filters to support its timing recovery loop. We have observed the same property of other shaped CPM signal sets in the MIL-STD-188.

There are two phase profiles, CW and CCW, in the binary CPM signal set. Thus there must be two matched filters one for each profile. Both filters must operate in each symbol interval and their outputs must be examined. Each filter can have a positive or a negative local maxima and the filter with the largest output is selected as the proper filter for the spin direction for that interval and hence identifies the logic level carried by that symbol interval. The top subplot of fig. 9 shows the time signal at the start of the preamble. Here we removed the Doppler offset to clearly show the matched filter responses. The two center subplots show the matched filter responses to the preamble sequence. Here we see the 2 responses to the 1 1 sequence in MF_1 and then the 2 responses to the 0 0 sequence in MF_0. The blue and red curves are the real and imaginary parts of the filter responses. Note that at the extrema points of each blue curve, the red curve goes through its’ zero crossings, hence is (within a sign change) the derivative matched filter response.

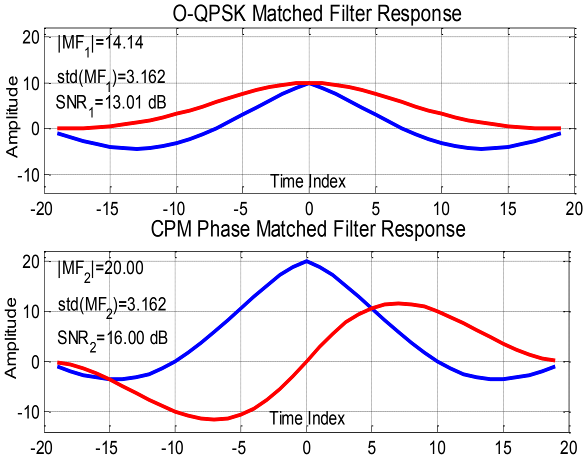


Figure 8 Matched Filter Response, OQPSK and CPM Process.

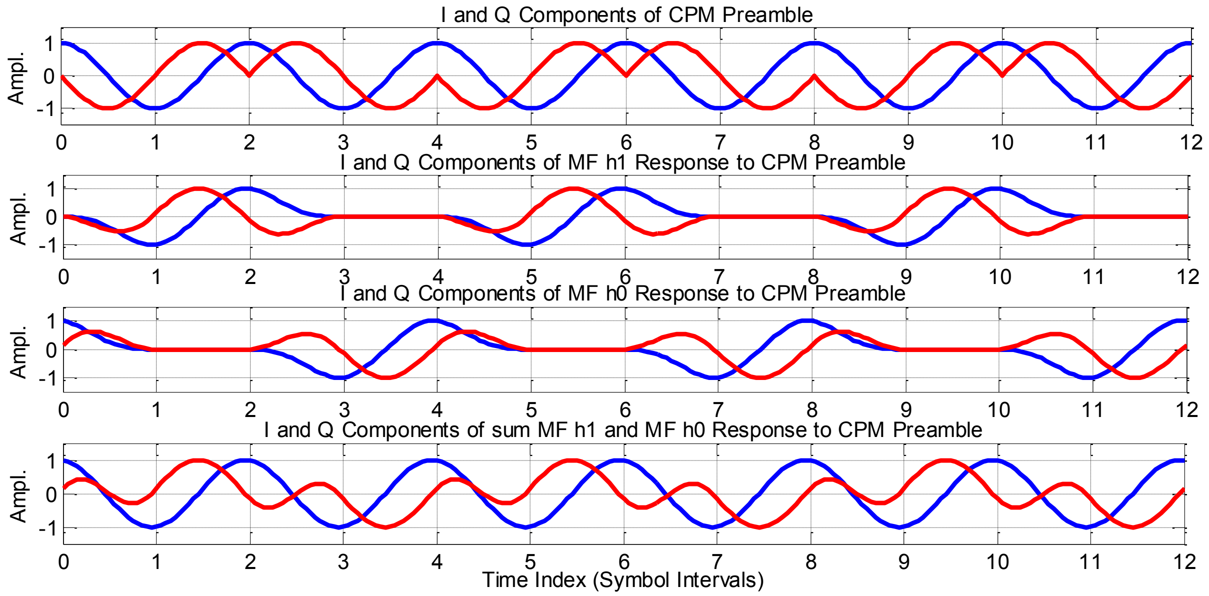


Figure 9. I & Q Input Signal and Two Matched Filter and the Sum of I & Q Responses of CPM Preamble Process.

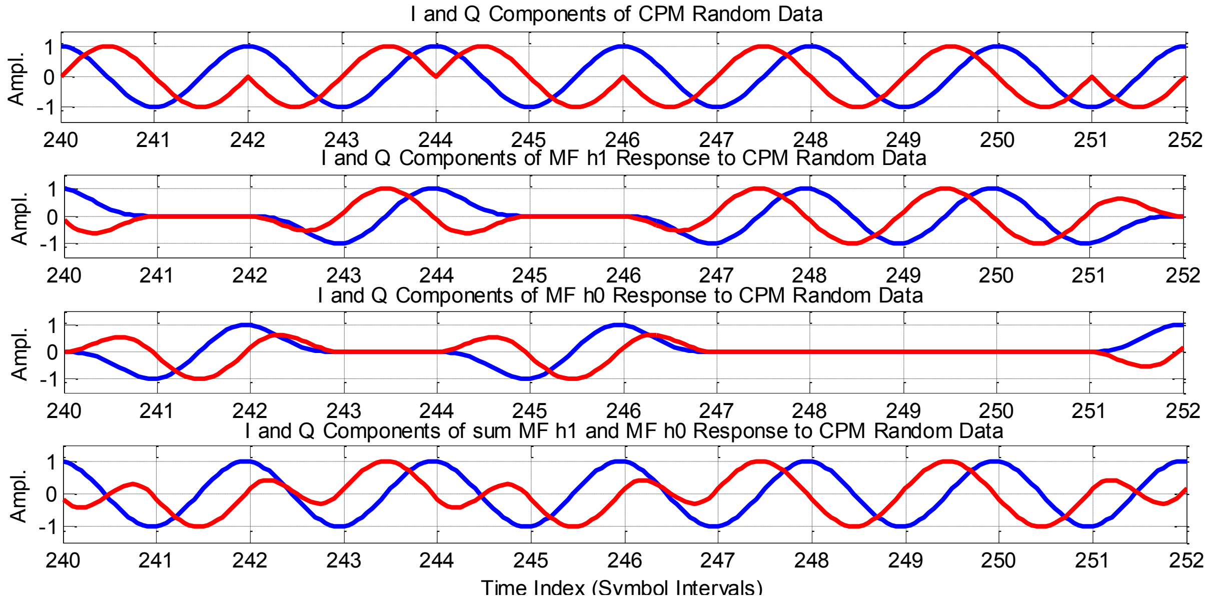


Figure 10. I & Q Input Signal, Two Matched Filters, and the Sum of I & Q Responses of CPM Random Data Process.

Note also in the two center subplots that at time tag 1 and 2 the matched filter h1 response peaks at -1 and +1 while the matched filter h0 response is 0 and 0. Then in the next interval time tags 3 and 4 the matched filter h1 response is 0 and 0 while the matched filter h0 response is -1 and +1. The responses of the 2 matched filters for CCW and CW rotation are seen to be orthogonal. This pattern repeats for the full preamble. What is remarkable is the sum of the matched filter response shown in the bottom subplot. The real part of the response is a cosine! Compare this response to the real part of the input signal in the top subplot. The detection process simply locates the location and value of the local extrema and then asks which of the two matched filter outputs is responsible for this extrema value?

The top subplot of fig. 10 shows the time signal at in the random data segment of the PCM time series. In practice, the Doppler offset had been removed by the Doppler processing the Doppler suppression in the preamble segment of the CPM signal. The two center subplots show the matched filter responses to the random data sequence. Here we see the responses to the 1 sequence in MF_1 and then the response to the 0 sequence in MF_0. The blue and red curves are the real and imaginary parts of the filter responses. We note that here too, at the extrema points of each blue curve, the red curve goes through its' zero crossings, hence is (within a sign change) the derivative matched filter response. Note also in the two center subplots that at time tag 241 and 242 the matched filter h0 response peaks at -1 and +1 while the matched filter h1 response is 0 and 0. Then in the next interval time tags 243 and 244 the matched filter h1 response is -1 and 1 while the matched filter h1 response is 0 and 0 The responses of the 2 matched filters for CCW and CW rotation are seen to be orthogonal for both the preamble and the random data sequences. What continues to be remarkable is the sum of the matched filter response shown in the bottom subplot. The real part of the response is a cosine! Again we should compare this response to the real part of the input signal in the top subplot. The detection process simply locates the location and value of the local extrema and then asks which of the two matched filter outputs is responsible for this extrema value? The sample values at symbol time of each filter can assume one of three values, +1, 0, and -1 and if one filter has value +1 or -1, the other will have value 0.Once the correct filter has been identified the y-y_{dot} product moves the sample index to the sample corresponding to the matched filter peak response. Once the peak sample position is identified, the phase angle of that index is used to control the phase PLL. Additional comment is presented in the following material.

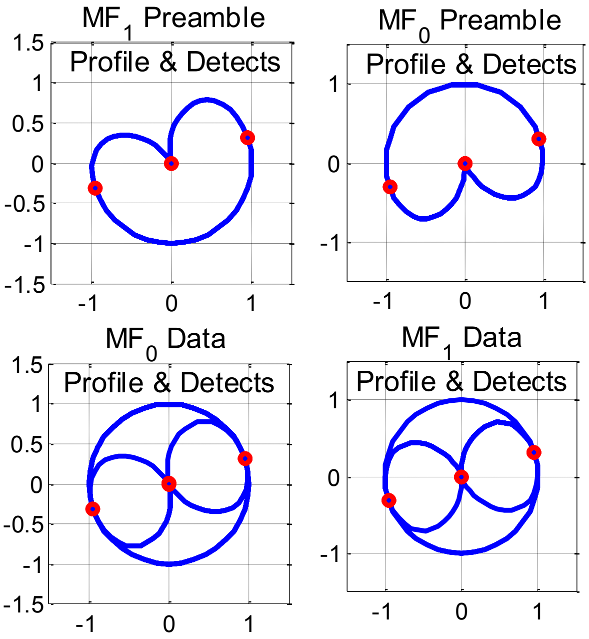


Figure 11. Phase Profiles and Detection Points for two Matched Filters in Preamble and Data Intervals.

Figure 11 shows the phase trajectories of the two matched filters along with the peak MF detection phase points on the profiles. Here we intentionally phase shifted the signal so that the detection points are off the horizontal axis. In this figure, the phase angle of the detection point supplies the angle errors for the phase PLL. The phase error seen here is typical of the residual phase after suppression of the frequency offset due to Doppler.

IV. CONCLUSIONS

We have presented in this paper a new approach to estimating and removing large Doppler frequency offsets for a Binary CPM Modulated signal. We then demonstrated the 3-dB SNR advantage of a phase profile matched filter based receiver compared to the more traditional O-QPSK approach to demodulating CPM signals. Finally we described and demonstrated some of the signal structures unique to the phase match matched filters with the filters offering detection information in their separate response while their sum provides the information for the timing recovery process.

V. ACKNOWLEDGEMENT

The Doppler estimation and removal process presented here was develop by the harris under a SPAWAR Contract to SDSU Foundation, contract No. N66001-15-D0099, "Tactical Secure Voice SDR Support (FY16)". SDSU is pursuing a patent based on the Doppler estimation process. The design and implementation of the modem using the Doppler estimation and the phase profile matched filters was an extension of this work started in 2016 and completed by the authors while harris was a SPAWAR 2017 ONR Summer fellow.

REFERENCES

[1] Pierre E Laurent, "Exact and Approximate Construction of Digital Phase Modulators by Superposition of Amplitude Modulated Pulses (AMP)", IEEE Trans. On Comm. VOL. Com-34, NO 2, Feb. 1986, pp. 150-160.

[2] Marvin. K Simon, "Carrier Synchronization of Offset Quadrature Phase-Shift Keying", Jet Propulsions Laboratory, California Institute of Technology, Telecommunications and Mission Operations (TMO) Progress Report TMO PR 42-133 Jan-Mar, 1998.

[3] Arjun Ramamurthy, fredric j. harris, "An All-Digital Implementation of Constant Envelope-Bandwidth efficient GMSK Modem using Advanced Digital Signal Processing Techniques", Wireless Personal Communications, Jan. 2010, VOL. 52, Issue 1, pp 133-146.

[4] Mohamed K. Nezami, "Techniques for Acquiring and Tracking MIL-STD-181B Signals", MILCOM-2002 Proceedings, Anaheim CA, 25-Feb. 2003, pp. 224-231.

[5] Earl McCune, "Practical Digital Wireless Signals", Cambridge University Press. March 2010, ISBN 10-0521516307, Chapter 6, Phase Modulation, Section 6.1, Author's Dilemma.

LICENSED SHARED ACCESS EVOLUTION ENABLES LOCAL HIGH-QUALITY WIRELESS NETWORKS

Seppo Yrjölä (Nokia, Oulu, Finland; seppo.yrjola@nokia.com); Heikki Kokkinen
(Fairspectrum, Helsinki, Finland; heikki.kokkinen@ fairspectrum.com)

ABSTRACT

This paper discusses the regulatory and standardization status of the Licensed Shared Access (LSA), compares it with the US Citizens Broadband Radio Service (CBRS) concept, and reviews results from the ongoing feasibility study in the European Telecommunications Standards Institute on temporary spectrum access for local high-quality wireless networks. Based on comparative analysis, a new LSA evolution concept and functional architecture is proposed, and the early results of the world first LSAevo e2e validation are presented. Introduced concept and system architecture can be applied to 3.4-3.8 GHz band so that current individual fragmentation challenges to take the band into 5G use in the European member states can be solved, while ensuring that the communication of the incumbent users, Fixed Wireless Access (FWA), fixed links, and satellite earth stations do not experience any harmful interference. In the e2e field trial, local high-quality wireless network use case for an industrial automation micro-operator on 3.4-3.8 GHz band is validated.

1. INTRODUCTION

The rapid growth in the number of mobile and wireless communication systems' users with a large range of diverse services, applications and devices [1] will require significantly more spectrum and wider continuous bandwidth than currently available [2] despite advances in spectral efficiency and network densification. In order to meet additional spectrum demands, besides identifying more dedicated spectrum, the regulators have globally shown growing interest in novel regulatory approaches related to spectrum allocation, utilization, and management. The Radio Spectrum Policy Group (RSPG) of the European Commission (EC) have identified 700 MHz, 3.4-3.8 GHz and 26 GHz spectrum bands as pioneer bands for 5th Generation (5G) in Europe, recommend the band 3.4-3.8 GHz as the primary band for introduction of services in its strategic roadmap [3], and call for industrial user experiments for the digitization of industry in its 5G Action Plan [4]. Furthermore, the Office of Communications (Ofcom) statement defines the same spectrum bands for the first wave of 5G in the UK [5]. Groupe Speciale Mobile Association (GSMA) recommends at least one frequency band allocated to 5G from each of the following frequency ranges: sub GHz, 1-6 GHz, and above 6 GHz [6]. The Global mobile Supplier Association (GSA) recommends 3.3-4.2 GHz frequency range, the Third Generation Partnership Project (3GPP) is working on in 5G New Radio (5G-NR) channel arrangement [7], as the primary band in the spectrum below 6 GHz, for the global introduction of 5G [8]. There is, on the other hand, a great variation of the current 3.4-4.2 GHz spectrum use and authorization in the EU member states as well as globally. The incumbents include, e.g., FWA, satellite communications, and fixed links, with highly varying expire dates of their radio licenses. Moreover, some of the member states plan to clear and auction at least parts of the band with nationwide licenses, while others have already prepared to have regional licenses on the band. There are member states, which plan on having primary and secondary 5G allocations with over 10 years and less than 2-year license periods, respectively.

From the above examples, it is apparent that Europe should prepare for a diverse 5G spectrum use on the primary 3.4-3.8 GHz band. The key objective of the European Telecoms framework is to provide a pro-investment framework to support 5G development through new bands, new users and usages, and increased more flexible use of spectrum. Proposed European Electronic Communications Code (EECC) framework promotes shared use of the spectrum [9].

Based on profound spectrum sharing work in policy, standardization and research, two novel licensing based sharing models have recently emerged, the Licensed Shared Access (LSA) [10] from Europe and the Citizens Broadband Radio Service (CBRS) from the US [11]. The two-tiered LSA builds on scale and harmonization in traditional exclusive licensing based regulation & standardization and leverages existing asset and capability base of Mobile Network Operators (MNOs). The CBRS on the other hand, extends dynamics through an opportunistic third "License by the Rule" GAA layer, fine-grained census tract based spectrum allocation, and sensing. Furthermore, the more dynamic CBRS concept was found likely to promote competition and foster innovation in the forms of new enabling technologies, novel ecosystem roles, and Internet era platform based business model designs [12]. The European Telecommunications Standards Institute Reconfigurable

Radio Systems Technical Committee (ETSI RRS) initiated a feasibility study "*temporary spectrum access for local high-quality wireless networks*" [13] in 2017 to study LSA evolution towards 5G spectrum, localization of spectrum for novel 5G use cases, and to enable horizontal sharing and sub-licensing for efficient use of the spectrum assets.

For these prominent spectrum sharing concepts currently under final stages of standardization and field trialing, there is not much prior work available in the field of comparative architecture analysis and common evolutionary scenarios. In the METIS II project [14] spectrum-sharing ecosystem evolution analysis was extended towards 5G, emphasizing potential changes in the roles, positions, and relationships of the key stakeholders in service delivery [14]. The Coherent project, stemming from the METIS, proposes a novel three-plane architecture which utilizes the available network graphs for spectrum usage and consists of spectrum management plane (spectrum management application), infrastructure plane (or equivalently data plane), and a central coordination and control plane [15]. LSA evolution towards dynamic modes of operation utilizing dynamic channel configuration through sensing and dynamic resource allocation algorithms was presented in [16]. The local high-quality wireless micro-operator network concept was introduced in [17]. To the best of authors' knowledge, the LSA evolutionary architecture concept and related e2e field trial validation has not been presented elsewhere. This paper seeks to answer the following research questions:

1) What are new requirements and amendments for the LSA spectrum sharing evolution to enable local high-quality wireless micro-operator networks?
2) What are the needed revisions in architecture and technology?
3) How could this be of help for key stakeholders and regulators in implementing LSA evolution?

The rest of the paper is organized as follows. First, the CBRS and the LSA sharing concepts are defined and their comparative analysis presented in section 2. Second, requirements for the LSA evolution are discussed, LSA evolution architecture concept proposed, and its validation presented. Finally, conclusions are drawn.

2. COMPARATIVE ANALYSIS OF THE SPECTRUM SHARING FRAMEWORKS

This section will introduce the CBRS and the LSA spectrum sharing concept, and provide comparative analysis.

2.1. Citizens Broadband Radio Service (CBRS)

In the US, the PCAST report [18] suggested a dynamic spectrum sharing model as a new tool to the US wireless industry to meet the growing crisis in spectrum allocation, utilization and management in 2012. The key policy messages of the document were further strengthened in 2013 with Presidential Memorandum [19] stating "*…we must make available even more spectrum and create new avenues for wireless innovation. One means of doing so is by allowing and encouraging shared access to spectrum that is currently allocated exclusively for Federal use. Where technically and economically feasible, sharing can and should be used to enhance efficiency among all users and expedite commercial access to additional spectrum bands, subject to adequate interference protection for Federal users.*"

In Figure 1, the US three-tier authorization framework with the Federal Communications Commission's (FCC) spectrum access models for 3550-3650MHz and 3650-3700MHz spectrum segments are illustrated. While the general CBRS framework could be applied to any spectrum and between any systems, the current regulatory efforts in the FCC are focused on the 3550-3700 MHz band [20].

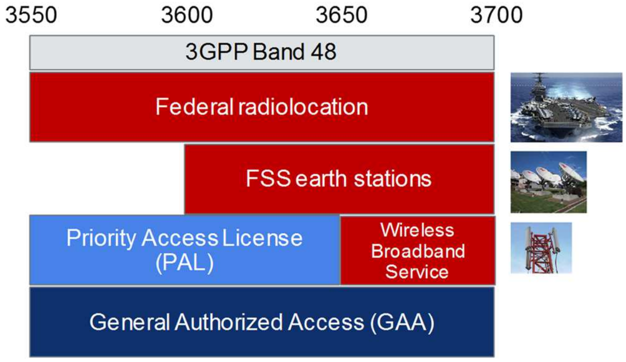

3550
3600
3650
3700
3GPP Band 48
Federal radiolocation
FSS earth stations
Priority Access License (PAL)
Wireless Broadband Service
General Authorized Access (GAA)

Figure 1. The US 3-tiered CBRS spectrum access model and band plan.

The standardization process for the CBRS is ongoing in the Wireless Innovation Forum (WinnForum) [21], and for the specific spectrum band in the 3GPP [22]. The three tiers depicted in Figure 1 are:
1) *Incumbent Access* (IA) layer consists of the existing primary operations including authorized federal users and Fixed Service Satellite (FSS) earth stations. The IA is protected from harmful interference from the CBRS users by geographic exclusion zones and interference management conducted by the dynamic *Spectrum Access System* (SAS),
2) *Priority Access* (PA) layer includes critical access users like hospitals, utilities, governmental users, and non-critical users, e.g., MNOs. PA users receive short-term priority authorization (currently, a three-year authorization is considered) to operate within designated geographic census tract with Priority Access Licenses (PALs) in 10 MHz unpaired channel. PALs will be awarded with competitive bidding, and with ability to aggregate multiple consecutive

PALs and census tracts to obtain multi-year rights and to cover larger areas. Any entity eligible to hold a FCC license could apply for a PAL and is protected from harmful interference from the General Authorized Access (GAA) layer.

3) *General Authorized Access* layer users, e.g., residential, business, and others, including Internet service providers are entitled to use the spectrum on opportunistic *license-by-rule* regulatory basis without interference protection. In addition to the defined 50% floor of GAA spectrum availability specified to ensure nationwide GAA access availability, GAA could access unused PA frequencies. GAA channels are dynamically assigned to users by an SAS. The addition of the third tier is intended to maximize spectrum utilization, and to extend usage from centralized managed Base Stations (BSs) to stand-alone GAA access points (CBSDs).

The SAS dynamically determines and assigns PAL channels and GAA frequencies at a given geographic location, controls the interference environment, and enforces exclusion zones to protect higher priority users as well as takes care of registration, authentication, and identification of user information [23]. In 2016, the FCC finalized rules for CBRS [20] and introduced the *light-touch leasing process* to make the spectrum use rights held by PALs available in secondary markets. Under the light-touch leasing rules, PA Licensees are free to lease any portion of their spectrum or license outside of their PAL protection area (PPA) without the need for the FCC oversight required for partitioning and disaggregation. This allows lessees of PALs to provide targeted services to geographic areas or quantities of spectrum without additional administrative burden. Coupled with the minimum availability of 80 MHz GAA spectrum in each license area, these rules will provide the increased flexibility to serve specific or targeted markets. Furthermore, the FCC will let market forces determine the role of an SAS, and as such, stand-alone exchanges or SAS-managed exchanges are permitted.

The *CBRS devices* (CBSDs) are fixed or portable base stations or access points, or networks of such, and can only operate under the authority and management of a centralized SAS, which could be multiple as shown in Figure 2. Both the PA and the GAA users are obligated to use only the FCC certified CBSDs, which must register with an SAS with information required by the rules, e.g., operator identifier, device identification and parameters, and location information. In a typical MNO deployment scenario, the CBSD is a managed network comprising of the *Domain Proxy* (DP) and NMS functionality. The DP may be a bidirectional information routing engine or a more intelligent mediation function enabling flexible self-control and interference optimizations in such a network. In addition to larger MNO-operated MBB networks, DP enables combining, e.g., the small cells of a shopping mall or sports venue to a virtual BS entity that covers the complete venue. The DP can also provide a translational capability to interface legacy radio equipment in the 3650–3700 MHz band with an SAS to ensure compliance with the FCC rules. A MNO could utilize a DP and/or operator-specific SAS in protecting commercially sensitive details of their network deployment data.

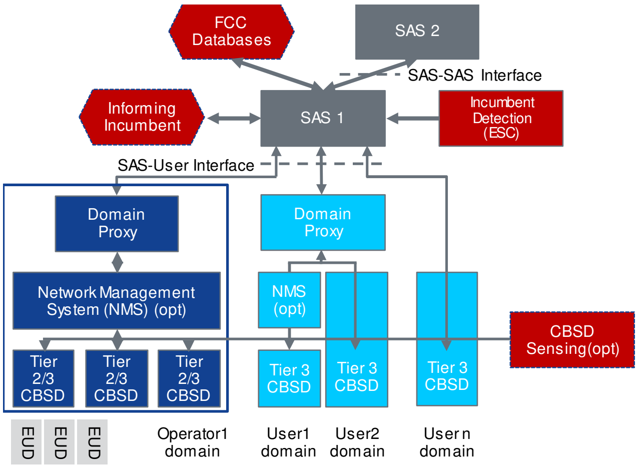


Figure 2. The US 3-tiered CBRS concept and functional architecture.

In the dialog between industries [11], the FCC and the main incumbent user, United States Department of Defense (DoD), it is assumed that in addition to informing database approach, there is a need to introduce a Non-Informing Approach, requiring Environmental Sensing Capability (ESC). The ESC architecture and implementation scenarios discussed include a dedicated sensing network for an SAS, collaborative sensing by commercial network BSs, or their combination. According to the FCC rules [20], the SAS must either confirm suspension of the CBSD's operation or its relocation within 300 seconds after the ESC detection communication, or other type of notification from the current federal user of the spectrum band.

The White House aims to expand wireless innovation in spectrum sharing further through identifying an additional 2 GHz of federal owned spectrum below 6 GHz for future commercial sharing [24]. The success of the CBRS is critical to future federal–commercial spectrum sharing. Moreover, the FCC has already proposed the use of the three-tier model and the SAS for 5G in several cmWave and mmWave bands.

