SECURE INTELLIGENT RADIO FOR TRAINS (SIRT)

by

K R Damindra Savithri Bandara
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Information Technology

Committee:

Dr. Duminda Wijesekera, Dissertation Director

Dr. Paulo Costa, Committee Member

Dr. Paul Ammann, Committee Member

Dr. Daniel Menasce, Committee Member

Dr. Stephen Nash, Senior Associate Dean

Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date: 4/11/17

Spring Semester 2017
George Mason University
Fairfax, VA

Secure Intelligent Radio for Trains (SIRT)

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

K R Damindra Savithri Bandara
Master of Science
George Mason University, 2014
Bachelor of Science
University of Peradeniya, Sri Lanka, 2009

Director: Dr. Duminda Wijesekera, Professor
Department of Computer Science

Spring 2017
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to my loving husband Erool and to my dear father and mother.

Acknowledgments

I am very grateful to my committee Dr. Duminda Wijesekara (Chair), Dr. Paulo Costa, Dr. Paul Ammann and Dr. Daniel Menasce for their support, advice, and encouragement during my research. My advisor, Dr. Duminda Wijesekara, has had a profound influence on my development as a researcher. Dr. Wijesekara has always challenged me to look at every last detail and ask the toughest questions. He also provided me with various resources to achieve my career and research goals as well as professional growth. I have found working with him very rewarding. My co-advisor Dr. Paulo Costa guided me with important technical aspects of my dissertation work and his suggestions significantly improved my final defense presentation.

My research work is funded by the Federal Railroad Administration Grant FR-TEC-0010-2015. I would like to thank my points of contact in the Federal Railroad Administration Dr. Mark Hartong and Mr. David Blackmore.

I would like to thank my group mates Andre Abadie, Tony Melaragno and Satish Kolli for providing me with ideas and support in the collaborative work we did at the RARE lab. I would also like to thank Dr. Andy Powell for providing me with valuable support in my preparation for the final defense presentation. Finally, I would like to thank my parents, my husband, my parents in law, my sister, my brother, my grandmothers and all my friends for their unconditional support and patience during this work.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

PTC Positive Train Control

WIU Wayside Interface Unit

CR Cognitive Radio

SDR Software Defined Radio

SNR Signal to Noise Ratio

BER Bit Error Rate

IDS Intrusion Detection System

QPSK Quadrature Phase Shift Keying

BPSK Binary Phase Shift Keying

CE Cognitive Engine

CRC Cyclic Redundancy Check

QoS Quality of Service

IDS Intrusion Detection System

MCE Master Cognitive Engine

SCE Spectrum Management Cognitive Engine

CCE Cryptographic Cognitive Engine

QPSK Quadrature Phase Shift Keying

8PSK 8 Phase Shift Keying

16QAM 16 Quadrature Amplitude Modulation

64QAM 64 Quadrature Amplitude Modulation

BPSK Binary Phase Shift Keying

SIRT Secure Intelligent Radio for Trains

USRP Universal Software Radio Peripheral

TESLA Time Efficient Stream Loss-tolerant Authentication

IV Initialization Vector

OFDM Orthogonal Frequency Division Multiplexing

CCJPA Capitol Corridor Joint Powers Authority

FRA Federal Railroad Administration

WIU Wayside Interface Unit

AAR Association of American Railroads

Abstract

SECURE INTELLIGENT RADIO FOR TRAINS (SIRT)

K R Damindra Savithri Bandara, PhD

George Mason University, 2017

Dissertation Director: Dr. Duminda Wijesekera

Positive Train Control (PTC) is a radio-based control system designed to ensure safe navigation of trains. Safety objectives of PTC are to avoid train to train collisions, train derailments and ensure railroad worker safety. According to the published specifications of Interoperable Electronic Train Management System (I-ETMS), the on-board PTC controller communicates with two networks; the Signaling Network and the Wayside Interface Unit (WIU) network to gather navigational information such as the positions of other trains, the status of critical infrastructure and any hazardous conditions along the train path. PTC systems are predicated on having a reliable radio communication network. Secure Intelligent Radio for Trains (SIRT) is an intelligent radio that is customized for train operations with the aim of improving the reliability and security of the radio communication network. SIRT system can (1) operate in areas with high spectrum congestion, different noise levels and interference conditions, (2) withstand jamming attacks, (3) improve data throughput and (4) detect threats and improve communication security.

My work includes (1) Analyzing the PTC system to identify communication constraints and vulnerabilities, (2) Designing SIRT to overcome them, (3) Developing a prototype of SIRT using Software Defined Radios and (4) Testing it under varying channel conditions, noise levels and attackers. My experiments show that SIRT dynamically chooses the best modulation schemes based on the channel noise level and switches channels in response to channel jamming. Also, it changes cryptographic key values using a scheme like Lamport scheme and detects replay and forgery attacks with an accuracy more than 93%.

Chapter 1: Introduction

Before 2008 most train lines within the USA depended on the human crew to obey safety rules when operating trains. They relied on voice radios to receive authorities to enter a block and used the same voice radio to inform when they left a block. According to the accident reports of the Federal Railroad Administration (FRA) [5], a significant portion of accidents during this time was due to human errors. Therefore starting from the 1990's the FRA invested on designing a safety system called Positive Train Control (PTC), which is a communications-based control system for scheduling trains. Safety objectives of PTC are to avoid train to train collisions, train derailments and provide railroad worker safety.

After a major train accident which happened in Chatsworth, California [6] in 2008, the US Congress passed a Railway Safety Improvement Act (RSIA-2008) which mandated that PTC should be implemented in all the train lines by 2015. However due to various reasons the PTC implementation deadline has been extended to 2020.

PTC system consists of trains with radio-based communication capabilities and two independent networks. The two networks are,

1. WIU Network: Which broadcasts the status of critical infrastructures such as switches and broken rail detectors.

2. Signaling Network: Which is responsible for communication permissions for trains to occupy track segment and enforce speed limits.

These two networks together provide information such as speed restrictions on track segments, location of other trains and switch positions (e.g. if a switch is open to the track) that are deemed necessary for the train to travel in a way that ensures safety requirements. PTC is designed so that if either network detects a potential threat, the PTC radio network

provides warnings to oncoming trains so that the train can take precautionary action. For the communication between the trains and the two networks the PTC system uses a Software Defined Radio (SDR) based radio system.

1.1 The Problem Statement

The safety of PTC systems depend on having a reliable radio communication network, although the system itself is designed as a fail-safe real-time distributed control system. One challenge is the proper management of the radio spectrum when the PTC network is expanded nationwide. If the spectrum is not correctly allocated there can be interference within the PTC network. This can lead to message collisions and reduce the throughput. Another problem is the usage of static radio parameters, which makes the radio communication susceptible to long-term interference and jamming. Also, using static parameters reduce the effective data throughput.

Therefore to ensure the reliability of PTC communication, the radio network should guarantee the following requirements. PTC radios need to,

R1: operate in areas with high train congestion and high channel congestion.

R2: operate in noisy and interference environments.

R3: maintain connectivity and improve data throughput.

R4: withstand jamming attacks.

R5: detect forgery and replay attacks.

1.2 Thesis Statement

It is possible to design a Software Defined Radio based intelligent radio communication network satisfying the requirements R1, R2, R3, R4 and R5.

1.2.1 Attainment of R1

This dissertation provides a comprehensive frequency analysis to identify the spectrum congestion issue for PTC communication (presented in Chapter 3). Based on these results I desgined a frequency planning architecture to reduce spectrum congestion. This includes the following.

1. **An extension to the current PTC protocol to dynamically allocate channels based on train density, channel availability, and train priority:** This channel allocation protocol is designed as part of the radio architecture and is tested using a prototype (presented in Chapter 4). In this prototype, I have tested scenarios such as signaling point communicating with two trains at the same time, inter signal point communication and train handover from one signaling point to another. A holistic approach to test the dynamic channel allocation algorithm may require moving radios or may need a combination of simulation and prototype testing.

2. **An algorithm to cluster WIUs based on the distance and propagation conditions:** WIUs in each cluster may connect to a single transmitter and transmit their status interleaved in time (presented in Chapter 3). One limitation with this algorithm is that it does not consider the type of critical infrastructure. This provides a more realistic solution to the problem because the frequency of sending WIU Status messages depend on the type of the critical infrastructure.

1.2.2 Attainment of R2, R3, R4

SIRT radio architecture is designed to fulfill the requirements R2, R3, and R4. SIRT continuously monitors the spectrum and reconfigures modulation and frequency to obtain optimal throughput in noisy environments and withstand jamming attacks. SIRT's throughput at noise levels less than -35 dB and between -18 dB to -20 dB is better than the current PTC radio communication. During the noise levels of -35 dB to -20 dB throughput is relatively

similar. However, this throughput can be improved if SIRT gets the spectrum measurements directly from the hardware. Noise level greater than -18 dB is considered jamming. In a jamming event, SIRT can hop the frequency to a different frequency and maintain connectivity. This work is presented in Chapter 4. The Software Defined Radio based prototype implementation is presented in Chapter 5.

In my prototype, I have tested the modulation changing between Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and 8 Phase Shift Keying (8PSK) modulations. The prototype could be improved to add higher order modulation schemes such as 16 Quadrature Amplitude Modulation (16QAM) and 64 Quadrature Amplitude Modulation (64QAM) to achieve better throughput.

My testbed has a couple of limitations. One of the limitations is that it does not have enough precision to vary the noise levels. More precision would allow me to capture the variations of CRC errors and the synchronization message losses more precisely. The second limitation is that the data collection was done in one-minute intervals. Therefore, if noise introduction time and data collection time do not coincide precisely, the data shows average error rate before and after the introduction of noise. Also, it measures the spectrum quality in the software. More precise measurement can be taken if there is a hardware probe to get the direct spectrum measurements from the Analog to Digital Converter.

1.2.3 Attainment of R5

Secure Intelligent Radio for Trains (SIRT) dynamically changes cryptographic materials based on an algorithm that uses Lamport scheme [7]. SIRT also has an Intrusion Detection System (IDS) to identify cryptographic attacks. This work is presented in Chapter 4.

Current SIRT design uses a threat detection module that was developed by one of my collaborators, Anthony Melaragno. However, SIRT has the capability to work with different cryptographic modules. Current threat detection is done by observing the packet content. This can be improved to use multiple data sources such as GPS location, and geolocation [8, 9]. Additionally, an accurate timing source is needed to be integrated into the radio

transceiver. I noticed that timing had a direct effect on the quality of detection, due to time-based cryptographic key changing.

1.3 Contribution

Published research in the area of using cognitive radios is extensive in dynamically allocating licensed spectrum for unlicensed users (so-called secondary users). There is very limited or no research known to me done in using software defined radio functionality to improve the reliability and security of radio communication for licensed users. To the best of my knowledge, there is even limited research done in improving the reliability and security of PTC communication. Also, most of the research work related to radio parameter reconfiguration (such as frequency hopping and adaptive modulation) are either theoretical analysis, numerical estimations based on models or simulations using tools such as Matlab, Simulink, and NS2 [10–13]. Prototype developments in this area are limited. In my opinion, prototypes provide first-hand data that augment or validate theoretical estimates and simulated value in realistic scenarios and come closer to real-life deployments. In my work, I developed a working prototype using software defined radios to validate the ideas. Therefore the work presented in this thesis is unique and opens doors for many areas of future research.

Following are the main contributions presented in this thesis.

A comprehensive frequency analysis to check how the spectrum congestion can affect PTC operations: This work was published in [14]. The paper explains how the PTC spectrum gets congested when the train density and speed increases. It presents the advantages of different modulation schemes, different channel bandwidths and dynamically allocating channels to reduce spectrum congestion.

A frequency channel allocation architecture: This architecture is designed to allocate frequency channels between the signaling network and the WIU network to reduce spectrum congestion. This work is presented in [15].

Design a radio architecture to improve the reliability and security of PTC operations:

1. The preliminary work that I did to identify the use of risk engines for PTC operations is presented in [16].
2. The architecture of SIRT is given in [17].

Prototype development and testing:

1. Development of the SDR's lower layers for message transmission and reception using GNURadio and Ettus USRPS. Details given in [18].
2. Prototype development and testing of the spectrum management cognitive engine is presented in [19].
3. Prototype development and testing of Master Cognitive engine.

1.4 Organization

The organization of the rest of this dissertation is as follows. Chapter 2 provides background and related work in PTC, network planning for trains and intelligenet radio technologies. Chapter 3 presents frequency analysis that I did to understand the PTC spectrum congestion problem and frequency planning architecture that I designed. Chapter 4 presents the design and architecture of SIRT radio system. In Chapter 5, I present the prototype development and testing on the proposed system. Finally in Chapter 6, I summarize the main conclusions.

Chapter 2: Background and Related work

Section 2.1 provides the background and related work of PTC. Section 2.2 describes work related to wireless network planning from trains. Section 2.3 describes the work related to intelligent radios.

2.1 Background on Positive Train Control System

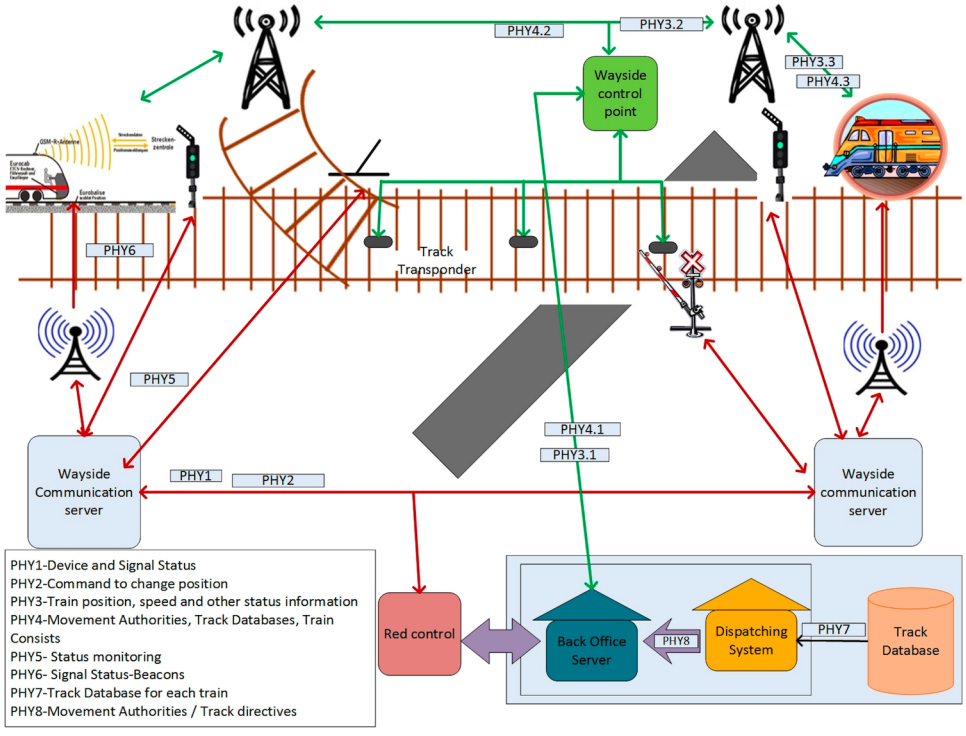

Figure 2.1: PTC Architecture

Figure 2.1 shows the architecture of the proposed PTC system. As shown in the top half of the diagram, on-track train movements are governed by authorities communicated through a system of networks connecting the back offices that are in charge of managing tracks. This network is referred as the *Signaling Network*. The logical connectivity of this network is shown as *green* lines. In addition to this envisioned PTC system, existing track mounted external, and in-cab signals provide movement authority and track condition notifications, including but not limited to switch positions using the *Wayside Interface Unit Network (WIU)* shown at the bottom of Figure 2.1 as *red* lines. In areas with in-cab signaling, the red network may use wireless communications, track mounted sensors (as in Amtrak's Northeast Corridor) or provide wired external signals. In either case, train movements can be controlled using signals and existing voice-based radio communication. The dependency of voice-based communication is expected to lose its importance with the full implementation of PTC as a vital system.

Figure 2.2: In-cab Signal Display [2]

2.1.1 The WIU Network

The Wayside Interface Unit (WIU) network consists of wayside interface devices that broadcast status of critical infrastructures such as switches, broken rail detectors or flood detectors. They periodically broadcast the device status to trains and also send a copy of the status messages to the back office. The WIU Status message numbers are 5100 and 5101 and the format is shown in Table 2.1 [20].

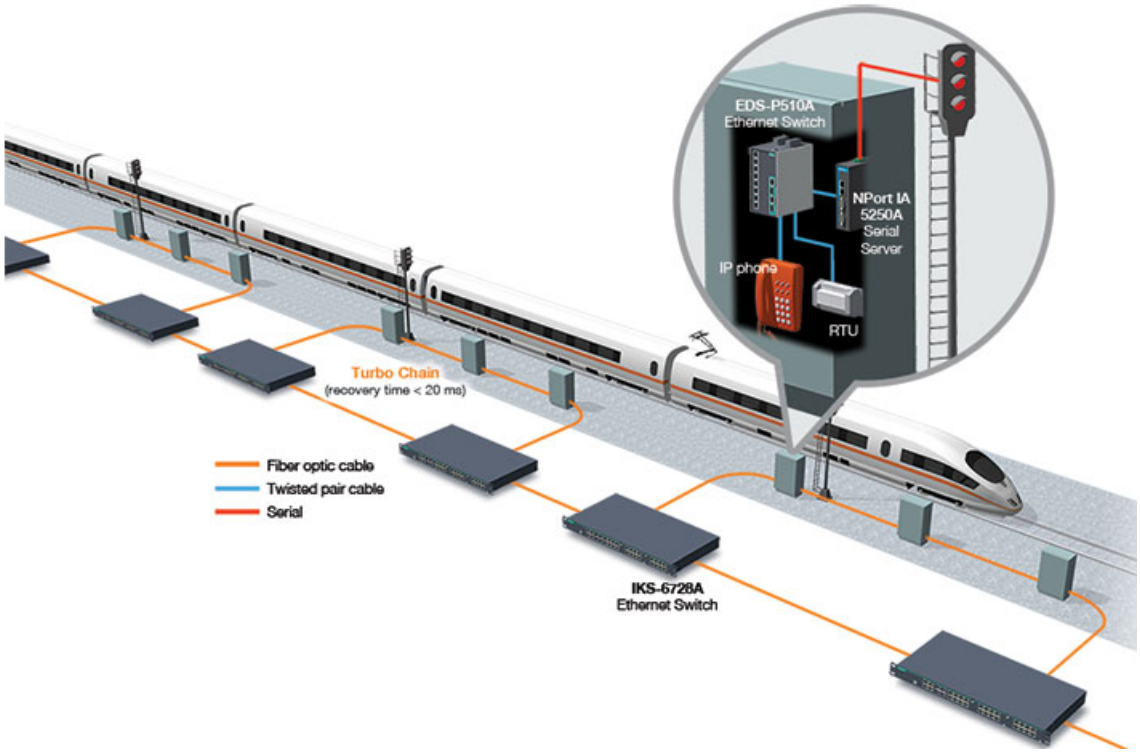

Figure 2.3: The WIU Network [3]

There are two types of messages sent from a train to the WIU. They are,

- **Beacon Request:** Request to begin periodic transmission of *WIUStatus* messages. Message type number is 5200. The message format is similar to *WIUStatus* message, but exclude message payload fields [20].
- **GetWIUStatus:** Request to return a WIUStatus immediately. Message type number is 5201. The message format is similar to WIUStatus message but excludes message payload fields [20].

Table 2.1: WIU Status Message Format

Field	Size (bits)	Description
WIU address	40	ATCS Type address
Beacon TTL	1	Beacon expiration
Vital message type	6	Defined by WIU
Vital message version	5	
mod 16 times	4	Modified time stamp
Message sequence number	8	0-255 binary
Device status	1-1944 bits	Generated by WIU
VDIV	32	HMAC

2.1.2 The Signaling Network

The signaling Network provides two-way communication between a train and the back office. The train should receive the authorization from a so-called signaling point to enter the next block. A signaling point is a dedicated radio that connects the train's on-board radio with the back office's signaling radio system. Signaling point communicates with the dispatching center (which resides in the back office) to obtain the dispatching details for the requesting train. Furthermore, the signaling network relays message communication between the train and the back office such as sending the train location reports to the back office and sending train consists from the back office to the train. The message format is shown in Table 2.2. There are two types of messages.

- Messages from the train to the back office - Message type from 01000 to 01123
- Messages from the back office to the train - Message type from 02000 to 02122 [21]

Table 2.2: Signalling Message Format

Field	Size (bits)	Description
Protocol version	1	Version of EMP header
Message Type(ID)	2	
Message version	1	
Flags	1	Time stamp Format, No encryption, No compression, ...
Data Length	3	
Message Number	4	Application Message Sequence Number
Message Time	1	Message creation time
Variable header size	1	Defined by length of source and destination addresses
Time to Live	2	Time to Live
Routing QoS	2	Quality of Service
Source	64 max	
Destination	64 max	
Data Integrity	4	Truncated Keyed Hashed Message Authentication Code.

2.1.3 PTC Controller

Messages received from the WIU and signaling networks are shown on the on-board display unit in the locomotive. The PTC controller accesses the risks of proceeding based on these received messages. If either of the networks indicates that there is a risk and if the operator fails to react, the PTC controller changes the train movements to avoid the risk. Some examples are given below.

- Signaling network indicates that the next block is occupied. If the train operator fails to apply brakes before the safe braking distance, the PTC controller assesses that there is a higher risk because the train can collide with another train or an obstacle

in the next block and applies brakes to stop the train.

- A WIU beacon indicates the train to slow down before it enters a switch. The controller monitors the operator action to ascertain if he/she applies brakes to reduce the speed to the specified speed and if not applies brakes to slow down the train.

The trains need to have active communication channels for both networks to facilitate this design. The onboard PTC controller assesses information supplied by the two networks continuously and displays warnings to the train operator using a display unit onboard the locomotive. The warning is sent with sufficient time for the operator to understand the risk and apply brakes to avoid it. This is shown in the *warning curve* (curve in yellow) in Figure 2.4. Then the PTC controller monitors if the operator applies brakes. If the operator does not apply brakes on time, the PTC controller automatically does apply brakes to control the train speed or stop the train. The *braking curve* is shown in Figure 2.4 (curve in red).

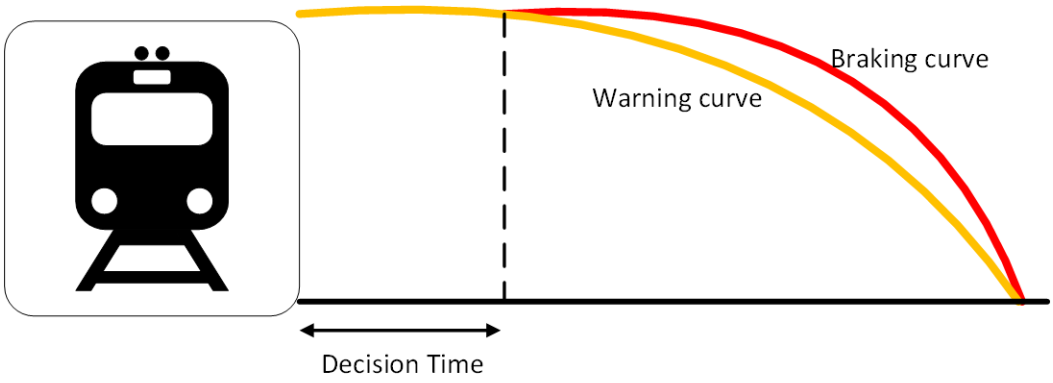

Figure 2.4: PTC Braking Curves

2.1.4 SDR based PTC Communication Network

PTC equipped trains and the controlling devices uses a SDR network to exchange messages. The class 1 railroads outsourced the development of PTC radio system to Meteorcomm LLC. that custom builds SDRs for PTC communications [22]. These SDRs consists of software modules that replace traditional hardware components such as filters, modulators/demodulators constructed in a general purpose computer. SDRs facilitates flexible design and manufacturing of radio components and changing components conveniently. Furthermore, SDRs allow signal processing components to change its radio communication parameters such as power levels, frequencies, and modulation schemes dynamically. However, PTC radios do not have any intelligence to utilize this flexibility in adapting to different environments.

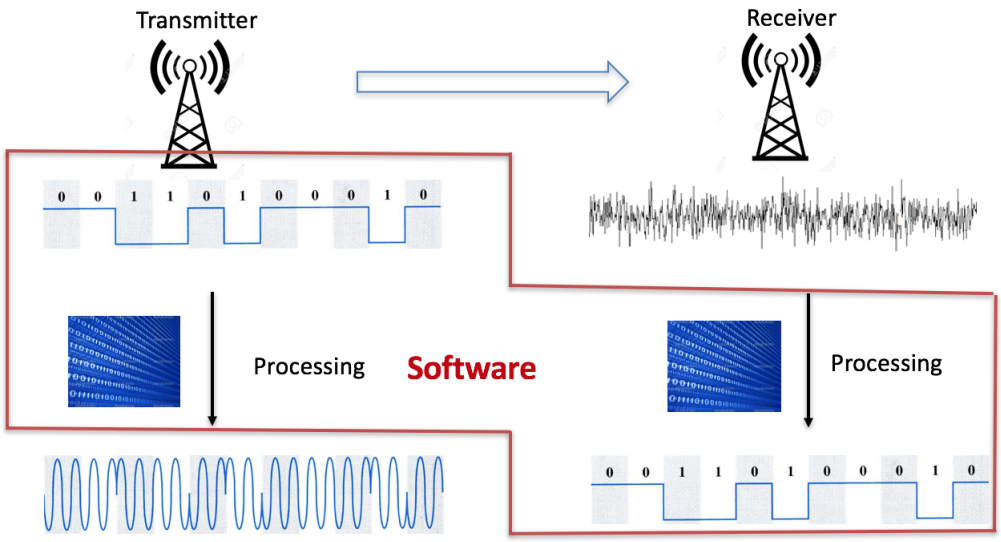

Figure 2.5: SDR Architecture

2.1.5 Related Work: PTC Communication

Hartong's thesis [23] presents a model for authentication, authorization, and scheduling for interoperable PTC operations. He identifies threats to PTC operation and provides trust management system based on PTC use cases and misuse cases.