The CBRS system has been validated in field trials in Finland and US. Architecture, implementation and field trial results are presented, e.g., in [25] and [26].

2.2. Licensed Shared Access (LSA)

The EC communication based on an industry initiative promoted spectrum sharing across wireless industry and diverse types of incumbents [27]. In 2013, the RSPG of the EC defined LSA as [28] "*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 LSA framework, the additional users are allowed 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 Quality of Service (QoS).*"

The recent development in policy, standardization and architecture has focused on applying the LSA to leverage scale and harmonization of the 3GPP ecosystem. This would enable MBB systems to gain shared access to additional harmonized spectrum assets not currently available on exclusive basis, particular the 3GPP band 40 (2.3-2.4 GHz) as defined by the European Conference of Postal and Telecommunications Administrations (CEPT) [29]. The European Telecommunications Standards Institute (ETSI) introduced related system reference, requirements, and architecture documents [30]-[32] from the standardization perspective. In the LSA concept, the incumbent spectrum user, such as a Program Making and Special Events (PMSE) video link, a telemetry system, or a fixed link operator, is able to share the spectrum assigned to it with one or several LSA licensee users according to a negotiated *sharing framework* (SF) and *sharing agreement* (SA). The *LSA License* (LL) model guarantees protection from harmful interference with predictable QoS for both the incumbent and the LSA licensee.

The LSA architecture consists of two new elements to protect the rights of the incumbent, and for managing dynamics of the LSA spectrum availability shown in Figure 3: the *LSA Repository* (LR) and the *LSA Controller* (LC). The LR supports the entry and storage of the information about the availability, protection requirements, and usage of spectrum together with operating terms and rules. The LC located in the LSA licensee's domain grants permissions within the mobile network to access the spectrum based on the spectrum resource availability information from the LR. The LC interacts with the licensee's mobile network to support the mapping of LSA resource availability information (LSRAI) into appropriate radio transmitter configurations via Operation, Administration and Management (OAM) tools, and to receive the respective confirmations from the network.

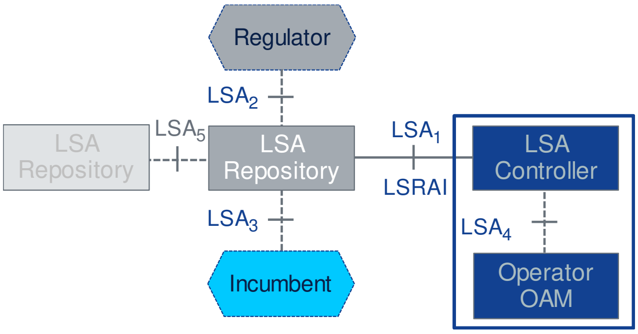


Figure 3. The LSA architecture reference model.

The LSA system for 2.3-2.4 GHz band has been validated in field trials in Finland, Italy and France. Architecture, implementation and field trial results are presented, e.g., in [33]-[36]. The second use case currently being considered in European regulation is the application of LSA to the 3.6-3.8 GHz band [37]. For this band, the incumbent usage is less dynamic, and the LSA band availability is guaranteed in the license area for a known period. This allows extension to more innovative use cases, such as local private networks using small cells, as there is no need for additional frequency resource or existing infrastructure to support dynamic handover.

2.3. Comparative analysis of LSA and CBRS

In this section, the CBRS and the LSA spectrum sharing concepts are summarized and compared. Comparative analysis of architecture, interface, functions, protection, and security is summarized in Table 1. The state diagrams for the CBRS and the LSA are depicted in Figures 4 and 5, respectively. The key difference is that the LSA doesn't allow dynamic spectrum grant and relinquishment for local usage. In the CBRS, the *Grant* is the authorization provided by an SAS to a CBSD, subject to a *Heartbeat* exchange, to transmit using specified operating parameters. Grants are identified by a unique Grant identifier. Once issued, a Grant's operating parameters are never changed; if new or modified operating parameters are required, then a new Grant must be obtained. The Grant's operating parameters are maximum Effective Isotropic Radiated Power (EIRP) and Channel. If the CBSD no longer needs access to the Grant prior to its expiration, the CBSD initiates the Grant *Relinquishment* procedure. A Grant can be in different states as depicted in the CBSD Grant State Diagram.

Table 1. Comparative analysis of the LSA and the CBRS spectrum sharing concepts.

	CBRS	LSA
Architecture	• SAS, DP, CBSD • Spectrum management decision entity in SAS supporting dynamic features • Spectrum management implementation entity as part of SAS (decision of CBSD operating parameters) • Flexible assignment of Controller function to domains: DP provides parts of the Controller function; other functions are part of SAS. • Multiple radio access support covered by CBRS on Incumbent side and CBSD side	• LR, LC, MFCN • Spectrum management decision entity in LR • Spectrum management implementation entity as part of LC and/or 3GPP OAM functionality. Enables use of efficient protection zone. • Flexible assignment of LC function to domains: In LSA phase 1 a fixed assignment to the LSA Licensee domain. • Multiple radio access support covered by LSA on Incumbent side and Licensee side: the LSA phase 1 foresees LTE as Licensee RAT.
Interfaces	• Defined enabling standardized interoperability. • SAS-SAS, SAS-CBSD • No direct access of Incumbents, ESC is used to detect Incumbent usage of spectrum. Informing incumbent as an option. • NRA access via SAS interface (SAS admin proprietary) • Spectrum user access via SAS-CBSD interface; DP proxy may handle multiple CBSDs	• High level requirements and frameworks only to date • LSA5, LSA1 • LSA3 covers Incumbent spectrum availability control, input of Sharing Agreement, and Reporting information • NRA access via named LSA2 interface (Proprietary as not defined in the standards) • Spectrum user access via LC using LSA1 (LC-LR interface) and optional LSA4 (OAM, LC- Mobile/Fixed Communications Networks (MFCN))
Function	• 3-tier sharing • Public, competitive suppliers of access control available to any user • Sharing Framework, PA License, GAA Registration at SAS • Sub-licensing of spectrum resources supported by the PPA concept allowing PAL users	• 2-tier sharing • Direct relationship between Incumbent and Licensee • Sharing Framework, Sharing Agreement and LSA License • Sub-licensing of spectrum resources currently not supported
Protection and exclusivity	• SAS introduces licensed like PAL spectrum resources and license exempt like "License by the Rule" GAA spectrum resources without guaranteed QoS. • Incumbent Protection via Rules by FCC; the protection is performed by SAS and translated in spectrum availability information, which is provided to the requesting PA or GAA user • Spectrum resource is shared between Incumbent, PA, and GAA users following the sharing rules of FCC, SAS may use additional rules to influence the spectrum resource assignment to a user to guarantee fairness. Multiple SAS operators in the same area allows CBSD operator to switch SAS operator • Finer granularity in geographic and temporal sharing condition, and broader scope of licenses enable enterprise/residential/small MNO deployments and third tier • SAS service is an advantage to unexperienced non-MNO operators	• LSA follows a licensing concept and provides QoS when spectrum is available for Licensees. • Incumbent protection via Sharing Framework and Sharing Agreement; both results in protection requirements, which are provided to the LSA Licensee • Spectrum resource is exclusive shared between Incumbent and a LSA Licensee; different LSA Licensees are protected by guards, which needs to be derived via the Sharing Framework and Sharing Agreement • Large blocks of nation-wide geographic long-term exclusivity favor wide-area MNOs
Security	• Comsec and Opsec	• Comsec and Opsec

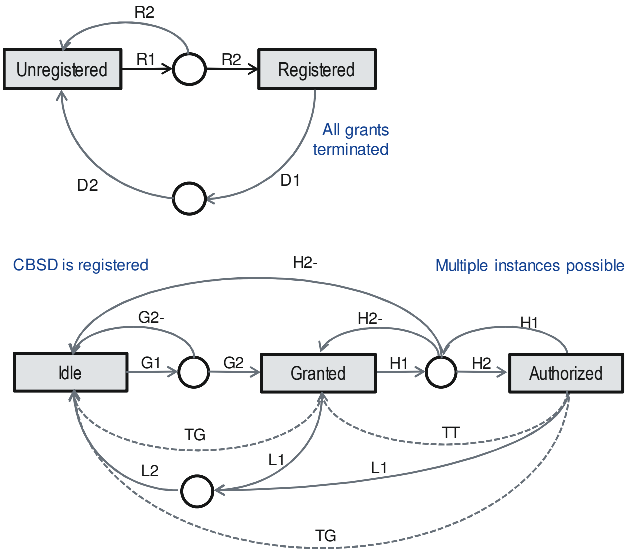


Figure 4. State diagrams for the CBRS Registration/Deregistration and Grant state.

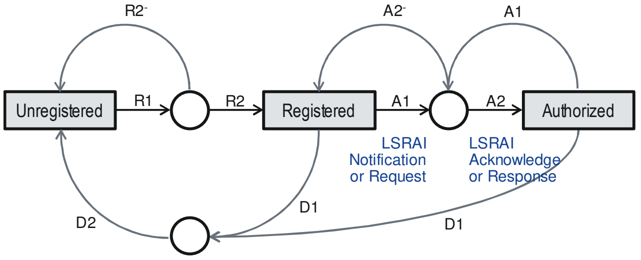


Figure 5. State diagram for the LSA.

3. LSA EVOLUTION

3.1. High level requirements

High level requirements for this LSA evolution concept study and validation were derived from the ongoing feasibility study in the ETSI RRS technical committee on temporary spectrum access for local high-quality wireless networks [13], and the research on micro-operator concept [17] and [38]. Additional evolution use cases under study includes: nomadic events, use under R&D licenses, and governmental use, e.g., local non-critical military use.

Micro-operator concept and Factory of the Future use case

The *micro-operator* concept was recently introduced for local service delivery in 5G to establish local small cell networks and offer context related services and content with guaranteed quality [17]. Deployment of local specialized networks requires changes to the current operational models and regulations to allow new agile players to deploy the RANs and deliver new services. Micro-operator use cases include deployments of ultra-dense specialized small cell RANs in distinct locations such as factories, campuses, malls, and sports arenas. With the help of network slicing and spectrum sharing techniques, the micro-operator can rapidly respond to local needs and provide high connectivity services [38]. A network slice can be tailored to support specific applications and services delivered over micro-operator's network where the RAN part of the slice is from the micro-operator and other parts can be from the owner of other physical network infrastructure.

A fundamental rethinking of the mobile network operations and management principles is needed to address novel requirements for diverse locations, services, use cases, and business models. The ability to identify and capture network resources and capabilities at a targeted geographical area can be combined with the ability to enable usage at the needed service level for the use case. The micro-operator ecosystem enables any party that would need telco grade wireless networking capacity in selected locations to build their own network solution or take it as-a-Service (aaS). Micro-operator concept can introduce a new control point in the digital value platform through the spectrum and network slice management in technology utilizing self-organizing network (SON), in management and orchestration (MANO), policy (brokerage) and business (aaS).

There will be numerous deployment scenarios. The micro-operator can install and manage itself a local wireless network and roaming contracts with MNO. Network infrastructure can be owned by the micro-operator or enterprise/vertical. Alternatively, in the Infrastructure as a Service (IaaS) model, a micro-operator offers multi-tenant infrastructure service with value added localized resource optimization to larger MNOs. A MNO provides applications, and manages and optimizes the service with components either instantiated in the micro-operator edge cloud, or in the MNOs cloud. In the Platform-as-a-Service (PaaS) or SW-as-a-Service (SaaS) model the micro-operator will operate and offer a hosted aaS functions or complete service components to MNOs. A MNO will manage e2e service leveraging micro-operator's function and services optimized and scaled for local dynamics in demand, resources and network status.

In *the Factory of the Future* (FoF) scenario, new business models can be built, e.g., around process-aaS, robot/machine-aaS, maintenance-aaS, or virtual network-aaS. Service vertical integration of networked factory allows for an optimized and more dynamic usage of resources, and calls linkage of manufacturing processes performed by multiple systems and providers inside the factory boundaries. This set stringent requirements for Service Level Agreement (SLA) processes, workflow integration, standardized interfaces, and networked services for security, trust and data analytics. Fundamentally, the aim of the FoF communication is to monitor and control real-world actions and conditions of the specific physical equipment. Industrial automation has a wide range of use cases with a unique set of communication requirements, particularly latency, reliability, availability, and throughput. These high-quality networks are typically geographically confined in area, serve heterogeneous professional applications requiring high predictable levels of service guarantee, and may require own network control and operation functions due to specific security standards and privacy requirements [39].

Temporary spectrum access for local high-quality wireless networks

ETSI RRS feasibility study on temporary spectrum access for local high-quality wireless networks [13], focus on spectrum sharing approaches offering new entrants (licensees) spectrum access rights, so that they are able to provide predictable levels of QoS to the end users on a local geographical area on short or longer-term basis. These spectrum access rights are described in the form of a sharing agreement that constitute the regulatory legal basis for ensuring a certain QoS level for all authorized users, including incumbents. Use cases prefer a private network deployment or hybrid with public network infrastructure and management, to implement needed security standards and privacy requirements. Initially, the LSA and the tier 2 of the CBRS sharing concepts are considered for the following scenarios:

Local high-quality wireless networks as private network areas scenario focus on vertical industries integration, and foresees the set-up and operation of private networks in a local and closed environment without the necessary direct involvement of a MNO. The Licensee may be, for instance MNO, Mobile Virtual Network Operator (MVNO), or a new vertical service provider entering the market owning access infrastructure and providing spectrum management services.

Local high-quality networks as in-standard service areas scenario considers an integration of local high-quality wireless networks into the MFCN ecosystem where the role of the Licensee is occupied by an MNO.

3.2. LSAevo concept and functional architecture

Based on the comparative analysis in Section 2.3 and requirements discussed in section 3.1, the following considerations and issues with current spectrum sharing concepts were found.

In general, the geographic scope of licenses should serve the needs of micro-targeted deployments as well as larger deployments, while guaranteeing the QoS of spectrum resources that may be impacted by dynamic spectrum sharing. Concept should enable neutral operator instances. The management of the spectrum resource at local level by the vertical should be guaranteed in case spectrum resource is provided by a MNO. Furthermore, security of sensitive network information of verticals should be guaranteed, e.g., SAS administrators should protect CBSD registration information.

LSA authorization process with regulator and incumbent is complex, lengthy, and vary from country to country. Incumbent interface is difficult to standardize generally, and possible only for a specific country and specific incumbent type. Furthermore, the interface between regulator and Spectrum Manager (SM) has been proprietary, exception being the Ofcom harmonized TV White Spaces (TVWS) in UK. Deployment durations are ranging from several hours to several years. LSA doesn't support of flexible grant and relinquishment procedures for LSA spectrum resource, neither support mutual renting. CBRS has interface for SM – Licensee/CBSD, while LSA1 interface is an internal in SM.

In the CBRS concept, GAA users have no interference protection. CBRS PAL License auction prefers MNOs and may lead to expensive PPA claims. Increasing the term for PALs with greater certainty will promote investment, and larger geographic scope of PAL will facilitate deployment.

Proposed LSAevo functional architecture

Proposed LSAevo functional architecture in Figure 6, builds on proven LSA benefits of leveraging scale and harmonization in regulation & standardization, and utilization of existing commercial assets and capabilities. Introduced new extensions to LSA architecture depicted in Figure 3, enables new frequency bands towards 5G, localization of spectrum with novel 5G use cases, e.g., for verticals, horizontal sharing & sub-licensing for efficient use of the spectrum assets, and as a recapitulation lowers entry barrier for new service providers through unbundling investments in spectrum, infrastructure and services. Identified initial features for the LSA evolution are:

- 2-tier sharing with deterministic and predictable channel arrangement to avoid complexity and to satisfy the stringent QoS requirements.
- Central management for spectrum and license handling CEPT has generic technical requirements for co-existence of different radio systems, they should be used as reference for SM protection.
- SM broker as additional operator type beside vertical and MNO (as SAS operator for CBRS)
- Hierarchical coexistence/interference management with possibility to negotiate and perform local adaptations at network and service level.
- Facilitates local network infrastructure, shared network infrastructure, e.g., MVNO, MNO or hybrid service.
- Re-use of the CBRS concept in modified SAS registration procedure for GAA to simplify LSA Licensing process.
- Extend the LSA1 interface to support spectrum resource grant and relinquishment (as shown in the CBRS state diagram in Figure 4) for a MFCN without violating sharing method specific rules.

- Utilize CBRS PPA in CBSD-SAS interface information exchange instead of CBSD detailed data.

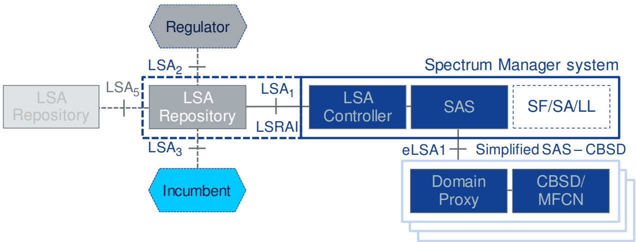


Figure 6. The LSAevo architecture reference model based on the LSA standards with CBRS extensions.

3.3. LSAevo validation

In this trial, we responded to the discussed early 5G deployment requests by demonstrating how LSA evolution can move towards more dynamic and flexible spectrum management concept. LSAevo concept and system architecture can be applied to 3.4-3.8 GHz band so that current individual fragmentation challenges to take the band into 5G use in the respective member states can be solved, while ensuring that the communication of the incumbent users, FWA, fixed links, and satellite earth stations do not experience any harmful interference. In the e2e field trial, an industrial automation FoF micro-operator use case was validated. In this use case, the objects, special needs (low latency), and other solutions are local, e.g., industrial machinery never leaves the site or they need special connectivity only when in the local defined area. This leads to a different type of network, opportunities, and requirements. Private micro-operator networks offer relatively speaking unlimited capacity and speed by tapping into large pools of local spectrum. Network architecture is built around distributed and edge clouds offering low latency and local content management to boost use case development with the domain specific ecosystem as illustrated in Figure 7.

Validation platform, depicted in Figures 7 and 8, utilizes open APIs, native could architecture, and leverages the sharing economy principles to create a sustainable business models across stakeholders and interfaces. The demonstrated Network as a Service (NaaS) deployment consists of commercial Long-Term Evolution (LTE) User Equipment (UEs), 3.5 GHz eNodeBs under LSAevo control, and virtualized hosted Evolved Packet Core (EPC). The implemented SM demo system runs on commercially available virtualized Network Management System (NMS) and Self Organizing Network (SON) platforms, and is built on synergies between the existing LSA and CBRS standards. The LTE test network is installed in the Nokia factory in Oulu, Finland. The incumbents were created for the demonstration purpose based on typical types and protection criteria in Finland and the EU member states [40].

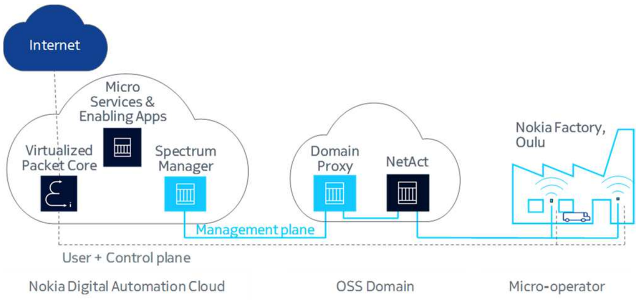


Figure 7. The LSAevo trial environment.

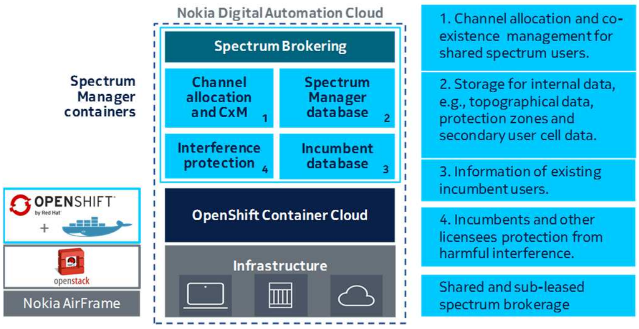

Nokia Digital Automation Cloud
Spectrum Manager containers
Spectrum Brokering
Channel allocation and CxM 1
Spectrum Manager database 2
Interference protection 4
Incumbent database 3
OPENSHIFT
openstack
Nokia AirFrame
OpenShift Container Cloud
Infrastructure
1. Channel allocation and co-existence management for shared spectrum users.
2. Storage for internal data, e.g., topographical data, protection zones and secondary user cell data.
3. Information of existing incumbent users.
4. Incumbents and other licensees protection from harmful interference.
Shared and sub-leased spectrum brokerage

Figure 8. The LSAevo Spectrum Manager architecture.

The LSAevo *Spectrum Manager (SM)* was implemented in Nokia Digital Automation Cloud service running the network as an on-demand private network. The hardware infrastructure is based on Nokia AirFrame running RedHat's openStack and OpenShift Docker-formatted containers. As container images include code, system libraries, and settings, containers isolate software from the processing infrastructure. The SM had two main functions resembling the LSA Repository and LSA Controller. Interference Protection ensures that the LSAevo controlled NaaS LTE network does not cause harmful interference to incumbent spectrum users. It could be considered as an electronic, automated version of radio regulation. Channel Allocation and Co-existence Management optimize the radio resource utilization of the NaaS LTE network within the limits defined by the Interference Protection. The data processed by the Interference Protection and Channel Allocation and Co-existence Management were stored in SM and Incumbent databases, respectively. The architecture contained also Spectrum Brokering to facilitate sub-leasing of radio licenses from a license holder to a temporary and regional spectrum user. The Spectrum Broker was not demonstrated in this trial. The incumbent information was provided by the Finnish Communication Regulatory Authority, Ficora. The largest number of incumbents were the FWA spectrum allocations registered in Radio communication sector of the International Telecommunication Union (ITU-R) database. ITU-R database contained a few FSS earth stations, which were considered and protected by LSAevo. R&D radio licenses and microwave links for broadcasting applications in Oulu area were the geographically closest incumbents.

Results of the initial e2e field trial demonstrated first time at the WInnComm-Europe 2017 [41] and EUCNC 2017 [40] are summarized in Table 2. The LSAevo band evacuation and reconfiguration process was implemented into Nokia e2e trial environment and initial performance measurement studies have been conducted to evaluate the involved time scales for the e2e network evacuation and reconfiguration in the LSA band due to incumbent's immediate LSA spectrum notification. Timestamps have been recorded at SM (Docker), DP (Eden-NET) and UE (Nemo Outdoor). In addition, OAM (NetAct CM Operation Manager) logs were used. DP is registered and authorized to SM. Heartbeat interval (HB) is 10 seconds.

Table 2. LSAevo band reconfiguration measurement results.

T	Cumulative time for workflow step T1-T8 in seconds	e2e
T1	Incumbent notification arrives at SM	0
T2	SM informs DP to vacate the spectrum	8
T3	DP sends relinquishment message for frequency in use. DP EMS configuration for freq. vacation starts	9
T4	UE LTE cell service dropped (Nemo Outdoor)	69
T5	EMS ends, DP asks new frequency grant from SM	81
T6	After HB, EMS configuration of new granted freq. begin	91
T7	UE LTE cell service received	154
T8	SM is informed EMS config. ready by DP via HB that	163

4. CONLUSIONS

This paper discusses the regulatory and standardization status of the Licensed Shared Access, compares it with the Citizens Broadband Radio Service concept, reviews results from the ongoing feasibility study in the European Telecommunications Standards Institute on temporary spectrum access for local high-quality wireless networks, proposes a new LSA evolution architecture, and presents the early results of the world first LSAevo e2e validation. In the trial, we responded to the early European 5G deployment requests by demonstrating how LSA evolution can move towards more dynamic and flexible spectrum management concept enabling integration of local vertical services. Introduced LSAevo concept and system architecture can be applied to 3.4-3.8 GHz band so that fragmentation challenges to take the band into 5G use in the member states can be solved, while ensuring that the communication of the incumbent users, fixed wireless access, fixed links, and satellite earth stations do not experience harmful interference.

In the e2e field trial, local high-quality wireless network use case for an industrial automation micro-operator on 3.4-3.8 GHz band was validated. Network architecture was built around distributed and edge clouds, offering low latency and local content management to boost use case development with the domain specific ecosystem. Proposed LSAevo builds on proven LSA benefits of leveraging scale and harmonization in regulation & standardization, and utilization of existing commercial assets and capabilities. Enhanced flexibility and dynamics in sharing stems from the CBRS framework. Introduced new extensions enable, new frequency bands towards 5G, locally-confined and temporarily-flexible spectrum with novel 5G use cases, horizontal sharing and sub-licensing for efficient use of the spectrum assets, and as a recapitulation lowers entry barrier for new service providers through unbundling investments in spectrum, infrastructure, and services. Performance validation was conducted by measuring the duration of the spectrum evacuation and the base station cell reconfiguration workflow steps in the LSA band due to Incumbent's immediate LSA spectrum resource availability notification. The measurement results revealed that both the emergency evacuation and the reconfigure operation can be done in a way that fulfills typical service incumbent's requirements in the Finnish sharing use case, and wider in a static and semi-static LSA use cases.

This study provides viewpoints about additional ingredients and revisions, which can be of help for key stakeholders and regulators for implementing LSA evolution. The successful deployment of the LSA evolution towards 5G needs will further improve the efficiency of the spectrum use, influence the management approach of other spectrum bands and create new business opportunities. This calls for a collaborative effort from the government, industry and academia to set the harmonized regulatory framework, agree on the standard, and prove the architecture and technology enablers in a pre-commercial trial with e2e ecosystem including novel incumbents and use cases, like private networks and micro-operators.

5. ACKNOWLEDGMENT

This work has been done in the Micro-operator concept for boosting local service delivery in 5G (uO5G) research project within the 5G programs of the Finnish Funding Agency for Technology and Innovation.

6. REFERENCES

[1] Cisco white paper, Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020. Feb. 2016.

[2] ITU Report M.2290-0, Future spectrum requirements estimate for terrestrial IMT, 2014.

[3] RSPG16-032. Final Radio Spectrum Policy Group strategic roadmap towards 5G for Europe. Opinion on spectrum related aspects for next-generation wireless systems, Nov. 2016.

[4] EC COM(2016) 588 final. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. 5G for Europe: An Action Plan, Brussels, Sept. 2016.

[5] Ofcom Statement. Update on 5G spectrum in the UK, 2017.

[6] GSMA Spectrum. 5G Spectrum Public Policy Position, Nov. 2016.

[7] 3GPP RP-170855, Work Item on New Radio (NR) Access Technology, 2017.

[8] GSA, The Future of IMT in the 3300-4200 MHz frequency range, June 2017.

[9] European Commission, Proposal for a Directive of the European Parliament and of the Council establishing the European Electronic Communications Code COM(2016) 590, Articles 2 (26), 45 (2) (e), 46 and 47, 2016.

[10] ECC Report 205, Licensed Shared Access, 2014.

[11] FCC, Amendment of the Commission's Rules with Regard to Commercial Operations in the 3550-3650 MHz Band, FCC, Docket 12-354, 14-49, 2014.

[12] S. Yrjölä, P. Ahokangas, and M. Matinmikko, "Evaluation of recent spectrum sharing concepts from business model scalability point of view," IEEE International Symposium on Dynamic Spectrum Access Networks, pp. 241-250, 2015.

[13] ETSI RRS, DTR/RRS-0148: Feasibility study on temporary spectrum access for local high-quality wireless networks, Early draft 0.0.6, June 2017.

[14] METIS II D1.1 Refined scenarios and requirements, consolidated use cases, and qualitative techno-economic feasibility assessment, Mobile and wireless communications Enablers for the Twenty-twenty Information Society-II. [Online] Available: https://metis-ii.5g-ppp.eu/wp-content/uploads/METIS-II_D1.1_v1.0.pdf, 2016.

[15] COHERENT Deliverable D4.1 Report on enhanced LSA, intra-operator spectrum-sharing and micro-area spectrum sharing, Coordinated control and spectrum management for 5G heterogeneous radio access networks. [Online]. Available: http://www.ict-coherent.eu/coherent/wp-content/uploads/2015/10/COHERENT_D4_1_v1.pdf, 2015.

[16] V. Frascolla et al., “Dynamic Licensed Shared Access – A New Architecture and Spectrum Allocation Techniques,” IEEE 84th VTC, Montreal, OC, pp.1-5, 2016.

[17] M. Matinmikko, M. Latva-aho, P. Ahokangas, S. Yrjölä, and Timo Koivumäki, "Micro operators to boost local service delivery in 5G," Wireless Personal Communications journal, Springer, May 2017.

[18] The White House, Realizing the Full Potential of Government-Held Spectrum to Spur Economic Growth, President's Council of Advisors on Science and Technology (PCAST) Report, July 2012.

[19] The White House, Expanding America's Leadership in Wireless Innovation, Presidential Memorandum, June 2013.

[20] FCC 16-55, The Second Report and Order and Order on Reconsideration finalizes rules for innovative Citizens Broadband Radio Service in the 3.5 GHz Band, 2016.

[21] WINNF Spectrum Sharing Committee, SAS Functional Architecture, [Online]. Available from: http://groups.winnforum.org/d/do/8512, 2016.

[22] The 3GPP, R4-168006: Relevant requirements for Band 48 introduction in 36.104, TSG-RAN4 Meeting #80bis, Ljubljana, Slovenia, Oct. 2016.

[23] M. Sohul, M. Yao, T. Yang, and J. Reed, "Spectrum Access System for the Citizen Broadband Radio Service," IEEE Commun. Mag., vol. 53, no. 7, pp. 18-25, 2015.

[24] The Federal Communications Commission Technical Advisory Council Advanced Sharing and Enabling Wireless Technologies (EWT) WG, Sharing recommendations [Online] Available:https://transition.fcc.gov/bureaus/oet/tac/tacdocs/meeting92314/TACMeetingSummary9-23-14.pdf, 2014.

[25] P. Aho et al., "Field Trial of Citizens Broadband Radio Service (CBRS). The 22nd European Wireless Conference (EW16), Oulu, Finland, May 2016.

[26] M. Hamblen, "Multiple U.S. trials underway for shared 3.5 GHz wireless spectrum," Computerworld, Inc. [Online]. Available:http://www.computerworld.com/article/3075748/mobile-wireless/multiple-u-s-trials-underway-for-shared-3-5ghz-wireless-spectrum.html, May 2016.

[27] EC, Promoting the shared use of radio spectrum resources in the internal market, COM (2012) 478, European Commission, Sept. 2012.

[28] RSPG Opinion on Licensed Shared Access. RSPG13-538, Radio Spectrum Policy Group, Nov. 2013.

[29] CEPT, Harmonized technical and regulatory conditions for the use of the band 2300-2400 MHz for Mobile/Fixed Communications Networks (MFCN), CEPT, ECC Decision (14)02, June 2014.

[30] ETSI, Mobile Broadband services in the 2300-2400 MHz frequency band under Licensed Shared Access regime. ETSI TR 103.113, 2013.

[31] ETSI, System requirements for operation of Mobile Broadband Systems in the 2300 MHz -2400 MHz band under LSA. ETSI TS 103 154, 2014.

[32] ETSI, System Architecture and High-Level Procedures for operation of Licensed Shared Access (LSA) in the 2300 MHz-2400 MHz band. TS 103 235, 2015.

[33] M. Matinmikko et al., "Cognitive radio trial environment: First live authorized shared access-based spectrum-sharing demonstration," IEEE Veh. Technol. Mag., vol. 8, no. 3, pp. 30-37, 2013.

[34] S. Yrjölä et al., "Licensed Shared Access (LSA) Field Trial Using LTE Network and Self Organized Network LSA Controller," Wireless Innovation Forum European Conference on Communication technologies and Software Defined Radio (WInnComm-Europe), pp. 68-77, Oct. 2015.

[35] ECC PT1, "World's first LSA pilot in the 2.3-2.4 GHz band," Input contribution to ECC PT1 #51, ECC PT1(16)028, 2016.

[36] RED Technologies, "Ericsson, RED Technologies and Qualcomm Inc. conduct the first Licensed Shared Access (LSA) pilot in France," [Online]. Available from: http://www.redtechnologies.fr/news/ericsson-red-technologies-and-qualcomm-inc-conduct-first-licensed-shared-access-lsa-pilot-france, 2016.

[37] ECC PT1, Operational guidelines for spectrum sharing to support the implementation of the current ECC framework in the 3 600-3 800 MHz", PT1(16)82 Annex 20, Apr. 2016.

[38] M. Kibria, G. Villardi, K. Nguyen, W. Liao, K. Ishizu, and F. Kojima, “Shared Spectrum Access Communications: A Neutral Host Micro Operator Approach,” IEEE Journal on Selected Areas in Communications. PP, No. 99, May 2017.

[39] S. A. Ashraf, I. Aktas, E. Eriksson, K. W. Helmersson, and J. Ansari: "Ultra-Reliable and Low-Latency Communication for Wireless Factory Automation: From LTE to 5G", Proc. of IEEE conference on Emerging Technologies and Factory Automation, Berlin, Germany, Sept. 2016.

[40] S. Yrjölä and H. Kokkinen, "e2e field trial of the Dynamic Spectrum Manager for a 5G industrial automation micro operator on 3.4-3.8 GHz in Europe," 2017 European Conference on Networks and Communications: Radio Access Technologies towards 5G (RAT), Oulu, June 2017.

[41] S. Yrjölä and H. Kokkinen, "Citizen's Broadband Radio Service enables micro-operators to provide Industrial automation," Wireless Innovation Forum European Conference 2017, Oulu, May 2017.

Coding-Scheme Classification with Applications to PHY-Layer Security

Garrett Vanhoy and Tamal Bose
Electrical and Computer Engineering
University of Arizona
Tucson, Arizona 85719
Email: {gvanhoy, tbose}@email.arizona.edu

***Abstract*—The exploitation of side-channel information (SCI) poses a threat to the security of even the most sophisticated systems. SCI generally refers to any information that is exposed from a system employing encryption other than the original or encrypted data. In wireless systems, this information can include signal attributes such as received signal strength, bandwidth, burst duration, modulation, and others. Although encryption prevents an eavesdropper from being able to completely understand traffic being generated between devices, SCI can be exploited to potentially circumvent encryption. Traffic patterns are an especially revealing form of SCI. For example, it has been shown that particular traffic patterns can be used to identify a web page that a user is currently browsing. Many existing techniques used to exploit or extract SCI require knowledge of the protocol being used between devices or being able to extract commonly unencrypted information from protocol headers. In this paper, we discuss how methods to hide physical layer parameters may still be overcome using classification techniques.**

I. Introduction

The fulfillment of security requirements such as data authenticity, integrity, and confidentiality, has always been a challenge for wireless communication systems. This is because in wireless systems, both malicious and legitimate users have access to the same media. In wired systems and malicious user needs to connect physically to the communication system in order to carry out attacks. With the proliferation of wireless access technologies such as Bluetooth and Wi-Fi, the threat of wireless the threat of wireless attacks has become a increasingly important issue. Addtionally, with the increasing availability of general-purpose software defined radios and sophisticated tools, the threat of wireless attacks has become more prominent. As a result there has been a substantial increase in the academic, government, and commercial communities in wireless security.

Communication systems typically adopt the OSI protocol architecture for transporting information which includes several layers. Information from one application to another typically traverses from the upper the layers of the protocol stack through the lower layers and then back up the stack again like that seen in Figure 1.

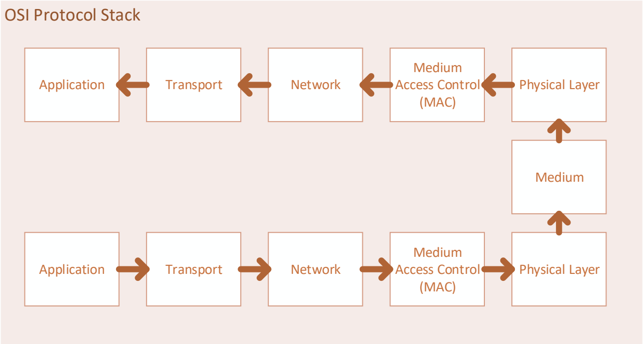


Fig. 1: OSI Protocol Stack

The protocol stack contains the application, transport, network, medium access control (MAC), and physical layers in decreasing level on the stack. Information from upper layers are always passed down to lower layers as “packets” of generic data to be moved and could take many different forms. Hence, layers are usually only aware of parameters related to their intended purpose and thus they present different vulnerabilities and require corresponding security measures to be in place. However, the physical layer can present vulnerabilities to other parts of the stack. The physical layer, which defines how a medium is used to relay bits that come from all other layers, is naturally more vulnerable to attacks than in wired systems [1]. It has been shown that through observing frame sizes at the physical layer, the type of application being used can be discerned [2].

There are several attacks that can be carried out against wireless systems that mirror that of wired systems. Eavesdropping, denial of service (DoS), spoofing, man-in-the-middle, and message injection are commonly discussed attacks in the realm of security. In wireless systems, since eavesdropping is easier to carry out, other attacks are become even more threatening as well as a result. For example, in a wired network, a common type of DoS attack simply overwhelms a server with many requests for a service, rendering it unable to properly serve many users. A malicous user might carry this out without any knowledge of the state of the server or other clients. In a wireless scenario malicious node can attempt to prevent service for a only for a particular client instead of all clients by understanding only manipulating some traffic.

In this work, background of the existing work in the field of physical layer security in Section II is provided. It is then shown how these existing approaches may be compromised using common classification techniques with cyclostationary features in Section III. An eavesdropping scenario is the simulated and the results are presented in Section V. Finally we conclude in Section VI.

II. Physical Layer Security

In the realm of wireless security, physical layer security has seen a substantial increase in interest as it has been shown that encryption alone is not enough to achieve security requirements. The physical layer of wireless systems is exposed to both legitimate and nefarious users and this presents two major threats to security: jamming and eavesdropping. In this work, we focus primarily on threats related to eavedropping.

Eavedropping presents a threat to information confidentiality and is generally combatted using encryption. The success of encryption lies entirely in the assumption that an eavesdropper does not have enough computational capacity to "break" the encryption scheme compared to the intended receiver that has additional information about the incoming message. There has been plenty of work with brute-force type attempts to breaking encryption methods. However, the more imminent threat to the efficacy of encryption is the leakage of side channel information (SCI). Side channel information refers to any information that is neither the encrypted payload data (bits) or the unencrypted payload data (also bits). For example, almost every protocol prepends a header to the data payload. This header has a known format and sometimes contains fields that are common or of predictable values. If such a field is encrypted and is known to the eavesdropper it may be possible to reverse-engineer the encryption key in a short time. This is called a known plain-text attack. Other types of SCI that have been exploited include signal strength, bandwidth, data rate, and modulation scheme. These parameters are generally relayed through PHY-layer headers in transmitted frames, but they can also be inferred without the use of these bits using various DSP or statistical techniques. Hence, the obfuscation of SCI or prevention of SCI leakage has become an important topic in wireless physical layer security.

A. Existing Countermeasures

Preventing SCI leakage has been carried out in a few ways. First, SCI leakage can be prevented through effectively reducing the signal to noise ratio (SNR) seen by the eavesdropper relative to the intended receiver. One effective technique for accomplishing this is the use of low probability of intercept (LPI) waveforms that are typically applied in the realm of radar. Since the intended receiver knows the precise method (sequence) by which the transmitted signal has been spread, it achieves a spreading gain over the eavesdropper. Friendly jamming or artificial noise generation also reduces the effective SNR at the eavesdropper by transmitting artificial noise along with its message. By transmitting noise in unused space, time, or frequency, the eavesdropper receives a disadvantage over the intended receiver.

A common and critical assumption of existing work addressing the threat of eavesdropping is that the eavesdropper has similar capabilities as the intended receiver. These results are relevant to common scenarios where commercially available receivers are used in malicious ways such as one cell phone eavesdropping on another. In scenarios where a receiver is specifically designed for surveillance purposes, it is unlikely to be so restricted. There are very few, if any, existing approaches to the issue of eavesdropping that can guarantee an SNR advantage for the intended receiver in the presence of an eavesdropper *with sufficiently capable hardware*.

One such technique that can be used to fulfill security requirements even in the presence of a more capable eavesdropper is full frame encryption [3]. header bits that can be used to determine important information about the frame such as its length, modulation, coding scheme, and the protocol are generally sent unencrypted. By encrypting these bits and obfuscating the modulation and coding scheme, it becomes much more difficult for an eavesdropper to effectively determine what may be happening at higher layers of the OSI stack.

III. Signal Classification

In this section, the method by which signals of the same modulation, but different coding scheme, can be classified using cyclostationary features and modern machine-learning techniques.

A. Cyclostationary Signals

Many signals and systems have been modelled as wide-sense stationary stochastic processes where second-order statistics of the signal remain constant with time, but whose autocorrelation is independent of time. However, many man-made signals exhibit a periodic or an almost-periodic autorrelation function because they contain various periodic structures in time. From a practical perspective, if the received signal is considered as a stochastic process that is a sum of both additive white Gaussian noise (AWGN) and the signals of interest, they each fall into the category of almost-cyclostationary processes. The distinction between cyclostationary (CS) and almost-cyclostationary (ACS) is an important distinction made in the literature and more details can be found in [4]. Such signals can be said to be cyclostationary or periodically correlated and the analysis of this class of processes has been ongoing for decades [5]. Over the years, cyclostationarity has been studied rigorously in continuous and discrete, real and complex, and stochastic and non-stochastic contexts [4].

As an example, let $x(t)$ be a continuous-time real-valued stochastic process. The process $x(t)$ can be called wide-sense cyclostationary if its autocorrelation function, $R_x(\tau) = \int_{-\infty}^{\infty} x(t-\frac{\tau}{2})x(t+\frac{\tau}{2})dt$, is periodic. Due to this periodicity, the process $x(t)$ can be expanded in a Fourier series such that

$$R_x(t,\tau) = \sum_{\alpha \in A} R_x^{\alpha}(\tau)e^{j2\pi\alpha t}, \tag{1}$$

and

$$R_x^{\alpha}(\tau) \triangleq \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} R_x(t,\tau)e^{-j2\pi\alpha t}dt \tag{2}$$

where τ is the lag parameter and A is the set of cycle frequencies α such that $R_x^\alpha(\tau) \neq 0$. Both the coefficients $R_x^\alpha(\tau)$, which are called the cyclic autocorrelation functions (CAF) and their Fourier transforms, which are called the cyclic spectra of the process $x(t)$, are useful in analysis and classification of signals and processes. The cyclic spectra is especially important for analysis as it represents the density of correlation between two spectral components of a process that are separated by α. This property is especially useful for detection because a process that produces additive white Gaussian noise (AWGN) contains no correlation between spectral components making it readily discernible from many signals.

The cyclic spectrum at cycle frequency α of the process $x(t)$ can be written

$$S_x^\alpha(f) = \int_{-\infty}^{\infty} R_x^\alpha(\tau) e^{-j2\pi f\tau} d\tau, \tag{3}$$

which can interpreted as the time-averaged statistical correlation of two spectral components seperated by cycle frequency α as the bandwidth of each spectral component approaches zero. For this reason, the cyclic spectrum can also be called the spectral correlation density (SCD). According to this definition $S^0(f)$ is actually the traditional power spectral density (PSD) of the process $x(t)$. The SCD has been used in a variety of signal processing and classification tasks in communications, radar, and others [6], [7], [8], [9]. This primarily stems from its ability to detect and characterize the presence of cyclic features such as cyclic prefix length, symbol period, or carrier frequency even in the presence of noise and other channel effects.

B. Estimation of the SCD

The difficulty with using this feature is that it is computationally complex to estimate for digital signals. A single estimate of the SCD of a reasonable resolution can require up to 65,536 of 32-point complex FFTs. The SCD is readily derivable in closed form for many continuous forms of communications signals and a substantial efforts has been made decades ago to estimate this quantity for a finite-duration digital signal. For a digital signal, estimating the SCD has two commonly-used methods which are optimized for computational efficiency. The FFT Accumulation method (FAM) [10] and Spectral Strip Correlation Algorithm (SSCA) [11] are the two variations to estimate the SCD. The FAM, due its data parallel computations and regular data access patterns, offers opportunities for exploiting parallelism [12], particularly on hardware architectures that allow fine-grained parallelism. The FAM is the most efficient computationally and is calculated as

$$X_{N'}(n,k) = \sum_{r=-N'/2}^{r=N'/2} a[r]x[n-r]e^{-j2\pi k(n-r)T_s}. \tag{4}$$

with

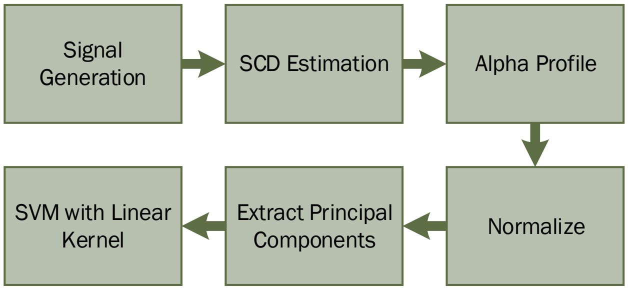


Fig. 2: Classification System

$$S_x^\alpha(n,k) = \frac{1}{N}\sum_{n=0}^{N-1} \frac{1}{N'} X_{N'}\left(n, k+\frac{\alpha}{2}\right) X_{N'}^*\left(n, k-\frac{\alpha}{2}\right). \tag{5}$$

where N' and N together determine a resolution in both the time and frequency domains and $a[n]$ is an arbitrary windowing function. Equation (4) is the sliding window discrete Fourier transform (DFT) with window $a[n]$.

C. Classification by Cyclostationarity

The proposed system to classify signals can be seen in Figure 2. The SCD is first estimated using the FAM method with parameters, $N' = 32$, $N = 256$, and $a[n]$ is a hamming window of length N'. Second, the maximum is taken along the α axis of estimate of $S_x^\alpha(f)$ to narrow down the number of features the classifier needs to train with. Next, these values are normalized with respect to the maximum value in each set so as to emphasize the relative values of all the features. Then, using principal component analysis, the 4096 features are narrowed down to 25 features that account for the majority of the variance between the classes. To classify among each class, a support vector machine with a linear kernel is trained on a subset of the provided training data.

IV. Simulations

Simulations were constructed using GNU Radio to generate signals for the classification system. The signals were generated with random data for every frame. Using different bits for every frame ensures that classification is not done based on the actual data in the frame. After this, several different channel codes were implemented including a convolutional code of rate 1/2, trellis code of rate 3/4, and no coding at all. These codes are commonly used to decrease bit error rates in wireless systems. Next, each bit was mapped onto a 16-QAM signal constellation and pulse-shaped with an interpolating root-raised cosine filter of transition with one-fourth the sample rate. To simulate the reception of a bandpass signal, the signal was then carrier modulated to a frequency of 1/4 the sample rate. The signal was then carried through an additive white Gaussian noise (AWGN) channel. The imaginary part of the signal was then discarded so that the SCD features that will be generated later are more rich.

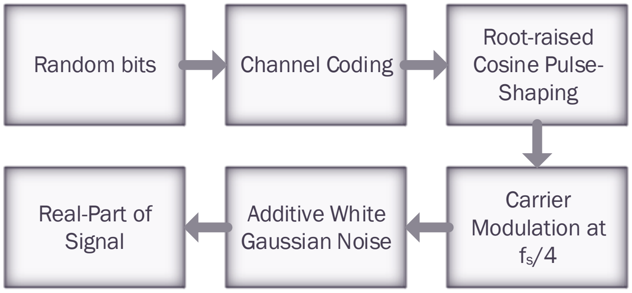

Random bits
Channel Coding
Root-raised Cosine Pulse-Shaping
Carrier Modulation at $f_s/4$
Additive White Gaussian Noise
Real-Part of Signal

Fig. 3: Simulated Wireless System

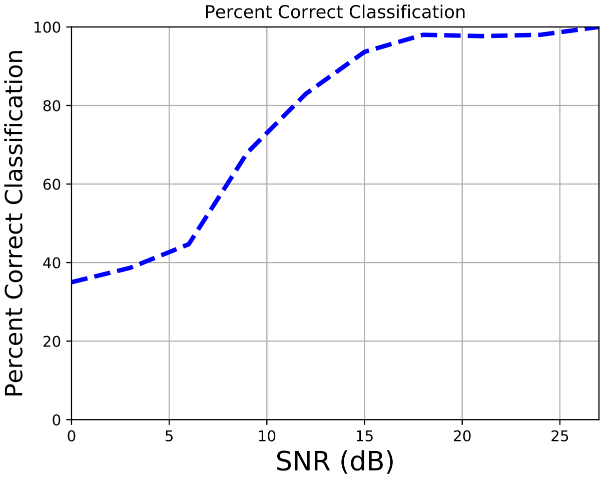


Fig. 4: Percent Correct Classification vs SNR

V. Results

To evaluate the ability of the proposed classification system, 100 signals of each type of channel coding were generated for each SNR tests. The classifier was trained on a random sample of 1/3 of the total samples generated. The results are shown in Figure 4. The results show that although the same modulation is sent, it may be possible to determine the channel coding with some degree of accuracy.

VI. Conclusions

In the field of wireless security, the exposition of side-channel information poses a threat to many systems. It has been shown in recent studies that even basic information about a waveform such as the number of bits in each frame can be used to compromise data confidentiality. Many techniques have been proposed to deter eavesdropping of malicious users that have similar hardware, but the scenario where eavesdroppers have more capable hardware has not been studied in detail. This work shows that even in the face of techniques such as full frame encryption, which hides information critical to determining the number of bits in a frame, it is possible to classify the channel coding being used. This suggests that if the dictionary of possible channel codes of obfuscating modes are known to an eavesdropper, that with enough time and samples the frame length could be determined with a high degree of certainty.

VII. Acknowledgements

This project was partially supported by the Broadband Wireless Access and Applications Center (BWAC); NSF Award No. 1265960.

References

[1] Y. Zou, X. Wang, and L. Hanzo, “A Survey on Wireless Security: Technical Challenges, Recent Advances and Future Trends,” vol. 104, no. 9, pp. 1–31, 2015. [Online]. Available: http://arxiv.org/abs/1505.07919

[2] S. Chen, R. Wang, X. F. Wang, and K. Zhang, “Side-channel leaks in web applications: A reality today, a challenge tomorrow,” *Proceedings - IEEE Symposium on Security and Privacy*, pp. 191–206, 2010.

[3] H. Rahbari and M. Krunz, “Full frame encryption and modulation obfuscation using channel-independent preamble identifier,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2732–2747, 2016.

[4] W. A. Gardner, “Cyclostationarity: Half a century of research,” *Signal Processing*, vol. 86, no. 4, pp. 639–697, 2006.

[5] H. L. Hurd, *An investigation of periodically correlated stochastic processes*. University Microfilms, 1970.

[6] Q.-Q. Lu, M. Li, and X.-J. Wang, “Improved method of spectral correlation density and its applications in fault diagnosis,” *Beijing Keji Daxue Xuebao/Journal of University of Science and Technology Beijing*, vol. 35, no. 5, pp. 674–681, 2013.

[7] D.-S. Yoo, J. Lim, and M.-H. Kang, “Atsc digital television signal detection with spectral correlation density,” *Journal of Communications and Networks*, vol. 16, no. 6, pp. 600–612, 2014.

[8] Z.-B. Zhang, L.-P. Li, and X.-C. Xiao, “Detection and chip rate estimation of mpsk signals based on cyclic spectral density,” *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, vol. 27, no. 5, pp. 803–806, 2005.

[9] L. Zhu, H.-W. Cheng, and L.-N. Wu, “Identification of digital modulation signals based on cyclic spectral density and statistical parameters,” *Journal of Applied Sciences / Yingyong Kexue Xuebao*, vol. 27, no. 2, pp. 137–143, 2009.

[10] S. R. Schnur, “Identification and classification of ofdm based signals using preamble correlation and cyclostationary feature extraction,” DTIC Document, Tech. Rep., 2009.

[11] D. Simic and J. Simic, “The strip spectral correlation algorithm for spectral correlation estimation of digitally modulated signals,” in *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 1999. 4th International Conference on*, vol. 1, 1999, pp. 277–280 vol.1.