In his thesis, Abadie [24] proposed a risk engine for PTC to ensure high-risk environments are avoided through adjustments in train operations. Abadie's thesis describes the risks of communication link availability due to environmental factors. Consequently, his work should be enhanced to incorporate the congestion in the radio bandwidth, one of the main components in my dissertation. Furthermore, I have improved his risk engine to detect any propagation conditions or jamming events that could effect PTC operation and included it as a part of SIRT.

2.2 Related Work: Wireless network planning for trains

A significant amount of research exists in the area of bandwidth provisioning for wireless customer services such as Internet access inside trains. In [25] Lin and Chang describe a conceptual architectural design of communication and entertainment services for onboard high-speed public transportation systems. In their approach, the transport system becomes an entity of the mobile network, and their external connectivity is provided using a terrestrial microwave or satellite communication system. Kanafani et. al [26] describe an architecture to provide Internet access on trains using WiFi and WiMAX protocols. They represent a survey conducted on trains managed by Capitol Corridor Joint Powers Authority (CCJPA) by offering trial Internet based on a low bandwidth communication infrastructure. These two papers discuss wireless bandwidth management for providing wireless services such as Internet access inside trains, but not train movement control.

In [27] Hui et. al describe GSM-R network planning for high-speed train operations in China, including quality of service of high-speed railway wireless communication systems according to carrier interference, bit error rate, and handover standards. Mihali et. al

proposes a railway network communication system based on the GSM-R standard in [28]. They report results of an experiment conducted on behalf of the Bucuresti Constana railway corridor, in Cernavoda, Romania. Their GSM-R based network planning will not address the frequency planning in USA due to bandwidth limitations. The architecture presented in my thesis is relevant to train operations in the USA because it addresses the issue of bandwidth limitations, capacity constraints and interference avoidance, and the dual network (signaling and WIU) design of PTC. The frequency planning for PTC communication network is work described in more details in Chapter 3.

2.3 Related Work: Intelligence to Radios

Building application intelligence into SDR systems was first presented by Mitola in [29], where he introduced the concept of Cognitive Radio (CR) which integrated agent-base control, natural language processing, and machine learning. He also proposed using CRs for spectrum pooling, the principal that allows multiple users to share a radio spectrum range. Today spectrum pooling is used for secondary users who do not own a licensed spectrum to gain access to parts of the licensed spectrum not used by primary users. When the primary user claims a channel, secondary users vacate them [30].

Dynamic Spectrum Access using CRs reduce the congestion in the unlicensed bands, because unlicensed users get opportunistic access to licensed spectrum. Such a model allows vendors to manage their wireless applications without purchasing licensed bandwidth for user communications. Therefore it has been used widely by different vendors in various applications including vehicular networks. In [31], Chen presents the use of dynamic spectrum allocation in vehicular networks and shows the use of creating a cooperative CR network between vehicles in fleets like Google driver-less cars. These cars can use a CR network to exchange information such as traffic conditions with other vehicles to increase driving efficiency and improve safety.

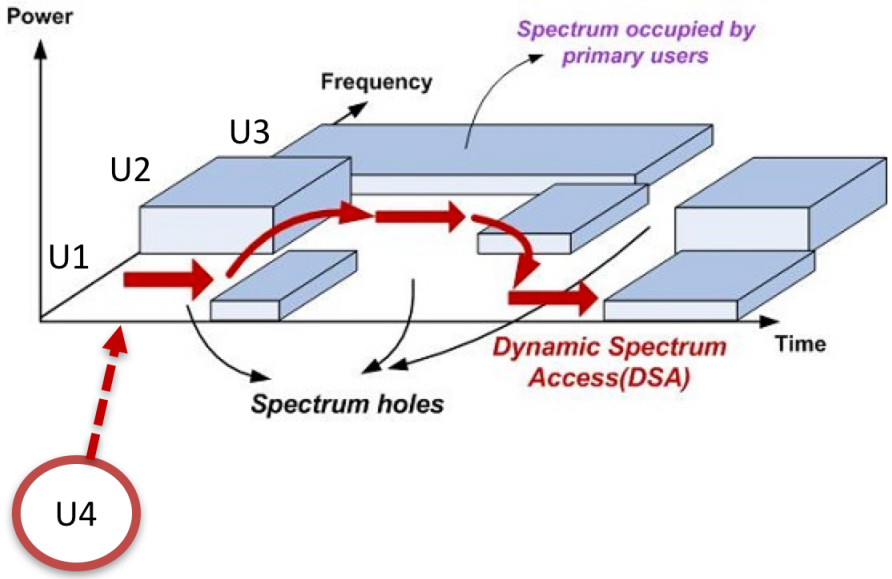

Figure 2.6: Dynamic Spectrum Access

Singh [32] presented the use of CRs in VANETs for V2V and V2I communication. He showed how the *Intelligent Transportation System's frequency spectrum* can be overcrowded when more vehicles use VANET applications and how using CRs can avoid congestion and increase the safety of traveling. In [13] Tabassum et. al formulated high-throughput channel allocation problem in cognitive radio based vehicular network as a mixed-integer linear programming (MILP) problem. Their objective was to maximize the throughput of the overall cognitive radio system while maintaining the required Quality of Service (QoS). Piran et. al presented *CR-VASNET* [33], which is a CR based wireless adhoc and sensor network. They attempt to optimized the distance between the sender and the receiver to reduce energy consumption and increase the network lifetime. They also presented a scheme to detect the primary user to mitigate interference from CR-VASNET users (secondary users) to the primary user.

To the best of my knowledge, CRs that have been developed so far only attempt to find white spaces in the unused licensed spectrum ([35], [30]) for dynamic spectrum access to use the license spectrum as secondary users. Thus their implementations includes spectrum sensing, using artificial intelligent to detect best frequency channel and frequency hopping.

Ammana et. al [34] showed how CR could be used to dynamically allocate spectrum for PTC and improve spectrum usage. They list out necessary PTC spectrum requirements and show how CR can be used to overcome them. Their work further showed how to detect better frequency channels in other users spectrum and use it when the PTC spectrum is congested. However using other users spectrum can have more overhead and security concerns.

The comprehensive frequency analysis that I did for PTC spectrum shows that proper frequency planning can reduce the spectrum congestion without using other users spectrum. SIRT is designed based on this study to use unique properties of PTC communication to intelligently distribute channels that is owned by PTC. SIRT network holistically manages the spectrum efficiently in addition to providing a CR based primary Intrusion Detection System (IDS) system for PTC systems. SIRT is more specialized for PTC operations because they,

1. Operates according to PTC protocol and spectrum planning uses unique properties in PTC network
2. Dynamic spectrum allocation within a licensed spectrum band
3. Improves both security and reliability of radio communication

Also, I developed a prototype of SIRT which includes spectrum monitoring, frequency hopping and adaptive modulation and tested it under varying noise conditions. Although spectrum management techniques used in SIRT have been extensively researched, most work provide theoretical analysis or numerical estimations based on models [36]. Other papers simulate these techniques using tools such as Matlab, Simulink, and NS2 [10–13], However, the prototyping efforts have been limited in this area. The advent of Software Defined

Radios makes such prototypes feasible, and prototypes provide first-hand data that augment or validate theoretical estimates and simulated value in realistic scenarios and come closer to real-life deployments. As Harris states [37], synchronization is a primary component of the radio transmission that is overlooked by many researchers. Most scientists assume that the system is already synchronized in their analytic or simulated models. Therefore my work provides valuable information for developing these systems in real world applications.

Chapter 3: Frequency planning for PTC network

American railroads have allocated the radio spectrum from 217MHz-219MHz and 221MHz-222MHz for PTC operations. To allocate this limited frequency spectrum between the signaling network and the WIU network a proper channel management architecture is required. Improper management of frequencies can lead to interference within the PTC communication network, which increases the Bit Error Rate (BER) and eventually reduces the system throughput. Therefore PTC controller may not receive critical messages on time to make decisions to avoid operational risks.

3.1 Constraints on Radio Bandwidth

To identify the spectrum congestion problem, I first analyzed the constraints on bandwidth allocation. For that study, I assumed the followings.

1. Train tracks follow a linear path
2. There is a continuous communication between train and the back office. That is, there are no dark territories

Following are the primary constraints.

1. **The bandwidth for signalling and WIU channels should be less than the total available bandwidth:** If b is the bandwidth of a channel, N is the total number of channels (Signaling channels and WIU channels) and g is the guard band, total bandwidth B follows the equation 3.1. The total number of signaling channels N_c and WIU channels N_w should be less than the total number of channels as shown in equation 3.2.

$$N \cdot b + (N+1) \cdot g \leq B \tag{3.1}$$

$$N \geq N_w + N_c \tag{3.2}$$

If br is the channel bit rate, mod is the modulation scheme and SNR is the signal to noise ratio, then bit rate of the channel can be represented as a function of the channel bandwidth and the modulation scheme as shown in 3.3. Also the maximum possible modulation is a function of Signal to Noise Ratio (SNR) as shown in 3.4.

$$br = f(b, mod) \tag{3.3}$$

$$mod_{max} = f(SNR) \tag{3.4}$$

If the roll-off factor of transmission band-pass filter is α, channel bandwidth is b and number of bits per symbol is $bits\_symbol$ then bitrate br can be represented as in equation 3.5.

$$br = (b \cdot bits\_symbol)/(1-\alpha) \tag{3.5}$$

The number of bits per symbol depends on the modulation scheme. For BPSK, the number of bits per symbol is 1, and for QPSK the number of bits per symbol is 2. The maximum possible modulation scheme that the signal can be modulated with depends on the Signal to Noise ratio (S/N) at the receiver. Using Shannon's theorem [38] on Channel capacity C and channel bandwidth b,

$$C = b \cdot log_2(1 + S/N) \tag{3.6}$$

Using Hartley's Law for a M-ary errorless channel [39],

$$C = 2 \cdot b \cdot log_2(M) \tag{3.7}$$

Comparing Shannons theorem with Hartley's law $b \cdot log_2(1 + S/N) = 2 \cdot b \cdot log_2(M)$

$$M = \sqrt{1 + S/N} \tag{3.8}$$

2. **Signaling channel capacity should support signaling messages between the back office and the train:** There are two types of signaling messages. They are,

   - sig_{ho} : Signaling rate during handover
   - sig_{normal} :Signaling rate at normal operations such as sending *position information* and *track consists*

   If signaling packet rate $rate_{sig} = max(sig_{ho}, sig_{normal})$ and signaling packet size is pc then bit rate br should be greater than the maximum signaling traffic rates as shown in equation 3.9. Also the signaling rate is a function of train speed $f(v)$ as shown in equation 3.10.

$$pc \cdot rate_{ctrl} \leq br \tag{3.9}$$

$$for \quad all \quad v : V[] ==> pc \cdot f(v) \leq br \tag{3.10}$$

3. **Interference between PTC elements should be minimized:** Train routes follow a linear path. Therefore all the signaling points and WIU towers follow a linear geometry as shown in Figure 3.1. In such a formation, there can be more than one WIU tower located in the cell area of a signaling point. Therefore, if the maximum frequency to avoid adjacent channel interference is $freq_{max}$, number of signaling points

in the track is n, and number of WIU towers in one signaling cell area is m the allocation of frequency channels should hold the constraints mentioned below.

- The interference between adjacent signalling point transmissions should be minimized:

$$for \quad all \quad i : [0 : n] ==> |f(C_i) - f(C_{i+1})| \geq freq_{min} \tag{3.11}$$

- The interference between signaling point transmission frequency and the WIU tower frequencies in that cell area should be minimized:

$$for \quad all \quad j : [0 : m] ==> |f(C_i) - f(W_j, C_i)| \geq freq_{min} \tag{3.12}$$

- Interference between the WIU towers at the boundary of two signalling cells should be minimized:

$$|f(W_m, C_i) - f(W_0, C_{i+1})| \geq freq_{min} \tag{3.13}$$

- The interference between WIU towers in the same signalling point area should be minimized:

$$for \quad all \quad j, k : [0 : m] \quad \& \quad j \neq k ==> |f(W_i, C_i) - f(W_k, C_i)| \geq freq_{min} \tag{3.14}$$

- Wayside devices are located close by. Therefore, interference avoidance between neighboring WIUs is an issue. Therefore the WIUs can be grouped and connect to a single tower that transmits the beacons from all WIU connected to that tower. The number of WIUs that can be connected to a tower should be less than the maximum number of beacons interleaved in the transmit packet. Therefore if r is the beacon rate of a WIU, p is the WIU packet size and M is number of WIUs

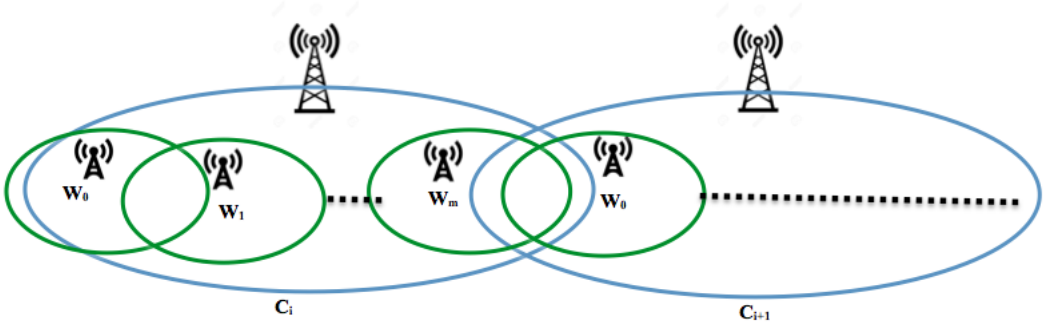

Figure 3.1: WIU and signaling point locations

that can be connected to one tower,

$$r \cdot p \cdot M \leq br \tag{3.15}$$

3.2 Enhancing the PTC system to encounter effects from propagation losses

SNR at the receiver depends on propagation characteristics of the medium. Therefore it is important that the baseline PTC architecture is designed to have sufficient SNR. Received power (P_{RX}) of a signal is a function of the transmission frequency (f), the distance between the transmitter and the receiver (d) and the propagation characteristics of the media ($Prop(.)$) as shown in equation 3.16.

$$P_{RX} = Prop(P_{TX}, d, f) \tag{3.16}$$

Therefore for proper transmission the distance between the transmitter (i.e the signalling point) and the receiver (i.e the train) r should be $r \leq Prop^{-1}(P_{TX}, f, P_{RX-min})$.

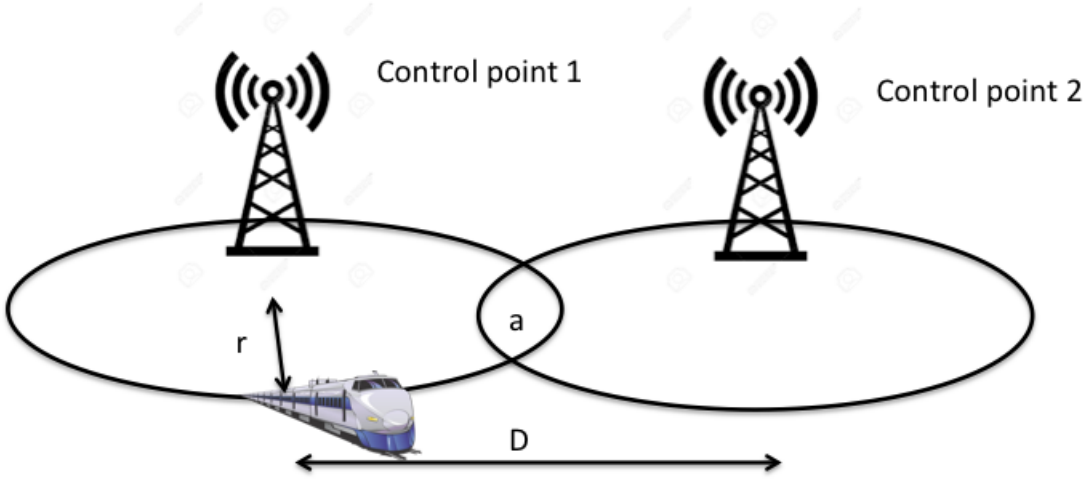

Figure 3.2: System design to minimize the propagation affect

In this scenario, r is the cell radius of the signaling point. The distance between two signaling points D is equal to $D = 2 \cdot r - a$ where a is the overlapping area between cell areas of two adjacent signaling points. Therefore,

$$D \leq 2 \cdot Prop^{-1}(P_{TX}, f, P_{RX-min}) - a \tag{3.17}$$

3.3 Analyzing Operational Constraints of PTC Equipped Trains

3.3.1 Constraints on train speed

In this section I have analyzed how the operational risk is increased when trains are travelling at high speeds. Two main factors limit the train's maximum speed. They are the handover time from one signaling point to the next and the safe braking distance of the train.

(a) The train should be able to complete the handover process from the current signaling point to the next signaling point before it exits from the coverage area of the current signaling point. I assumed that the train triggers the necessity to handover when it detects the signals from the next signaling point and makes an informed decision to switch. The

handover process should complete within the time the train moves through the overlapping area to maintain continuous connectivity. If the signaling packet size is p, the bit rate is b, trains travel at a constant velocity v and the number of handover messages required is n, then distance needed to exchange handover a message d in seconds can be obtained from equation 3.18. Therefore, for a reliable handover d should be less than the overlap distance between the two signaling cells.

$$d = \frac{n \cdot p}{b}.v \tag{3.18}$$

According to the Meteorcomm specification (the main official manufacturer of PTC radios), the signaling network (that is the messages from the train to the back-office and vice verse) uses 16 kbps and 32 kbps bit rates with the DQPSK (Differential Quadrature Phase Shift Keying) modulation scheme [1]. I varied the overlapping area length from 0-0.4 miles in steps of 0.01 miles and the number of handover packets from 50 to 200 in steps of 5 and calculated the maximum train speed for different bit rates. Results are given in Figure 3.3. Packet loss and signal drop-outs are most likely during train route. Therefore the number of handover packets is varied to simulate the retransmission overhead due to packet loss.

Figure 3.3 shows that if the data rate is 16 kbps, then for the train to operate up to 400 mph the number of handover packets should be less than 50 and overlapping area length should be greater than 0.3 miles. Increasing the data rate will allow the communication channel to have more retransmission and consequently shorter overlapping area. When the number of packets per handover increases, it is difficult for the trains to operate at high speeds even when the bit rate is higher. For example, when the overlapping area length is 0.2 miles, the bit rate is 32 kbps, and the number of packets is 50, the train can attain a maximum speed of 379 mph. When the bit rate is 64 kbps with the same bit rate and overlapping area length the speed will be doubled (758 mph). But with 64 kbps bit rate and 0.2 miles overlapping area length the maximum speed drops to 190 mph when the number

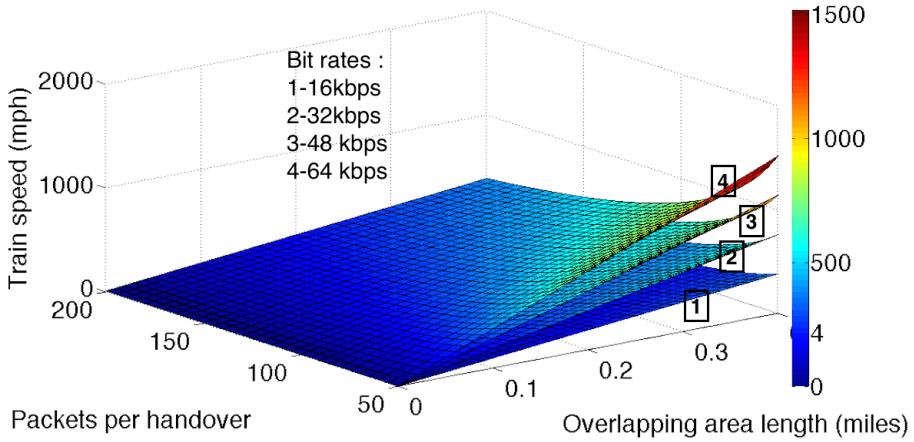

Figure 3.3: Speed with the number of handover packets and overlapping distance

of packets per handover becomes 200. With the limited bandwidth availability for train operations, it was difficult to increase the bandwidth for a channel. But a channel with the same bandwidth can achieve higher data rates by using a higher order modulation scheme. If the modulation scheme is BPSK (where a symbol carries 1 bit), a 12.5kHz channel can operate at 16 kbps data rate. But if the modulation scheme is QPSK (where a symbol carries 2 bits) the data rate is doubled.

(b) The braking distance of the train should be less than the cell radius of the wayside communication device. This ensures that the train can receive wayside status messages promptly so that it has enough time to come to a complete stop before it reaches the obstacle. According to [40] the braking distance of a train can be calculated using equations 3.19.

$$s = \frac{\phi \cdot V^2}{1.09375 \cdot \frac{\lambda\%}{100} + 0.127 - 0.235 \cdot (\pm i) \cdot \phi} \tag{3.19}$$

In equation (3.19), s is the braking distance, λ is the braking efficiency, i is the gradient and ϕ is a speed dependent constant. We take the λ to be 15% and i to be 0.

Cell radius of the wayside device depends on its transmit power, the minimum possible

receive power the locomotive receiver can handle and the propagation characteristics of the location. To calculate the cell radius, I conducted a link budget analysis using equation (3.20).

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_{TX} - L_{RX} - L_p - L_r - L_m - L_f \tag{3.20}$$

$P_{RX} = Received \quad power$

$P_{TX} = Transmission \quad power$

$G_{TX} = Transmit \quad antenna \quad gain$

$G_{RX} = Receive \quad antenna \quad gain$

$L_{TX} = Transmit \quad feed \quad loss$

$L_{RX} = Receive \quad feed \quad loss$

$L_m = Antenna \quad miss - pointing \quad losses$

$L_p = Path \quad loss$

$L_r = Rain \quad attenuation$

$L_f = Foliage \quad effects$

Each of the components in equation (3.20) is analyzed below to see how the losses and gains can affect the link I consider [41] [42].

1. **Transmission Power:** Power ratings from the Meteorcomm radio equipment specifications are given in Table 1 [1]. For the link budget calculations along the train path, I considered power values less than the peak WIU transmit power level - adjusting based on the height of the WIU antenna. I use the logic that higher placed antenna in a tower should have lower power because of limited obstacles. Lower antennas are more likely to have degraded signals due to physical interference.

2. **Antenna Gain:** The mobile antenna on the train should be omnidirectional because the train should be able to communicate with a wayside device in any direction. The

most suitable antenna for wayside devices and signaling points are 180^o low directive antennas to cover the train track.

3. **Feed Loss:** Feed loss occurs at feed runs between the high power amplifier and the transmitting antenna at the transmitting end and the receiving antenna and the high power amplifier at the receiving end. The values depend on the length of the feeding cable and its material. A feed made of coaxial cable CL- 50086 with a length of 12 feet has a feed loss of 1dB.

4. **Antenna Mis-pointing Loss:** Because both the mobile and fixed antennas are low directional; the antenna mispointing loss can be neglected from the calculations.

5. **Path Loss:** Usually, trains communicate over irregular terrain. Therefore, the multi-path fading caused by the phase differences between the waves (direct wave and waves reflected by obstacles) arriving at the receiving antenna need to be taken into account when calculating the path loss. In urban areas where several obstacles are present, multipath fading will adversely affect the signal. It is possible that many reflected waves from buildings and other structures interfere with the primary signal. But in rural areas where there are fewer obstacles than urban areas, the effect of multipath fading is less. I used the Hata model [43] to calculate the path loss; an experiential formulation of the Okumara model [43] valid from 150MHz to 1500MHz frequencies. According to the model, median path loss in urban areas is given by [43] [44].

$$L_P = 69.55 + 26.16 \cdot log_{10} h_B - C_H + (44.9 - 6.55 \cdot log_{10} h_B) \cdot log_{10} d \tag{3.21}$$

In equation 3.21, L_p is the path loss in dB, f is the frequency in MHz, h_B is the height of the fixed transmitter in meters, C_H is an antenna height correction factor, and d is the distance between the transmitter and the receiver in kilometers. The mobile antenna height correction factor is calculated based on the urban area density [44]. For small to medium sized cities, antenna height correction factor (C_H) is given by

equation (3.22).

Table 3.1: Transmit power specification from Meteorcomm [1]

TX power	WIU	Locomotive	Signaling Point
Peak(W)	25	50	75
Average(W)	12.5	25	37.5

$$C_H = 0.8 + (1.1 \cdot log_{10} f - 0.7)h_M - 1.56 \cdot log_{10} f \tag{3.22}$$

For large cities C_H is given by equation (3.23).

$$C_H = 8.29 \cdot log_{10}(1.54 \cdot log_{10} h_M)^2 - 1.1 \tag{3.23}$$

The path loss in suburban areas can be calculated using a modified Hata formula given by equation (3.24).

$$L_P = L_P(urban) - 2 \cdot (log_{10}(f/28))^2 - 5.4 \tag{3.24}$$

6. **Rain Attenuation:** Attenuation due to rain fade can be calculated by using the equation (3.25).

$$L_{prep} = 10^{1.203 log_{10}(f) - 2.290} R^{1.703 - 0.493 \cdot log_{10}(f)} \tag{3.25}$$

In equation (3.25) L_{prep} is the attenuation due to rain fade, f is the frequency in GHz and R is the rain density in mm/hr [45].

7. **Vegetation:** Rain and wind can cause variations of the propagation channel when heavy foliage is present. This is a predictable geographic phenomenon because train routes are known. A standard calculation for predicting the increase in loss due to propagation through trees is the exponential decay model presented by LaGrone [46]. The loss caused by vegetation L_{veg} is computed using equation (3.26) where f is the frequency in GHz and D_f is the foliage depth in meters.

$$L_{veg} = 0.26 \cdot f^{0.77} D_f \tag{3.26}$$

For the train to react to the message in the WIU beacon correctly, the message should be received such that the train has sufficient time to apply the brake and bring the train to a complete stop. Therefore the braking distance of the train should be less than the cell radius of the wayside interface unit.