[12] A. R. Castro, L. C. Freitas, C. C. Cardoso, J. C. Costa, and A. B. Klautau, “Modulation classification in cognitive radio.”

VALIDATION OF A CRV MODEL USING TVWS MEASUREMENTS

Khalil Anderson(a158@umbc.edu)[1], Lauren Lusk(l.o.lusk7@ou.edu)[2], Marti Hands(marti.hands@ttu.edu)[3], and Garrett Vanhoy(gvanhoy@email.arizona.edu)[4]

[1]University of Maryland, Baltimore County: Baltimore, MD, US
[2]University of Oklahoma: Norman, Oklahoma, US
[3]Texas Tech University: Lubock, Texas, US
[4]University of Arizona: Tucson, AZ, US

ABSTRACT

As autonomous vehicles advance, their commercial popularity will rise. This growth will increase the need for wireless transmission of data to these cars not only to drive more efficiently, but to also entertain the driver. As people are freed from the task of driving, the demand for in-car internet applications, such as Netflix or Skype, will grow. Currently, autonomous vehicles are allowed to transmit using the band specified by the IEEE protocol 802.11p. While vehicles can transmit data using the 5.9 GHz band (5.850-5.925 GHz), the band may not support wireless transmission of media to vehicles' infotainment systems. This requires an alternative. With the switch from analog to digital television, the government has vacated the analog TV bands. These bands provide a possible solution to the limitations of 802.11p transmissions. The vacated space is called TV white space. One proposed use of this white space is to provide Wi-Fi. This idea has been called White-Fi. According to our research, researchers have measured whether the specific frequencies are occupied but do not provide the unprocessed data. With this in mind, we measure the occupancy of the TV white space and we simulate how a network using this band would perform under the multiple scenarios of everyday driving.

1. INTRODUCTION

Although humans have developed many autonomous and semi-autonomous systems such as self-guided rockets, autopilot for airplanes, and cruise-control for cars, fully autonomous vehicles are still unproven. The idea of a self-driving car has been around almost as long as the automobile itself. In 1939, General Motors (GM) created its Futurama ride for the World's Fair. The ride allowed people to observe GM's vision of 1960, which included automated highways modeled after railroads. However, that vision was only science-fiction because computers were still in their infancy and did not have the necessary computational power to implement the vision. Nevertheless, today, faster computers and better sensors have brought self-driving cars closer to a reality with each prototype.

As companies such as Tesla and Google become successful in implementing autonomous vehicles, the demand for reliable wireless internet access within vehicles will rise for the following two reasons: first, vehicles will need to transmit and receive data from their surroundings for safety precautions; second, people riding in cars will presumably desire on-demand entertainment during their travels. While Wi-Fi has been considered, it is unlikely that this band will be an effective solution in high-bandwidth, high traffic situations.

Currently, 75 MHz of the 5.9 GHz band has been allocated by the FCC for the use of wireless access in vehicular environments (WAVE) that is controlled by IEEE 802.11p. 802.11p has been used for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Although it performs well for these purposes, through literature review, it appears unlikely that the 802.11p protocol would be able to handle streaming video and audio [1] [2] [3] [4].

In 2009, the US Government vacated the bands between 470 MHz and 890 MHz that had previously been used to transmit analog television. These bands, referred to as Television White Space (TVWS), may be a solution to the limitations of 802.11p transmissions. One of the proposed uses of TVWS is the implementation of Wi-Fi across the bands, which has been dubbed White-Fi. While white space occupancy has been measured many times, the unprocessed data found by the authors is, if it exists, not easily available. Another issue with preceding research in vehicle transmission applications is that it does not address spectrum availability at highway speeds.

In order to gauge the viability of White-Fi for streaming purposes, we measure the TVWS from Tucson to Phoenix, Arizona. Afterwards, we create a White-Fi simulation, using background noise levels based on collected data.

2. BACKGROUND

2.1. Measurements

While researchers may choose slightly different approaches, the data-gathering process is mostly consistent. Researchers who desire to gather spectrum data chooses a channel or set of channels to measure, then use an antenna to gather aforementioned data. With the invention of software-defined radio (SDR), it is possible to accurately measure a large amount of channels during a short period. Since its implementation, many researchers have used SDR to determine spectrum occupancy in different areas of the spectrum. While several articles focus on spectrum occupancy from a driver's point of view, the majority gather data from a set location.

In the literature review, spectrum measuring processes were considered from a wide range of sources. Spectrum measurements in the United States were considered, and international data-collection was considered as well. Research conducted in Metro Cebu, Philippines, Melbourne [5], Australia [6], and Singapore [7] were just a few of the places in which TVWS data was collected.

One instance of spectrum measurement was done in Singapore by [7]. Over the course of 12 weekdays, the authors measured the spectrum from 80 MHz to 5.85 GHz in 60 MHz bands using a spectrum analyzer and a directional antenna. During the 12 days, a total of 1248 samples were taken for each channel. After the measurements were taken, the authors used a threshold power level to differentiate ambient noise from an actual signal. Using the threshold power level, the authors determined spectrum usage. As is the case with many spectrum measurements, the results cannot be used to verify the viability of TV White Space for two reasons: first, the measurements were taken in Singapore, where analog TV is still being broadcast from 494 MHz to 680 MHz. Second, because the measurements were taken from a still location, the measurements may differ from a vehicle that is driving at highway speeds. Therefore, another approach must be considered.

While spectrum measurements are taken worldwide, there are also many researchers who concentrate their measurements in the United States. Data has been gathered in Green Bank, West Virginia [8], Denver, Colorado, San Diego, California, and Los Angeles, California [9], but other than these instances, it is difficult to find recent TVWS measurements in the United States. This has led us to conclude that the measurements are sparse or undocumented. Another difficulty was discovering what other researchers used as their threshold values to determine occupancy. The inaccessibility of the unprocessed data does not allow other researchers to verify results.This is a potential problem because if the researcher picked a threshold value that was too high, it could have returned that that frequency was unoccupied when it actually was.

2.2. Simulation

The current segment of spectrum allocated to vehicles is 75MHz. While 75MHz sounds adequate for most purposes, this is offset by the fact that only 40MHz is useful for things other than safety or control messages. Also, service channels are together in pairs so only two can be bonded together for a 20MHz channel. The remaining segments of the channel are two safety channels, a control channel and a guard band at the beginning. Each channel with a bitrate of 27Mbps [3]. This means with channel bonding you can achieve maximum of 54Mbps throughput or a total of 108Mbps throughput if the road side unit(rsu) has multiple antennas.

According to Netflix help site, they recommend 3Mbps for standard definition and 10Mbps for high definition. This would mean that a single piece of infrastructure could only handle 10 cars in HD and 36 in SD. While 35 seems like sufficent amount of cars, during a traffic jam the density can be between 185-250 vehciles per lane per mile [10]. WAVE would not be able to handle 10% of the vehicle in HD. This also fails to factor in packet error or collisions on the medium.

With the vacating of the Analog TV bands, 420MHz of spectrum space has been made available. Previously Analog TV channels were 6MHz. This means in the range 470MHz-890MHz, there are 70 channels open. Not all of these channels are still open as the government has reallocated the space, but most of the band is still unlicensed. One of these channels has a capacity of 80Mbps at 40Mw of power [11]. While research has been done to show the reliability of WAVE [1], none, according to our research, has compared White-Fi in VANETs to WAVE through simulation to show, empirically, the extra capacity that the would actually be gained by using White-Fi for VANETs.

3. METHODOLOGY

3.1. Measurements

The measurements taken in this paper were made using a Universal Software Radio Peripheral (USRP) a USRP N200 with a LTE Dipole antenna attached. Since the antenna has a range of 698-2690 MHz, instead of measuring the full TVWS band, data was collected from 35 channels ranging from 680-890 MHz. The USRP's parameters were configured using a GNU Radio program that controlled the N200's sample frequency, sample rate, and the number of samples taken per channel. The information written to the file that is later analyzed includes the frequency being measured, the position at which the frequency is being sampled, and the signal power.

The GNU Radio program implemented one original block and several blocks from UHDGPS. The original blocked used, labeled Sweeper, periodically sends a message to the UHD: USRP Source block that tells it to change the frequency being sampled. While the UHD: USRP Source block was used, it was slightly modified from the original. The modification allowed the program to send a tag downstream every time the frequency changes. One of the other blocks used was the Trigger Sample-Timer Event block, which allowed the program to use 256 samples out of every 30 kilo-samples. The average power from the samples used was found using the CPDU Average Power block, and then the output from that was written to a JSON file along with the location, time, and frequency information.

Before gathering the actual experimental data, a few tests were conducted to check how the receiver measured the power and gain levels relative to the transmitted levels. First, two USRP N200s were connected by a coaxial cable: a receiver and a transmitter. Next, using GNU Radio, a signal was transmitted from one to the other, and the gains were compared using QT GUIs that are built into GNU Radio. After repeating the test several times with different signal strengths, it was concluded that the gain offset is approximately -10 dB.

The experiment was conducted in the following way: the USRP was powered through a power strip that drew power from the cigarette lighter and placed between the driver and passenger seats. A laptop, connected to the USRP via an Ethernet cable, was held by someone in the passenger seat of the car and used to monitor the USRP. The LTE Dipole antenna was threaded through the car's rear window and taped to the outside of the vehicle. During the experiment, all 35 channels were measured every 175 milliseconds, and the USRP's sampling rate was set at 6 MHz. The GNU Radio program was set to change frequencies every 5 milliseconds, or every 30 kilo-samples. Out of the 1000 samples taken per channel per scan, the first 300 samples were discarded, and the following 256 were used to determine the signal power at that time. The calculated signal power was then written to a JSON file. Once all of the hardware and software was configured, we drove from Tucson to Phoenix, Arizona.

3.2. Simulation

The simulation portion of this paper was done using OMNeT++ [12], a network simulation framework, that has the ability to simulate all levels of the OSI model. The framework uses module which can be built on an extended. The behavior of the modules are coded in C++ while the language that describes the modules and their connections is called NED(Network Description Language). Once the network is setup up, a configuration file is setup which can change the run time variables of the network. This configuration file can have multiple configurations and run numbers setup to allow for easy organization of multiple circumstances.

Along with OMNeT++, there are multiple frameworks that have been created through it to ease the future development of projects. The two framework that I build off of are called INET and Veins[13]. INET provides a very detailed breakdown of each layer of the OSI model from the physical layer to the application layer. It allows a user to customize background noise, bitrate, carrier frequency, packet size and more. It also has built applications such a UDP video stream client and server that we use in the simulation. This UDP video stream client and server are setup to stream a specific video size at a particular send interval with a packet size that is set at run time. The video size was set at 2GB with a packet length of 2000B which is below the default MTU which will allow for a the fastest transmission rate. The send interval is 25 microseconds for analog and 100 microseconds for WAVE since both speeds are faster than the fastest bit rate for each mode.

INET also has a module called a WirelessHost which has the basic setup for more wireless simulations. This module is configured for IEEE 802.11g, and while the carrier frequency and bit rate can be changed, the module has many submodules which demand the parameters be fit to 802.11 WiFi standard. To overcome this, the radio type had to be changed from IEEE 802.11 radio to APSK radio. The default mac type also had to be change since it required the radio of an 802.11 mode.

Once these modules were changed, the mac was configured with the default values for WAVE EDCA mac[14]. The power level was set to 1W so that area is covered regardless of the center frequency of the radio. Rayleigh fading was used for the path loss type and two separate radio configurations were made: one for Analog and the other for WAVE. The analog frequency is 528MHz and WAVE is at 5.9GHz. The bit rates for analog are 80Mbps, 160Mbps, 240Mbps, 320Mbps. The WAVE bit rates are 27Mbps, 54Mbps, 81Mbps and 108Mbps. Multiple bit rates are used to mimic a road side unit simultaneously using multiple channels to transmit and recieve.

Veins was the other framework was used. Most of veins wasn't used but the mobility model was used. Veins uses Sumo to simulate traffic. Sumo can be used to take maps from OpenStreetMap to generate traffic. A local server listens for the a sumo.cfg file, and when an OMNeT++ network is run that loads that file, it creates a connection to that server. The server handles the navigation and insertion of vehicle nodes in the simulation.

4. NUMERICAL RESULTS

4.1. Measurements

The drive from Tucson to Phoenix, Arizona, resulted in 85,167 power samples across the 35 measured channels - an average of 2,433 samples per channel. The data is processed in the following way: the data is separated into five-mile intervals. The percentage of occupied bands is then computed over each interval. The threshold value was ascertained by comparing the average power in each band in Tucson and Phoenix to known occupied bands. Based upon the comparison, it was visually concluded that -100db was approximately where the bands became unoccupied. The resulting percentages were then graphed as a line plot showing the occupancy throughout Tucson and Phoenix.

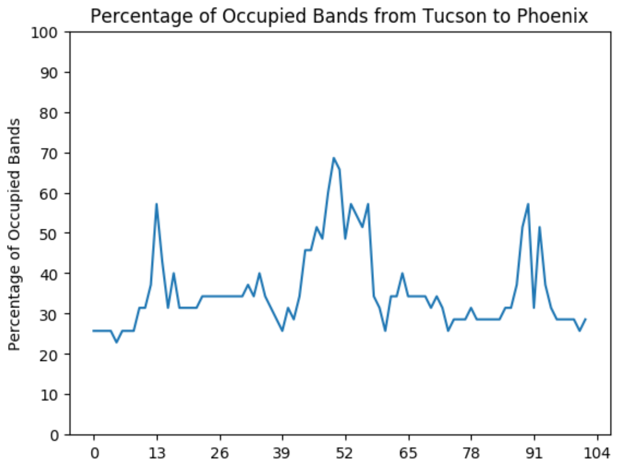


Figure 1

As figure 1 shows, there is some white space available. The highest occupancy level is around 70%, which leaves 30% of the 210 MHz free (about 63MHz).

4.2. Simulation

Table 1: The bitrate of a single RSU in each run to 200 vehicles in a 64,000m^2 area

Run Type - Amount of Channels	Throughput(MBps)
WAVE - 1	1.512
WAVE - 2	2.816
WAVE - 3	3.900
WAVE - 4	4.348
Analog - 1	5.711
Analog - 2	7.315
Analog - 3	9.624
Analog - 4	11.463

In the simulation, it record the bytes received and sent by each node. Using this we calculated the throughput of the RSU in each run type. As shown in Table 1 The 80Mbps analog run has a faster throughput than the 108Mbps WAVE run. The reason this most likely happens is the greater effect that the amount of radios transmitting in the small area have on the signals. This interference will hinder the ability of signals to reach their recipient properly.

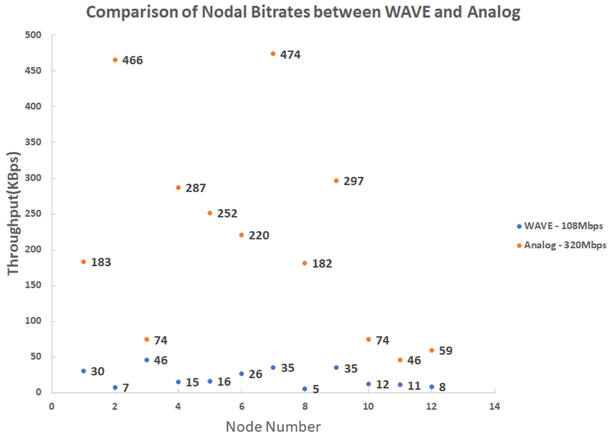


Figure 2

As figure 2 shows the comparison of the bitrates between the 4 channel Analog vs WAVE. You can see that most data points show similar trends as each other since each numbered node should enter and exit the simulation at the same time. This means they will send they ARP request at the same time, but since they are different bit rates and carrier frequencies, one transmission may take longer which why the trends are not exact. The timeouts for ack and arp requests are the same but the transmission time are different for each bit rate.

5. CONCLUSIONS

The need for the transmission of data and entertainment services to vehicles will require an abundance of bandwidth in order to service the amount of vehicles present in everyday traffic. Each passenger will want the ability to stream their favorite movies, work, or listen music while their car drives itself. The current 5.9GHz band does not have enough bandwidth to service one lane of a traffic, but the analog TV bands can provide better service. Based upon spectrum data gathered from Tucson to Phoenix, there is enough vacant space in the analog TV bands to create a sufficient amount of channels to provide smooth connection to vehicles from open to congested areas. That said, before this can become a reality real world tests need to be done to confirm how well the analog TV bands works in a vehicular network. The data does give an idea of how well the TV white space can be used to connect autonomous vehicles to the internet.

6. ACKNOWLEDGEMENTS

This project was supported by the National Science Foundation and the University of Arizona in the CAT Vehicle REU under Tamal Bose and Larry Head.

REFERENCES

[1] Y. Yao, L. Rao, and X. Liu, "Performance and Reliability Analysis of IEEE 802.11p Safety Communication in a Highway Environment," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4198–4212, 2013.

[2] A. T. Giang, A. Busson, A. Lambert, and D. Gruyer, "Spatial Capacity of IEEE 802.11p-Based VANET: Models, Simulations, and Experimentations," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, vol. 65, no. 8, pp. 6454–6467, 2016.

[3] X. Ma, X. Chen, P. Hightower, M. Abdul-hak, N. Al-Holou, U. Mohammad, K. Sjöberg-Bilstrup, E. Uhlemann, E. G. Strom, A. M. S. Abdelgader, W. Lenan, S. M. Nazir, and R. Rastogi, "The Physical Layer of the IEEE 802 . 11p WAVE Communication Standard : The Specifications and Challenges," *Electric Vehicles - Modelling and Simulations*, vol. II, no. 3, pp. 1–5, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5503941%5Cnhttp://www.intechopen.com/books/electric-vehicles-modelling-and-simulations/predictive-intelligent-battery- management-system-to-enhance-the-performance-of-electric-vehicle%5Cnhttp:/

[4] F. A. Teixeira, V. F. e Silva, J. L. Leoni, D. F. Macedo, and J. M. S. Nogueira, "Vehicular networks using the IEEE 802.11p standard: An experimental analysis," *Vehicular Communications*, vol. 1, no. 2, pp. 91–96, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.vehcom.2014.04.001

[5] G. J. M. Llames and A. S. Banacia, "Spectrum sensing system in software-defined radio for determining spectrum availability," *International Conference on Electronics, Information, and Communications, ICEIC 2016*, no. February, pp. 1–4, 2016.

[6] A. Al-Hourani, V. Trajkovi??, S. Chandrasekharan, and S. Kandeepan, "Spectrum occupancy measurements for different urban environments," *2015 European Conference on Networks and Communications, EuCNC 2015*, pp. 97–102, 2015.

[7] M. H. Islam, C. L. Koh, S. W. Oh, X. Qing, Y. Y. Lai, C. Wang, Y.-C. Liang, B. E. Toh, F. Chin, G. L. Tan *et al.*, "Spectrum survey in singapore: Occupancy measurements and analyses," in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*. IEEE, 2008, pp. 1–7.

[8] M. A. Mchenry and K. Steadman, "NSF Spectrum Occupancy Measurements Project Summary," Tech. Rep., 2005. [Online]. Available: http://www.sharedspectrum.com/inc/content/measurements/nsf/NSF_Project_Summary.pdf

[9] F. Sanders, "Broadband spectrum surveys in Denver, CO, San Diego, CA, and Los\nAngeles, CA: methodology, analysis, and comparative results," *1998 IEEE EMC Symposium. International Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.98CH36253)*, vol. 2, pp. 988–993, 1998.

[10] V. L. Knoop and W. Daamen, "Automatic fitting procedure for the fundamental diagram," *Transportmetrica B: Transport Dynamics*, vol. 5, no. 2, pp. 133–148, 2017. [Online]. Available: http://dx.doi.org/10.1080/21680566.2016.1256239

[11] A. B. Flores, R. E. Guerra, E. W. Knightly, P. Ecclesine, and S. Pandey, "IEEE 802.11af: A standard for TV white space spectrum sharing," *IEEE Communications Magazine*, vol. 51, no. 10, pp. 92–100, 2013.

[12] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 60:1–60:10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416290

[13] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[14] K. L. Yeung, "Link Quality Based EDCA MAC Protocol for WAVE Vehicular Networks," pp. 1–6.

AN EVALUATION OF ADAPTIVE EQUALIZERS ON NON-LINEAR HF/IONOSPHERIC CHANNEL MODELS

Noel Teku (nteku1@email.arizona.edu)[1], Garrett Vanhoy (gvanhoy@email.arizona.edu)[1], and Tamal Bose (tbose@email.arizona.edu)[1]

[1]Dept. of Electrical and Computer Engr. The University of Arizona, Tucson, AZ 85721-0104

ABSTRACT

HF communications (3-30 MHz) have been a popular medium for emergency, military, and hobbyist applications because they generally do not require significant architecture or equipment. For long range HF communications, which are common, this involves having signals reflect off the Earth's ionosphere. However, the ionosphere is highly unstable, varying with respect to space, time, and frequency, which has prompted research into compensating the effects of the ionosphere in the receiver to maintain robust communications even when faced with harsh channel conditions. This instability in the ionosphere causes the channel to change between non-linear and linear models at different times. Thus, the objective of this paper is to utilize and evaluate the performance of two non-linear adaptive equalizers in recovering distorted signals transmitted through a non-linear HF channel. The performance of the equalizers will be characterized using mean squared error (MSE).

1. INTRODUCTION

The High Frequency (HF) band (3-30 MHz) is very popular among radio enthusiasts. Unlike mainstream communication architectures that require a significant amount of equipment to maintain (i.e. internet, satellites, fiber optic cables, etc.), transmissions in the HF band are made by only using radios to transmit signals by reflecting them off of the ionosphere, which enables long-range communications at a low cost. My work in [1], for example, showed that such communications were possible over the HF band, as our team (stationed in Tucson, Arizona) was able to hear communications from Tokyo, Japan. Because of this, in addition to being inexpensive, HF systems are used in emergency crises more than mainstream communication systems. One example of this is Hurricane Katrina, where HF systems were used to organize "rescue and recovery operations" as a result of damaged, regularly used communication infrastructures [2]. In addition, the HF band is used for various military applications, such as enabling troops to establish communication links in different locations [3]. However, while the ionosphere enables long-range/low-power communications over the HF band, it itself is not a stable medium. As [4] elaborates, the ionosphere varies significantly as a result of different atmospheric variations (i.e. solar radiation, sunspot cycles, seasonal changes), which can result in the transmitted signal suffering from effects like multipath and fading. This has been the motivation for the development of ionospheric models, with the most heralded of these developed by Clark Watterson. As will be elaborated on later, his model assumes that the channel is stationary in time and frequency making it only accurate for small bandwidths [5]. Thus, it is desirable to design a system capable of obtaining an adaptive, real-time model of the ionosphere that can be used to improve transmissions made over the HF frequency band.

Equalizers are an ideal signal processing tool for this application because they take the received signal from the channel as input and output an estimate of the transmitted signal by "creating an inverse model of the transmission channel" [6]. The error between this estimate and the transmitted signal is calculated and, in the case of adaptive equalization, is fed into a learning algorithm that utilizes the error to fine tune the equalizer's coefficients. Thus, the objective of this paper is to survey/compare the performance of different equalizers that have been implemented specifically for the HF channel. These equalizers are specifically the Least Mean Squares Decision Feedback Equalizer (i.e. LMS-DFE) and Constant Modulus Algorithm equalizer (i.e. CMA). The structure of the paper is as follows. Section 2 provides background on previous implementations of these equalizers for the HF channel. Section 3 provides an explanation on the functionality of a general Decision Feedback Equalizer (DFE). Section 4 provides an explanation on LMS-DFEs and CMA equalizers specifically. Section 5 elaborates on the Watterson and non-linear models used in this effort. Section 6 describes how the experiments were implemented and provides an analysis on the results obtained. Section 7 summarizes the work completed in this effort and provides next steps we will take in future projects.

2. BACKGROUND

Historically, different types of equalizers have been implemented in the context of HF channels. The authors of [7] showed that decision feedback equalizers (DFEs) were better suited for the HF channel than maximum-likelihood sequence estimation (MLSE) equalizers, due to having a similar performance but much simpler complexity. As such, different types of DFEs have been implemented for the HF channel. One type of equalizer that has been implemented frequently for HF environments is the Kalman Decision Feedback Equalizer (i.e. Kalman-DFE), where different variations of the standard Kalman state-tracking algorithm have been used to update the taps of the DFE. In [8], the recursive least squares (RLS) algorithm was utilized to update the taps because its usage in adaptive equalization is analogous to tracking states with standard Kalman filtering. In [9], a square root Kalman algorithm is utilized to update the weights of the DFE. In [10], a fast recursive least squares (FRLS) DFE is implemented and shown to have significantly better performance than a LMS-DFE and a similar performance to the DFE formed in [9].

In addition, different types of equalizers have been used in the HF channel. In [11], various blind equalization algorithms, including the Constant Modulus algorithm, were implemented over the air. While the above algorithms require sending a training sequence over the channel, to allow the equalizers to adapt the taps, the main attraction of blind equalization algorithms is that they don't require training sequences - allowing for more useful data to be sent at higher speeds [12].

3. DFE FUNCTIONALITY

The DFE consists of a feed-forward and feedback filter. Different adaptive algorithms are used to update the taps of the DFE, with the mean squared error (MSE) algorithm being one of the most common algorithms utilized. A standard DFE can be expressed as follows [13]:

$$\hat{I}_k = \sum_{j=-K_i}^{0} c_j v_{k-j} + \sum_{j=1}^{K_2} c_j I\check{}_{k-j} \tag{1}$$

where the indices of j and c_j represent the taps of the DFE, v represents the sequences received from the channel, $\check{I}$ represents symbols decoded in previous iterations, and $\hat{I}_k$ represents the output of the equalizer. In equation 1, the first summation represents the feed-forward filter and the second represents the feedback filter. As [13] describes, if the mean squared error is being used to update the weights, the following cost function is minimized as shown in equation 2:

$$E(K_1, K_2) = |\hat{I}_k - \check{I}_k|^2 \tag{2}$$

where $(K_1 + 1)$ and K_2 represent the number of taps used in the feed-forward and feedback filter respectively. However, as will be shown in section 4, this criteria will vary based on the learning algorithm used.

4. LEAST MEANS SQUARES AND CONSTANT MODULUS ALGORITHMS

4.1. LMS

Similar to equation 2, [14] defines the error of the equalizer to be the following:

$$e(n) = d(n) - y(n) \tag{3}$$

where e represents the error, d represents the desired signal, and y represents the equalized output as given in equation 4:

$$y(n) = \hat{w}^H(n)u(n) \tag{4}$$

where $u(n)$ is the input sequence, $\hat{w}$ represents an estimate of the ideal tap vector (i.e. w), and H represents the Hermitian transposition. Contrary to equation 2, the LMS algorithm assumes that the function to be minimized is as shown in equation 5 [14]:

$$J(n) = E[|e(n)|^2] \approx e(n)e^*(n) \tag{5}$$

where $E[]$ denotes the expectation value and $e^*(n)$ is the complex conjugate of the error vector. As [14] elaborates this removal of the expectation is used to make the estimation more feasible to adapt to a varying environment. To find the subsequent tap vector that is most effective in optimizing the above cost function, equation 6 is used to update the taps at each iteration of the adaptation process:

$$\hat{w}(n+1) = \hat{w}(n) + \mu u(n)e^*(n) \tag{6}$$

where μ is a step-size parameter. Thus, the LMS algorithm would be an alternative to the MSE algorithm for updating the weights of the DFE. An important attribute to note about the LMS algorithm is that it does have knowledge about attributes of the signal (i.e. modulation) prior to equalization, unlike the CMA algorithm.

4.2. CMA

The Constant Modulus algorithm (CMA) is commonly used as a learning algorithm for blind equalizers. It is ideally used when managing signals with a constant amplitude. Similar to the LMS and MSE algorithms, it also has a cost function as shown in equation 7 [15]:

$$D^{(p)} = E(|z(n)|)^p - R_p)^2 \tag{7}$$

where D is referred to as the dispersion of order p, $z(n)$ represents the equalizer output, and the meaning of R_p will be explained later. As [11] summarizes, steepest descent can be used to optimize equation 7 resulting in the following update equation:

$$e_{n+1} = e_n - \mu \frac{\partial D^{(p)}}{\partial e}|_{e=e_n} \tag{8}$$

where e represents each of the taps of the equalizer, μ is a step-size parameter, and $\frac{\partial D^p}{\partial e}$ is given by the following expression:

$$\frac{\partial D^{(p)}}{\partial e}|_{e=e_n} = 2pE[y_n^*|z(n)|^{p-2}(|z(n)|^p - R_p)] \qquad (9)$$

As [11] indicates, a specific form of R_p can be derived making the following two assumptions. The first is that the equalized signals, $z(n)$, are assumed to have the form $a(n) * e^{j*(\phi+2\pi\delta fnT)}$. The second is that $\frac{\partial D^{(p)}}{\partial e}$ is set to zero. Plugging this information into equation 9, it can be shown that R_p is given by the following expression:

$$R_p = \frac{E[|\mathbf{a(n)}|^{\mathbf{2p}}]}{E[|a(n)|^p]} \qquad (10)$$

5. CHANNEL MODELS

5.1. Watterson Model

As explained in the introduction, Clark Watterson's model, referred to as the Watterson model, is the most frequently used ionospheric model in HF experiments. The model represents the ionosphere's effects on transmitted signals as a tapped delay line as shown in figure 1 [16]. Each tap modulates the transmitted signal in amplitude and phase to simulate Gaussian scattering effects using a complex gain function specific to each tap, represented in figure 1 as $G_i(t)$. Each function has a bi-variate Gaussian power spectrum, which enables the model to capture the effects of Rayleigh fading on the outputted signal [17].

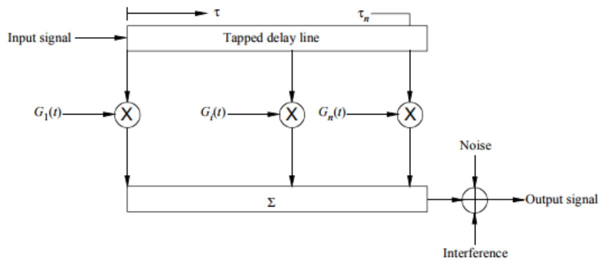


Figure 1: Watterson Model - Tapped Delay Line

Each main tap of the delay line shown in figure 1 is created using a series of taps generated using equation 11 [17]:

$$h_n(t) = ke^{-\pi^2 f_j^2 n^2 T_s^2}; -N < n < N \qquad (11)$$

where f_j represents the Doppler spread, k is used to preserve a unity gain, n represents the tap index, N represents the total length of the filter, and T_s represents the sample period. For this effort, the above tap generation was implemented as shown so that it could accept the delay spread, Doppler spread, and number of taps as parameters. Complex Gaussian noise is filtered through these taps and sampled, with these final values used as the main taps of the Watterson model [17].

The International Telecommunications Union (ITU) [18] defines different HF channel conditions based on the delay spread and Doppler spread inputted into the Watterson Model, which are displayed in tables 1 and 2. As will be shown in later sections, these values were used in the simulations to simulate the ionosphere under different conditions.

Table 1: ITU Poor Channel Conditions

Channel Condition	Poor
Delay Spread (ms)	2
Doppler Spread (Hz)	1

Table 2: ITU Moderate Channel Conditions

Channel Condition	Moderate
Delay Spread (ms)	1
Doppler Spread (Hz)	0.5

5.2. Non-Linear Channel Model

While the Watterson model has been experimentally verified and used in multiple projects analyzing the HF band, as stated earlier, the assumptions Watterson made when constructing this model makes it valid only for small bandwidths [5]. In addition, because of the ionosphere's instability, it is possible for it to exhibit non-linear attributes at different instances, which the Watterson model does not neccesarily capture. However, Watterson's model did verify certain attributes of the ionosphere that are accurate; specifically, the channel having a Gaussian distribution and Rayleigh fading. Thus, in addition to using Watterson's original model, a non-linear model was also implemented to simulate such effects. To our knowledge, there is not an existing non-linear Ionospheric model that has been verified/heralded as the Watterson model. Thus, the following model, shown in equation 12 is our initial attempt to simulate non-linear effects in the HF channel:

$$y = x^2 + x + n \qquad (12)$$

where y represents the signal received by the equalizer, x is the signal corrupted from the Watterson model, and n represents Gaussian noise. Thus, for this project, both Watterson's original model and the above non-linear model were used in the experiments for both equalizers.

6. RESULTS

6.1. System Architecture

The equalizers explained in section 4 were simulated using existing implementations in GNU Radio, an open source language used for rapid prototyping of communication systems/algorithms [19]. As explained earlier in section 4.1., the LMS equalizer does have knowledge about the incoming signal unlike the CMA equalizer. GNU Radio accounts for this by having one of the inputs to its LMS decision-directed (i.e. LMS-DD) equalizer block be the constellation of the incoming signal. In replacement of this parameter, the CMA equalizer block takes as an input the desired modulus (i.e. amplitude) that will drive the signal's equalization. Despite this distinction both blocks have the following parameters: number of taps, gain of the update loop, and samples per symbol rate of the incoming signal, which is used to apply proper down-conversion if the signal was interpolated at an point in the system. During the simulations, each equalizer had 4 taps, a gain of 0.01, and a sample per symbol rate of 2. A vector sink block was used to store the equalizer's output, and Python was used to calculate the subsequent MSEs.

8-PSK was the only modulation used in all experiments. Complex Gaussian noise was filtered through 101 taps used to generate a Watterson model with two main taps on its delay line, as explained in section 5.1. Both the Watterson and non-linear model were simulated using GNU Radio and Python.

6.2. MSE Results

Two different experiments were executed to gauge the performance of the two equalizers. The first involved observing the MSE of each equalizer as the SNR of the Watterson model was varied from 10 to 50 dB, with a step size of 5 dB. Each point was averaged over 5000 trials. Figures 2 and 3 show the results of these experiments for poor and moderate channels.

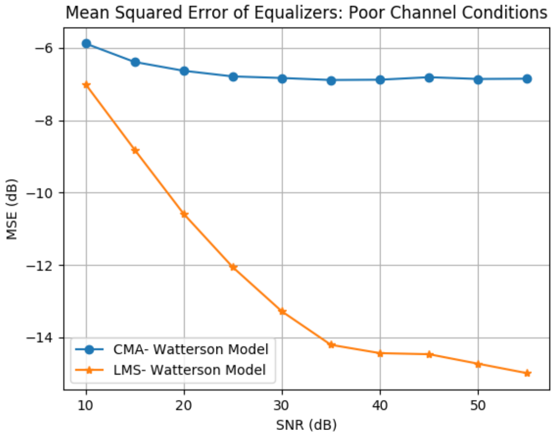


Figure 2: MSEs of LMS and CMA for Poor Channel Conditions as a Function of SNR

As the figures show, for both poor and moderate channel conditions, the LMS-DD equalizer has a much smaller MSE compared to the CMA equalizer. This may be due to the LMS-DD equalizer having "a-priori" knowledge of the signal's modulation, giving it an advantage over the CMA equalizer. However, for both equalizers, the MSE does decrease as the SNR of the channel is increased, as expected.

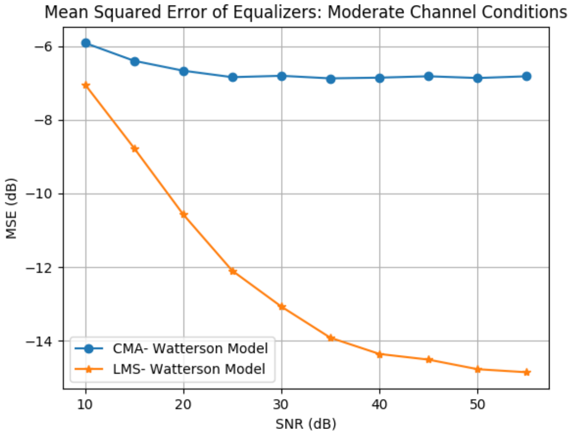


Figure 3: MSEs of LMS and CMA for Moderate Channel Conditions as a Function of SNR

As stated earlier, a vector sink block in GNU Radio was used to store the equalizer's output. A head block was then used to limit how many samples were stored in the vector. Subsequently, the second experiment involved observing the MSE of each equalizer as the number of samples outputted from the equalizers were varied. These trials were performed with the channel having an SNR of 10 dB, under poor and moderate conditions, with a range of 50-300 samples and a step size of 10 samples. Similar to the first experiment, each point was averaged over 5000 trials. The results of these experiments are shown in figures 4 and 5.

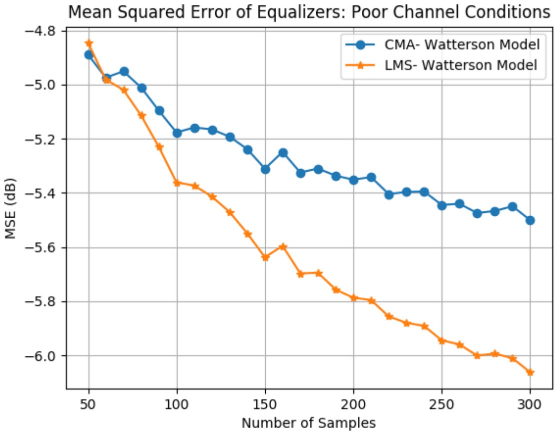


Figure 4: MSEs of LMS and CMA for Poor Channel Conditions as a Function of Number of Samples

Similar to the results from the first experiment, the LMS-DD equalizer has a smaller MSE than the CMA equalizer, under

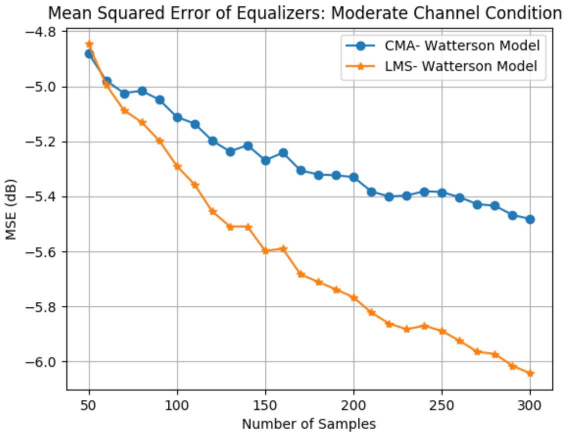


Figure 5: MSEs of LMS and CMA for Moderate Channel Conditions as a Function of Number of Samples

both poor and moderate conditions. As a sanity check, the MSE does decrease as the number of samples is increased again, as expected. However, it is noteworthy that at a low number of samples, the equalizers seem to have a similar performance under poor and moderate conditions. The above experiments were then repeated to obtain MSE results from usage of both the Watterson and non-linear channel models. Figures 6 and 7 show the MSE results when the Watterson and non-linear model are used as the SNR of the channel is varied. Figure 8 shows the MSE results in a similar experiment, with the number of samples sent to the equalizer being varied, but the channel set at an SNR of 10 dB under poor conditions.

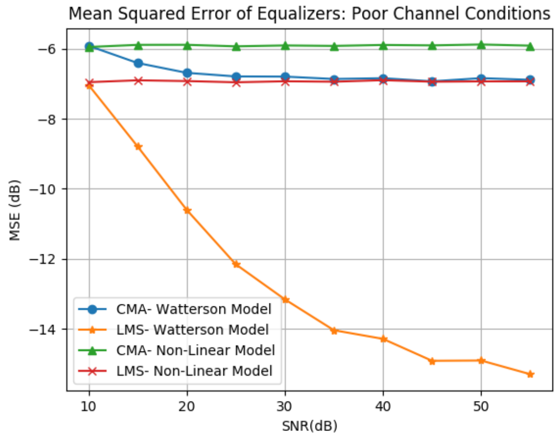


Figure 6: MSEs of Equalizers Under Poor Linear/Non-Linear Conditions as a function of SNR

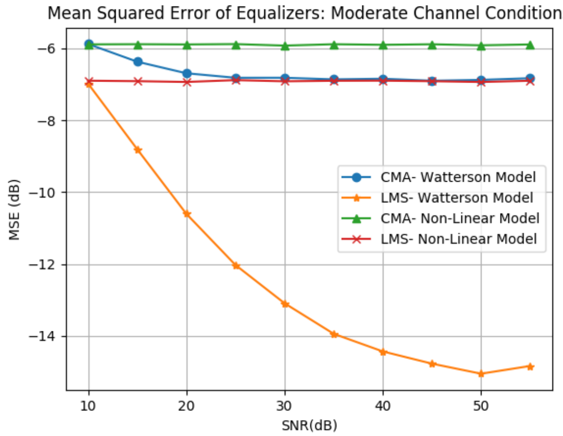


Figure 7: MSEs of Equalizers Under Moderate Linear/Non-Linear Conditions as a function of SNR

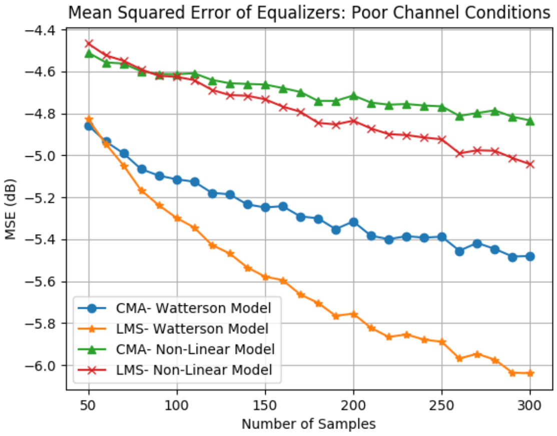


Figure 8: MSEs of Equalizers Under Poor Linear/Non-Linear Conditions as a Function of Number of Samples

Figures 6, 7, and 8 show that using the non-linear channel model produces a higher MSE than using the Watterson model. This is to be expected as the non-linear model provides more distortion to the signal prior to being processed by the equalizer. It's interesting to observe, however, that despite this added complexity the LMS-DD equalizer still produces an MSE smaller than that of the CMA-equalizer under non-linear conditions. Figures 6 and 7 also indicate that the LMS equalizer under non-linear conditions has an MSE that is comparable to that of the CMA equalizer when using the Watterson model, which again affirms the LMS equalizer's superiority in these experiments. In addition, it seems that increasing the SNR is not effective towards reducing the MSE in this scenario. Figure 8 affirms these results but provides some additional insights as it appears the equalizers seem to again have a similar performance when the number of samples collected from the output of the equalizer is small. However, as the number of samples is increased, despite the LMS-DD equalizer having a better performance, figure 8 indicates there is not a significant reduction in the MSE when using the non-linear model. These results imply that further signal processing techniques will need to be implemented to compensate for non-linear effects.

7. CONCLUSION

The objective of this paper was to observe the performance of LMS and CMA equalizers in restoring signals transmitted through linear and non-linear Ionospheric models. It was shown in all experiments that the LMS-DD equalizer had a smaller MSE than the CMA equalizer, which could be due to the LMS knowing the constellation of the transmitted equalizer whereas the CMA did not, due to it being a blind equalizer. As expected, it was shown that usage of a non-linear Ionospheric model produced higher MSEs compared to using the Watterson model. However, it's noteworthy that despite these added complexities the LMS equalizer had a better performance than that of the CMA equalizer. These experiments also indicated that increasing the SNR, as well as the number of samples outputted from the equalizer, are not sufficient means for handling these non-linear effects. This implies that additional signal processing techniques are required to fully recover a signal corrupted due to non-linearities in the HF channel.

One of our next steps will be incorporating different adaptive equalizers/equalization techniques; such as, Minimum Mean Square Equalizers (MMSEs), Neural Networks and Volterra Equalizers, and observing their performance over similar experiments. As in this paper, the effectiveness of these equalizers will be classified based on their average MSE as well as additional metrics such as BER and convergence rate. We will research/investigate the ionosphere's unstable behavior to craft a more realistic non-linear model. In addition, we will work towards having analysis of the equalizers' performances when transmitting over the air, to capture the real-time impact of Ionospheric reflections on a signal transmitted through the HF band.

8. ACKNOWLEDGEMENTS

This project was partially supported by the Broadband Wireless Access and Applications Center (BWAC); NSF Award No. 1265960.

9. REFERENCES

REFERENCES

[1] N. Teku, G. Gulati, H. Asadi, G. Vanhoy, A. H. Abdelrahman, K. Morris, T. Bose, and H. Xin, "Design of a long range cognitive hf radio with a tuned compact antenna," *International Telemetering Conference 2017*, pp. 1–10.

[2] M. Uysal and M. R. Heidarpour, "Cooperative communication techniques for future-generation hf radios," *IEEE Communications Magazine*, vol. 50, no. 10, pp. 56–63, October 2012.

[3] A. D. Sabata and C. Balint, "Structure of signal received by passive ionospheric sounding in the hf band at the location of timisoara, romania," in *2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)*, Oct 2016, pp. 55–58.

[4] F. H. Raab, R. Caverly, R. Campbell, M. Eron, J. B. Hecht, A. Mediano, D. P. Myer, and J. L. B. Walker, "Hf, vhf, and uhf systems and technology," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 888–899, Mar 2002.

[5] C. Watterson, J. Juroshek, and W. Bensema, "Experimental confirmation of an hf channel model," *IEEE Transactions on Communication Technology*, vol. 18, no. 6, pp. 792–803, December 1970.

[6] Z. Zerdoumi, D. Chikouche, and D. Benatia, "Adaptive decision feedback equalizer based neural network for nonlinear channels," in *3rd International Conference on Systems and Control*, Oct 2013, pp. 850–855.

[7] A. Bartlett, S. M. Brunt, and M. Darnell, "Comparison of DFE and MLSE equalisation in a HF serial tone modem and implications for frequency selection," in *IEE Colloquium on Frequency Selection and Management Techniques for HF Communications*, 1999, pp. 15/1–15/7.

[8] F. A. Faik Eken, Erol Hepsaydir, "Performance Study of Kalman Adaptive Equalizer for High Speed Data Transmission over the HF Channel," 1988.

[9] F. Hsu, "Square root Kalman filtering for high-speed data received over fading dispersive HF channels," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 753–763, sep 1982.

[10] E. Eleftheriou and D. Falconer, "Adaptive Equalization Techniques for HF Channels," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 238–247, feb 1987.

[11] R. B. Casey, "Blind Equalization of an Hf Channel For A Passive Listening System," Ph.D. dissertation, Texas Tech University, 2006.

[12] N. Miroshnikova, "Adaptive Blind Equalizer for HF Channels," 2017.

[13] D. Brilyantarto, I. Kurniawati, and G. Hendrantoro, "Early results on the design of adaptive equalizer for HF communications system on equatorial region," in *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*, aug 2014, pp. 1–4.

[14] S. Haykin, *Adaptive Filter Theory*. Pearson, 2014. [Online]. Available: https://books.google.com/books?id=J4GRKQEACAAJ

[15] G. H. Godard, "Self-recovering equalization and carrier tracking in two dimensional data communication systems," *IEEE transactions on communications*, vol. 28, no. 11, pp. 1867–1875, 1980.

[16] I.-R. F.1487, "Testing of HF modems with bandwidths of up to about 12 kHz using ionospheric channel simulators," *Group*, vol. 1487, 2000.

[17] J. M. Wilson, "A Low Power HF Communication System," Ph.D. dissertation, 2011.

[18] I. Recommendation, "520-1 Use of high frequency ionospheric channel simulators," *Recommendations and Reports of the CCIR*, pp. 5–8, 1994.

[19] *GNU Radio*, available at http://gnuradio.org/.

LOW-POWER, LOW-COST SOFTWARE DEFINED RADIO AND ITS APPLICATIONS

Xiaodong Zhang, General Processor Technologies, Beijing, China, xdzhang@hxgpt.com
John Glossner, General Processor Technologies, Tarrytown, NY, glossner@hxgpt.com

ABSTRACT

Currently, many Software Defined Radio (SDR) products in China focus on research and prototyping of communications systems. However, when mass production is required, the practical design and codes should be optimized to fit into new platforms, e.g. ASICs. Such a development flow requires significant investment and time to enter into the market. In this paper, we introduce China-based production environments of the Sandbridge Sandblaster SB3500 chip and system. It is a heterogeneous SDR platform for wireless communications. The platform consists of heterogeneous radio hardware programmed using mainstream high-level C language tools. The main features include: 1) unified source programming composed of host CPU and kernel DSP codes, 2) flexible and powerful instruction set architecture optimized for digital-communications, 3) unified address space between multiple DSP and CPU cores, 4) unified profiling across heterogeneous cores for functional and timing verification using a system simulator, 5) micro-architectural support for instruction-level, data-level and thread-level parallelism, and 6) extremely low-power interleaved hardware multithreaded implementation. The SB3500 SDR platform has been used in research, prototyping, and most recently in China-based production systems. The software programming model for product development has reduced the typical time to market by more than six months. To-date, many types of custom chips and terminals have been designed for the China market including GPS, PDT (Police digital terminal), LTE, BPLC (Broadband power line communication), and NB-IOT, LORA, Radar and even embedded AI (Artificial Intelligent) chips.

1. INTRODUCTION

In recent years, some important trends have emerged in the design and production of custom chips for dedicated communications, localization/positioning and AI recognition applications, especially with the introduction of wide-area IoT (Internet of Things) networks. The motivation behind such trends usually lies in the fact that custom chipsets will reduce the size of user equipment, significantly lower the overall power consumption and in addition, minimize the total cost for mass production. As a consequence, such a solution will become more and more competitive once the market volume gets large enough.

Unfortunately, most applications do not have enough volume to support a custom ASIC. Hence if each of them requires a customized chip design, there will be little possibility to deliver cost effective products. Instead, a different way should be followed for such cases to provide low-cost chips accordingly; that is, the well-known Software Defined Radio (SDR) solution.

In practice, if SDR is considered for implementation, there are many conditions that need to be taken into account, i.e. small development team, for example, less than 10 engineers; short time to market and limited capital investment. As a feasible and competitive candidate, the Sandbridge Sandblaster SDR platform is able to meet such strict requirements simultaneously. In fact, with the Sandblaster solution, many types of dedicated chips, terminals and devices have been designed for the China market including GPS/Beidou, PDT (Police digital terminal), LTE, BPLC (Broadband power line communication), NB-IOT, LORA, Radar and even embedded AI specific chips that are already delivered in different volumes to the customers.

In Section 2 we describe the Sandblaster SB3500 SDR platform. Section 3 introduces the parallel programming development methodology. Section 4 outlines some applications cases designed with the SB3500 and finally in Section 5, we summarize and draw conclusions for this paper.

2. HETEROGENEOUS SDR PLATFORM

Figure 1 shows a chip diagram of the Sandbridge Sandblaster SB3500 chips [3] . The chip includes three DSP cores and one CPU core integrated together. The cores are interconnected with a high-speed ring network. The original application market for SB3500 was commercial wireless handsets. Therefore, the peripheral interfaces are very complete for many interesting applications and include an LCD display, camera, DigRF, GPIO, SPI, I2S, I2C, UART, USB 2.0, etc., which are connected to the internal CPU and DSP cores via ARM AMBA buses.