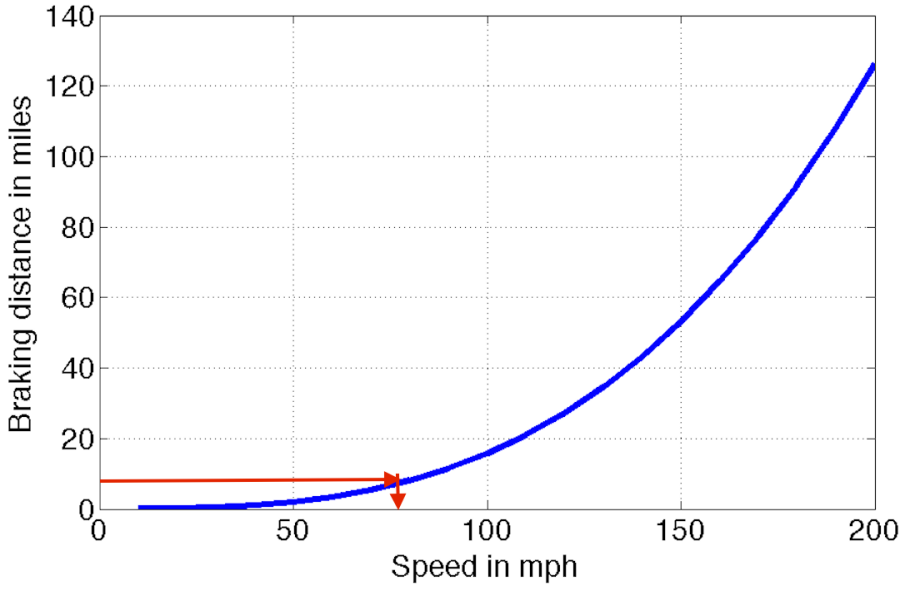

Figure 3.4: Speed restrictions based on braking distance

I considered $\phi = k \cdot V$. Then by using a known braking curve, I calculated k. By using k, I calculate the braking distance for different train speeds ([40]) as shown in Figure 3.4.

Then I calculated the maximum possible cell radius of a WIU, using peak transmit power of 25W (44 dBm) and minimum transmit power of -110 dBm using the equations (3.20)-(3.26). I used propagation characteristics for a hypothetical train line using existing maps. I observed that the cell radius should be less than 7 miles. By using the braking curve in Figure 3.4, the train to have a braking distance less than 7 miles the train should operate at speed less than 78mph. The braking curve I considered is from an old cargo train, which has a significant braking distance. Therefore a high-speed train with a small braking distance should be able to achieve higher speeds without violating the WIU power limitations. A system can support higher train speeds by increasing the communication rate through an adjustment of the modulation scheme. Furthermore, high-speed trains should be designed with weight characteristics that correlate to a small braking distance to reach maximum speed yet maintain the channel bandwidth reserved for PTC. These findings were published in [15].

3.3.2 Calculating the Guard Band

The guard band is calculated to overcome the Doppler effect. Due to the potential speed of trains, the well-known Doppler shift should be addressed. The guard band is a vacated portion of the spectrum to allow for possible frequency shift due to Doppler, protecting the adjacent channel from interference. To ensure its functionality, the guard band should be at least as twice the value of the calculated Doppler frequency shift. The frequency shift is calculated using Equation (3.27):

$$\frac{\Delta f}{f} = \frac{V_t}{C} \tag{3.27}$$

I varied the speeds from 50mph to 400mph and calculated the corresponding Doppler shift. As shown in Figure 3.5, the frequency shift is linearly proportional to the speed of

the train. I assumed a maximum train speed of 400 mph and calculated the Doppler shift to be 131Hz. Then I doubled that value to 262Hz and recognizing the potential for slight variations set the final guard band size to 300Hz.

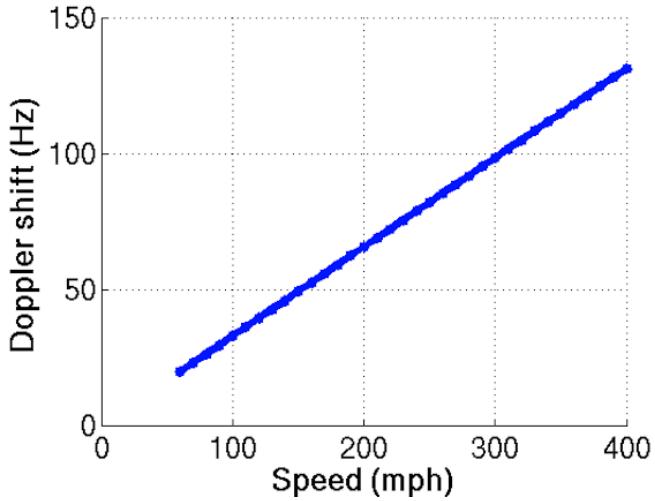

Figure 3.5: The Doppler Shift (Hz) Vs. Speed(mph)

3.3.3 Calculating the Maximum Number of Packets per Channel

Train operations occupy 217-219MHz for the uplink and 221-222MHz for the downlink. A channel is 25 kHz wide and uses QPSK modulation. The base and locomotive radios can support data rates up to 32 kbps, while the wayside radio can support data rates up to 16 kbps. The maximum WIU message size is 2040 bits (see Table 2.1) and the maximum signaling message size is 1216 bits (see Table 2.2). I used these maximum packet sizes and calculated the maximum number of packets per second that a channel can use. This result is tabulated in 3.3. It shows that with the current PTC system with 32 kbps data rate signal channel can use up to 26 packets per second and beacon channel can transmit up to 15 packets per second.

Then I analyzed how the bit rates change with the bandwidth and the modulation

scheme, and for selected bit rates I calculated the number of packets each channel can handle per second. If the roll off of the filter is α, Symbol rate is S then the occupied bandwidth (B) can be calculated using equation 2. $\alpha = 0.5$ is the variation of channel capacities with different bandwidths. The modulation schemes are shown in Table 3.2. This work is published in [14].

$$B = S \cdot (1 + \alpha) \tag{3.28}$$

Table 3.2: Channel Capacities with Different Bandwidth and Modulation Schemes

Bandwidth(kHz)	BPSK	QPSK	8QAM
12.5	8 kbps	16 kbps	24 kbps
25	16 kbps	32 kbps	48 kbps
37.5	24 kbps	48 kbps	72 kbps
50	32 kbps	64 kbps	96 kbps

Table 3.3: Maximum Possible Packets per Channel per second for the Signalling Network and the WIU Network

Bit rate(kbps)	Signal	Beacon
16	13	7
32	26	15
48	39	23
64	52	31
80	65	39

3.3.4 Calculating the Maximum Number of Channels

The number of channels that can be allocated using 3 MHz band can be calculated by using equation 3.29. B is the channel bandwidth in kHz, and g is the guard band in kHz. n is the number of channels.

$$n \cdot B + (n+1) \cdot g = 3MHz \qquad \rightarrow \qquad n = \frac{3000 - g}{B + g} \tag{3.29}$$

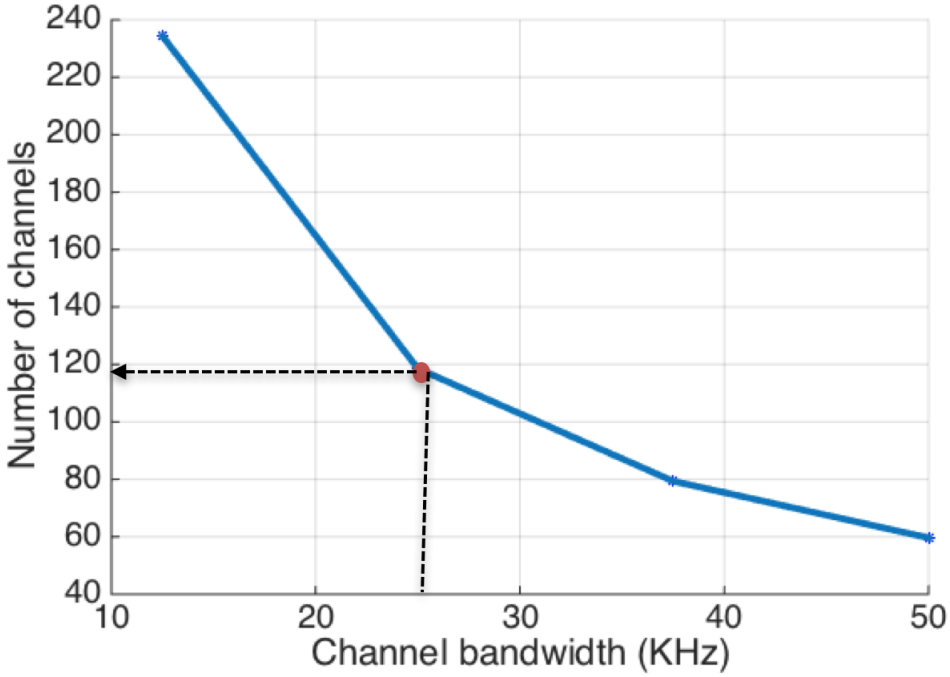

Figure 3.6: Maximum possible channels with bandwidth

3.4 Cell Planning

Due to limited bandwidth availability, static allocation of frequency channels for PTC devices is inefficient. It will limit the maximum number of trains that can be operated at the same time. Therefore the channels should be dynamically allocated. To dynamically allocate channels for the signaling point to train communication, I developed a cell-based movement authority granting system which is an extension of the block based movement authority proposed in [23].

It divides the train line into segments so that there is one signaling point transceiver for each segment. Trains will request and obtain movement authority to enter the next segment by communicating through the existing transceiver connection. Consequently, a train will be granted entry into the next track segment only when another train does not

occupy that segment, and the signaling transceiver of that segment has a vacant channel to accommodate the requesting train. As shown in Figure 3.7, when a train reaches the overlapping region it is expected to start obtaining permission to enter the next block and establish a connection with the approaching signaling transceiver.

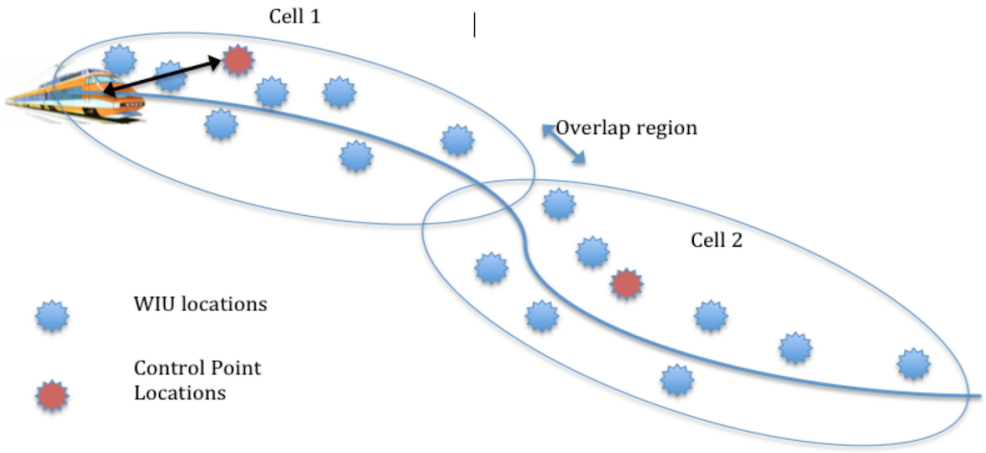

Figure 3.7: Wayside and signaling network layout

3.4.1 Cell-based Movement Authority

Figure 3.8 shows the scenario where the train is requesting to enter the next block. As shown in message 1, the train should request movement authority to enter the next cell from the current signaling point. The current signaling point communicates with the next signaling point to see whether there are available channels (message 1.1). If channels exist, the signaling point assigns an available channel to the train and hand over the train to the next signaling point (messages 1.2 -1.4). If there are no channels available, signaling point checks if the train is a priority train such as high-speed trains (message 1.3). If so, movement authority is granted to the priority train, and the bandwidth allocated to the low priority trains is reduced to accommodate the high priority train as shown in messages

1.3B to 1.3B.2. If channels are not available and the oncoming train has a low priority the train is requested to stop (messages 1.3A.1-1.3A.3a). PTC mechanisms will ensure that the train comes to a complete stop if the operation is not safe, by applying automatic brakes. After stopping the train, the system alerts all adjacent signaling points to warn oncoming trains of the delay.

Figure 3.9 shows the scenario when a train is handed over to the next signaling point. When handover happens, the signaling point releases the channel allocated to the train. This will free a channel in the channel pool of that signaling point. This freed channel can be allocated to any stopped trains waiting for a channel as given in message 1.3 or if there are no waiting trains the channels can be kept in reserve (message 1.3B). Furthermore, if the bandwidth allocated for non-high-speed trains are reduced to accommodate a high priority train by reducing the train speed or decreasing modulation scheme, the system can increase the channel bandwidth to the default value (i.e., for 25kHz for the current PTC operations) (in message 1.3A.1). This is a decision that could be taken by the signaling point. This system allows a limited number of channels to be allocated to the signaling network while still allowing the high-speed trains to operate smoothly (without applying unnecessary braking).

3.4.2 Computing the Bandwidth Capacity of signaling Points

The capacity of a signaling point is designed to support the maximum number of trains to minimize the number of trains waiting for a signaling channel at any given time. If the signaling point has to support more trains, the system can operate by stopping some trains or reducing the channel bandwidth. To consider the scenario where two trains occupy parallel tracks and train junctions where multiple trains may be present (as shown in Figure 3.10), I designed every signaling point to support a minimum of n trains at any given time. Because every train requires an uplink and a downlink, a signaling transceiver that can service n trains needs $2{\cdot}n$ signaling channels. Consequently, the signaling point capacity is C is $C = 2{\cdot}n$.

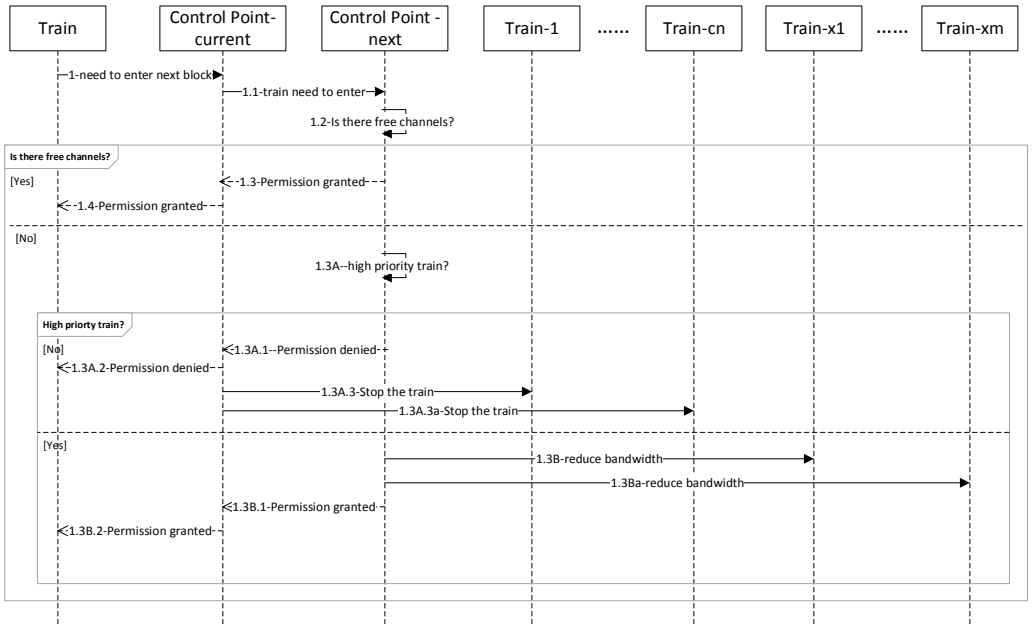

Figure 3.8: Train requesting to enter next block

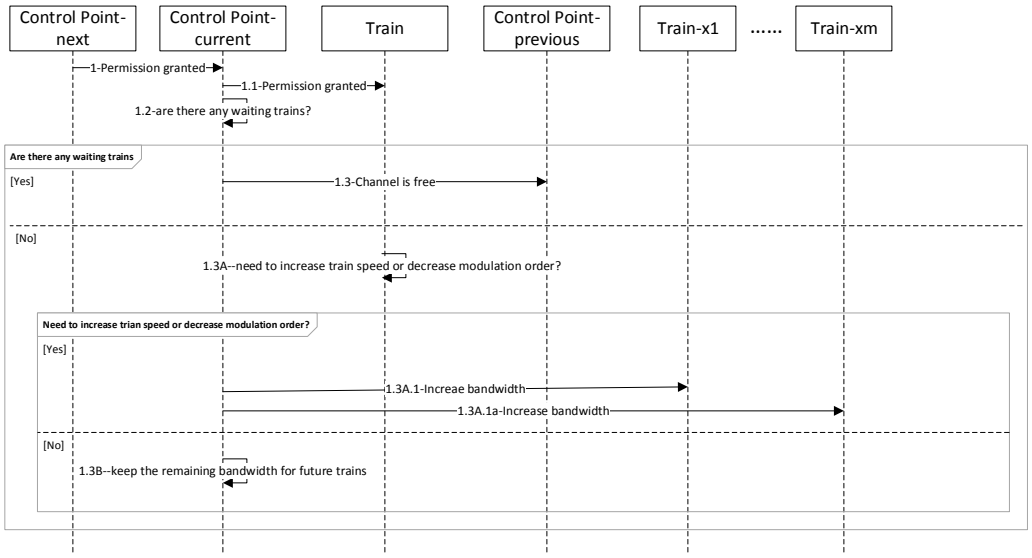

Figure 3.9: Train leaving signaling point

Figure 3.10: An intersecting track geometry

Cell planning for the signaling network is less complicated because of its linear topology. As a result, I only need two sets of frequencies to cover the signaling points on a rail line interchangeably as shown in Figure 3.11, where each of these sets should have a $2 \cdot n$ number of channels. Channels within one set should have enough frequency separations within themselves so that they do not interfere. Therefore there should be $4 \cdot n$ channels for the signaling network.

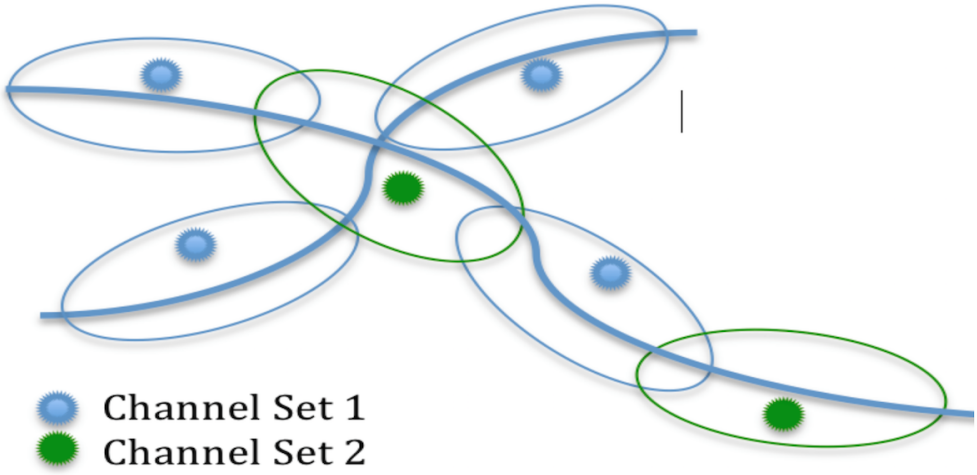

Figure 3.11: Signaling cell planning basic layout

3.4.3 Frequency Allocation for WIUs

After allocating the channels to the signaling network, the rest of the channels can be divided among the WIU. The location of WIUs is static and known because WIUs and switches are based on rail line typologies. However assigning channels to this network should be coordinated properly because of the possibility of having many wayside devices located close by (say, located within 5 miles) and they may transmit independently.

In urban areas where trains are frequently operated, WIUs operate in periodic mode. In *periodic* mode the WIU broadcasts a periodic beacon. Therefore WIU requires channels only for its downlink to broadcast the beacon status of the device. If there are a N number of total channels, the number of channels for the WIU network m can be given using the equation 3.30.

$$m = (N - 4 \cdot n) \tag{3.30}$$

If trains do not operate regularly in a route, the WIUs can operate in *on demand* mode.

When the WIU is operating in *on demand* mode, an oncoming train has to request the device status from the WIU. Upon receipt of a request, the WIU starts beaconing until a pre-set timer expires. Therefore a WIU operating in *on demand* mode requires an extra channel for receiving the beacon request message. Because the beacon request messages are not sent very regularly, I assumed that it is reasonable to have one channel for all the WIUs.

Channels should be allocated for wayside devices so that co-channel and adjacent channel interference is avoided. To do so, I divide the channels assigned to the WIU network into two groups as shown in Figure 3.12. I can connect wayside interface units located very close to a single transmitter to transmit the status of these devices interleaved in time and allocate more than one channel from the frequency set to that transmitter. This method allows the transmitter to use fewer channels to transmit the status of many wayside devices and hop between the frequencies of the allocated channels to avoid interference. To do so, I clustered the transmitters along the route based on the distance and assign frequencies from one set to one cluster interchangeably. WIU channel allocation is done in a way similar to the channel allocation in signaling points, but based on the number of wayside devices located in a region. Therefore the wayside unit clusters and the signaling point location can be independent of each other as shown in Figure 3.13.

Figure 3.12: Allocating channels for two sets

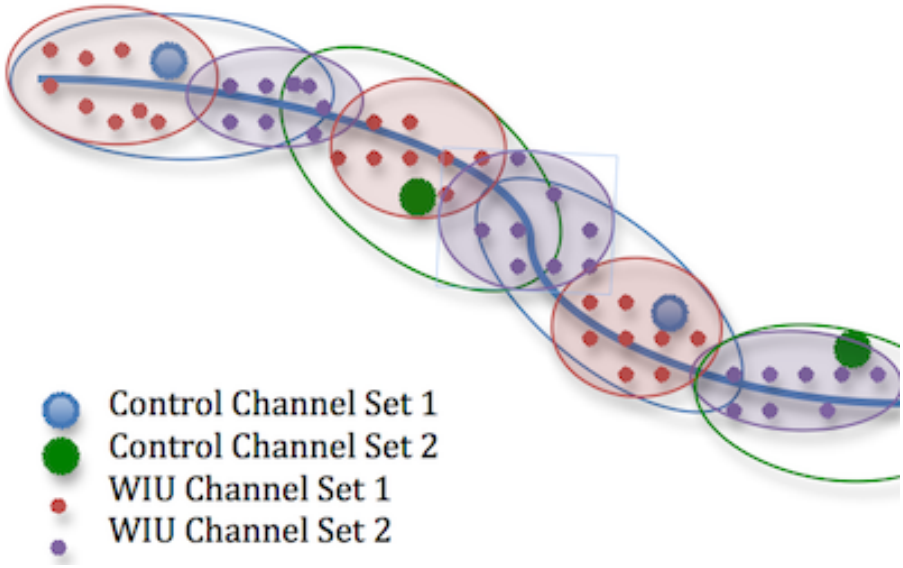

Figure 3.13: WIU and signaling network planning

3.4.4 Estimating Static Radio Locations

The signaling point should be able to communicate with a train uninterrupted until it is handed over to the next signaling point. Therefore, according to the transmit power limitations, I had to limit the distance between two signaling points such that the received power level is more than the minimum received power acceptable by the demodulator. I calculated the minimum distance between the signaling point and the train (D_{RX}), based on the transmit power value, the losses along the train line and the minimum received power the receiver can handle. The train should maintain communication with a signaling point all the time. The signaling points are located close to the train track. Therefore the distance between two signaling points can be considered as closer to $2{\cdot}D_{RX}$ as shown in Figure 3.14. I considered the distance between two signaling points to be $2{\cdot}D_{RX}$.

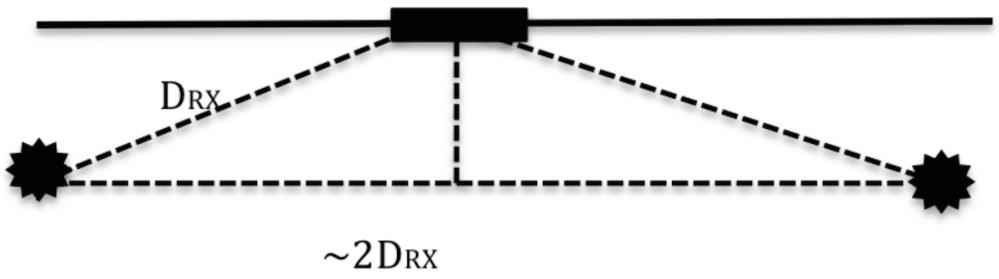

Figure 3.14: Signaling point positioning based on RX power

I calculated D_{RX} value along the entire train route according to the propagation characteristics of the railroad I consider and select the cell radius to be less than the minimum $2 \cdot D_{RX}$ value I obtained.

3.4.5 My Case Study

According to the Meteorcomm radio specifications [1], the proposed PTC system may use 25 kHz channels with a 32 kbps data rate and QPSK modulation scheme. Therefore, considering 25kHz channel bandwidth and a 300Hz guard-band, I can get 118 channels using the 3MHz band for train operations. These 118 channels should be efficiently shared between the WIU network and the signaling network [14].