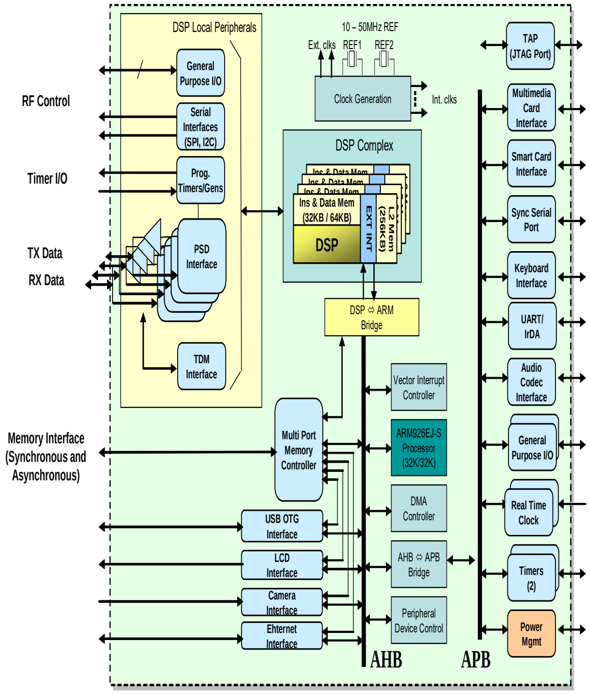


Figure 1. Sandblaster SB3011 Chip

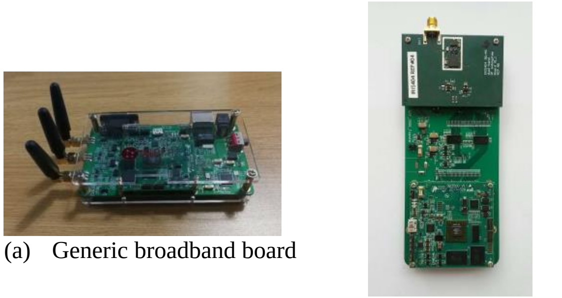
Technology	65nm
Processor Clock	600MHz
Power Dissipation	75mW @ 1V, 25C
On-chip Memory	1.5Mbytes
Peak DSP performance	9.6 GMACs

Table 1. Key Sandblaster Parameters (per core)

.

With SB3500 chips and appropriate RFICs, we have designed different development boards according to different applications' requirements. However, to make sure the final delivered products is the most competitive and cost effective for mass production, it is preferred that the integrated RFIC and analog convertors are available commercially. For example, for broadband wireless modem development, the RFICs that have been adopted for 3GPP terminals should be used.

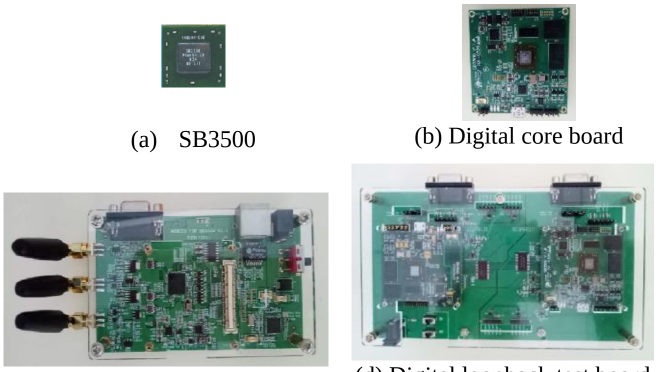
Figure 2 and Figure 3 show the SDR development boards being supplied to developers. Usually they are further separated into their composite series, i.e. the digital core board, the RF/analog board and the digital loopback test board, etc. That is, for generic wireless communications, we present a commonly used development board that is comprised of a digital core and a broadband RF/analog board. But for low-power IoT developments, the wideband RF board must be replaced with appropriate ones for different bandwidths and different radio frequencies. In particular, its sleep mode power management strategy must be carefully optimized for longer battery operation, which is always specified by certain publicly released standard documents. For example, for China Mobile operated NB-IoT terminals, the maximum power consumption that cannot be exceeded for the sleep mode is about 3-5uw in practice.



(a) Generic broadband board

(b) IoT board

Figure 2. SDR development boards for generic wireless communications and IoT



(a) SB3500

(b) Digital core board

(c) RF/Analog board

(d) Digital loopback test board

Figure 3. The SB3500 chip and the associated digital, RF/analog and loopback test boards

With low-cost production in mind, a formal design flow has been established for SB3500 chips, which consists of three stages: 1) fixed-valued algorithm simulation using the SB3500 simulator, 2) prototyping with the SDR board and small-scale field trials, and 3) multi-chip packaging for mass production. These stages are classified as ***Simulator-level***, ***Board-level*** and ***Package-level development stages***, which are illustrated in figure 4. For the algorithm programming and system trials that are encountered in the first and second stages, we encourage our users to use generic development boards and their associated simulation tools, which are already available worldwide. With these boards user coded programs can be profiled and optimized conveniently and quickly to meet real-time requirements. In our experience, the human resource needed for the development of a GPS-like receiver usually requires 3-5 man-months of effort. Compared to classical hardware design this is a very small investment. In addition, even for small-scale field trials, the number of SDR boards required for system testing is very limited and their price is rarely a concern for most users. As a result, the generic SDR boards will always accelerate the development of most applications, such as the signal processing we apply to communication systems.

Once a design enters into mass production, the development then goes into its third stage, where the packaging of multiple chips starts to become an emphasis. Usually the motivation for this lies in cost reduction and yield improvements of the final products. To achieve such goals, we often use either SIP (System in Package) or MCM based packaging technologies. In reality, most Chinese customers prefer to follow this design methodology because compared to conventional board-level integration such a solution has become more and more competitive in their respective markets. As an example, we have shown a SIP packaged SB3502 with one SB3500 and two AFE dies placed inside in figure 5.

As a result, when SB3500 based packaged chips are used in terminals, our clients are able to directly compete with many mainstream chip vendors in the market, whether in the early low-volume sails or later mass productions.

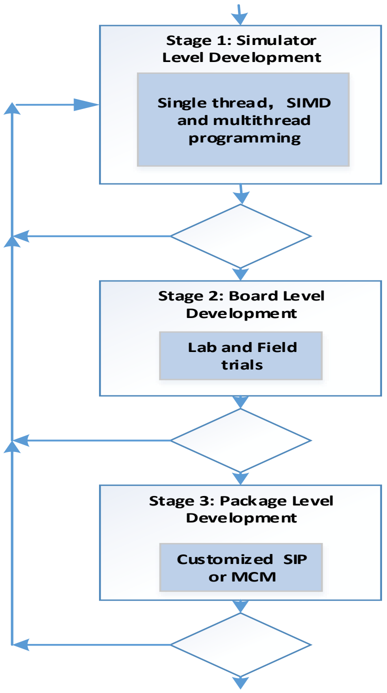

Stage 1: Simulator Level Development
Single thread, SIMD and multithread programming
Stage 2: Board Level Development
Lab and Field trials
Stage 3: Package Level Development
Customized SIP or MCM

Figure 4. Product Development Flow

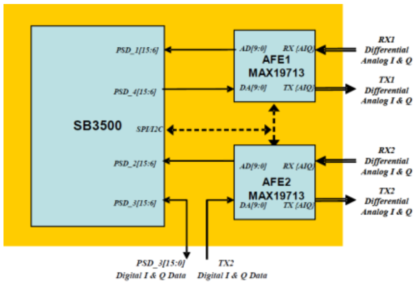


Figure 5. SIP of SB3502 with SB3500 and AFE dies

To-date we have experience with a considerable number of SDR projects using the Sandbridge SB3500 chips. In addition to simplified technology development, providing technical support developers is important. To help our users in the China market we have developed a special website with many design examples, libraries, on-line discussion zones as well as real-time video conference systems, which are shown in figure 6. It is open to anyone requiring technical support at the website www.sdrerc.com.

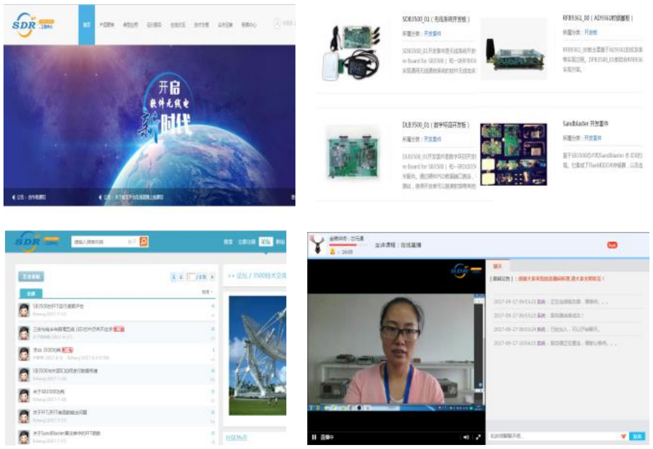
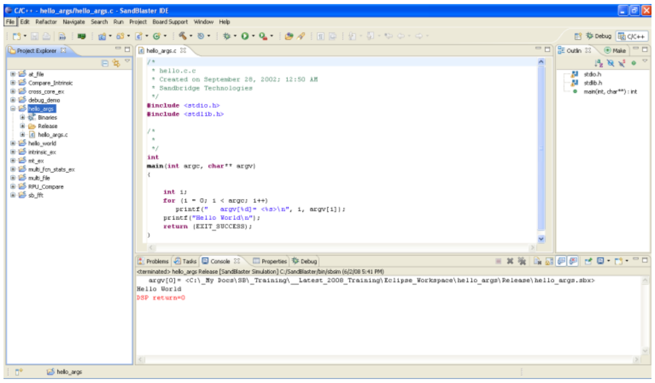


Figure 6. The SDR website (www.sdrerc.com)

3. PARALLEL DEVELOPMENT FLOW

We have developed a tool-chain based on Sandbridge Technologies' tool chain to facilitate the development of SDR applications on the Sandblaster Platform [3] . The software tool chain is primarily dedicated towards generating and simulating efficient code for the Sandblaster processor. The basic philosophy behind the tools is that the user should program in a higher-level language such as C and not need to use any assembly language code. The tool chain consists of an Integrated Development Environment, ANSI C compiler, functional simulator, and a real time operating system.

Figure 7 shows the Integrated Development Environment (IDE). It provides an easy to use graphical user interface to all the software tools. The IDE is based on the open source eclipse integrated development environment [3] . The IDE is the graphical front end to the C compiler, assembler, simulator and the debugger. The IDE provides the ability to create, edit, build, execute and debug an application. In addition, it provides the ability to mount a file system, access CVS, access the web and communicate with the Sandblaster hardware board.



```
/*
 * hello.c.c
 * Created on September 28, 2002; 12:50 AM
 * Sandbridge Technologies
 */
#include <stdio.h>
#include <stdlib.h>

/*
 *
 */
int
main(int argc, char** argv)
{

    int i;
    for (i = 0; i < argc; i++)
       printf("   argv[%d]= <%s>\n", i, argv[i]);
    printf("Hello World\n");
    return (EXIT_SUCCESS);
}
```

```
<terminated> hello_args Release [SandBlaster Simulation] C:/SandBlaster/bin/sbsim (6/2/08 5:41 PM)
   argv[0]= <C:\_My Docs\SB\_Training\__Latest_2008_Training\Eclipse_Workspace\hello_args\Release\hello_args.sbx>
Hello World
DSP return=0
```

Figure 7. The Sandbridge IDE

The software tool set also consists of an optimizing ANSI C compiler. This C compiler performs many high performance optimizations, including the generation of saturation instructions from out of the box C code. This obviates the need to write assembly language code or to use machine specific intrinsics for saturation code. The compiler has proprietary semantic analysis techniques, which eliminate the need for intrinsics. A programmer writes C code in a processor independent manner and the compiler converts the C code into a dependence flow graph, analyzes the range of the arithmetic operations (specifically the sign bit) in the emulation code, propagates it across code segments, determines if it is a saturating or non-saturating operation and emits the correct assembly code.

Another important technique used by the compiler is the exploitation of SIMD instructions. The Sandbridge architecture uses SIMD instructions to implement vector operations. The compiler performs high performance inner and outer loop vector optimizations that use SIMD instructions to exploit the data level parallelism inherent in signal processing applications. These optimizations include vector load, store and multiply-add-reduce-saturate. In conjunction with loop optimizations, these provide very efficient and tight loops that can provide as many as 16 RISC operations in a single cycle.

Sandbridge has also developed an ultra-fast cycle counting accurate simulator, which improves the programmer productivity. The simulator uses architecture description of the underlying DSP and provides close to accurate cycle counts, but does not model the external memories or peripherals. However, the information provided by it is sufficient to develop the first executable version of an application. The simulator is based on Just-in-Time code generation technology, which has been developed in house.

The tool set also consists of an operating system kernel which provides access to the resources (multiple threads, peripherals, memories etc.) on the processor. This is provided via POSIX threads interface. Keeping with the philosophy of having a standard, user-friendly and efficient programming model, this interface is based on ANSI C and is commonly used in multi-threaded/multi-processor environment.

3.1 Serial to Parallel Code Development

Most users start programming algorithms using serial C code since parallel multithreaded coding can be complicated. This provides quick development but is not performance optimized.

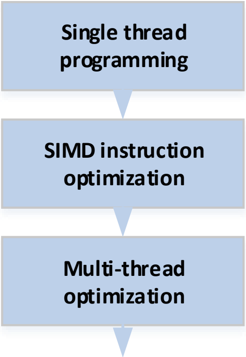

Single thread programming
SIMD instruction optimization
Multi-thread optimization

Figure 8. The development from serial to parallel codes

Figure 8 shows a serial to parallel code development flow which significantly lowers the threshold of parallel coding of programs. Using this development flow, users can start from their familiar serial code, and then gradually convert their algorithms into high performance parallel codes. The first optimization is the use of VLIW instructions. Instruction-level parallel operations are performed within each clock cycle. The second parallelization step is the use of intrinsic functions to execute SB3500 data-level SIMD operations. The third step is to transform the code to multithreaded code. The hardware has multiple threads that accelerate multithreaded software. However, to balance the processing workloads between successive threads, the users have to profile and balance the execution cycles and latencies between them, whether it is within a single core or across multiple cores.

3.2 Heterogeneous System Computing

As a further enhancement, the SB3500 also provides a heterogeneous computing environment composed of DSP and CPU cores with one ARM CPU and three SBX DSP cores inside its die. To aggregate all the computing capabilities, Sandbridge provides a hardware architecture and software tools optimized for heterogeneous computing. A single unified address space between all processors is provided along with a high-level C language compiler to program user code, and concurrent profiling and computing engines to optimize upper-layer applications, etc. In fact, this solution has already benefited most clients since it makes user-level programming much easier and shortens the time to market dramatically.

4. APPLICATIONS CASES

In this section, we present four application that are implemented by our clients in China: 1) LTE modem, 2) China Mobile operated IoT devices, 3) high-precision GPS/BD positioning terminals, and 4) China Grid sponsored BPLC terminals and 5) the video recognition CNN terminals.

4.1. Custom 3GPP LTE modems

Figure 9 shows a well-known 3GPP LTE modem. It can be extended in different special-purpose wideband wireless modems such as real-time image transmissions used for UAV communication systems.

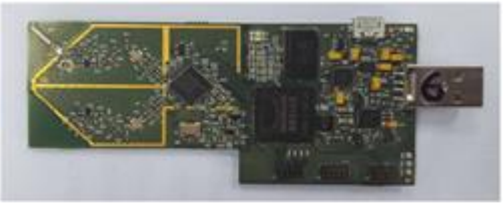


Figure 9. LTE module used in UAV

4.2. Narrow-band IoT devices

A new series of wide-area narrow-band IoT systems have been introduced into the 3GPP standards. Their associated communications, operations, devices, and several types of applications have emerged as a new wave of industrial products that are comparable to or even greater than the scale of the currently deployed 4G systems.

To meet these system requirements imposed by IoT applications, we have collaborated with distinguished Chinese companies to offer high-quality, low-cost NB-IoT terminal chips. These chips enable multimode communications, satellite positioning and artificial intelligence algorithms to execute simultaneously.

Figure 10. Prototype of NB-IoT terminal

4.3. GPS/BD positioning devices

GPS positioning functionality has become one of the most important services for smart terminals. However, to offer high-precision positioning information to Chinese clients we usually have two choice – one based on the world-wide GPS or the quickly expanding BD system. Consequently, we have developed code optimized for both GPS and BD receivers inside relevant SDR terminals for our clients.

Figure 11. Dual mode GPS/BD module

4.4. BPLC driven by China Grid

China Grid has launched an aggressive plan in China on energy Internet, to significantly promote their services to nation-wide industries and end users. However, to support such a project, the last-mile accesses from all devices as well as power meters must be guaranteed. Otherwise large portions of the grid could eventually fail. Hence China Grid released its latest broadband power line commutation (BPLC) standards in 2016 and later in this year, 2017, some promising BPLC chips emerged in the Chinese markets. Among them, the SDR based solutions have outperformed most of their competitors so far. In the following figure, we present some of the representative BPLC products that are developed using SB3500 chips that are scheduled to enter mass production in 2018.

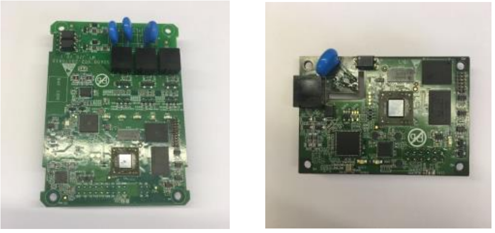


Figure 12. BPLC modules

4.5. Embedded AI accelerator

Recently, artificial intelligence services have shown promise in many IoT applications. In order to lower power consumption for radio transmissions as well as to minimize data throughput transferred from IoT terminals, we suggest that some embedded AI acceleration codes should be coupled closely with satellite positioning, wireless and wireline communication libraries that are present in typical IoT terminals. Using SB3500 DSP cores, our partners have developed their own AI SDK and its associated libraries. For example, the well-known CNN and RNN codes will be available for video and voice detection over SDR terminals. Below is an SDR development environment that is used for CNN acceleration purposes. It is scheduled to be released to the public in the first half of 2018.

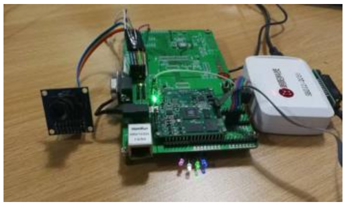


Figure 13. DEMO of embedded AI

5. CONCLUSIONS

In this paper, we have introduced a sophisticated SDR solution, the Sandblaster Sandbridge platform and analyzed its basic features. First, the Sandbridge SB3500 chips can be used for designs from the initial prototyping to the final mass production at competitive unit costs. Second, with the powerful heterogeneous computing architecture and a cycle-accurate simulator, users can easily develop and optimize their codes with a small team. Third, many types of reference designs have been released and mature products developed with the SB3500 are available including SIP/MCM packaged chips to small-size terminal modules. Finally, to support product developers, a specific SDR website for the public was developed proving technical support for all users of SB3500 chips.

6. REFERENCES

[1] M. Moudgill, J. Glossner, S. Agrawal, and G. Nacer, “The Sandblaster 2.0 Architecture and SB3500 Implementation”, Proceedings of the Software Defined Radio Technical Forum (SDR Forum '08), Washington DC, October, 2008.

[2] S. Jinturkar, V. Ramadurai, S. Shamsunder, M. Moudgill, J. Glossner, “Software Centric Approach to Developing Wireless Applications”, Proceedings of Software Defined Radio Technical Forum, Volume C, pp. 169-173, 16-18 November, 2004, Scottsdale, Arizona.

[3] http://www.eclipse.org/ as accessed on 9/3/2017.

DEEP LEARNING-BASED COGNITIVE RADAR SYSTEM FOR MICRO-UAS DETECTION AND CLASSIFICATION

Gihan J. Mendis (ijm11@zips.uakron.edu), Jin Wei (jwei1@uakron.edu), and Arjuna Madanayake (arjuna@uakron.edu)

The University of Akron, Akron, Ohio, USA

ABSTRACT

This paper proposes a low-cost cognitive radar system that exploits deep learning and 2.4 GHz continues wave Doppler radar sensing techniques for detecting and identifying micro unmanned aerial systems (micro UASs). The proposed architecture, employes the spectral correlation function (SCF), which is a Doppler radar-based method that has high resilience to environmental noise, to generate the unique pattern signatures for the individual micro UASs. Furthermore, a low-complexity binarized convolutional neural network (CNN) is designed to detect and identify the micro UASs by recognizing the SCF-based pattern signatures. By employing the low-complexity CNN, the computationally costly 617632 floating point multiplication operations required in the conventional CNN are represented by zero computational cost no connections, simple connections, negation operations, bit-shifting operations, and bit-shifting with negation operations. The simulations evaluate the performance of the proposed Low-complexity CNN in detecting and identifying the micro UASs by comparing the accuracy achieved by the proposed method with that is obtained by using the conventional CNN.

1. INTRODUCTION

Micro unmanned aerial systems (micro-UASs) are remotely controlled or autonomous small scaled aerial systems. They are low-cost devices with small dimensions and weights and also referred as "drones". In recent years, micro-UASs are becoming increasingly popular among hobbyist and also for a variety of applications such as performance art, aerial photography and video, search and rescue mission, precision agriculture, mapping and surveying, and package delivering [1]. Despite these useful applications, misuses of micro-UASs are also becoming a concerning threat to the public. Micro UASs have been reported to interfere with aircraft [2, 3]. A micro-UAS was crashed at the White House, raising concerns about security risks [4]. The micro-UASs can also be used as spying devices which violate the privacy of civilians and the government and private organization.

Radar sensing is one of the most effective methods adopted by defense mechanisms for detecting aerial systems. However, because of the small size and slow moving speed, the micro UASs are undetectable by the conventional radar systems designed to detect larger and fast-moving aerial systems. As one type of non-radar techniques, acoustic signal processing has been widely used in detecting micro-UASs [5]. Acoustic signal processing-based techniques are effective but not resilient to environmental noise. Another existing non-radar technique is to detect the radio frequency signals that are used for the remote control of the micro-UAS [6]. This method may not be appropriate for fully autonomous micro-UASs. Some of the other existing non-radar strategies include video-based detection and thermal based detection [6]. Recently some radar-based systems have also been proposed for detecting and identifying micro-UASs. In [7], Drozdowicz *et al.* discussed an experiment conducted for the detection and tracking of micro-UASs using a radar system. In [8], Shin *et al.*, proposed a K-band radar system with fiber-optic links for detecting micro-UASs. In [9], Jahangir *et al.* used 2-D L-Band receiver arrays to detect micro-UASs. In this method, decision tree machine learning technique was utilized to reject other targets. In our previous work, we proposed a low-cost 2.4 GHz continuous-wave Doppler radar sensor built using commercially available RF components along with a signal processing mechanism that uses spectral correlation function (SCF) to generate noise-resilient and distinguishable 2-D signature patterns and deep belief network (DBN) based classifier for detection and identification of micro-UASs [10, 11].

Deep learning methods are artificial neural network (ANN) based machine learning techniques having multiple layer hierarchies of ANNs. These techniques are more effective in extracting hierarchical features from raw data [12]. Deep learning methods have been used for pattern recognition in various application areas [13–17]. Convolutional neural networks (CNNs) are one of the most successful deep learning techniques inspired by the neuron arrangement of the visual cortex of mammals [18]. CNN-based methods are widely used for image classification tasks including radar signature analysis [19–22].

The main challenge of implementing deep learning methods is the high computation-complexity that increases the power and area cost of digital implementations for deep learning based classifiers. High computation-complexity is a result of the high number of floating-point multiplications operations. In our previous work, we proposed a multiplierless low-complexity DBN with direct mapping to binary logic circuits. In this work, we use a multiplierless CNN-based classifier for detecting and identifying micro-UASs SCF patterns. In [23], Lin *et al.* proposed a binarization method for backpropagation algorithm which produces binary weights $\{-1, 0, 1\}$. In this paper, we adopt this method to realize the multiplierless low-complexity CNN.

In the following section, we provide an overview of the proposed system. Sections 3, 4, and 5 briefly discuss the proposed radar sensor, SCF pattern generation, and the low-complexity CNN, respectively. The conducted experiment and the results obtained are summarized in Section 6. In Section 7, the conclusions and future work are presented.

2. OVERVIEW

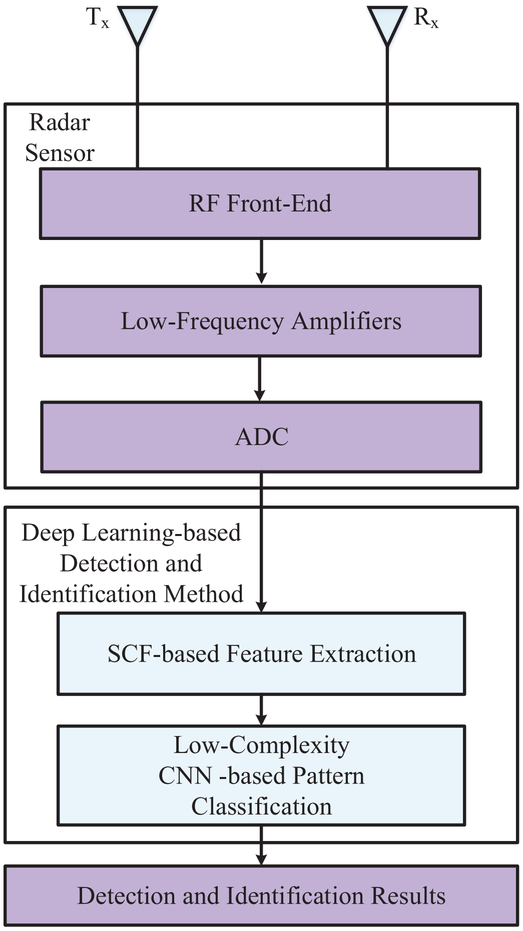

Tx
Rx
Radar Sensor
RF Front-End
Low-Frequency Amplifiers
ADC
Deep Learning-based Detection and Identification Method
SCF-based Feature Extraction
Low-Complexity CNN -based Pattern Classification
Detection and Identification Results

Figure 1: Overview of the proposed system.

As shown in Fig. 1, the proposed system consists of the radar sensor and the deep learning-based detection and identification method. The radar sensor is formed with a Doppler radar RF front-end, low-frequency amplifiers, and an analog to digital converter (ADC) system. Relevant low-frequency Doppler shifts are amplified with low-frequency amplifiers and ADC is used to sample the signals for further digital signal processing. Deep learning-based detection and identification method contains a spectral correlation function (SCF)-based feature extraction method followed by the low-complexity CNN-based pattern classifier, which classifies the generated SCF patterns to detect and identify micro UASs presence within the radar beam.

3. RADAR SENSOR

Figure 2(a) illustrates the block diagram of the radar sensor presenting the details of the radar front-end. The front-end of the radar sensor is a Doppler radar system for capturing frequency shifts induced on reflected electromagnetic waves. 2.4GHz continues-wave interrogation waveform is transmitted through the transmitter Tx and the reflected waves are captured by the receiver Rx. The moving propellers of micro-UASs cause mechanically induced phase modulations on the reflected signal that produces Doppler shifts.