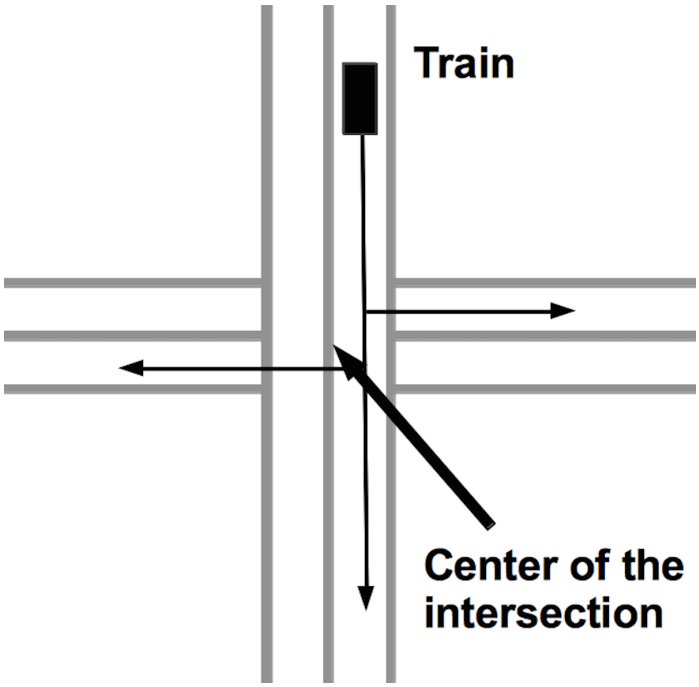

Figure 3.15: Considered Train Intersection

My case study considers a hypothetical train intersection with four train lines as shown in Figure 3.15. I used a hypothetical line because high-speed lines do not exist in the USA. I added wayside device locations (longitude, latitude), propagation factors such as urbanization, vegetation, and precipitation for points along the train track according to the data obtained from existing maps. Foliage depth is considered as the average foliage depth between the transmitter and the receiver. I took the frequency as 220 MHz, locomotive (mobile) antenna height as 5m and WIU (fixed) antenna height varies between 10-35m to be consistent with the terrain features. Transmit power was changed between 12.5W and 25W along the train line to be compatible with the antenna height. The minimum received power that can be handled by a receiver is considered as -110dBm. I considered a 180^o low directional antenna with 12dBi gain [47].

Scenario 1: Track intersections such as the one shown in Figure 3.15 could result in

having many WIUs closers to each other. Therefore it may be advantageous to cluster them and have a single antenna for all the WIUs belonging to the same cluster. Now my objective is to determine a suitable set of WIUs that can be clustered and the best location to place the signaling points. I do so assuming that a maximum of 8 trains would be operating simultaneously (two per each arm of the intersection) through the intersection.

Experimental Methodology

1. Using Equation(3.30), compute the maximum number of channels allocated for WIUs (m in Equation(3.30)) using the total number of available channels N as 118 and the number of trains n as 8. Then, limiting the number of WIUs per cluster to 40 (explained shortly), cluster the WIUs using algorithm 1. The reason I use this algorithm is that the standard clustering algorithms do not provide the ability to limit the number of WIUs belonging to a cluster.

2. Calculate the minimum required distance between the train and the signaling point(D_{RX}) using equations (3.20) to (3.26) for along all train lines. Select the minimum distance D_{RX_min} from these points, because the connection between signaling point and the train should not drop at any point during the train operation.

Algorithm 1 clustering WIUs

procedure CLUSTERING

define $start\_point$;
for all $i : WIU\{$
    $WIU(i).d = distance(WIU.locations(i), start\_point)$;
    % compute distance using (latitude,longitude)
}
$sort \quad WIU \quad based \quad on \quad WIU(i).distance$;
define $count = 0$;

define $cluster = 0$;
if $count < cluster\_limit\{$
    $WIU(i).cluster = cluster$;
    % assign the WIU to a cluster

    $count++$;
}
else {
    $cluster = 1 - cluster$;
    % Alternate between two clusters

    $WIU(i).cluster = cluster$;

    $count = 1$;
}

My Experimental Outcomes

To support eight trains the center signaling point requires 16 signaling channels. Because the same channels can be used in alternating signaling points the total number signaling channels is 32. Allowing for further expansions, I allocate 38 channels for signaling network. This leave 80 channels for WIU network (according to equation (3.30)). Therefore the total number of WIUs per cluster is 40. Figure 3.16 shows the WIU clusters when the WIUs per cluster is set at 40, where blue colored points represent WIUs, with channels assigned from set 1 and green color represents WIU's with channels assigned from set 2. In Figure 3.16, the horizontal and vertical axes represent latitudes and longitudes of the WIU locations. As shown in Figure 3.16, the first cluster contains WIUs belonging to all eight tracks. This is important because WIUs in the center cluster need to communicate with trains traveling on all the four tracks. But other clusters contains WIUs belonging to a single track.

I also computed D_{RX} values for points along the train lines, and they vary between

18.0256km (11.2 miles) and 20.1481km (11.8 miles) with a mean of 18.5181km (11.5 miles), and the standard deviation is 0.04266km (0.265 miles). Therefore the distance between two signaling points is calculated to be 22 miles. Figure 3.17 shows the location of signaling points for the train track provided in Figure 3.16 as black stars.

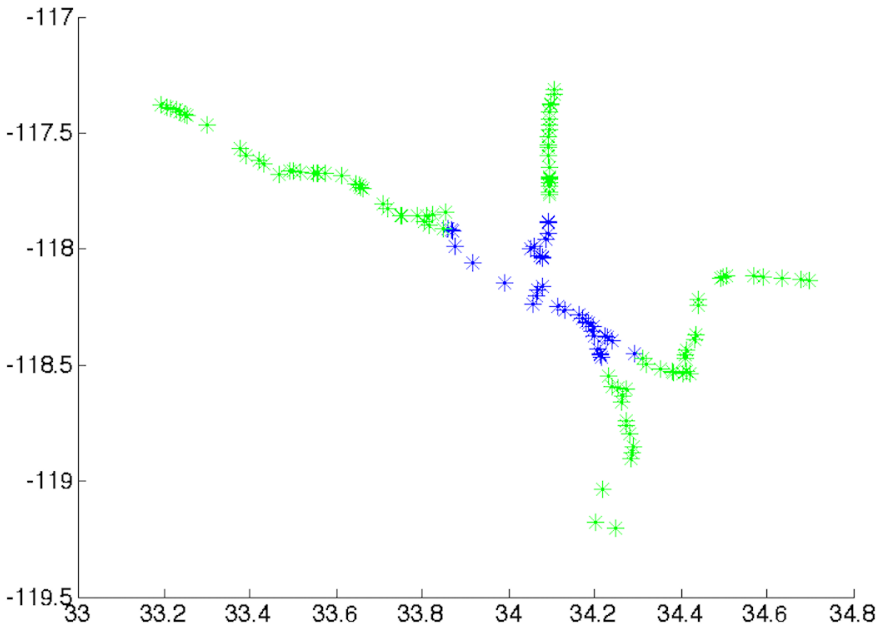

Figure 3.16: Dividing WIUs into clusters when WIU per cluster is set to 40

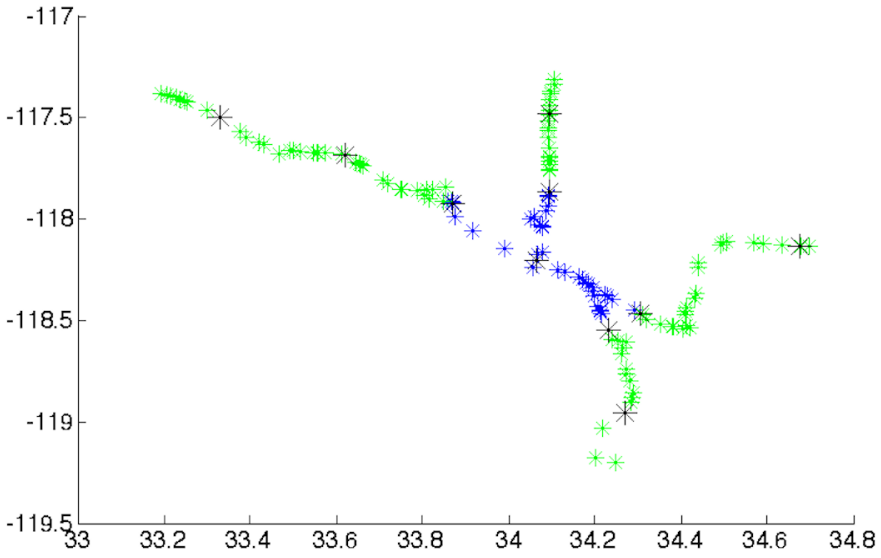

Figure 3.17: Signaling Point Locations

Scenario 2: This scenario extends the first one by doubling the number of trains that can approach the intersection simultaneously to determine the effect of high train volume on WIU clustering. The layout remains same but the trains are frequently operated, and therefore there are two trains on one track, due to having two blocks per track near the intersection. Because the number of trains is doubled, the system requires 76 signaling channels leaving only 42 channels for the beaconing network. Therefore I selected the cluster size to be 20. The resulting cluster assignment is shown in Figure 3.18. Figure 3.16 and Figure 3.18 show that when the number of trains that a signaling point has to support increases the clustering of WIUs becomes more complicated. Therefore train traffic density can determine if there are a sufficient number of channels available for WIUs. These results are published in [15].

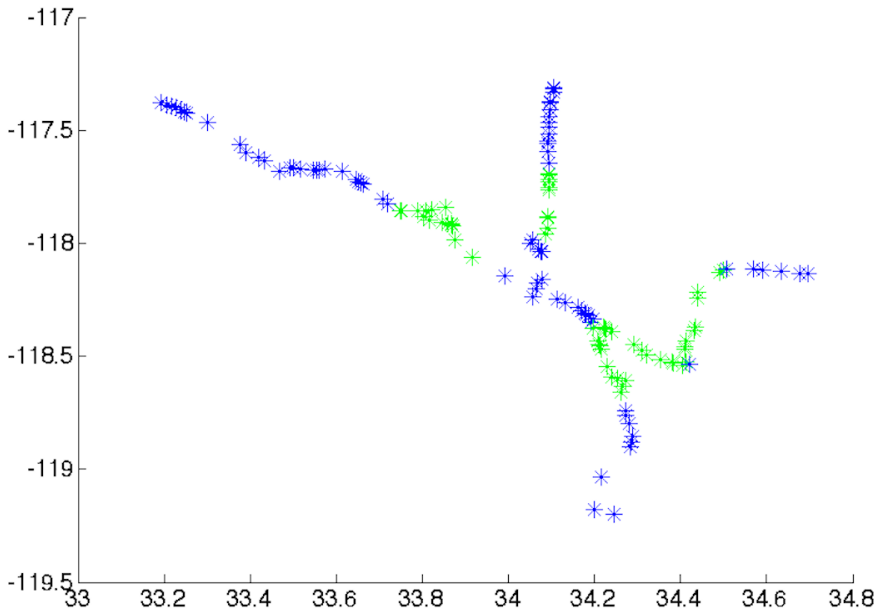

Figure 3.18: Dividing WIU into clusters when WIU per Cluster is set to 20

Chapter 4: Internal Architecture of SIRT

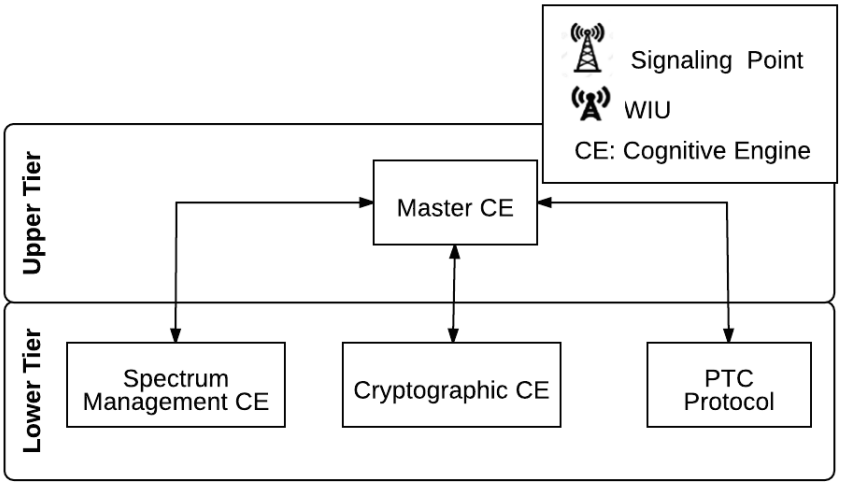

Figure 4.1: Internal Architecture of a SIRT Node

As shown in Figure 4.1 SIRT has two tiers. The upper tier has the Master Cognitive Engine (MCE) that estimates potential risk to the PTC communication and takes mitigating actions. To decide whether there is a risk, the MCE communicates with the lower tier cognitive engines, the Cryptographic Cognitive Engine (CCE), and the Spectrum Management Cognitive Engine (SCE).

- Spectrum Management Cognitive Engine (SCE) : The functions of this CE is two fold. First is to monitor the received signal variations and alert the Master CE about any risks (eg: drop in the SNR or increase in BER). Second is to adjust radio parameters (i.e frequency, power and modulation) according to the decision given by the master CE.

- Cryptographic Cognitive Engine (CCE) : This CE is responsible for detecting any malicious activities. If a malicious attack is detected it informs it to the master CE.

If the mitigation action is related to Spectrum Management (e.g., change frequency, increase power level), MCE sends the response to the SCE. If it is related to train controlling (e.g., Apply brake), the message is forwarded to the PTC controller.

4.1 The Master Cognitive Engine

The Mater cognitive engine (MCE) determines the main actions of the SIRT's dynamic behavior. The behavior of the MCE depends on the expected PTC behavior. To do so, the MCE continuously interact with two lower tier cognitive engines, access various databases and process information from the other radio nodes in the network.

4.1.1 Master Cognitive Engine Rules

Dynamic Channel Allocation

The main functionality of the MCE is dynamic channel allocation based on the train density and the train priority. I provided a preliminary schema for dynamic channel allocation that accommodates PTC's varying bandwidth needs in [15], which is described in Section 3.4.1. As shown in Figure 4.2 the train communicates with a network of signaling points while on route to get the information on how to proceed. The area that is covered by a signaling point is considered as a *Cell*. As shown in Figure 4.3, each cell includes more than one *Block*.

Following rules capture the dynamic channel allocation strategy describe above.

Rule 1(a): If the next block is within the cell area of the same signaling point the movement authority is given based on the block availability.

Rule 1(b): If the next block is not within the cell area of the same signaling point the

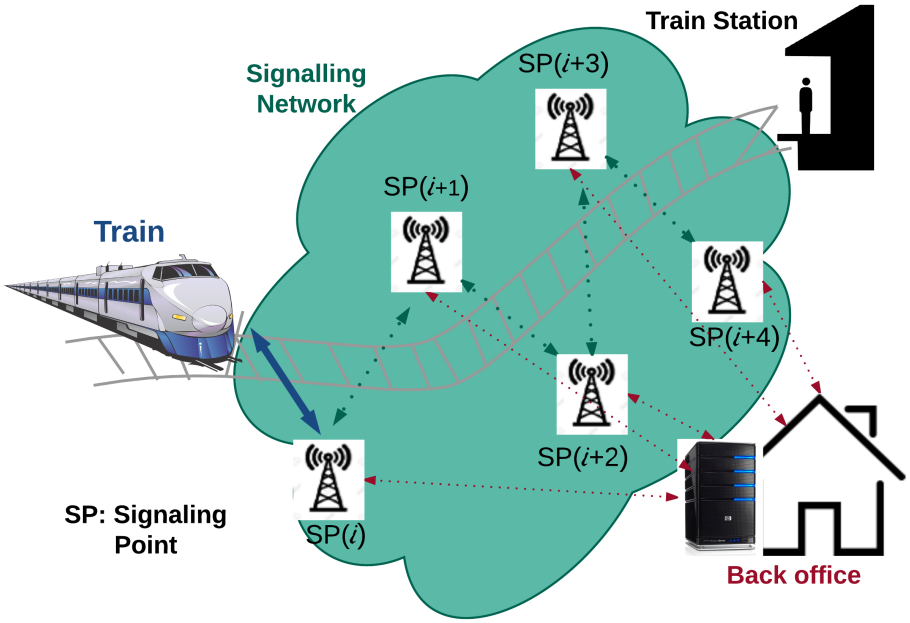

Figure 4.2: Signalling Network Architecture

movement authority is given based on the block availability and the channel availability of the handover signaling point.

Rule 1(c): High priority trains get channels even if there are no vacant channels. To do so, the signaling point reduces the bandwidth allocated to the low priority trains.

Rule 1(d): For nonpriority train if channels are not available, the movement authorization will be rejected.

Based on the rules explained above the signaling points and trains contribute to dynamic channel allocation as described below.

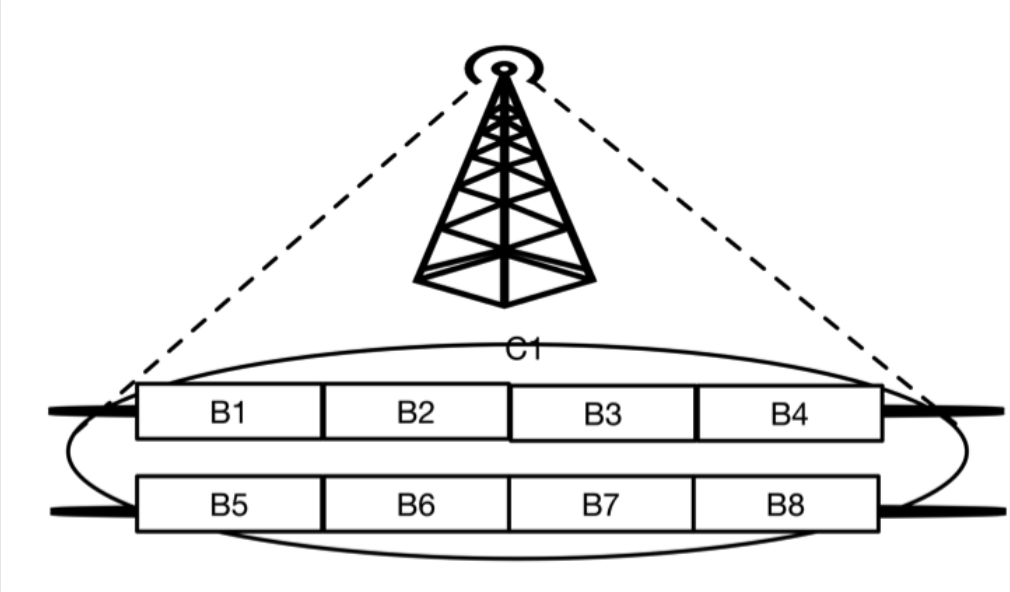

Figure 4.3: Control Cell and Block Layout

Signalling Points: As shown in Figure 4.4, every *signaling point n* has a wired connection with *signaling point n-1* and *signaling point n+1*. When the signaling point receives a *movement authority request* from the train, it checks with the back office whether the block that the train is trying to enter is available. If not occupied by some other train and the block is in the signaling point's region, it will grant movement authority to the train. If the block is occupied by another train, the signaling point sends a *movement authority reject* message to the train. If the next block is under the region of a different signaling point (*signaling point n+1*; For example in Figure 4.3 the train is currently at B4 or B8 the next block is controlled by a different signaling point), the signaling point forwards the message to *signaling point n+1*.

Once the request is received, the *signaling point n+1* checks for channel availability. If there are vacant channels, the *signaling point n+1* accepts the handover requests and send a handover accepted message to the *signaling point n* (signaling point that generated

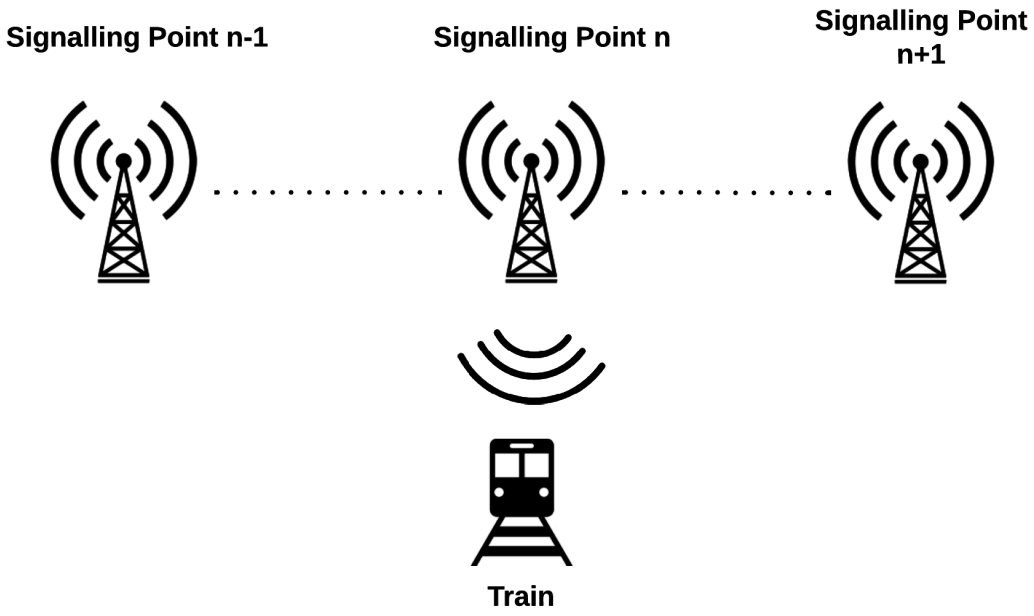

Figure 4.4: Signalling communication

the request). Once the train is handed over to *signaling point n+1*, the 'signaling point n' checks for waiting trains. If any trains are waiting, the signaling point grants the movement authority to the first train in the queue (by sending movement authority grant to *signaling point n-1*). If the next block is not available or no channels available the 'signaling point n+1' rejects movement authority. The 'signaling point n' stops the incoming train and also notify the *signaling point n-1* that there is a stopped train and make all the following trains to stop to avoid train collisions. It also sends a message to signaling point n-1. Based on the channel availability of signaling point n-1 it may inform it to *signaling point n-2*, this will follow until the signaling point can accommodate a new channel.

When a *signaling point n* receive a message from *signaling point n+1* that a train has stopped, it checks for any trains trying to enter that block. If so, it stops those trains

and indicates that to the *signaling point n-1*. If no trains are trying to enter, the signaling point ignores the message. If the block is available, but there are no vacant channels, then the signaling point checks the priority of the train. If the train has a higher priority (i.e. high-speed train), then the control point reduce the bandwidth of the non-priority trains that it is currently serving. Table 4.1 shows the functions of this module.

Table 4.1: Dynamic Channel Allocation functionality at the signalling point

Event	Action
Movement authority request: from train	Send handover request :to signaling point
Handover request: from signaling point	Check block availability: to back office
Handover accept: from signaling point	Send movement authority grant: to train
Handover reject: from signaling point	Send movement authority reject: to train
Block available: from back office	Check channel availability
Block not available: from back office	Send handover reject: to signaling point
Channels not available & priority train	Send handover grant/send change bandwidth(f1,f2) to non priority trains
Channels not available & non priority train	Send handover reject: to signaling point

Trains: Train radio stores the status received from the WIU network. Every time it receives a WIU status message, the stored value is updated according to the status of the critical infrastructure. When the train starts its journey, it checks if the nearest signaling

point has any vacant channels and if there are any trains in the block that the train is operating. Based on that it gives a signal to the operator as to whether the train should proceed. The train maintains the position of it with respect to the beginning of the block, and when the train reaches the end of the block, it sends a movement authority request to the signaling point. If the train receives movement authority to enter the next block, it checks the status of the most recent WIU status and decides whether to proceed to the next block.

If the train receives movement authority rejection from the signaling point, the PTC controller sends a warning to the operator, requesting to stop the train. Also if the instructions from the signaling network and the status of the most recent WIU contradict (i.e. when signaling network is instructing to proceed while according to WIU status it is not), the PTC controller sends a warning. The operator applies the brake and stops the train. If the operator does not apply the brake the PTC controller monitors the train movement, and after it reaches safe braking distance, it automatically applies the brake to stop the train.

When movement authority is granted the current signaling point (*signaling point n*) assigns two control channels for the bi-directional communication between the train and the *signaling point n+1*. After that the train maintains bi-directional communication with the *signaling point n+1*. For example, the signaling point requests position reports from the train periodically. Table 4.2 shows how the dynamic channel allocation functionality is programmed in the train radio.

Table 4.2: Dynamic channel Allocation functionality at Train

Event	Action
Movement authority grant: from sp & WIU status == PROCEED	Proceed to the next block
Movement authority grant: from sp & WIU status == STOP	Stop the train
Movement authority reject: from sp	Stop the train

Detecting PTC safety violations

Another significant functionality of the MCE is detecting conditions that can violate PTC safety such as a tampered or replayed message and lost messages. When the CCE detects an insecure message it informs the MCE which sends a warning to the operator.

Also when the train sends a movement authority request, the MCE sets up a timer. If it does not receive a response before the timer expires, the MCE sends a warning to the operator.

Rule 2(a): If a train receives movement authority and the CCE determines that the message is insecure, inform the PTC controller against entry.

Rule 2(b): If a train requests movement authority and it does not receive an acknowledgment before there is sufficient time to brake, inform the PTC controller against entry.

Table 4.3: PTC safety violation detection functionality at Train Radio

Event	Action
Movement authority grant: from sp & CCE decision==Insecure	Stop the train
MA timer expires & ACK received == False	Stop the train

Network level intrusion detection

The third functionality of the MCE is network level intrusion detection. All SIRT radios in the PTC network (i.e., the trains, signaling points and the WIUs) form a combine intrusion detection system exchanging information about potential threats. The MCE in each SIRT calculates risks based on the information it collects from other SIRT nodes.

One functionality of this intrusion detection system is detecting locations of attacker radios as indicated by Rule 3. Based on the signal strength of the received signal and the propagation conditions of the environment, a signaling point detects the distance between the transmitter and the signaling point. However, to pinpoint the exact transmitter location, the signaling point need the distance measurements from at least two other signaling points in the network. Once it has gathered sufficient distance measurements from other signaling points, the signaling point uses a triangulation method to find the location of the transmitter. Also, the signaling point keeps track of the trains expected location using the train's location updates and the equation of motion. If the location estimated by triangulation and the expected location does not deviate significantly, the signaling point can validate that the messages are from an authorized train. If not it detects a possible intrusion from an attacker trying to mimic a train.

Rule 3: If the location of the train is different from its expected location detect it as suspicious behavior.

4.2 Spectrum Management Cognitive Engine

The SCE is responsible for providing a reliable communication medium for PTC communications. The primary functions of the SCE are spectrum monitoring and reconfiguring radio parameters to maintain the most stable communication connection under varying channel conditions.