A bandpass filter is used to filter out the noise in the received signal through Rx. Low noise amplifiers (LNAs) are used to boost the filtered received signal. The amplified signal is mixed with in-phase (I) and quadrature (Q) components of the transmitted signal. A phase-shifter (90-degree hybrid) is used to generate the I and Q components of the transmitted signal. The I/Q based method is employed to eliminate the possible effects of null points. In order to extract the induced Doppler shifts, lowpass filters are applied on the mixer output signals. Low-frequency amplifier stage with lowpass filtering of 100 Hz cut-off frequency is used to extract and boost the useful Doppler frequency shifts for micro-UAS detection and classification. ADC is used to sample and record the boosted signals for further processing. The practical implementation of the radar sensor is shown in Fig. 2(b). Doppler radar front-end is implemented using commercially available RF components, low-frequency amplifiers are implemented with low-cost analog components. National Instruments data acquisition unit is used as the ADCs.

4. SCF SIGNATURE PATTERNS

Cyclic Autocorrelation Function (CAF) is defined to quantize the amount of correlation between different frequency shifted versions of a given signal and represent the fundamental parameters of their second order periodicity [24]. CAF is calculated as follows:

$$R_x^\alpha[l] = \left[\lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]x^*[n-l]e^{-j2\pi\alpha n}\right] e^{-j\pi\alpha l} \tag{1}$$

Where $x[\cdot]$ is the given signal and $\alpha = m/T_0$ is the cyclic frequency, when T_0 is the process period, and m is an integer. Spectral correlation function (SCF) is the Fourier transform of CAF, f the temporal frequency of the given signal SCF is calculated as follows:

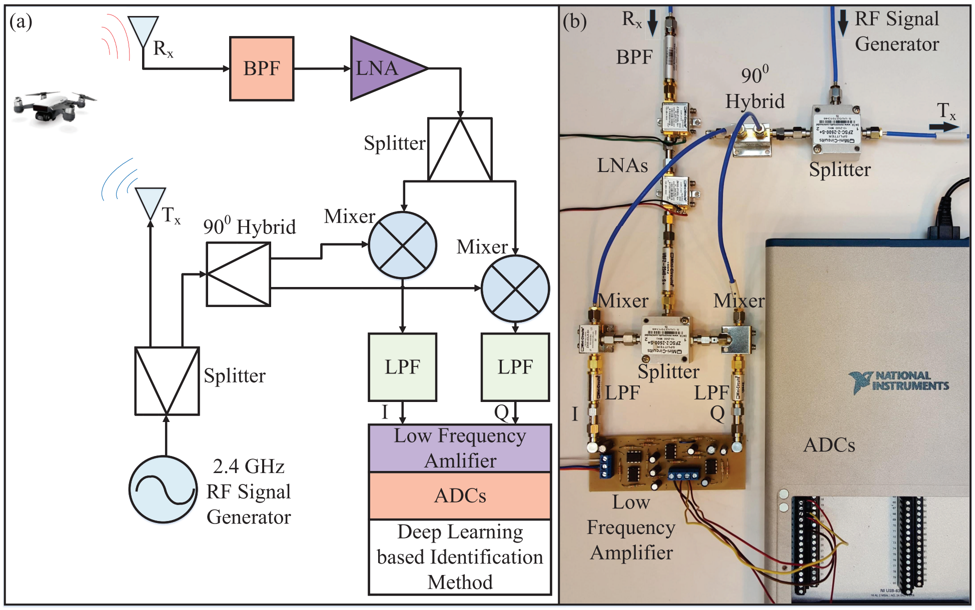


Figure 2: (a) System architecture of our proposed deep learning-based AMC method; (b) Implementation of radar sensor.

$$S_x^\alpha[f] = \sum_{l=-\infty}^{\infty} R_x^\alpha[l] e^{-j2\pi f l} \tag{2}$$

For signals that have different modulation schemes, it has been shown that the SCF provides unique peak profiles because of their modulation types [24]. Because that the propeller motions of micro UASs induce the Doppler effect mechanically, which can be considered as the phase modulation characterizing the unique physical properties of different types of micro UASs, it is possible to observe a distinguishable peak profile on SCF of the extracted Doppler radar signal for each type of micro UASs [25].Another advantage of using the SCF patterns is since the SCF suppresses stationary features, our method is resilient to stationary impairments such as additive white Gaussian noise (AWGN) [24].

5. LOW-COMPLEXITY CNN

As shown in the Fig. 3, the CNN designed in our work consists of 3 convolution layers, 2 pooling layers, a fully connected layer with rectifier linear units (ReLU), and a softmax-based output layer. The inputs to the CNN are 2D images having the size of 24×24. The first convolution layer evaluates 32 features with 5×5 kernel size. Maximum pooling is performed after the first convolution layer with the kernel size of 2×2, which reduces the image size to 12×12. The second convolution layer evaluates 32 features with 5×5 kernel size along with 2 maximum pooling, which reduces the size of the image to 6×6. The third convolution layer evaluates 32 features with 2×2 kernel size. The outputs of the 32 kernels of the third convolution layer are reshaped and combined to form a vector of the size $6 \times 6 \times 32 = 1152$.

Table 1: The floating-point multiplication operations required in CNN.

Layer	Floating point Multiplications
Convolution layer 1	$24 \times 24 \times 5 \times 5 \times 32 = 460800$
Convolution layer 2	$12 \times 12 \times 5 \times 5 \times 32 = 115200$
Convolution layer 3	$6 \times 6 \times 2 \times 2 \times 32 = 4608$
Fully connected ReLU	$1152 \times 32 = 36864$
Fully connected softmax	$32 \times 5 = 160$
The Number of Required Multiplications	617632

A fully connected layer with 1024 ReLU units is added on top along with a softmax layer for classification.

If the weights of the convolution layers and fully connected layers remain as floating-point numbers, the total number of floating point multiplication operations required to perform in a single iteration of testing is shown in Table 1. We assume the number of class labels as 5. Since floating point multiplication is computationally expensive in digital logic and the number of the total multiplications required for the CNN is very high, the deployments of the above CNN becomes a hardware-expensive task. By modifying the backpropagation algorithm of the CNN as shown in Table 2, we replace the floating-point weights of the CNN by using five possible values $-2^p, -1, 0, 1, 2^p$, where p is a positive integer. By doing so, we reduce the hardware-expensive floating-point multiplications to the operations that are much less costly in the digital hardware as shown in Table 3.

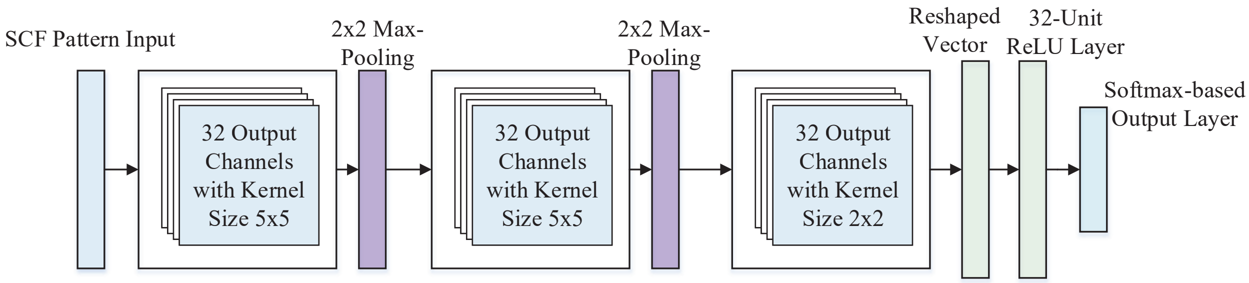

SCF Pattern Input
32 Output Channels with Kernel Size 5x5
2x2 Max-Pooling
32 Output Channels with Kernel Size 5x5
2x2 Max-Pooling
32 Output Channels with Kernel Size 2x2
Reshaped Vector
32-Unit ReLU Layer
Softmax-based Output Layer

Figure 3: The structure of CNN.

Table 2: The training algorithm for updating our low-complexity CNN.

Operators and functions:

$.\geq$: the elementwise more than or equal comparison of two matrices.

$.\times$: the elementwise multiplication of two matrices.

$y = sign(x)$: if $x < 0$, $y = -1$, else $y = 1$.

$y = absolute(x)$: if $x < 0$, $y = -x$, else $y = x$.

$\mathbf{Y} = rand(\mathbf{X})$: randomly assigns $y_{ij} \in [0, 1]$ and $dim(\mathbf{Y}) = dim(\mathbf{X})$.

$y = cast(x)$: if $x = true$, $y = 1$, else $y = 0$.

$\mathbf{W} = backprop(\mathbf{W}, f)$: applies the gradient descent based backpropagation algorithm to fine-tune the weight matrix $\mathbf{W}$, where f is a batch of training data.

$f = nextbatch(\mathbf{F}, batchsize)$: returns the next batch of training data according to batch size, where $\mathbf{F}$ is the training data set.

$\mathbf{W_c} = clipping(\mathbf{W}, L)$: clips the element values of weight matrix $\mathbf{W}$ to be in the range $[-L, L]$ where L is a predetermined scalar.

Inputs: L-clipping level, $\mathbf{W}$-initial weight matrix , $\mathbf{F}$-training data, $\mathbf{T}$-labels corresponding to training data, N-number of training iterations.

Output: $\mathbf{W_b}$-binarized weight matrix ($w_{ij} \in \{-1, 0, 1\}$)

Steps:

```
For $epoch \leqslant N$
    $f = nextbatch(\mathbf{F}, 50)$
    $\mathbf{W} = backprop(\mathbf{W}, f)$
    If $(mode(epoch, 100) = 0)$
        $\mathbf{W_c} = clipping(\mathbf{W}, L)$
        $\mathbf{S} = sign(\mathbf{W_c})$
        $\mathbf{P} = absolute(\mathbf{W_c})/L$
        $\mathbf{T} = \mathbf{P}. \geq rand(\mathbf{P})$
        $\mathbf{W_b} = cast(\mathbf{T}). \times \mathbf{S}$
    End
End
```

Table 3: Digital logic mapping of multiplications with low-complexity weights.

Weight Value	Mapping
0	No connection
1	Connection
-1	Negation
2^p	Right shift by p bits
-2^p	Right shift by p bits and negation

6. EXPERIMENT AND RESULTS

The proposed radar sensor is set up in the laboratory environment and the experiment is conducted using four types of micro-UASs. First, the Doppler radar sensor is operated with the micro-UASs whose positions are fixed in front of the radar beam. The time series data collected from the ADC are used to generate a set of reference SCF patterns that is used as the pattern signature for the scenario in which no micro-UASs presented. Then micro-UASs are clamped in front of the radar beam by a distance of 3 meters from the transmitter and receiver antennas to a reasonable far-field approximation at the radar frequency of 2.4 GHz. The time series are collected while the propellers of each micro-UAS are in motion. The collected time series are used to generate a set of SCF pattern signatures for each type of micro-UAS. Figure 4 shows the micro-UASs used in the experiment and Fig. 5 shows the example SCF patterns for reference and for each micro-UAS. Furthermore, in our experiment, SCF patterns are generated using a MATLAB Communications System Toolbox functions on experimentally collected data [26].

Pre-processing and training

The gray-scale images of the SCF patterns are resized to be 48×48 images. Fast Fourier transform (FFT) based method is used for image scaling. A 2-dimensional (2D) FFT operation is implemented on the original grayscale images and a 48×48 pixel square is selected from the center of the FFT transformed image. Then inverse 2D FFT is performed to achieve the scaled down image. By doing so, high-frequency components of the original image are filtered out, and thus high-frequency noise

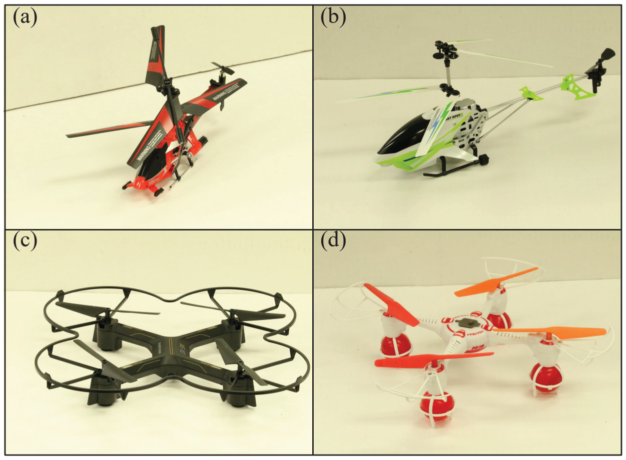


Figure 4: Micro-UASs used in the experiment; (a) Type 1; (b) Type 2; (c) Type 3; (d) Type 4.

is removed from the scaled down image. Considering the symmetry of the patterns, a quarter of the pixels from the resized images is used as the input for the low-complexity CNN classifier. Therefore, the input size of the low-complexity CNN is 24 × 24.

The low-complexity CNN is trained using 1000 SCF pattern data that includes 200 patterns corresponding to each micro-UAS and the reference. As a comparison, a conventional CNN, which has the same structure but uses floating-point accurate weights, are also trained with the same 1000 training data. Another 50 patterns from each category are used to evaluate the performance of the CNN. The classifiers based on low-complexity and conventional CNNs are implemented using TensorFlow APIs [27]. In the simulation, we set the number of iterations as 1000 and the batch size to be 20. At each 100th iteration of backpropagation training, binarization is performed for the low-complexity CNN. The simulation results are illustrated in the following subsection.

Results

Tables 4 and 5 present the confusion matrices for the TensorFlow implementations of conventional and low-complexity CNNs, respectively. In Tables 4 and 5, the rows represent the actual class that each testing SCF pattern belongs to and the columns are the classes identified by using the conventional and low-complexity CNNs.

The overall detection accuracy of micro-UASs by using conventional CNN based classifier is 98%, and that achieved by using the low-complexity CNN based classifier is 96.5%. The rates of false alarm for the conventional and low-complexity CNNs are 0.5% and 2%, respectively. Figure 7 compares the performance of the these two considered CNNs in classifying micro-UASs.

Based on the simulation results shown in Figs. 6 and 7, we can observe that the low-complexity CNN achieves comparable accuracy in detection and identification of micro UASs compared with that achieved by conventional CNN. Furthermore, our proposed CNN outperforms the conventional CNN in low computational complexity. Overall, our proposed CNN achieves a good tradeoff between the performance and the computational complexity.

Table 4: Classification of SCF patterns for micro-UAS detection and identification using TensorFlow implementation of conventional CNN.

Actual	Classification from CNN				
Pattern	Type 1	Type 2	Type 3	Type 4	Ref.
Type 1	48	0	2	0	0
Type 2	0	48	0	1	1
Type 3	0	0	50	0	0
Type 4	0	0	0	50	0
Ref.	1	0	0	0	49

Table 5: Classification of SCF patterns for micro-UAS detection and identification using TensorFlow implementation of low-complexity CNN.

Actual	Classification from Low-Complexity CNN				
Pattern	Type 1	Type 2	Type 3	Type 4	Ref.
Type 1	46	0	2	0	2
Type 2	0	48	1	0	1
Type 3	0	0	50	0	0
Type 4	0	0	0	49	1
Ref.	1	1	0	0	48

7. CONCLUSION

In this paper, we propose a deep learning-based cognitive radar system for detecting and identifying micro-UASs by using SCF function and low-complexity CNN method. The proposed system consists of a low-cost radar sensor and a digital signal processing subsystem that exploits the noise resilient SCF to generate unique signature patterns of the individual micro-UASs and uses the low-complexity CNN to classify the patterns. Our proposed low-complexity CNN has the advantage of containing no multipliers while a conventional CNN with the same structure requires performing 617632 floating-point multiplication operations. As illustrated in the simulation results, although the low-complexity CNN shows lower accuracy compared with the conventional CNN the accuracy of detection and identification is are acceptable especially considering the low computational cost.

During the experiments for the work in this paper, we kept the micro UASs immobile and leverage the Doppler shifts induced by propeller movements. In our future work, we plan to conduct the experiments by using moving micro UASs and we also plan to set up the radar system for real-time detection of micro-UASs.

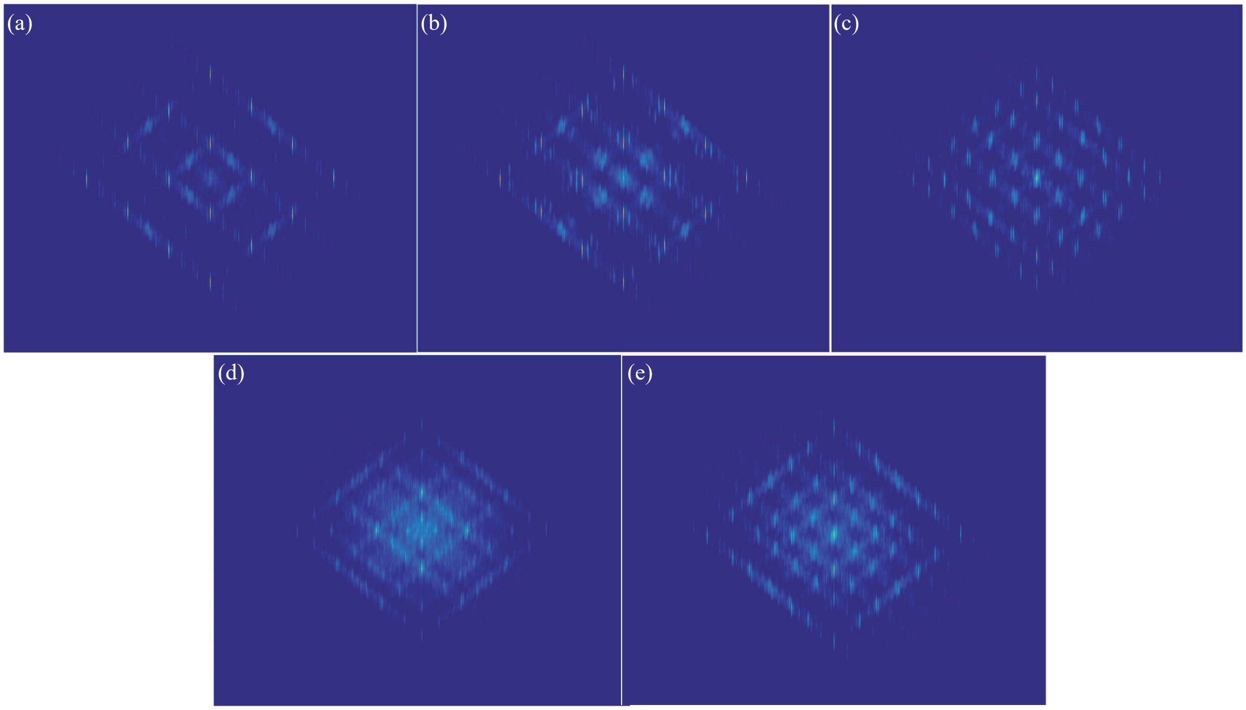


Figure 5: Example SCF patterns for (a) Reference (When there is no micro-UAS); (b) Type 1; (c) Type 2; (d) Type 3; (e) Type 4.

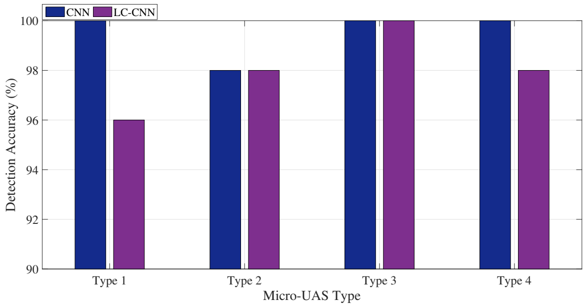


Figure 6: Comparison between the accuracies in detecting each type of micro-UAS achieved by using the conventional CNN and low-complexity CNN.

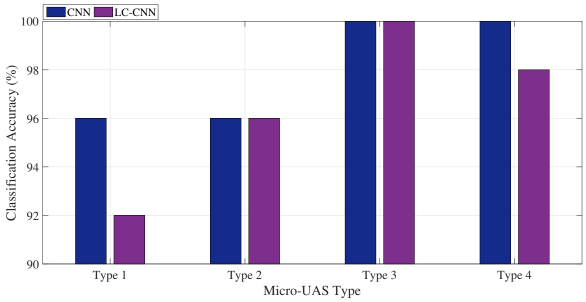


Figure 7: Comparison between the accuracies in identifying each type of micro-UAS achieved by using the conventional CNN and low-complexity CNN.

REFERENCES

[1] Z. Liu, Z. Li, B. Liu, X. Fu, I. Raptis, and K. Ren, "Rise of mini-drones: Applications and issues," in *Proceedings of the 2015 Workshop on Privacy-Aware Mobile Computing*. ACM, 2015, pp. 7–12.

[2] P. McGreevy, "Private drones are putting firefighters in âĂŹimmediate danger,âĂŹ california fire official says," LA Times, August 2015.

[3] J. Serna, "Lufthansa jet and drone nearly collide near lax," LA Times, March 2016.

[4] B. Jansen, "Drone crash at white house reveals security risks," USA Today, January 2015.

[5] "Droneshield," https://www.droneshield.com/how-droneshield-works.

[6] https://www.helpnetsecurity.com/2015/05/28/drone-detection-what-works-and-what-doesnt/.

[7] J. Drozdowicz, M. Wielgo, P. Samczynski, K. Kulpa, J. Krzonkalla, M. Mordzonek, M. Bryl, and Z. Jakielaszek, "35 GHz FMCW drone detection system," in *2016 17th International Radar Symposium (IRS)*, May 2016, pp. 1–4.

[8] D. H. Shin, D. H. Jung, D. C. Kim, J. W. Ham, and S. O. Park, "A distributed FMCW radar system based on fiber-optic links for small drone detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 2, pp. 340–347, Feb 2017.

[9] M. Jahangir and C. Baker, "Robust detection of micro-UAS drones with L-band 3-D holographic radar," in *2016 Sensor Signal Processing for Defence (SSPD)*, Sept 2016, pp. 1–5.

[10] G. J. Mendis, T. Randeny, J. Wei, and A. Madanayake, "Deep learning based doppler radar for micro UAS detection and classification," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, Nov 2016, pp. 924–929.

[11] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning cognitive radar for micro UAS detection and classification," in *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, June 2017, pp. 1–5.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[15] A. Venkataraman, "Deep learning algorithms based text classifier," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, July 2016, pp. 220–224.

[16] Q. Weng, Z. Mao, J. Lin, and W. Guo, "Land-use classification via extreme learning classifier based on deep convolutional features," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 704–708, May 2017.

[17] K. Matsumoto, Y. Tajima, R. Saito, M. Nakata, H. Sato, T. Kovacs, and K. Takadama, "Learning classifier system with deep autoencoder," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 4739–4746.

[18] D. D. Cox and T. Dean, "Neural networks and neuroscience-inspired computer vision," *Current Biology*, vol. 24, no. 18, pp. R921–R929, 2014.

[19] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, Jan 1997.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[21] J. Yang, Y. Zhao, J. C. W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2016, pp. 5079–5082.

[22] Y. Kim and T. Moon, "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, Jan 2016.

[23] Z. Lin, M. Courbariaux, R. Memisevic, and Y. Bengio, "Neural networks with few multiplications," *arXiv preprint arXiv:1510.03009*, 2015.

[24] W. A. Gardner, A. Napolitano, and L. Paura, "Cyclostationarity: Half a century of research," *Signal processing*, vol. 86, no. 4, pp. 639–697, 2006.

[25] T. Randeny, "Multi-dimensional digital signal processing in radar signature extraction," Master's thesis, The University of Akron, 2015.

[26] "P25 spectrum sensing with synthesized and captured data," https://www.mathworks.com/help/comm/examples/p25-spectrum-sensing-with-synthesized-and-captured-data.html.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

Spectrum Data Mining: Measurement-Driven Insights for Sustainable Spectrum Management

Amir Ghasemi
Communications Research Centre Canada
Email: firstname.lastname@canada.ca

***Abstract*—Communications Research Centre Canada (CRC) has deployed a network of spectrum sensors with high time and frequency resolution for continuous monitoring and analysis of spectrum activity in different bands. In the present work, the radio frequency (RF) measurement data collected over a long period of time is used to characterize the spectrum usage of incumbent users in Land Mobile Radio (LMR) bands. Statistical analysis of measured data indicates daily and weekly spectrum usage patterns, especially in the channels used by public safety users. The identified activity patterns are then leveraged to predict future spectrum usage, thereby enabling a data-driven proactive approach to spectrum assignment and management, where spectrum shortage and oversupply across different networks can be predicted and managed. We further investigate correlations between spectrum activity and external factors such as severe weather and special events. The findings confirm existence of such correlation and provide further insights for dynamic spectrum assignment in Land Mobile Radio bands.**

I. Introduction

Land Mobile Radio (LMR) systems are often used by government agencies, municipal services, and a variety of commercial users to enable (mostly push-to-talk) voice communications among a group of users. These systems are critical, especially in public safety and emergency response operations, where multi-cast voice communications (e.g., between first responders and a dispatch centre) are needed.

Typically, frequency channels in the LMR bands are assigned to different licensees based on the information received from the users when they apply for a licence (e.g., transmitter location, antenna height, emitted power, etc.). In addition, regulators primarily have to rely on predefined models for the (voice) traffic load and propagation loss to estimate the impact of a potential assignment on other users. In the absence of measurement data showing where, when, and how different parts of the spectrum are being used, regulators have to be more cautious in reusing the spectrum in order to avoid interference among different radio systems.

To address the knowledge gap described above, Communications Research Centre Canada (CRC) has developed a prototype spectrum monitoring system which, using multiple sensors distributed over a geographical area, continuously monitors the spectrum usage in selected frequency bands below 6 GHz, and uploads the collected measurement data to the cloud for further processing and analysis [1]. Since its deployment in Ottawa, Canada in February 2016, this system has gathered terabytes of data about spectrum usage at different locations, times, and frequencies, thereby enabling data-driven approaches to spectrum assignment, sharing, and interference resolution [2].

The work presented herein describes preliminary research toward a better understanding of spectrum usage, and some of the underlying factors driving it, by exploiting historical RF data collected by CRC's spectrum monitoring system. The exploratory analysis is focused on a subset of these measurements in the 800 MHz LMR band (specifically the public safety subband at 866-869 MHz). As we will illustrate later, LMR spectrum usage is non-uniform across different spectral ranges and exhibits time and frequency patterns that could be exploited to improve users' timely access to the spectrum. This paper's focus is on temporal characterization of these bands and exploring the existence of correlations between intensity of spectrum usage and external factors, such as special events or severe weather.

The remainder of this paper is organized as follows. Section II describes the spectrum measurement setup for the bands of interest in this study. Section III discusses the spectrum occupancy analysis and prediction, as well as some external factors affecting the spectrum usage. Finally, conclusions are presented in Section IV.

II. Spectrum Measurement System

CRC has developed a spectrum monitoring system designed to provide detailed spectrum usage information for research purposes. The prototype includes a wide range of both fixed and mobile sensors covering different frequency ranges to provide a continuous and wideband monitoring capability below 6 GHz [1]. The system's sensors are networked and their near real-time RF measurements are reported to a cloud-based processing entity for subsequent retrieval and analysis work.

As the focus of the present work is on the time and frequency dimensions, spectrum data used for the analysis was collected from a single, fixed monitoring station. The exact measurement location is shown in Fig. 1 (denoted by the red marker). Operationally, this sensor is a FFT-based receiver covering 138 MHz to 952 MHz

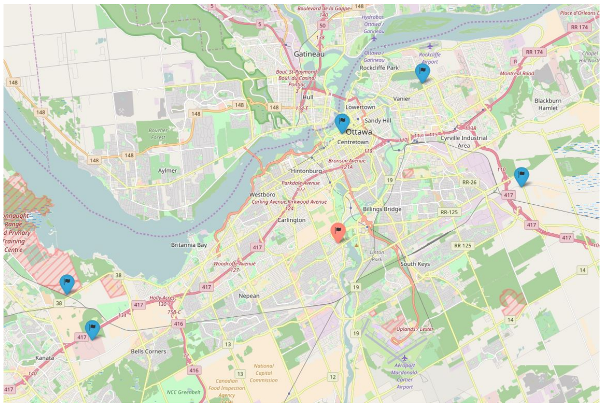


Fig. 1. Location of fixed measurement sites covering the LMR bands

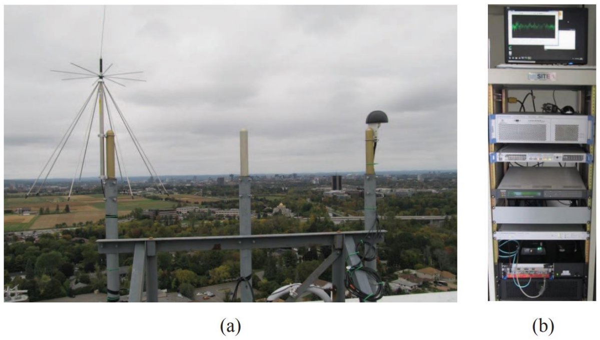


Fig. 2. View of (a) line-of-sight from measurement antenna towards downtown and (b) spectrum monitoring station equipment.

with a fast digital sweep and an instantaneous bandwidth of 24.76 MHz. For the bands of interest, it uses an omnidirectional and vertically polarized wideband discone antenna installed at a height of 80 m above ground on the roof of a building approximately 5 km south of downtown Ottawa.

Fig. 2 shows the view from the antenna towards downtown Ottawa. For the measurements used in this study, the average band sweep time was approximately 0.3s with a resolution bandwidth of 1.984 kHz. Such a sweep time provides sufficient time resolution to capture a variety of LMR traffic patterns. For each measured frequency channel, the detection power threshold was set to a margin over the estimated noise floor to account for temporal noise variations. The threshold was set using the method in [3] based on deeming a probability of false alarm of 10^{-5} to be the maximum acceptable error.

Fig. 3 shows a 5-minute snapshot of high-resolution spectrum measurements for downlink public safety LMR band (866-869 MHz) in Ottawa. Most radio systems operating in this band are trunking systems providing voice communications to a variety of public safety and municipal users. The continuously active channels in Fig. 3 (i.e. the uninterrupted horizontal lines) typically represent the control channels of trunked radio systems used to transmit control signalling required for the network's operation. The 0.3s time resolution of the measurements for each channel allows the temporal activity pattern of voice channels to be accurately characterized.

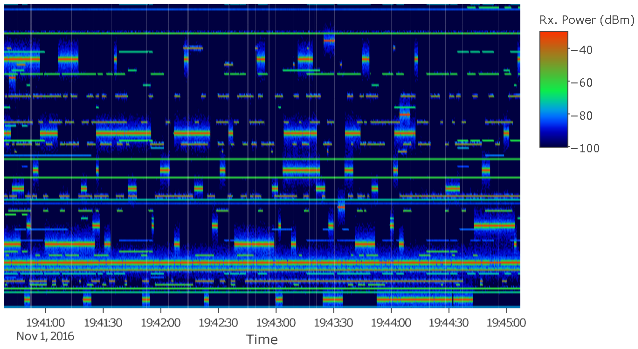


Fig. 3. A 5-minute snapshot of the received-power levels for LMR channels in 866-869 MHz.

Fig. 4 shows a 10-hour snapshot (approximately 120,000 samples) of the temporal pattern of idle (burst-off) and busy (burst-on) states for a typical public safety voice channel. The empirical distributions of idle and busy state durations as well as log-normal fits for them (obtained via maximum likelihood estimation) are also shown in Fig. 4. We tried fitting a variety of common distributions such as exponential, log-normal, weibull, and pareto to the empirical data and found out that the log-normal distribution typically provided the best fit for trunked voice channels with many users, corroborating the results of [4].

Such data-driven modelling of incumbent users' spectrum activity patterns allows for more realistic simulation studies of spectrum sharing opportunities as we have shown in [2]. In the present work however, we focus on longer-term spectrum usage patterns which offer insights enabling a more proactive approach to spectrum management.

III. Spectrum Occupancy Analysis

We define spectrum occupancy as the percentage of time a frequency channel is detected to be in use with its received power being above the detection threshold as defined earlier. In this section we describe some of the occupancy patterns observed and how these might be leveraged for spectrum assignment. To facilitate the analysis over longer time periods, we generate aggregate *hourly* statistics from the high resolution measurement data. These include the hourly received-power histogram, idle and busy state duration histograms, and hourly occupancy of each channel, among others. Given the large size of high-resolution data (as shown in Fig. 3), continuous production of hourly statistics is performed on an Apache Spark cluster [5].

Figs. 5 and 6 show two typical plots of the hourly occupancy versus time for different LMR ranges for a

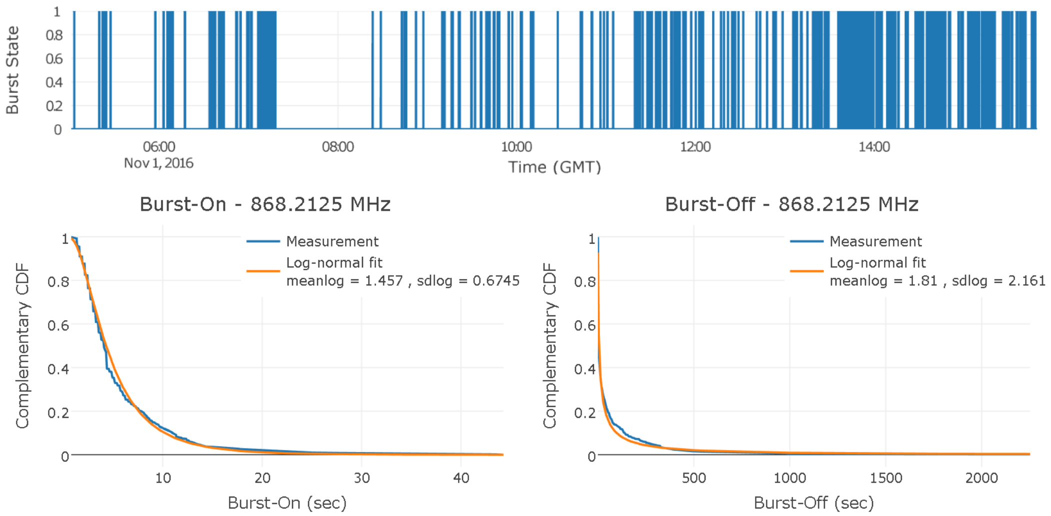


Fig. 4. Top: Channel idle (burst-off) and busy (burst-on) states pattern for a public safety voice channel over a 10-hour period; Bottom: Empirical distribution and log-normal fit for the idle and busy state durations

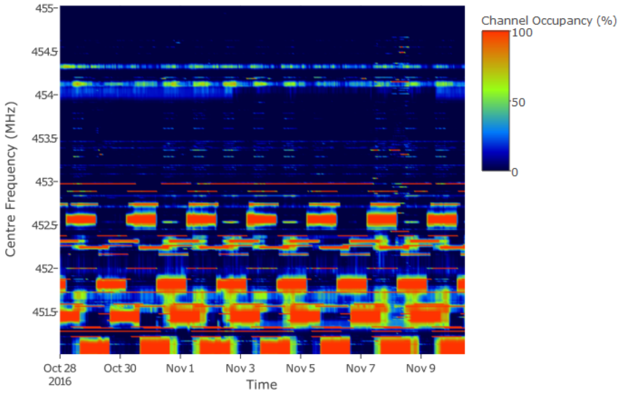


Fig. 5. Hourly occupancy of LMR channels in 451-455MHz range over a period of two weeks.

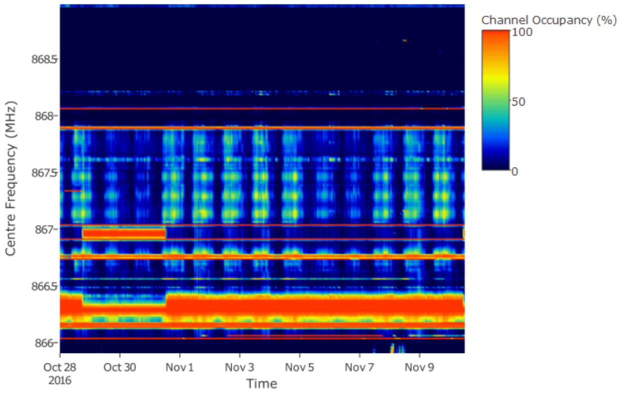


Fig. 6. Hourly occupancy of public safety LMR channels in 866-869MHz range over a period of two weeks.

2-week period. For display clarity, each plot has been limited to 4 MHz which still covers a large number of channels occupied by diverse LMR technologies and services, including commercial, public safety, government, and municipal public service users. The plots show that spectrum utilization is not uniform across the LMR bands, with some portions congested, while other portions are being either lightly used or completely unused. Some LMR traffic is observed to exhibit a periodic occupancy pattern that is repeated on a daily and weekly basis, with lower usage observed at night and on weekends. Such trends arise from social behaviour and habits and corroborate patterns observed for LMR use in prior work [6]. We also observe that in the Ottawa area, many LMR channels are not in use most of the time, both in the 450 MHz band and in the upper range of the 800 MHz band.

A. *Usage Pattern Seasonality and Prediction*

Inspection of the spectrum occupancy patterns in Fig. 6 suggests that hourly occupancy of LMR voice channels is both daily and weekly seasonal. Further analysis of the auto-correlation function (acf) for the hourly occupancy time-series, shown in Fig. 7, confirms this observation as the acf oscillates between its peaks and minima every 12 hours. Therefore, time series forecasting models such as seasonal ARIMA may be used to predict future occupancy patterns for these channels. For the aforementioned channels however, the augmented exponential smoothing of [7] was found to perform better. Unlike conventional seasonal ARIMA models, this model supports multiple seasonal periods (e.g., both daily and weekly patterns in our case) and allows the seasonality to vary slowly with time for a better fit.

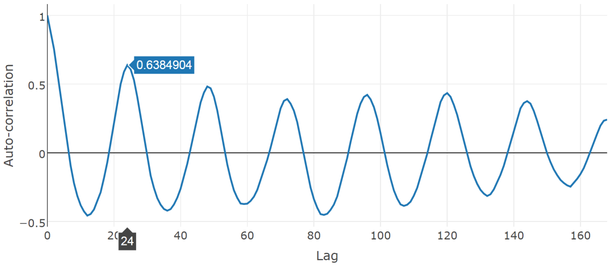


Fig. 7. Auto-correlation function of a typical LMR public safety voice channel.

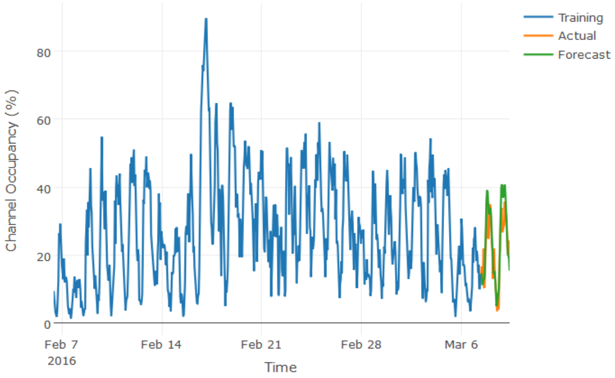
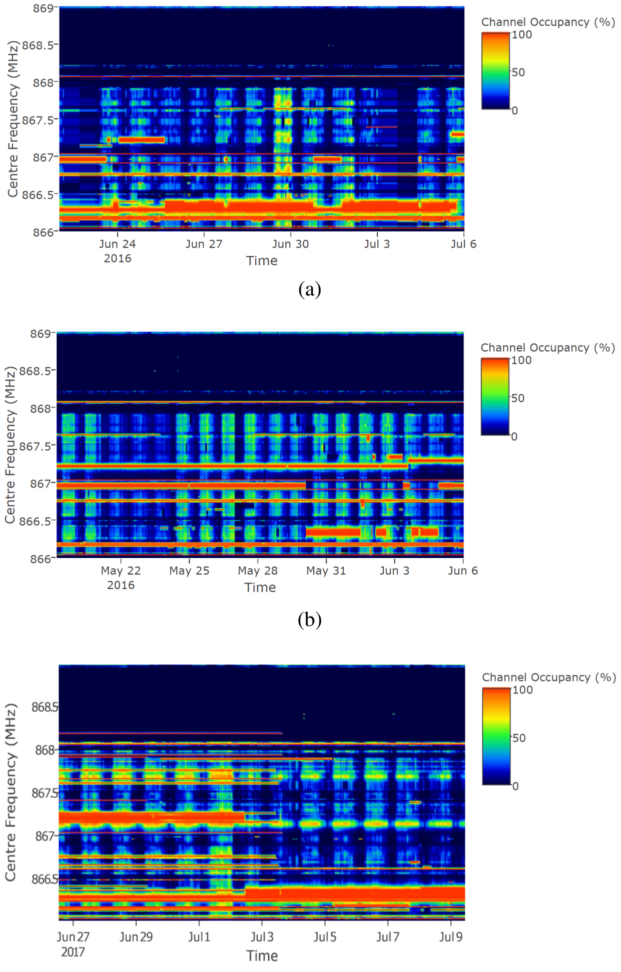


Fig. 8. 48-hour spectrum occupancy prediction for 867.8875MHz.

Fig. 8 shows an example of spectrum occupancy prediction using the above method for one of the public safety voice channels in Ottawa. In this case, one month of measurement data has been used as training to successfully predict the occupancy pattern over the next 48 hours. Having the ability to predict spectrum occupancy enables a more dynamic approach to spectrum assignment where spectral resources are assigned based on forecasted demand as opposed to over-provisioning for the worst-case scenario. For such predictive analytics to be viable however, it has to be able to account for occasional surges in the demand due to various external factors as we will discuss next.

B. *Impact of External Factors*

While daily and weekly patterns can be successfully used to model spectrum occupancy of most public safety LMR channels, there are occasional anomalies which can not be explained by the aforementioned seasonal effects. In many cases, such anomalies are due to external factors such as major events which drive up the communications needs of public safety users. Fig. 9 shows three examples of such anomalies: a) president of the United States visiting Ottawa on June 29, 2016, (b) Ottawa ”Race Weekend” which includes many running races spread over two days (May 28-29th, 2016) and is the biggest event of its kind in Canada, (c) Canada Day on July 1st, 2017, which coincided with Canada’s 150th birthday and drew very large crowds.



(a)

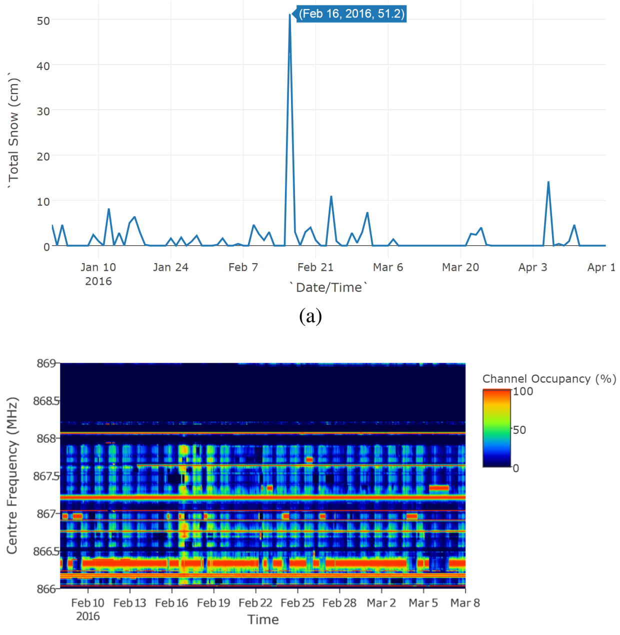
(b)

(c)

Fig. 9. Impact of special events on public safety LMR spectrum usage: (a) a visit by the US president (b) Ottawa Race Weekend (c) Canada Day 2017

In all three cases, there were many road closures, changes to traffic patterns, and concentration of people in certain areas of the city core requiring extensive coordination and communications among public safety users. As illustrated in Fig. 9, there is a marked increase in spectrum occupancy compared to similar weekday or weekends for most public safety LMR channels. Fig. 9b for instance shows that the Race Weekend event on May 28-29th increased weekend spectrum usage almost to the level of a typical weekday. In case of Canada Day, with this being the largest Canada Day event in the history, there were extensive preparations in the days leading to Saturday, 1st of July, which is mirrored in increased occupancy for some LMR channels in the last week of June as seen in Fig. 9c.

Severe weather is another external factor affecting public safety spectrum demand. Fig. 10 shows how a snowstorm on February 16, 2016 (more than 50cm of snowfall in the span of 24 hours) triggered a few days of increased spectrum occupancy for Ottawa public



(a)

(b)

Fig. 10. Impact of severe weather on public safety LMR spectrum usage: (a) Total daily snowfall reported at Ottawa International airport (b) Hourly occupancy of Ottawa public safety LMR channels in 866-869MHz range

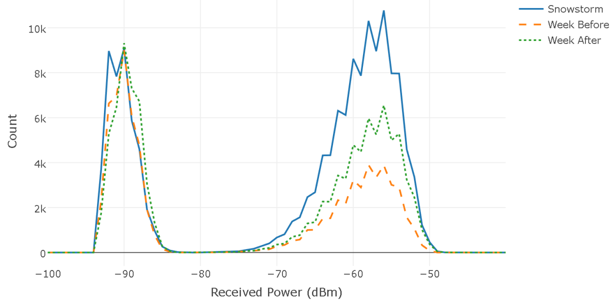


Fig. 11. Histogram of received power during a severe weather event (snowstorm) compared with those of regular days

safety users. A similar observation was reported in [9], where the spectrum occupancy of police channels over regular days was compared with the occupancy during a blizzard. It is worth noting that features of weather data other than total daily snowfall shown in Fig. 10 may be more important. Specifically, we have observed instances of increased public safety LMR spectrum usage on days with only 3-5cm snowfall (which is fairly typical for Ottawa during winter). Further inspecting other weather attributes indicated blowing snow and low visibility which had resulted in numerous emergencies. Therefore, care must be taken when fusing spectrum measurements with other sources of data to include the most relevant features.

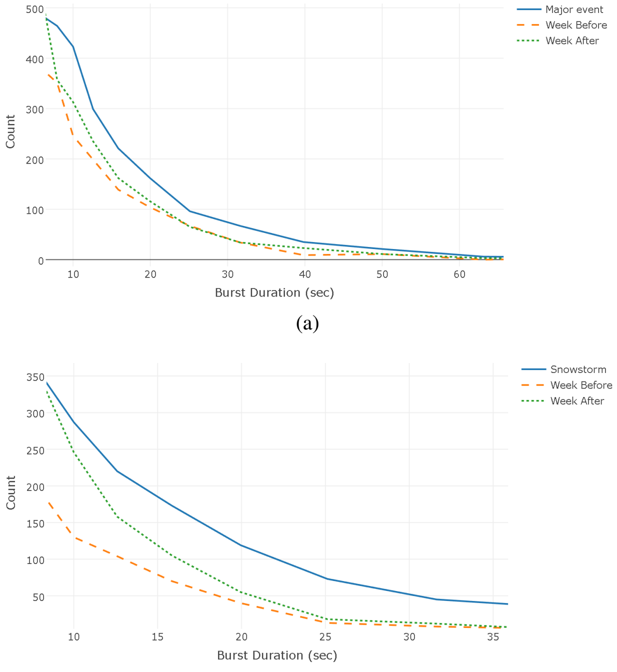
Authors in [8] also have studied the impact of major events on spectrum usage. In particular they modeled the *received-power* on a given LMR channel with a Gaussian Mixture Model (i.e. a mixture of Gaussian random variables a.k.a. GMM) and then showed that a model can be trained to distinguish event days by comparing the estimated parameters of the GMM model with those obtained on regular non-event days. This approach is predicated on there being changes to the received-power footprint due to big events. In the example presented in [8], the LMR channels belong to a transit company which runs additional buses and modified routes during major game events which can explain the variations in the received-power distributions. For many LMR channels though, the location of transmitters (and therefore shape of the received-power distribution) does not change from one day to another. This is illustrated in Fig. 11 which compares the measured daily histogram of received-power during a severe weather event (snowstorm) with those of regular days (7 days before and after). Given the two licensed transmitters on this channel are at fixed locations, their average received-power levels and shape of the distribution observed at the sensor do not change significantly. The intensity of the histogram however clearly indicates unusual activity for one of the transmitters due to severe weather. Therefore, the GMM approach used in [8] may need to be modified to use other *features* of the measurement data (e.g., occupancy as shown in Fig. 10).

Fig. 12 further shows an example of the impact of a major event or severe weather on the underlying characteristics of public safety users' activity. In particular, the right tail of the channel busy-duration histogram indicates a larger number of longer bursts which ultimately leads to a higher percentage of spectrum occupancy (as observed in Figs. 9 and 10b). Understanding the impact of external factors on the distribution of channel busy and idle durations allows more accurate modeling of spectrum sharing scenarios, where decisions on sharing a channel would depend on the characteristics of the gaps in spectrum activity [2].

IV. Conclusion

In this work, we presented preliminary exploratory analysis of LMR spectrum measurement data collected in Ottawa, Canada, which was shown to have distinct time and frequency patterns which could be used to train predictive models. We further illustrated existence of correlations between spectrum occupancy of public safety users and external factors, such as major events and weather.

The ability to predict future spectrum occupancy while taking into account other sources of data, such as weather forecast and upcoming events, allows a more proactive approach to spectrum assignment. Under such a paradigm, capacity bottlenecks are predicted and managed, before they occur, with minimum human



(a)

(b)

Fig. 12. Tail of the histogram for burst (channel busy) duration compared on regular days with that of (a) a major event (U.S. president visit) (b) a severe weather event (snowstorm)

intervention. As the size of spectrum measurement data grows, so does our understanding of the fundamental factors driving the demand for various users and services, thereby enabling us to transition toward a paradigm where spectrum shortage and oversupply across different networks can be predicted and managed proactively.

Results presented herein were based on data obtained from a single sensor. Given the availability of spatial data from the network of sensors shown in Fig. 1, future work will explore the spatial properties of the measurements with potential applications to data-driven propagation models, interference management, and spectrum sharing.

Acknowledgment

The author would like to thank Kareem Baddour, Paul Guinand, and Humphrey Rutagemwa for helpful discussions and feedback during the course of this work.

References

[1] L. Li *et al.*, "A cloud-based spectrum environment awareness system," *in proc. IEEE International Symposium on Personal, Indoor, Mobile Radio Communications (PIMRC)*, Montreal, Canada, October 2017.

[2] H. Rutagemwa, K. E. Baddour, A. Ghasemi, "Spectrum sharing opportunities in land mobile radio bands: A data-driven approach," *in proc. IEEE PIMRC Workshop on Cognitive Radio and Innovative Spectrum Sharing Paradigms for Future Networks*, Montreal, Canada, October 2017.

[3] S. Wang, F. Patenaude, R. Inkol, "Upper and lower bounds for the threshold of the FFT filter bank-based summation CFAR detector," *in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, pp. 289-292, May 2006.

[4] T. Taher, R. Bacchus, K. Zdunek, D. Roberson, "Empirical modeling of public safety voice traffic in the land mobile radio band," *in. Proc. Int. Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, 2012.

[5] Apache Spark: Lightning-fast cluster computing, https://spark.apache.org/

[6] T. Taher, R. Bacchus, K. Zdunek, D. Roberson, "Long-term spectral occupancy findings in Chicago," *in Proc. IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DYSPAN)*, pp. 100- 107, June 2011.

[7] A. M. De Livera and R. J. Hyndman, "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513-1527, 2011.

[8] A. Abdallah *et al.*, "Detecting the Impact of Human Mega-Events on Spectrum Usage," *13th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, Las Vegas, January 2016.

[9] T. Taher, R. Bacchus, K. Zdunek, D. A. Roberson, "Dynamic spectrum access opportunities for public safety in land mobile radio bands," *in proc. Int. Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, June 2011.