For example, if the communication between the train and a particular signaling point uses the same frequency all the time, that channel becomes susceptible to long-term interference and intentional jamming attacks. My SCE uses frequency hopping to avoid interference and intentional jamming that would interrupt PTC communication.

Also, freight trains that travel long distances over different terrains encounter different propagation conditions. Therefore statically choosing the modulation scheme before the train begins its journey does not result in optimal usage of the allocated radio spectrum and may lead to exposing exploitable security vulnerabilities. Therefore SCE changes the modulation scheme adaptively to take advantage of the best modulation based on the radio spectrum and required data rate at any given time and region.

4.2.1 Spectrum Monitoring

The SCE monitors the radio spectrum to identify signal quality variations. As shown in Figure 4.5 I used a combination of the CRC error rate at the decoder and the packet losses due to synchronization as the metric of quality of the channel. The two factors are explained as follows.

Consecutive CRC Failures: Indicates worsening signal quality resulting in bit flips. I added a CRC value during packet formation at the data link layer, which is validated at the receivers' data link layer. When bits flip occurs due to bad channel condition,

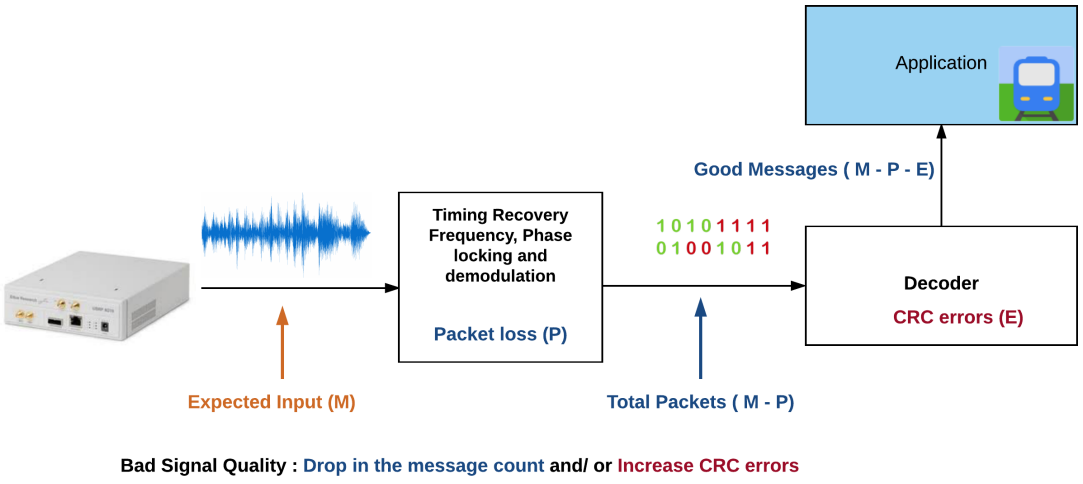

Figure 4.5: Spectrum Monitoring

the CRC validation fails.

Packet Losses before Decoding: Phase modulation I used (common in most telecommunication systems) encounters three levels of synchronization at the receiver: (1) frequency synchronization, (2) phase synchronization and (3) time synchronization. Consequently, high SNR results in synchronization block's inability to lock into the received signal and introduces bit flips. This phenomenon results in the decoder not being able to recognize the bits as part of a packet (bit flips in the packet header) and discard them. Hence high SNR's result in message losses during synchronization.

I measured the message arrival rate (per second) and estimated the synchronization message loss rate inside the packet decoder. I also measured the CRC error rate (per second). These measures are used to trigger radio parameter reconfigurations. Threshold values that are used are explained in section 3.3 (*Spectrum Management Functionality*).

4.2.2 Radio Parameter Reconfiguration

Based on spectrum monitoring, SCE change the transmission frequency and modulation. Once such a decision is made, the receiver sends a control message to the transmitter with the next value and the time to switch using the reverse channel. The switching time is determined based on the estimated message propagation time between the transmitter and the receiver so that the transmitter and the receiver can be switched at the same time to reduce the packet loss during the synchronization. Here the receiver waits until the switch time occurs and changes the agreed upon radio parameters. Once the message is received, the recipient extracts new parameters and change them at the specified time.

4.2.3 Spectrum Management Functionality

The functionality of the SCE is shown in Algorithm 2. The SCE monitors the number of packets received and the number of messages with errors and retains their statistics such as the total number of messages, the total number of erroneous messages, average message rate and erroneous message rate within the last one minute and five-minute intervals (line 1 of Algorithm 2). The SCE calculates the percentage of errors for one minute and five-minute running intervals (lines 2-3 of Algorithm 2). The SCE uses these values to make the reconfiguration decisions.

The threshold values that I used to trigger radio parameter changes are computed based on the operational environment. My experimental setup used the following threshold values. To go from a higher order modulation to a lower order modulation (due to signal quality degrading) threshold value was selected as an increase in error/message ratio greater than 10%. When going in the opposite direction (i.e., from lower order to higher order), the error rate for 1-minute interval and 5-minute interval both have to be less than 5%. These values were chosen to ensure the signal quality is stable before switching to a higher order modulation. When the message rate drops below 50% (that is, it becomes difficult for the receiver to maintain phase synchronization) independent of the CRC error rate, I changed the channel frequency. I implemented three main rules in my initial design of the SCE.

Rule 1(a): Hopping the frequency when the channel is jammed: shown in lines 4(a) to 4(d) in Algorithm 2. If the message rate drops below a specified threshold, the SCE flags a jamming or an interference event. Consequently, the SCE selects the next channel frequency and informs the recipient of the next frequency and the time to switch. Transmitter and receiver switch the frequency at the specified time.

Rule 1(b): Lower the modulation scheme when SNR is low: shown between lines 5(a) and 5(d) in Algorithm 2. The process is similar to Rule 1(a). The next modulation is selected by downgrading the current modulation by order of 2.

Rule 1(c): Raise the modulation scheme when SNR is high: shown between lines 6(a) and 6(d) in Algorithm 2. The process is similar to the one used for Rule 1(a). The next modulation is selected by upgrading the modulation by order of 2.

4.3 Cryptographic Cognitive Engine

The functionality of the cryptographic cognitive engine is to generate cryptographic material, verify message integrity, and communicate detected threats to the MCE.

4.3.1 Cryptographic key generation

The integrity of all PTC messages should be preserved so that it is not possible for an attacker to tamper with a message. PTC specifies a 32-bit HMAC field to detect any tampering attempts. It further uses a 8-bit sequence number to avoid replay attacks. Since the 8-bit time sequence number repeats every 256 message (i.e., because the range of the sequence number is 0 to 255 $(2^8 - 1)$), an attacker can replay a status message every second until it matches with the correct sequence number. For example, when the train is expecting a *STOP* status, an attacker can replay an earlier message with *PROCEED* status and can cause the train to collide with another train.

The replay attack can be minimized by changing the seed of the hash function frequently. To modify the key based on a key chain, an algorithm based on the Time Efficient Stream Loss-tolerant Authentication (TESLA) [48] is used. That decision was based on its low computational and transmission overhead, as well as on its high tolerance to packet losses.

TESLA Protocol

In TESLA, the sender first decides on the number of keys N and computes a chain of keys based on the one-way key generation algorithm shown in Equation 4.1. This algorithm is inspired by Lamport's One-Time Password Scheme [7].

$$K_0 = IV \quad \&\& \quad \forall\, i > 0 \quad K_i = F(K_{i-1}) \tag{4.1}$$

The total transmission period is divided into N periods, and each period is assigned with a key starting from the last key of the key chain. That is, the first time slot is assigned with the key k_N, the second slot with the key k_{N-1} and so on. The transmitter uses the key that is assigned to the current timeslot and uses it to generate the hash for transmission. Once the receiver gets a message, it puts in a buffer, since it does not have the key to authenticate the associated packet. A short time later, the sender discloses the key, and the receiver validates the packet based on that key. The approach of deriving the key chain using a forward function and using it backward is meant to avoid any attacker from obtaining the next key using the disclosed key.

The PTC usage scenario is different from the general broadcasting scenario because,

1. Objective of PTC communication security is to preserve the integrity and avoid replay attacks. The TESLA algorithm's primary objective is to ensure authentication of the sender.

2. It is safe to assume all trains to be trustworthy.

3. Trains can store a chain of keys in their on-board database

4. Messages should follow PTC packets specification. Therefore, transmitting the key has to be done as an additional transmission, and this will add a huge overhead.

To account for these differences and their effects, the TESLA algorithm is enhanced to suit for the PTC use case.

Enhanced TESLA protocol

The main differences between TESLA and Enhanced TESLA protocol are:

1. In Enhanced TESLA protocol, the transmitter and receiver generate the key chain. Therefore, no key transmission is required.

2. Enhanced TESLA protocol changes both the salt and the hash generation algorithm randomly with time.

Seed chain is derived using a Lamport scheme similar to TESLA. The derivation of algorithms in enhanced TESLA follows a similar approach to the seed derivation. Algorithm 4.3.1 describes the process used in deriving the algorithm. The derivation process starts with an algorithm derivation seed being provided as part of the initialization vector. This algorithm derivation seed is then hashed and rehashed successively following the same approach as traditional TESLA in deriving salts. Each generated result is then operated via the modulus operation with the modulus exponent equated to the total number of available hashing algorithms. The process continues for the entire time range and during each time step within the communication period.

$$\begin{aligned} seed_0 = IV_A \&\& \\ \forall \quad i > 0 \quad (seed_i = hash(seed_{i-1}) \quad \&\& \quad AlgoSelect(i) = seed(i)\%n) \end{aligned} \tag{4.2}$$

The derived seed chain and the equation sequence are loaded to the train's on-board

database. The train's CR uses the seed and the algorithm assigned for the message generation time to proceed with integrity signing and verification. The complete Enhanced TESLA algorithm is shown in Algorithm 4.3.1.

4.3.2 Threat Analysis

Another functionality provided by the cryptographic CE is to analyze potential threats to messaging communication. Fig. 4.6 shows the threat detection procedure of the current threat module. The threat analysis module uses the Cyclic Redundancy Check (CRC) and hash validation to identify attacks. For each message, the threat analysis module checks the CRC and the hash value. If both the CRC and the hash values are correct, the message is identified as a proper message. If either the CRC or the hash value is incorrect, it compares the message with the previous messages to assess whether there is a potential replay attack. The current threat module can detect the following three types of attacks.

1. **Type 1 - Replay Attack**

   In a replay attack, the attacker captures a previously sent message and transmits it later to the intended receiver. The original PTC specification defines the usage of a static salt value for the hash function, which makes it harder to detect a replay attack. In contrast, SIRT includes changing the cryptographic seed value with time, which makes it much more efficient in detecting the replay attempts since when the replayed message arrives, the system will likely be using a different hash key. The threat analysis component is described in more detail in [49].

   As shown in Equation 4.3, a message is identified as a potential replay if the message is syntactically correct, has a valid CRC value, but the hash value does not match the hash value corresponding to the key used by the system at the time of the message reception. CCE then check the hash value with the stored historical values to see whether the hash is generated using a previous salt value, which indicates a potential replay attack.

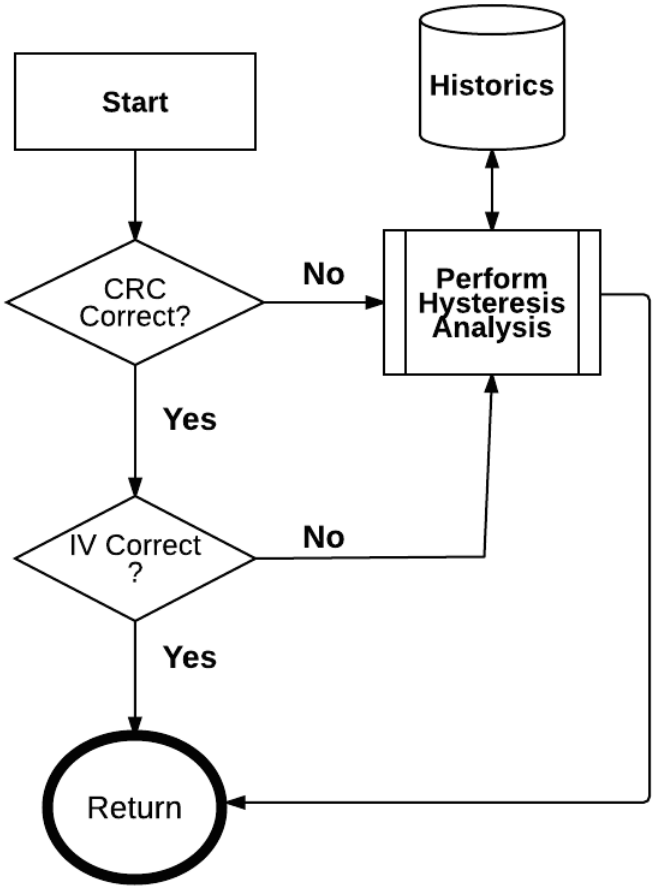

Figure 4.6: Detection Process

$$ReplayDetected : (CRC = VALID)\&(hash = INVALID)\& \\ \&(M = SyntacticallyCorrect)\& \exists t < t_{current} \quad hash_t = VALID \tag{4.3}$$

2. **Type 2 - Message Corruption Attack**

   In the message corruption attack, the attacker captures a message, modifies its content and transmits it back to the intended recipient. This attack is a special case of a message modification attack performed by an unsophisticated attacker that does not possess the ability to modify the message content while avoiding a CRC corruption.

As shown in Equation 4.4, the message corruption is detected by CRC invalidation. Our current message corruption detection algorithm does not have sufficient intelligent to distinguish between intentional message corruption or unintentional message corruption.

$$MessageCorruption : (CRC = INVALID)\&(hash = VALID|INVALID) \\ \&(M = SyntacticallyCorrect) \tag{4.4}$$

3. **Type 3 - Message Guessing Attack**

   Similar to the message corruption attack, the attacker tries to modify the content of the message and send it back to the receiver. However, in this case, the attacker is more sophisticated and capable of producing modified messages that do not fail the CRC check. Such an attacker has information about the CRC generation and key generation algorithms, but does not have the initial seed value to generate the key chain. Therefore, the attacker guesses the initial key, generates key chains, and uses it for message transmission.

   Because the attacker knows of all the algorithms, he can generate a syntactically correct message. However, as mentioned earlier CCE the keys are changed frequently, forcing the attacker to guess the key used at the period in which the modified message will be received. As a result, the attacker's modified message will fail the integrity validation process. The logic used to detect this kind of attacks as shown in Equation 4.5, in which hash invalidation happens while the message is syntactically correct, it has a valid time stamp and a valid CRC value.

$$GuessingAttack : (CRC = VALID)\&(hash = INVALID) \\ \&CurrentTimestamp\&(M = SyntacticallyCorrect) \tag{4.5}$$

Algorithm 2 Spectrum Management Algorithm

Symbols

- EMR_1 = Average message error ratio within last 1 minute (ratio between number of error messages to total number of messages per second)

  EMR_5 = Average message error ratio within last 5 minutes

  CMod = Current modulation scheme (2 for BPSK, 4 for QPSK and 8 for 8PSK)

  Nmod = Next modulation scheme

  CFreq = Current frequency channel

  NFreq =Next frequency channel

  MR =Message rate

  Rx= Receiver (where the spectrum is adjusted)

  TX= Transmitter

  $T_{monitor}$ =Spectrum monitoring interval

Algorithm

While(True) {

1. $MR_1, MR_5, ER_1, ER_5 := Monitor\_spectrum()$
2. $EMR_1 = ER_1/MR_1$
3. $EMR_5 = ER_5/MR_5$
4. **Rule 1(a): Hop the frequency when the channel is jammed**
   if $(MR_1 \leq MR_{threshold})$
   (a) $NFreq := find\_non\_congested\_channel()$
   (b) $RX \rightarrow TX : Change\_TX\_Freq(NFreq)$
   (c) $RX.set\_rx\_freq(NFreq)$
   (d) CFreq:=NFreq
5. **Rule 1(b): Lower the modulation scheme when SNR is low**
   else if $(EMR_1 \geq MER_{threshold1})$
   (a) $NMod := CMod/2$
   (b) $RX \rightarrow TX : Change\_TX\_Mod(NMod)$
   (c) $RX.set\_rx\_mod(NMod)$
   (d) CMod:=NMod
6. **Rule 1(c): Raise the modulation scheme when SNR is high**
   else if $(EMR_1 \leq MER_{threshold2} \&\& MER_5 \leq MER_{threshold3}$)
   (a) $NMod := CMod * 2$
   (b) $RX \rightarrow TX : Change\_TX\_Mod(NMod)$
   (c) $RX.set\_rx\_mod(NMod)$
   (d) CMod:=NMod
7. Sleep($T_{monitor}$)}

Algorithm 3 Enhanced TESLA protocol

Symbols

Tx = Transmitter

Rx = Receiver

AlgoSelect = Algorithm Selection

IV = Initialization Vector

n =length of the salt chain and the algorithm chain

Preconditions

1. An Algorithm seed, IV_A and salt derivation seed IV_S, is provided as part of the bootstrap process
2. The communication devices are utilizing indirect time synchronization such as the Global Positioning System (GPS)

Algorithm

1. $\rightarrow Tx, Rx : IV_S, IV_A$ Securely
2. The generate salt chain using Equation 4.1 where $F() = hash()$ and algorithm chain using Equation 4.2.
3. Assign salt values and algorithms to time slots such that $\forall\, i = 1 : n, Time\_slot_i \leftarrow salt_{n-i}, Algorithm_{n-i}$

Chapter 5: Implementation and Testing

5.1 Prototype development

To validate the usability of SIRT, I tested the functionality and the performance of the SIRT system by prototyping the train radio, the signaling point radio and the WIU radio as SDR applications. The detailed design architecture of SIRT is shown in Fig. 5.1. As the diagram show, I implemented the different Cognitive Engine (CE)s in SIRT as separate modules. Each module runs as separate sub-process communicates through inter-process communication.

For the communication between the physical layer and the sub-processes of SIRT and the communication between the sub-processes, I used an open source middleware called REDIS [50]. REDIS is a publication subscription service that can be utilized as an *in memory* message passing.

I used GNURadio [51] to develop the physical layer. GNURadio is an open source software development toolkit that provides signal processing blocks required to implement Software Defined Radio applications. Signal processing blocks in GNURadio are developed in C++. Connections between signal processing blocks are programmed using Python. I developed two GNURadio blocks that provide the link between the physical layer to the REDIS middleware which is *redis source v* and *redis sink v* explained in Section 5.1.2.

5.1.1 REDIS middleware as the transport layer

In REDIS, I defined three channels as shown below.

1. TxCh: This channel is used to transmit application messages to a different radio. Here the applications inject message that needs to be sent to the 'TxCh'. The *redis*

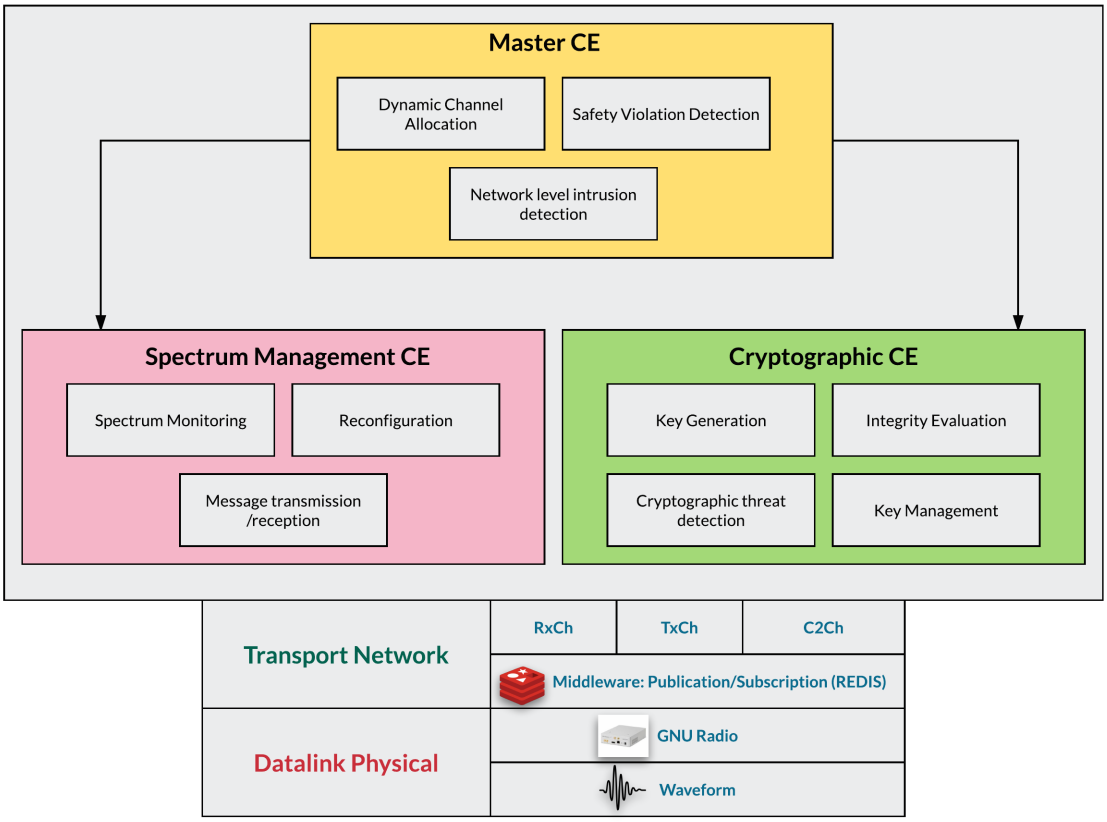

Figure 5.1: Design Architecture of the Cognitive Radio

source v block implemented in GNURadio fetches messages from 'TxCh'. The GNU Radio blocks process the message and transmit it.

2. RxCh: This channel is used for an application to get the messages received from other radios. Once a message is received, the GNURadio layer demodulates, decodes and adds it to the channel Redis 'RxCh'. The applications subscribing to the RxCh channel receives the message.

3. C2Ch: Transfers control messages between cognitive engines. When the receiver application gets that message from the RxCh channel, the receiver's MCE categorize it as a user message or a control message. If it is a control message, then the message

is published to the 'C2Ch'.

5.1.2 Physical layer implementation

Physical layer implementation was done in two stages. At first, I implemented a phase modulation based transceiver system. However, due to some of the limitations that occurred in that system (which will be explained momentarily), I decided to use a tagged stream and OFDM-based transceiver system.

Setup 1: Phase modulation transceiver

This transceiver has a transmitter and a receiver that runs in parallel using two different center frequencies. The application layer adds messages to the REDIS Channel. GNURadio layer gets the message from the REDIS channel, encode it, modulate it and pass it to the USRP interface block. This block transmits the message over the air. I used three modulation schemes that can be changed dynamically while the radio is in operation. The modulations are the BPSK, QPSK and 8PSK.

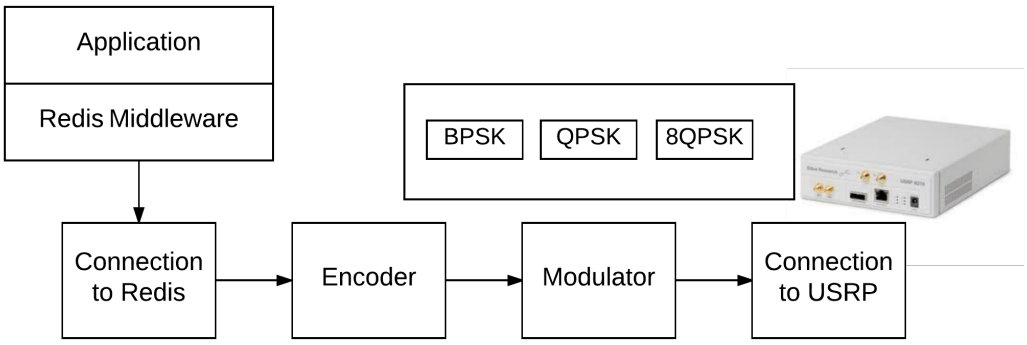

Figure 5.2: Transmitter Architecture

The GNUradio flow graph corresponding to the transmitter path is shown in Figure 5.3. The packet encoder used here convert the message to a bit stream, add the CRC value and add required packet header fields. The constellation modulator modulates the formatted

packet based on the selected constellation object (this can be either BPSK, QPSK, and 8PSK and can be changed dynamically). When a modulation change happens, the excess bandwidth of the RRC filter also has to be changed so that the signal is formatted correctly to have fewer propagation losses.

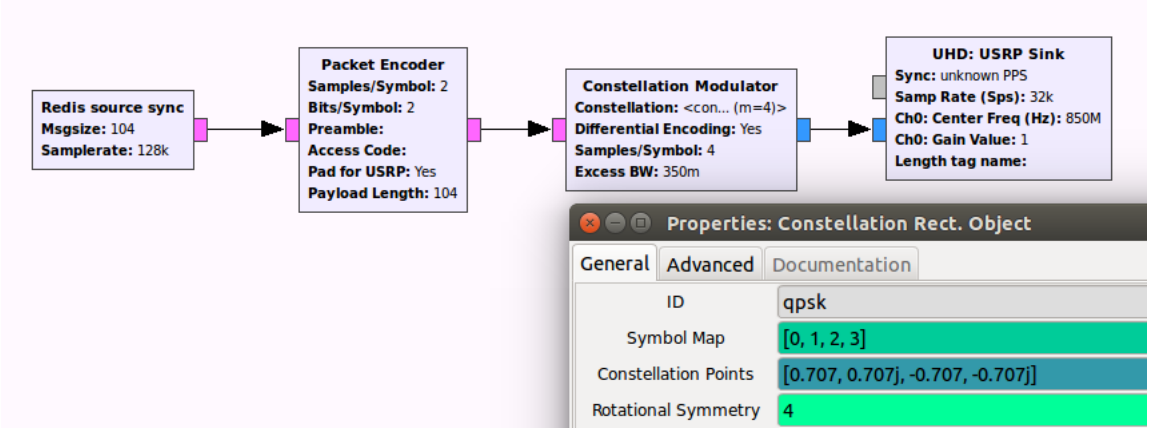

Figure 5.3: Transmitter flowgraph in GNURadio

Figure 5.4 shows the receiver end radio layer. The receiver's flow-graph starts with a block that interface to the USRP. The flowgraph follows up with the demodulator, packet decoder and the interface to the REDIS channel.

Fig. 5.5 shows the receiver flowgraph. Due to the characteristics of the phase modulation, the receiver requires precise time, frequency and phase synchronization. In the current receiver path, I implemented a *polyphase clock synchronization* for clock synchronization and a *Costas loop* to synchronize and lock to the correct phase and frequency. Further, a *15-tap constant modulus algorithm (CMA) equalizer* is used to remove any multipath effect. Once properly synchronized, the signal is demodulated, decoded and written to a REDIS channel. Writing to the REDIS channel is done by the *Redis sink v1* block. The application level cognitive engines can read this message from the REDIS channel.

Polyphase clock synchronization: Objective of this block is to find the best time to

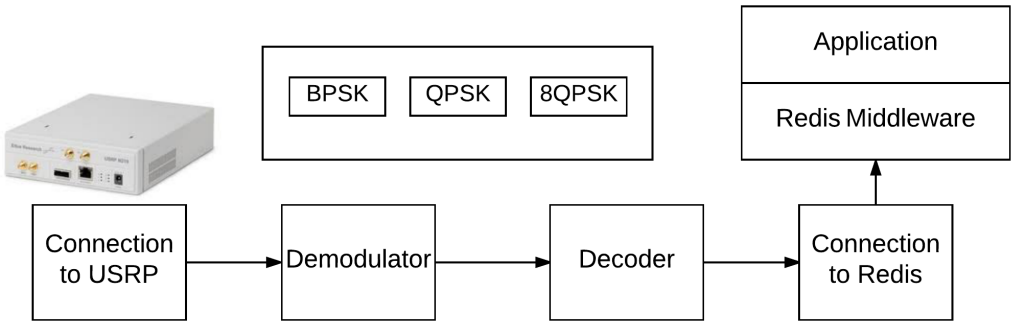

Figure 5.4: Receiver Architecture

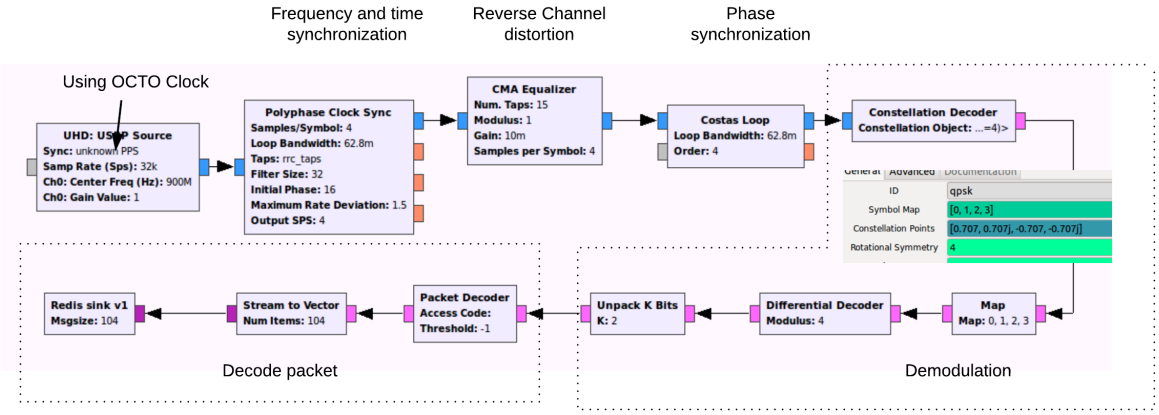

Figure 5.5: Receiver implementation in GNURadio

sample the signal. This block has two filter banks. In time domain the 1st bank has Signals pulse shaping matched filter, and the 2nd bank has the derivatives of the first filter bank. In time domain first filter bank has sync shaped signals and the second filter bank has a zero at the peak of the sync signal (because the derivative of the sync signal has zero at its peak). However, when the signal has a timing offset derivative

filter does not show a point at zero. Therefore it uses a series of filters with different phase values. The block has a second order control loop that starts at one of the filters and calculates the output as the error signal. It then moves its way up or down the bank of filters proportionally to the error signal until the error is close to 0. The filter value close to 0 is the ideal time for sampling. Apart from the time synchronization, this block removes inter-symbol interference and down samples the signal to 1 sps [52].

CMA equalizer : This equalizer uses the Constant Modulus Algorithm that works on signals with constant amplitude. This block has taps to utilize in the equalizer and uses a second order control loop to adjust the taps to invert the channel effect by enhancing or suppressing certain frequencies. This equalizer converges all the phase values to a unit circle [52].

Costas loop : CMA equalizer does the coarse frequency correction. However, the signal still needs fine frequency correction and phase correction. This functionality is done by the Costa's loop. Costa's loop has a second order control loop that keeps updating the frequency and phase until the error compared to expected constellation is minimum. This can lock into BPSK, QPSK and 8PSK modulations [52].

Reconfiguration of the radio parameters is done in two stages.

- When the receiver detects a need for reconfiguring a parameter it sets up a timer, adds the timer value and the necessary reconfiguration details in a request message and sends it to the transmitter.

- Once the transmitter receives the message, it checks the timestamp and extracts the reconfiguration parameters. At the time specified in the timestamp both the transmitter and the receiver reconfigures the radio parameter to the same value. This is done to reduce the packet losses due to having two different parameters at the two ends. The parameter reconfiguration is done as shown in algorithm 4.

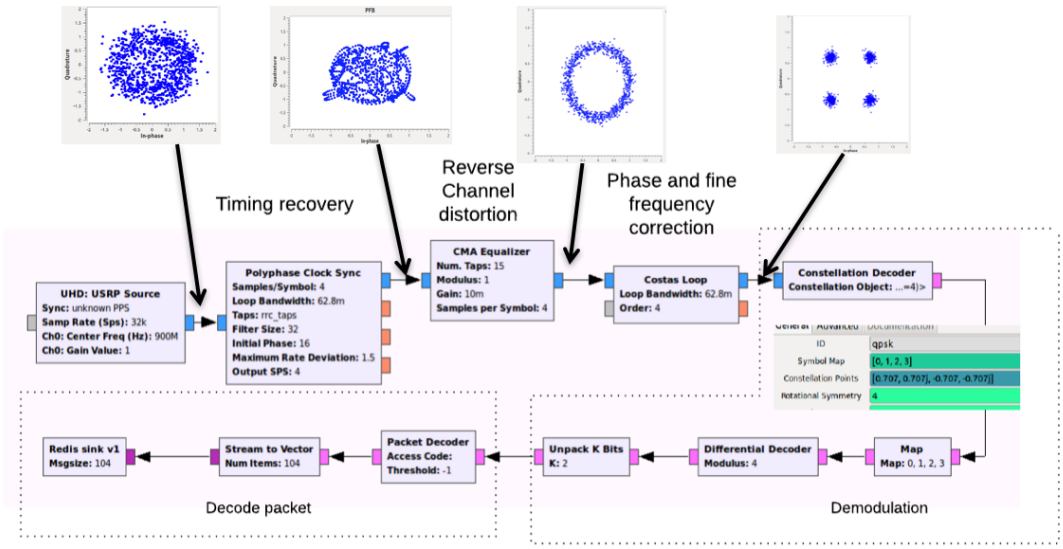

Figure 5.6: Receiver with constellation diagrams at different points

Algorithm 4 Radio parameter reconfiguration in GNURadio

flowgrpah.lock()

block().*disconnect*

change_parameters()

block().*reconnect*

flowgraph.unlock()

Limitations with the transceiver

Although the transceiver implementation described above provided me with an underlying platform to test the SIRT functionality, there were some restrictions in the design. The main limitation was the packet encoder block. As shown in Figure 5.7 a general GNURadio block has an input buffer and an output buffer. The input buffer of one block is connected to the output buffer of the next block. If there is no space in the output buffer, it does

not take input. This process provides a back pressure to the previous block so that the processing speed of the blocks are controlled based on the speed of the USRP (hardware speed).

However, the encoder uses a FIFO buffer. FIFO does not indicate that it is full. It keeps growing until it is full and starts dropping packets. In the original design, the queue length was set to 2. Therefore flow graph began to drop packets at the interface between the *redis source* and the encoder. To improve this, I increased the queue length of the encoder to a larger value. Although this stopped the packet drop at the interface, this resulted in an increase in the total propagation delay due to the increase in the queuing delay.

Figure 5.7: GNURadio encoder

The Figure 5.8 shows the Packet loss percentage for every 1000 packets with time when the queue length was 2. As shown in the figure BPSK has losses closer to 15%, and QPSK has loss rate closer to 10%. As shown in Figure 5.9 the propagation delay for BPSK is around 9mS. The propagation delay for QPSK is around 6mS.

When the queue length was increased to 20000, the packet loss percentage drops significantly. As shown in Figure 5.10 the loss percentage dropped closer to 0% for all the three modulations. However as shown in Figure 5.11 when the queue size was increased the propagation delay started to grow significantly.

GNURadio platform was originally designed to handle stream data, and it does not consider managing packetized data. However, there is a newly added feature in GNURadio

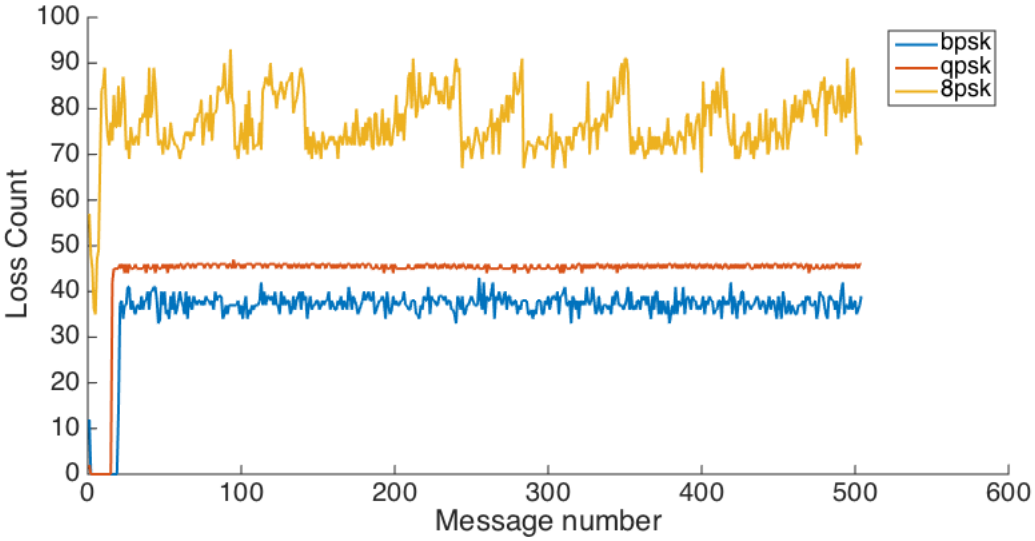

Figure 5.8: Packet loss before the increase of the queue size

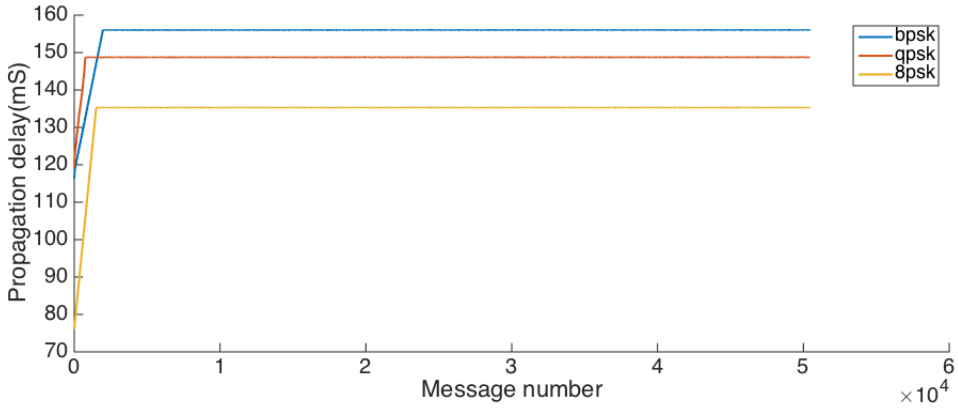

Figure 5.9: Propagation delay before the increase of the queue size

called 'tagged streams' [53] that can work on packetized data. Therefore I decided to improve my transmitter and receiver using the tagged stream based approach.

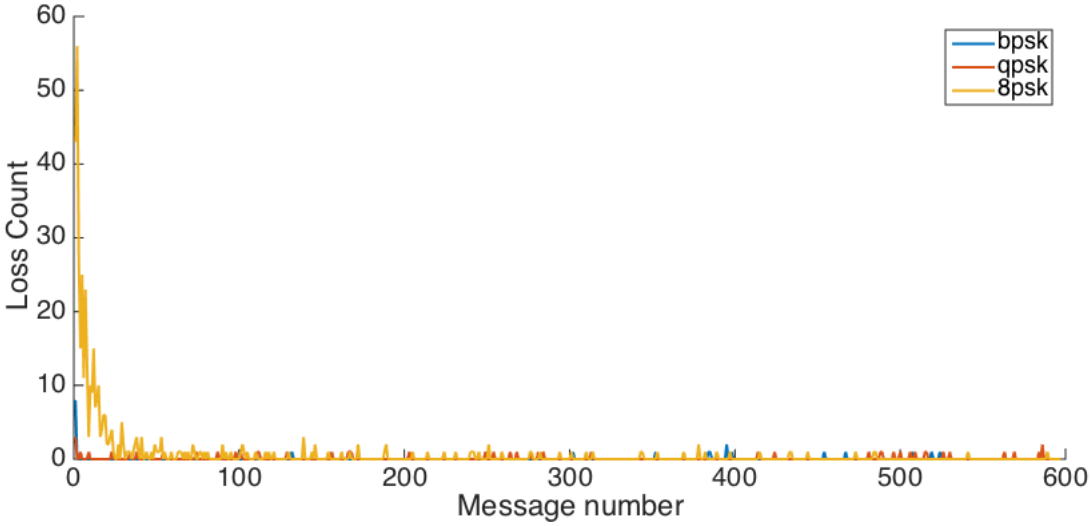

Figure 5.10: Packet loss after the increase of the queue size

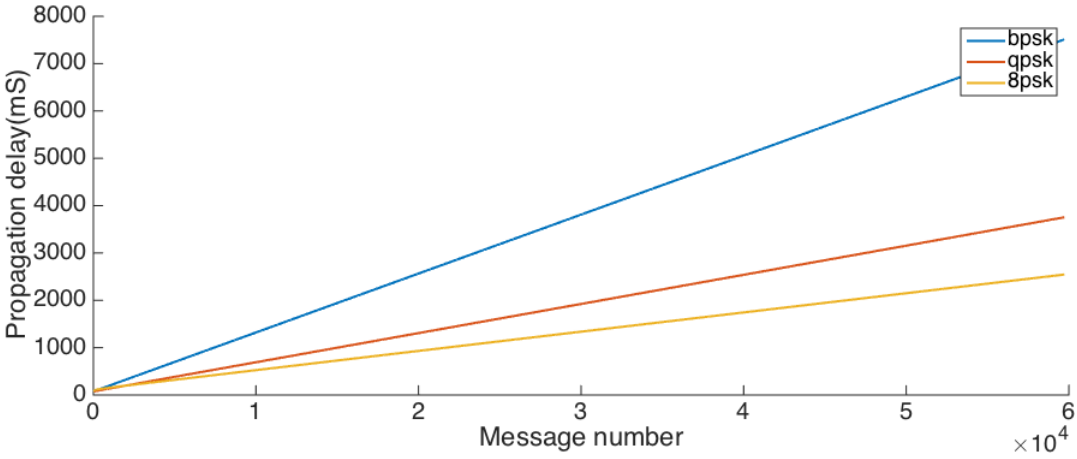

Figure 5.11: Propagation delay after the increase of the queue size

Setup 2: Tagged Stream and OFDM-based Transceiver

I decided to use a tagged stream based OFDM transceiver and receiver developed by Matt Ettus et. al [54]. It solved the queuing problem in the earlier system (given in 5.1.2)

and also increased the data rate significantly using the same bandwidth. The transmitter architecture is shown in Figure 5.12 and the receiver architecture is shown in Figure 5.13.

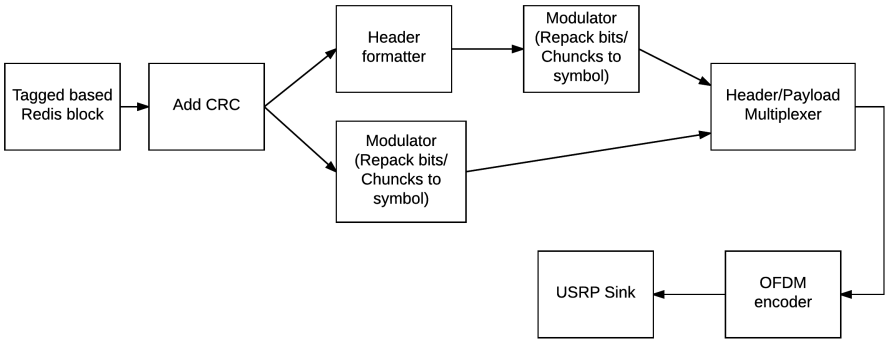

Figure 5.12: Tagged stream based transmitter architecture

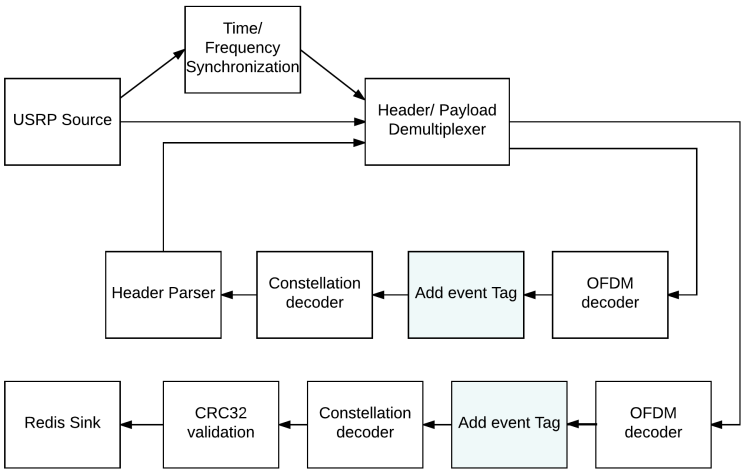

Figure 5.13: Tagged stream based receiver architecture

As shown in Figure 5.12 the transmitter flowgraph starts with a tagged stream based REDIS block. Similar to the earlier implementation this block provides the interface from

the application layer messages to the GNURadio radio layer. In this implementation, I modified it to add a $'length\_tag'$ to the message. Packetized messages inside the stream are identified using these tags, and consequent blocks use these to process the message.

Next block *Add CRC* extract messages in the tag, calculate and add a CRC value to it and update the *length_tag* value by adding the CRC length. The output of this block is directed to two paths. In the first path, there is a *Header formatter* block which generates the packet header based on the header format we specified. One unique fact about this implementation is that it separately modulates the header and payload. Therefore it is possible to use two different modulation schemes for the header and the payload. The modulation is done using *repack bits* and *chunks to symbols* blocks. The separately modulated two streams are then multiplexed at *Header/Payload Multiplexer*.

After multiplexing the signal goes through the OFDM encoder. The functionality of this block is explained in more details in 5.1.2. The output of the OFDM encoder is fed to the *USRP sink* for transmitting.

The receiver flowgraph starts with *USRP Source* which receives messages. The output of the *USRP Source* is directed to two paths. In the first path, it goes to time and frequency synchronization block. The output of this block is used as a trigger to the *Header/Payload demiltiplexer* block to identify the beginning of the packet. The second output goes directly to the *Header/Payload demiltiplexer*. This block extracts the header stream (using OFDM decoder, constellation decoder, and header parser) and uses it as a feedback loop to decode the payload. The block *Add event tag* is a block that I have created to add reconfiguration tags to the flow. More details about this are described in section 5.1.2. The *OFDM decoder* block decodes the OFDM symbols (more information about how this is done is explained in section 5.1.2) . The constellation decoder block demodulates the signal based on the selected modulation scheme and the *Header parser* block extract the header from the data stream.

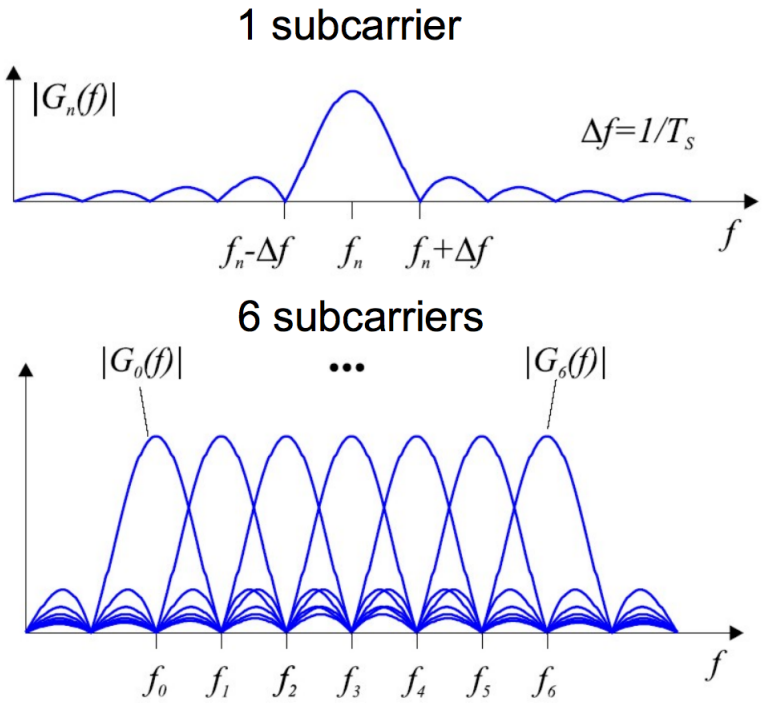

Figure 5.14: Carrier allocation in OFDM [4]

Orthogonal Frequency Division Multiplexing (OFDM) encoder and decoder

OFDM systems share data among several sub-carriers within the same channel and transmit them simultaneously. Each carrier can be modulated with a conventional modulation scheme (such as BPSK,QPSK, etc.) at a low symbol rate. The sub-carriers are orthogonal to each other as shown in Figure 5.14 [54, 55].

Mathematically a carrier can be represented as in equation 5.1

$$s(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k t} \qquad t \equiv [0, T_{os}] \tag{5.1}$$

For two signals to be orthogonal to each other, the product of the two signals should be

equal to zero as shown in equation 5.2.

$$\int_0^{T_{OS}} g_1(t).g_2(t)dt = 0 \tag{5.2}$$

If we formulate the orthogonality constraints for two OFDM sub-carriers p,q $p \neq q$ using the above two equations it can be shown as in equation 5.3.

$$\int_0^{T_{OS}} e^{j2\pi f_p t}.e^{-j2\pi f_q t}.dt = 0 \tag{5.3}$$

The OFDM encoder encapsulates three blocks. They are,

1. OFDM Carrier allocator: Distribute a single bit stream to multiple streams and allocate them to different sub-carriers. Also, add pilot symbols to help channel estimation at the receiver.

2. FFT (reverse): This block performs the Inverse Fast Fourier transformation on each bit stream to achieve orthogonality between subcarriers.

$$x(n) = IFFT\{X(k)\} n = 0, 1, 2....N-1 \tag{5.4}$$

3. OFDM Cyclic Prefix: Add a cyclic prefix to the OFDM signal. Cyclic Prefix is a guard interval added to avoid inter-symbol interference between two OFDM symbols [56]. It also performs pulse shaping. Using a cyclic prefix at the transmitter allows using a simple equalizer in the frequency domain at the receiver.

$$x_f = \begin{cases} x(N+n), n = -N_g, -N_g+1, ..., -1 \\ x(n), n = 0, 1, ..., N-1 \end{cases} \tag{5.5}$$

The OFDM decoder encapsulates the following blocks.

1. FFT: Perform Fast Fourier transformation on the OFDM waveform to convert the output to the frequency domain.

$$Y(k) = FFT\{y(n)\} k = 0, 1, 2, ...N - 1 \tag{5.6}$$

2. OFDM channel estimation and frame equilization:

   The transmitted signal $x_f(n)$ changes its properties according to the transmission medium h(n). In addition it gets noise w(n) added to it. Therefore the signal the reciever sees y(n) can be represesnted as in equation 5.7.

$$y_f = x_f(n) \otimes h(n) + w(n) \tag{5.7}$$

   When performing the Fast Fourier transformation and convert into frequency domain the signal can be represented as equation 5.8.

$$Y(k) = X(k)H(k) + I(k) + W(k) k = 0, 1, ..., N - 1 \tag{5.8}$$

   The objective of channel estimation is to invert the effect of the channel and get an estimate for the transmitted signal as given in equation 5.9.

$$X_e(k) = \frac{Y(k)}{H_e(k)} \qquad k = 0, 1, ...N - 1 \tag{5.9}$$

   GNURadio uses a pilot symbol based equalization scheme. Pilot symbols are sent periodically in the time domain and provide an estimate of the channel at given locations within a subframe [57].

3. OFDM serializer: Performs the inverse operation of the carrier allocator. That is it combines all the parallel streams and create one serial stream.

Changing GNURadio block codes to allow reconfiguration based on an event

The main challenge faced with developing the tagged stream based transceiver was its incompatibility with module reconfiguration. The program failed when I lock the flowgraph to do the reconfiguration. The problem was that the tagged stream blocks do not add any tag to the data stream that processes during the locking period, and the following blocks that expect tags to process data fail because of the missing tag. To solve this problem, there were two approaches that I could take. The first approach is to reconfigure parameters without stopping (locking) the flowgraph. Second is to re-engineer all the blocks that process tags to ignore any packets without tags.

I decided to go with the first approach since the second method is inefficient. One challenge that I had to face with this approach was the synchronization between different blocks when a particular reconfiguration required modifications in more than one block. To do that I created a block that I added to the flowgraph ahead of the blocks that need reconfiguration. This block listens to a 'REDIS' channel. When there is a parameter reconfiguration request, I add an event to this channel. Based on the event from the 'REDIS' channel this block generates a tag with the necessary change. The tag propagates through the flowgraph. The blocks that perform reconfiguration reads this tag value and change its parameters based on the tag at the specified time. Eg: To change the modulation I modified the *repack_kbits*, *header_formatter* and *chuncks_to_symbols* blocks at the transmitter and *OFDM_equilizer*, *repack_kbits*, *constellation_decoder* block at the receiver.

5.2 Experimental Validation

5.2.1 Experimental validation of the Spectrum Management Cognitive Engine

To validate the functionality of the SCE, I used three Universal Software Radio Peripheral (USRP) N210 (SDRs developed by Ettus Research and National Instrument [58]) to represent a train, a signaling point, and an attacker/noise generating source. I conducted

my experiments inside a noise cancellation chamber to isolate the test from the uncontrolled noise and to provide clock synchronization between the USRPs I used Ettus GPSDO clock. My experimental setup is shown in Figure 5.15.

Trains and signaling points have two-way radio communications. Therefore the receiver can send spectrum management control messages to the transmitter using the reverse channel. As the first step of my experiment, I ran the communication with BPSK, QPSK and 8PSK modulation schemes under different noise levels to get the decision-making values for spectrum monitoring and to calibrate the spectrum management cognitive engine. I used the third radio as the noise generator that adds noise to the channel and finally jams the channel with a high powered signal.

As shown in Figure 5.16, all three modulations have proper separation between their constellation points (Symbol values in the phase diagram) when the experiment is run without noise. However, the symbols start to disperse when I added noise. Because BPSK has only two symbols, it works better even at higher noise levels. The noise immunity decreases when the order of the modulation increases.

I set up the train radio to transmit packets using 100kHz bandwidth with BPSK, QPSK and 8PSK modulations. I generated Gaussian noise and applied it to one channel at a time with varying noise levels between no noise to -10 dB. I observed the total message rate (messages/minute) at the decoder and the total CRC error rate (messages/minute) at the decoder for each modulation scheme under different noise levels.

Figure 5.17 shows received message rate and the received message error rate variation with time and Figure 5.18 shows the cumulative message count and error message count with time for BPSK modulation. I set the message transmission rate to be 12 messages per second as shown in Figure 5.17. The BPSK modulated signal channel does not show any errors until a noise floor of -18 dB occurs. At -18 dB the channel starts to degrade, and as shown in Figure 5.17, the packet error rate increases, and the total message rate decreased as noise increases. Also, the BPSK modulation was able to recover and regain normal operation when the noise is fully removed.

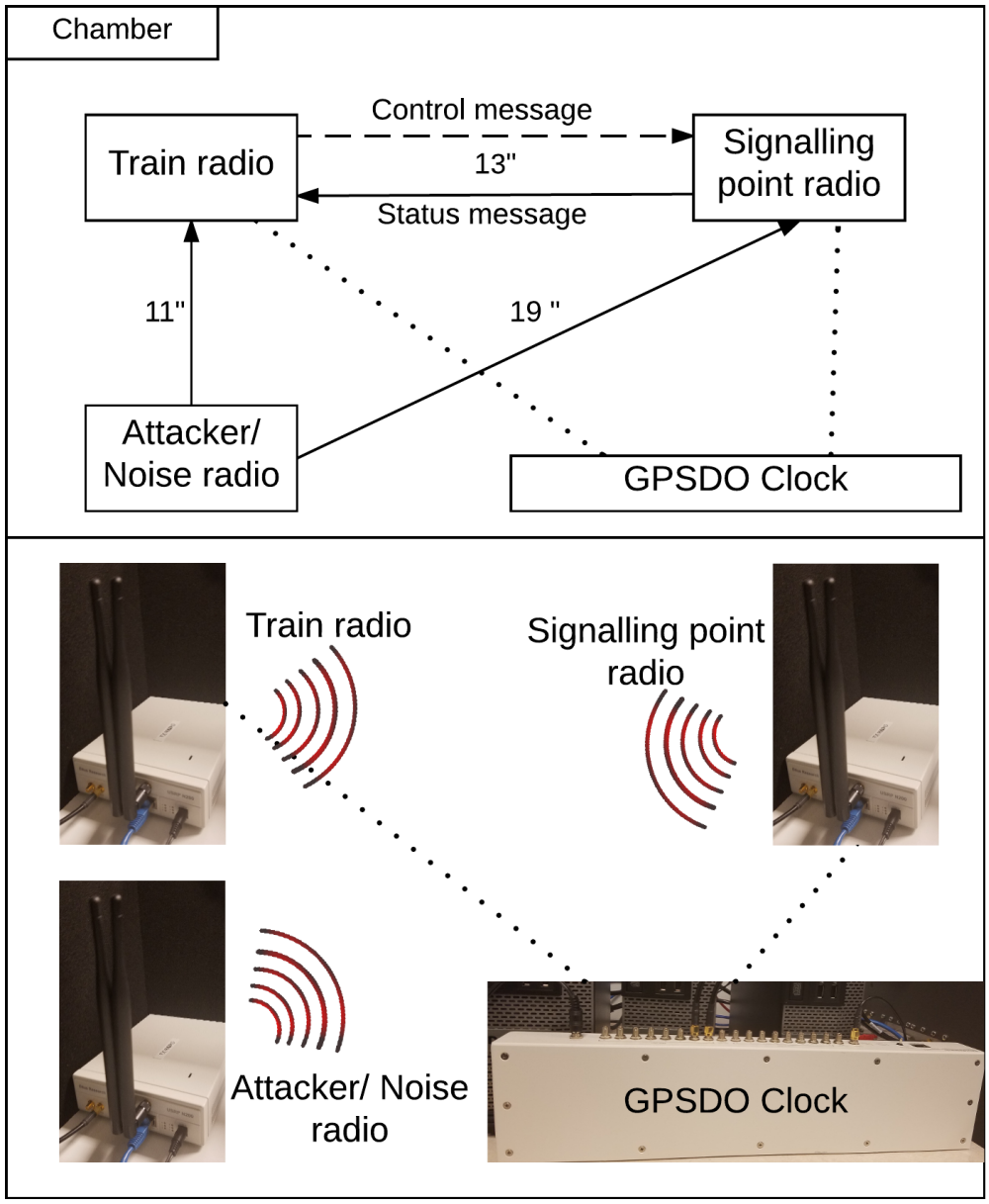

Figure 5.15: Test Setup

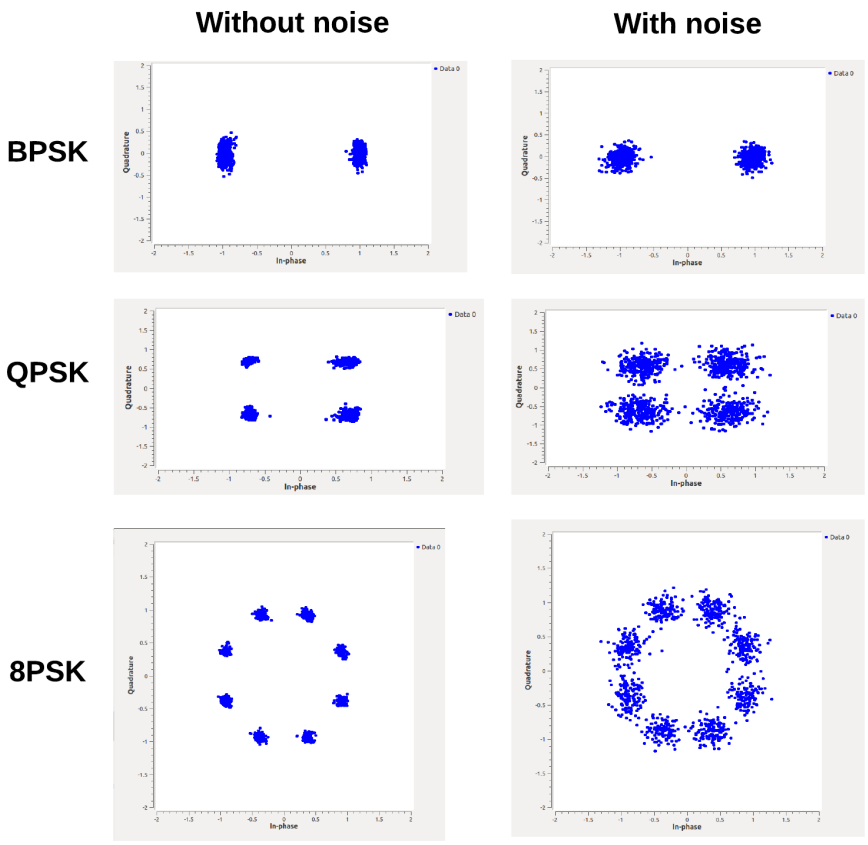

Figure 5.16: Different modulation schemes without noise and with noise

The QPSK modulation doubled the message rate using the same channel bandwidth, as shown in Figure 5.19. However, it was stable only until the noise level reached -25 dB. At the -25 dB noise level, QPSK suffered packet errors. However, the error rate was less than 10%, and the channels were stable. When the noise level was increased to -20 dB, the error rate increased and at -18 dB noise level, the message rate dropped significantly up to 2 messages per second due to synchronization losses. The QPSK modulated channel was able to recover after the noise was removed. Figure 5.20 shows the cumulative message count and error message count with time for QPSK modulation.

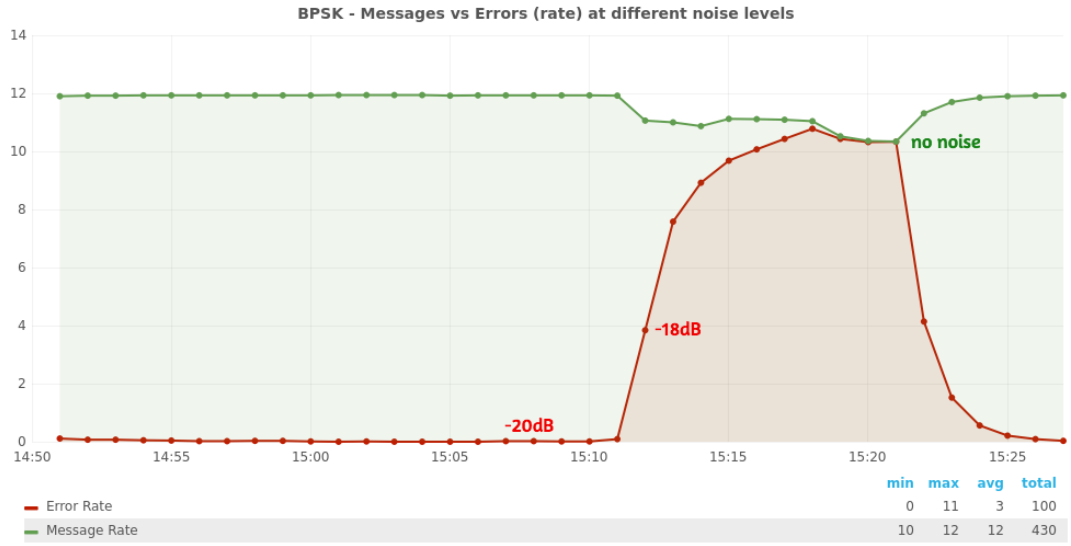

Figure 5.17: Message rate and error rate for BPSK under different noise levels

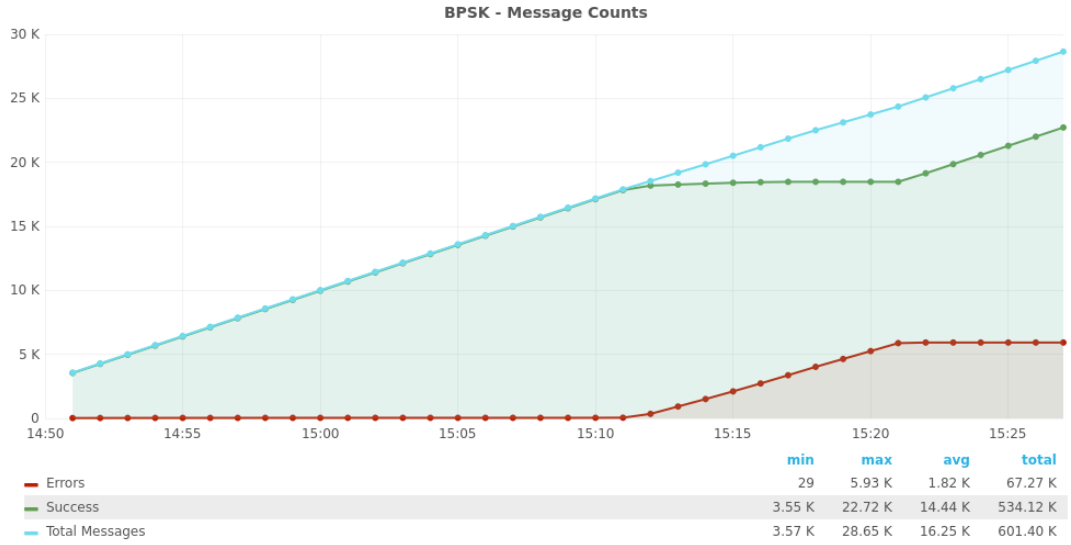

Figure 5.18: Message count and error count for BPSK under different noise levels

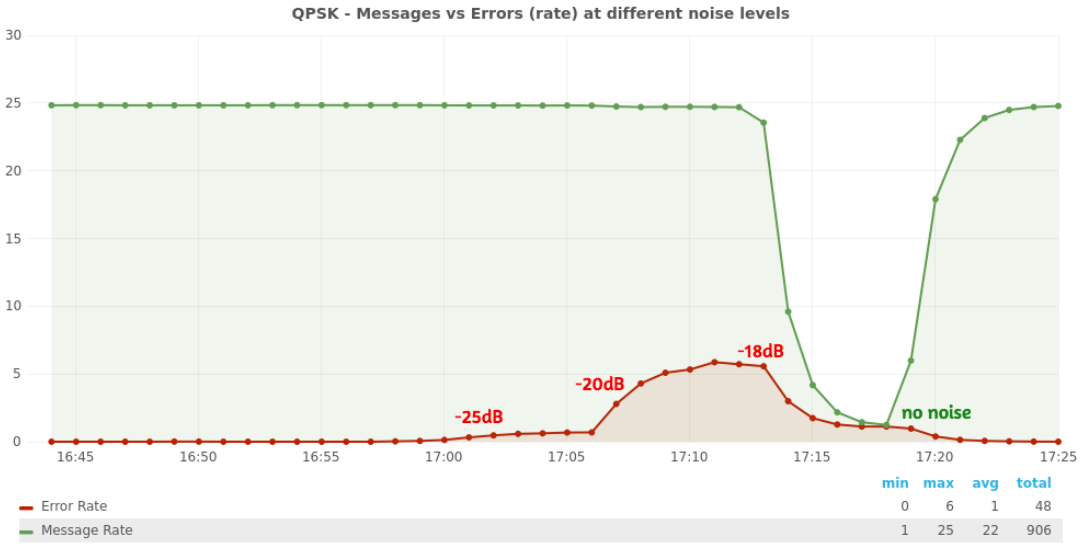

Figure 5.19: Message rate and error rate for QPSK under different noise levels

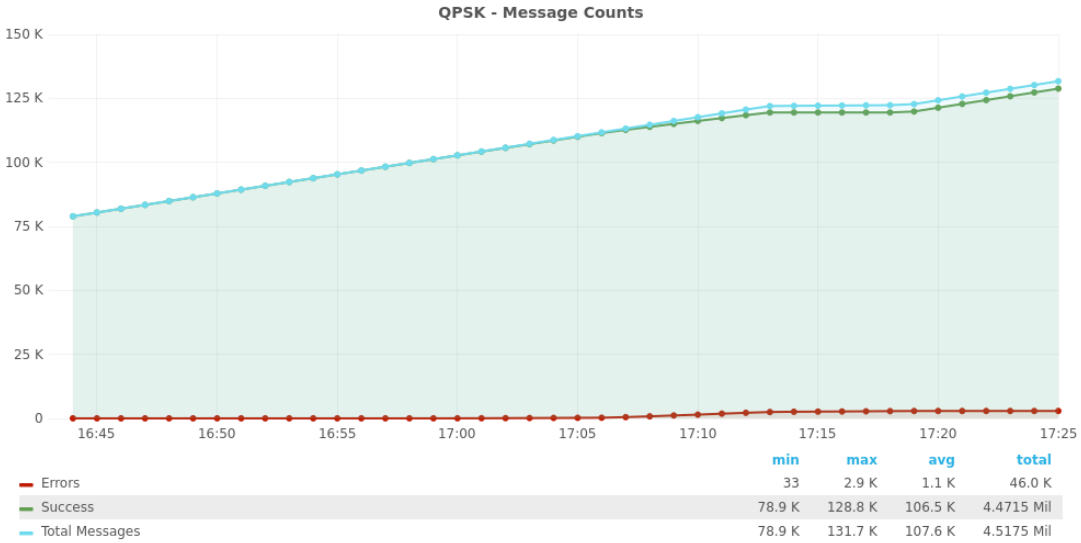

Figure 5.20: Message count and error count for QPSK under different noise levels

8PSK modulation scheme has a higher message rate compared to BPSK and QPSK modulation schemes and behaves well when the noise level is low as shown in Figure 5.21. However, the message error rate starts to increase at -30 dB. The channel kept receiving packet until the -20 dB noise level, although more than 70% of packets had errors. The receiver could not stay synchronized and stop receiving packets at the -20 dB noise level. Furthermore, the 8PSK demodulator could not recover even when the noise was removed. Figure 5.22 shows the cumulative message count and error message count with time for 8PSK modulation.

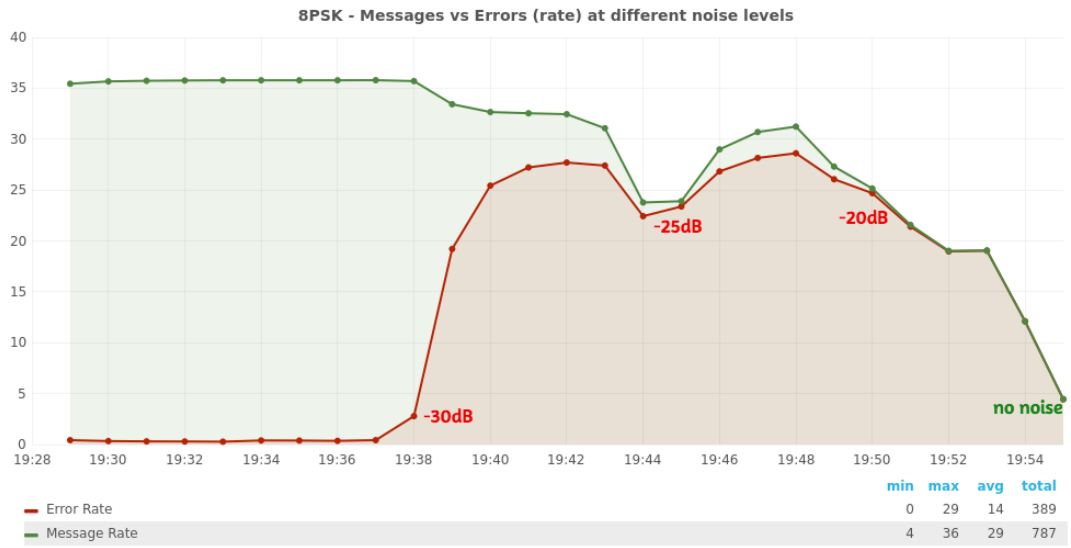

Figure 5.21: Message rate and error rate for 8PSK under different noise levels

Then I implemented the spectrum management functionality described in Algorithm 2 in the USRP and tested its performance under different noise levels. Figure 5.23 shows the behavior of the spectrum management cognitive engine. At the time (a random start time) 15:33 the transmission began with 8PSK modulation with a message transmission rate of closer to 36 messages/second. At 15:36, I started to introduce a noise level of -30 dB. As explained, 8PSK modulation is not stable under the -30 dB noise level. Accordingly,

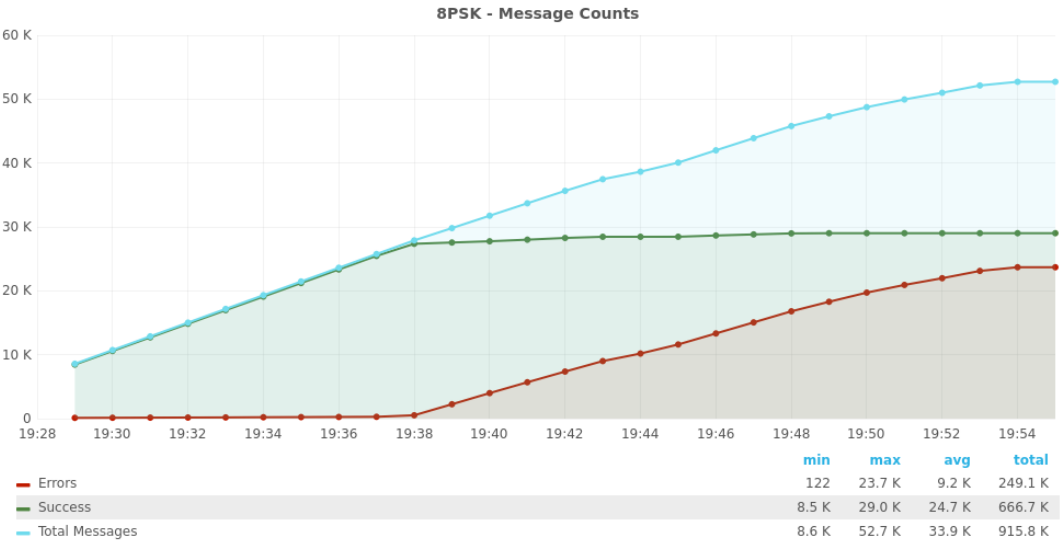

Figure 5.22: Message count and error count for 8PSK under different noise levels

I began to observe packet errors. When the error rate started to increase the SCE detects it and lowers the modulation to QPSK, which was stable at the -30 dB noise level. As observed, the error rate reduced (noise level of -30 dB) between times 15:38 to 15:41. Now, the message rate is around 25 messages/second, and the error rate is almost zero. Then, I gradually started to increase the noise level. At 15:42 (where the noise level is -20 dB) the channel began to introduce packet errors. The SCE detected this and switched to BPSK modulation. Now the message rate is around 12.5 messages per second, and the error rate approaches zero with BPSK under a noise level of -20 dB. At time 15:47, I introduced a noise level of -10 dB that completely blocks the 850MHz channel. The SCE detects this and change the channel frequency to a channel that is not jammed (in our example to 950MHz). In the new channel, the SCE starts with a BPSK modulation with a message rate of 12.5 message/second and zero error rate. SCE upgrades the modulation to achieve higher data rates because the error rate is zero the. At time 15:54, the modulation changes to QPSK. After monitoring the for three more minutes, because QPSK modulated

signals do not encounter measurable packet errors, the modulation turned to 8PSK at time 15:58. However, the channel could not stay stable at 8PSK under the channel condition and switched back to QPSK at time 16:00. Because the channel was stable at QPSK, the radios maintained the QPSK modulation.

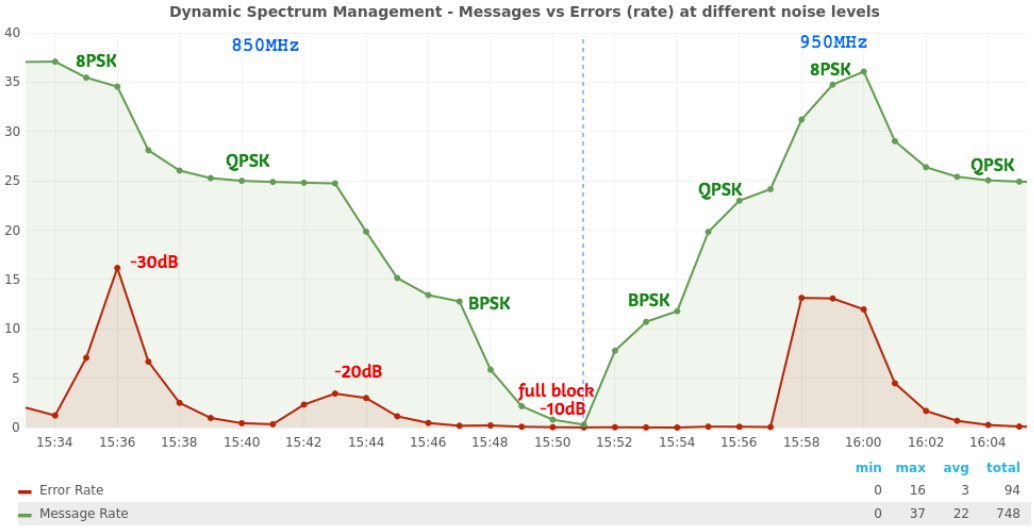

Figure 5.23: Behavior of the SCE with varying noise conditions

Comparison of SIRT with current PTC modulation at different noise levels

As shown in Figure 5.24 current PTC communication radio produces 24 messages per second when the channel noise is less than -35 dB. At the same noise level, SIRT has an average message rate of 35 messages per second.

When the noise level is between -20 dB and -35 dB the useful message rate for PTC radio and SIRT is similar. PTC communication maintains 24 messages per second consistently (Figure 5.26). SIRT oscillates between 8PSK and QPSK, with an average message rate of 29 messages. However during this noise level there is an error rate of 5 errors per second. This is shown in Figure 5.27. When operating at QPSK spectrum monitoring detects that there are

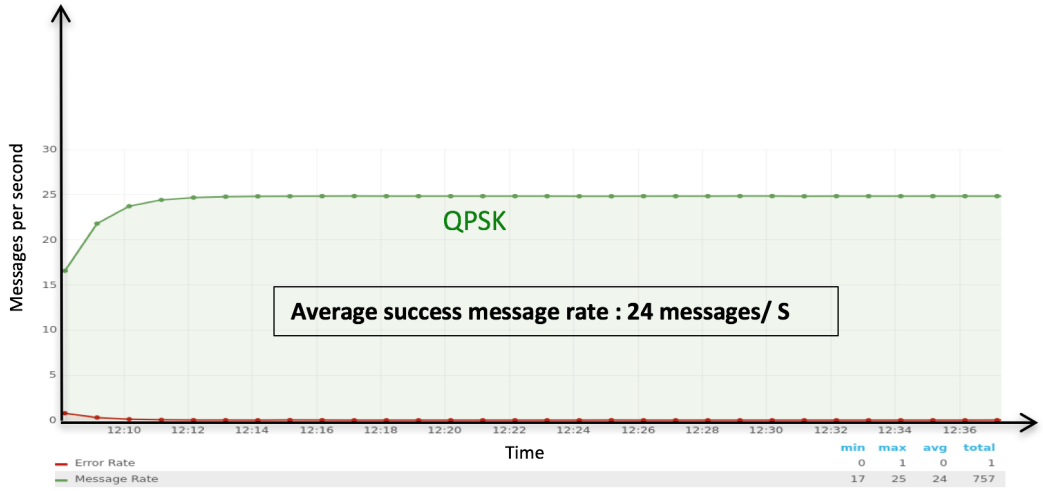

Figure 5.24: Behavior of current PTC radio at noise level under -35 dB

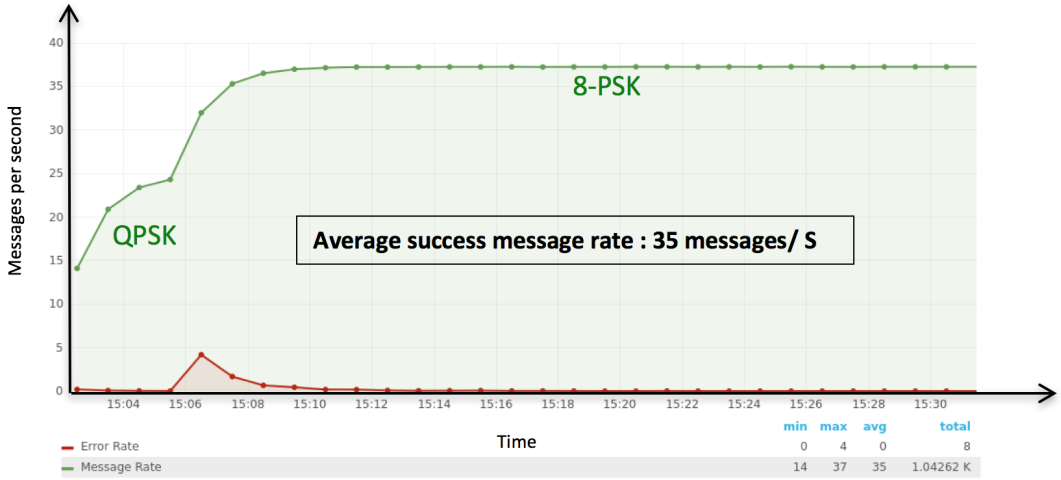

Figure 5.25: Behavior of the SCE at noise level under -35 dB

no errors and the channel is stable. So SIRT tries to change the modulation to 8PSK. But during this noise level 8PSK is not stable, so errors increase. After observing this behavior, SIRT downgrades the modulation to QPSK. The reason for this modulation oscillation problem is that SIRT gets the spectrum quality measurements inside the Software. With these metrics, the channel quality measurements are observed on different modulations. To solve this problem direct spectrum measurements from the Analog to Digital converter is required. As a software level solution to this issue, I added a timer to SIRT that gets invoked when the modulation is changed from higher to lower. As shown in Figure 5.32 now the modulation oscillation stoped and maintained QPSK modulation until the timer expires.

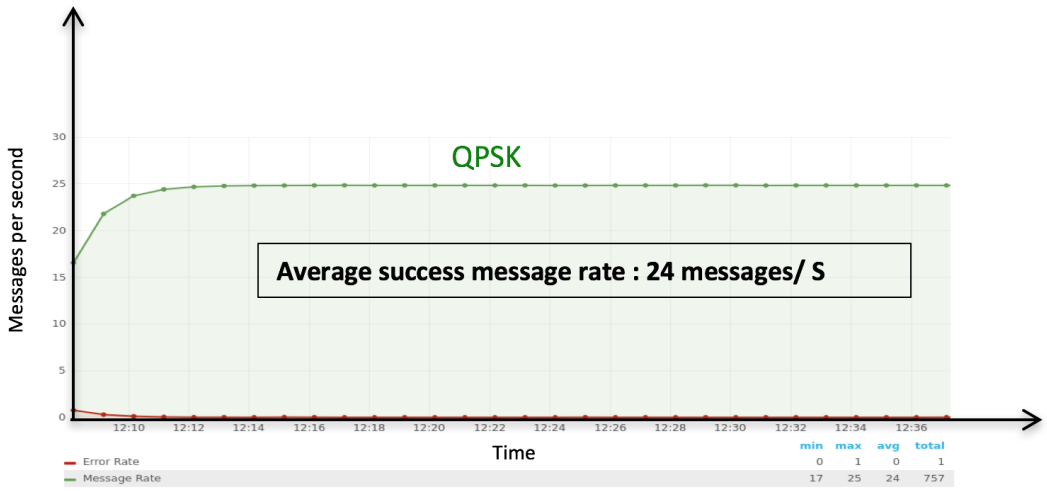

Figure 5.26: Behavior of current PTC radio at noise level between -35 dB -20 dB

SIRT has a similar behavior during the noise levels of -18 dB to -20 dB, where it oscillates between BPSK and QPSK as shown in Figure 5.29. However, in this noise range current PTC communication does not provide any useful throughput. As shown in Figure 5.28 after the noise is introduced at 19:55 all the received messages are identified as errors.

Noise levels greater than -18 dB are considered as a jamming event. As shown in

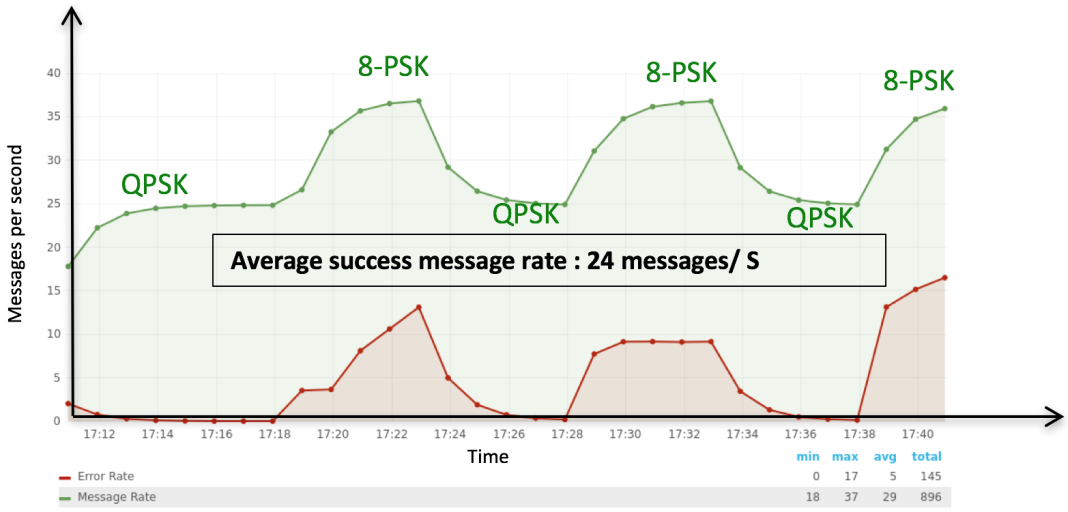

Figure 5.27: Behavior of the SCE at noise level between -35 dB -20 dB

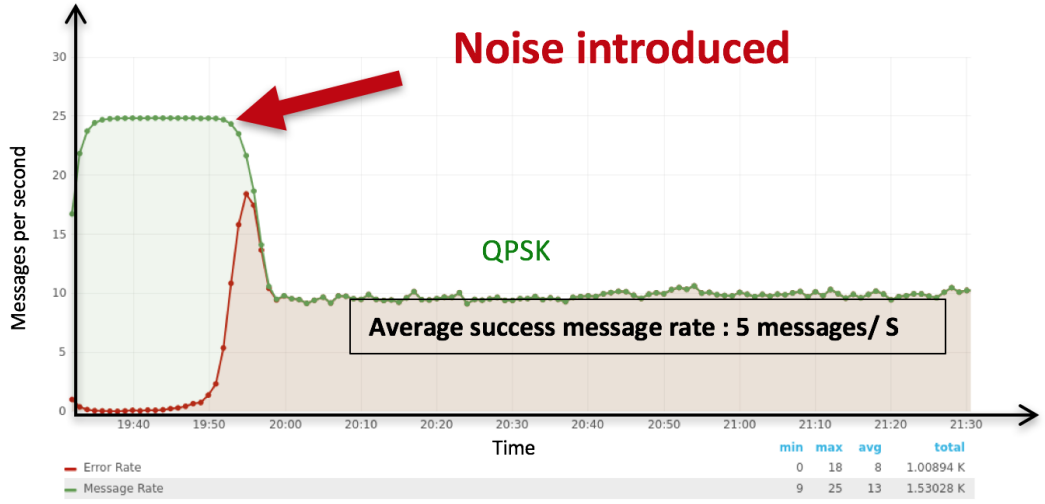

Figure 5.28: Behavior of current PTC radio at noise level between -20 dB -18 dB

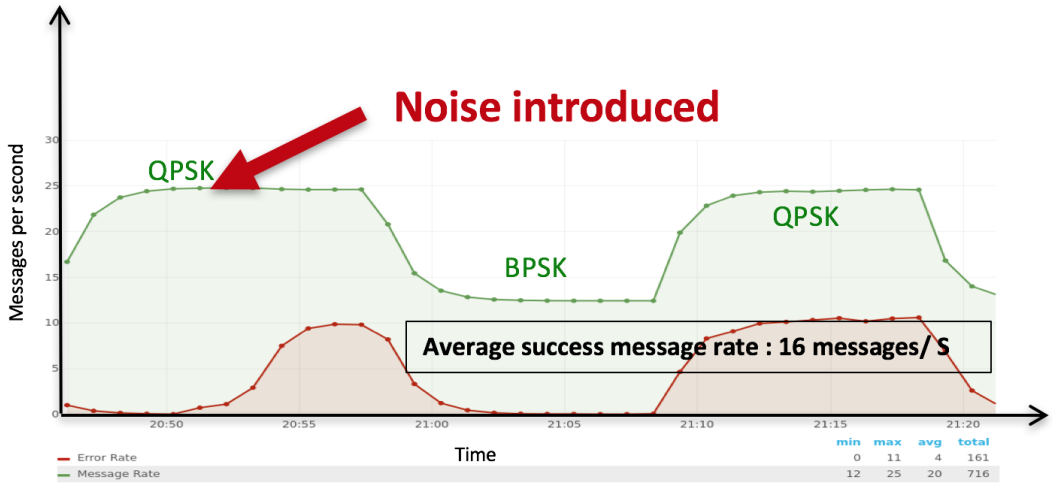

Figure 5.29: Behavior of the SCE at noise level between -20 dB -18 dB

Figure 5.30 PTC cannot maintain communication after jamming starts at 18:06. But as shown in Figure 5.31, SIRT changes frequency to an unjammed channel and maintain connectivity.

Estimating Synchronization delays during Modulation Changes

An issue that arises during dynamic spectrum adaptation is the synchronization delays encountered in locking into expected constellation points of the chosen phase modulation scheme. During periods where the modulation schemes are attempting to lock, application data streams encounter losses. In this experiment, I measured two types of delays. (1) Locking delay which represents the time until the link starts to receive packets and (2) Synchronization delay which represents the time until the link becomes fully stable.

To estimate the locking delay and the synchronization delay I used a timer that collects the rate of messages (message rate per second) and rate of message errors (error rate per second) every 1 Second. Figure 5.33 shows locking delays and synchronization delays for different modulation schemes under differing noise levels that were under -35 dB. Similarly,

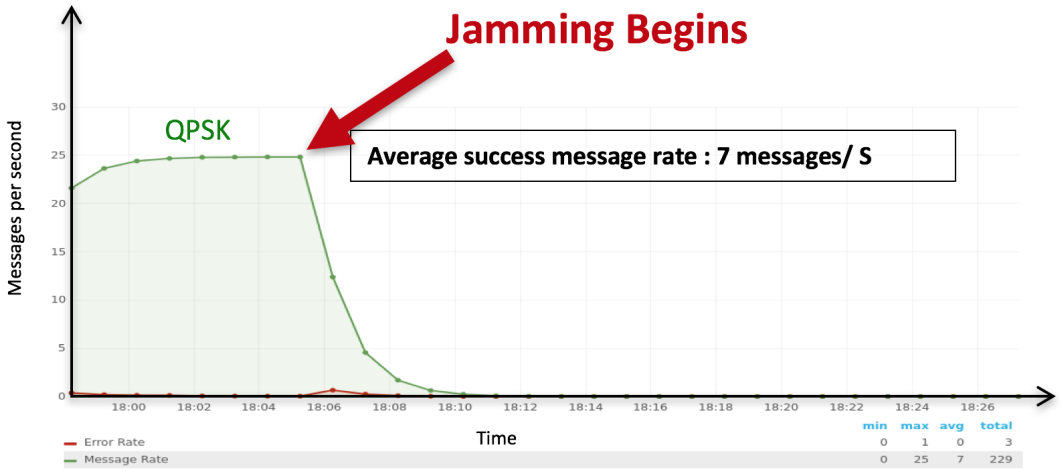

Figure 5.30: Behavior of current PTC radio at noise level more than -18 dB

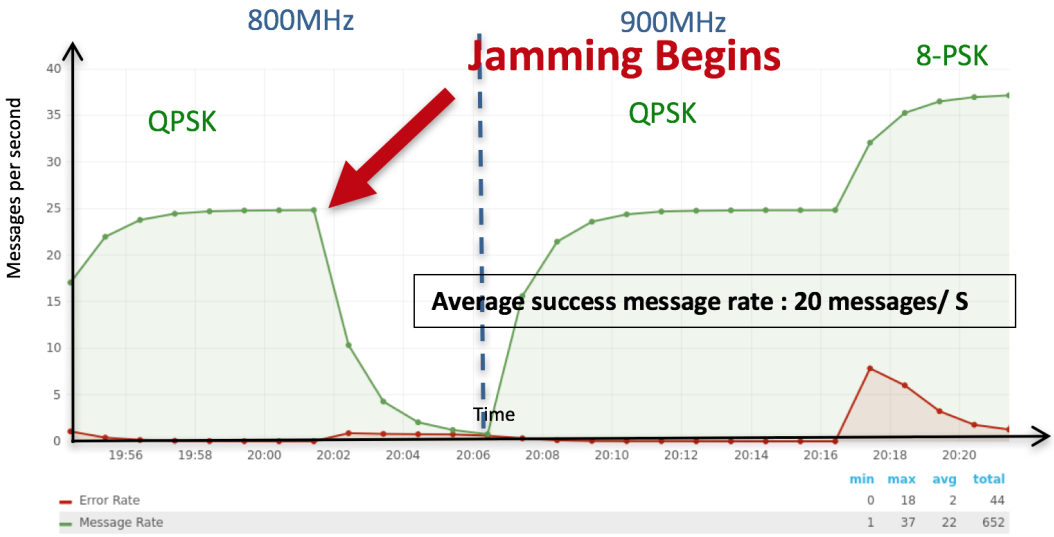

Figure 5.31: Behavior of the SCE at noise level more than -18 dB

I plotted graphs for various noise levels to estimate the locking delays. Because our timer is invoked every 1 Second, our delay estimates are accurate up to 1 Second. Summary of

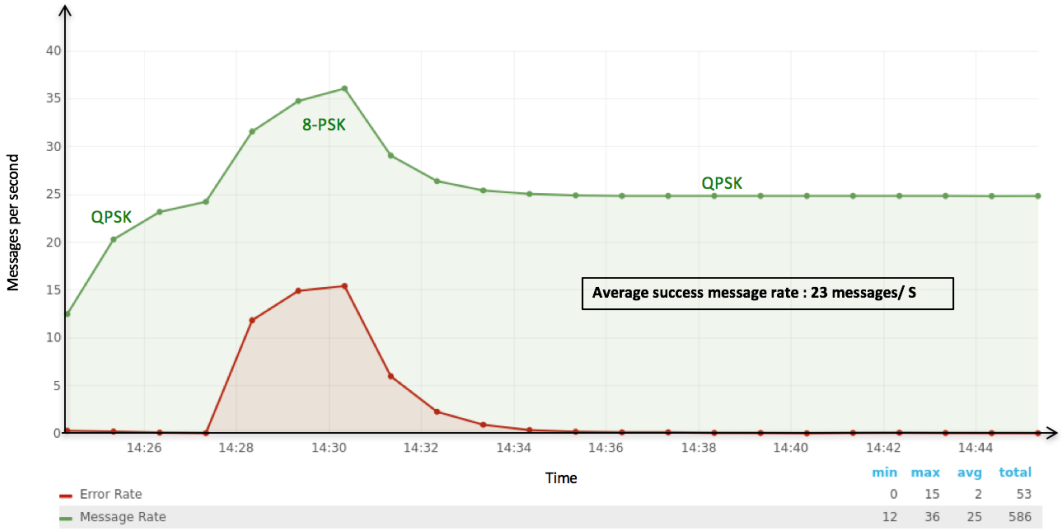

Figure 5.32: Behavior of the SCE at noise level more than -20 dB to -35 dB with a timer

locking delays and synchronization delays for different modulation schemes at different noise levels are shown in Table 5.1. As shown in the table, the locking delay is negligible, and synchronization delays increase with noise. Also, Synchronization delay increases on the constellation points of modulation schemes. However, 8PSK modulation has fewer delays compared to QPSK when there is no noise. I assumed that this is because I used higher order filters to maintain the 8PSK synchronized when the noise levels increase. However, 8PSK only could survive up to -35 dB noise level.

5.2.2 Experimental validation of the Cryptographic Cognitive Engine

To test the functionality of the cryptographic CE I used three Ettus N210 USRPs [58], one simulating the locomotive radio, another simulating the WIU radio and third simulating an attacker radio.

When setting up the experiment, I first defined the cryptographic key chains (i.e. the salt sequence and the algorithm sequence) for the locomotive and the WIU. After generating

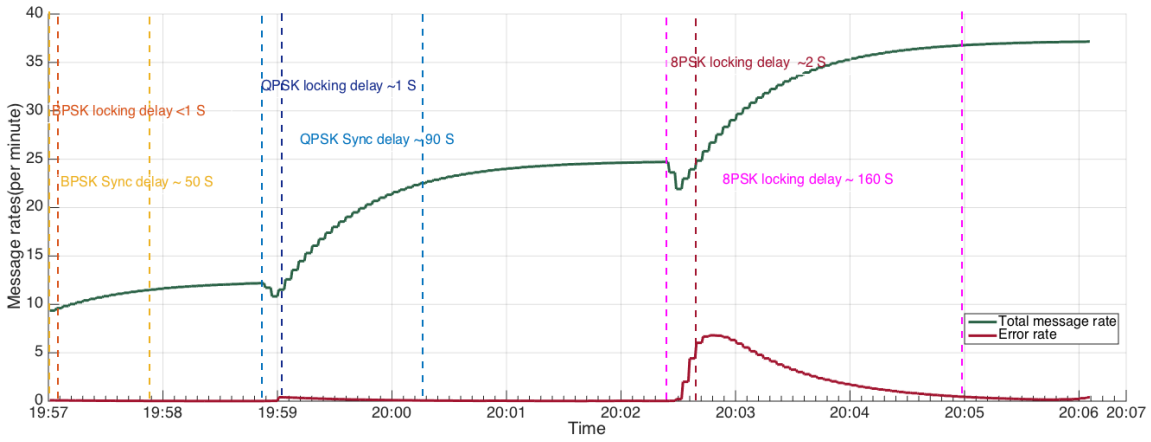

Figure 5.33: Synchronization delays at -35dB noise level

the cryptographic material, I started the 'GETWIUStatus' function at the locomotive. This initial experiment consists of sending 500 'GETWIUStatus' messages separated by 10 S intervals. These application level messages are encoded and modulated using QPSK modulation and transmitted via the USRP. Once a message is received at the WIU USRP, the unit first demodulates, decodes, and submits it to the threat analysis module. The threat module evaluates the message with respect to the Association of American Railroads (AAR) specification [20], as explained in Section 4.3.2. If the message is correct, the CR will generate 5 WIU status beacons, send it to the message transmission function, which will transmit it to the locomotive radio.

If the message does not follow the AAR specifications, the threat analysis module evaluates the message to categorize between the attack types mentioned in Section 4.3.2. The threat analysis module records the status of all the messages to an SQL database. These stored data is used by the threat analysis module later to detect replay attacks. The following experiments were conducted to determine the effectiveness of the cryptographic key generation and threat analysis components.

- **Normal Operation**: Experimental and performance results describing the regular communications between a WIU and a locomotive.

Table 5.1: Locking and Synchronization delay with different noise levels

Modulation	**Without Noise(S)**		**Noise(S)**							
			-35 dB		**-30 dB**		**-25 dB**		**-20 dB**	
	L	S	L	S	L	S	L	S	L	S
BPSK	<1	40	<1	50	∼1	80	∼1	180	∼1	210
QPSK	<1	80	∼1	90	∼1	150	∼1	170	∼1	230
8PSK	∼1	70	∼2	160	-	-	-	-	-	-

L-Locking delay(Time where no messages received)

S-Sync delay(Time until link become 100% stable)

- **Normal with different cryptographic rollover period**: Experimental and performance results illustrating the effect of the *roll over* cryptographic time boundaries as salts are expiring.
- **Message Guessing Detection**: Experimental and performance results describing the capability to detect an attacker guessing at a cryptographic salt to create WIU messages.
- **Message Replay Detection**: Experimental and performance results representing replay attack against the WIU.
- **Message Corruption**: Experimental and performance results describing the capability to detect corrupted messages.

Test Case 1 - Normal Operation

This test was conducted to measure the detection rate of the threat module during the normal operations using enhanced TESLA. It created cryptographic material as described in Section 4.3.1. The keys were generated for a total duration of 1 day and the time is divided into 40-time segments, where each segment lasts for 2160 seconds or 36 minutes. 40 salts

were generated and assigned to each time slot according to enhanced TESLA algorithm. Similarly, the algorithms are assigned to each time slot. The transmitter and receiver salt chain and algorithm generation are started approximately at the same time. The detection pattern is shown in Fig. 5.34 as *Normal 2160 S*.

Test Case 2 - Normal with different cryptographic rollover period

The objective of this test is to determine the effect of multiple cryptographic rollover periods on salts and algorithm changing procedure because the number of false positives increases near the rollover boundaries.

The experiment consists of transmitting and receiving approximately 500 messages at a rate of one message every 10 seconds. The cryptographic rollover periods were then shortened until the bit error rate increased. Fig. 5.34 *Normal 216 S* plots the effects of decreasing the rollover period to 216 seconds, in which the false positive rate increased.

The experimental data lead to the conclusion that the false positive rate depends on the key generation starting time in the transmitter and the receiver and on time between the cryptographic rollover.

Test Case 3 - Message Guessing Detection

The objective of this test was to validate the capability to detect an attacker who can derive algorithms and seeds. The test assumes that the attacker has the complete implementation details of the locomotive, and can derive algorithms and seeds similar to a valid locomotive. The attacker guesses the Initialization Vector (IV) values, initializes their radio, and begins transmission. The result is shown in Fig. 5.34 *Random Guess* illustrates that the threat module detects the attacker either as a guessing attacker or miss-categorized as a replay attacker.

Test Case 4 - Message Replay Detection

The objective of this test was to validate the detection capability during a replay attack. Due to the limitations of the testing setup, the locomotive acts as the attacker. The first message is copied and then replayed 500 times. The replay transmission rate is one replayed message every 10 seconds. The result is shown in Fig. 5.34 labeled as *Replay*.

Test Case 5 - Message Corruption Detection

The objective of this test was to determine the threat module's capability of detecting an intentional data modification or environmental data corruption by evaluating the CRC. In this scenario, the attacker's radio corrupts a field in the message. The receiver then invalidates the message when performing the CRC check. Results are shown in Fig. 5.34 labeled as *Message Corrupt*.

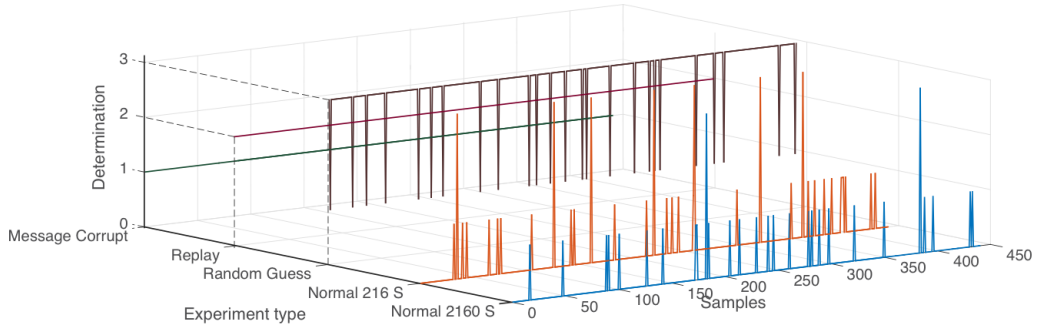

Figure 5.34: Threat module determination for different test cases

Analyzing Experimental Outcomes

Table 5.2 summarizes the accuracy and the error rates for the test cases mentioned above. In this table, accuracy refers to the system's ability to correctly classify the type of operation, while error rate includes any miss-categorization. Types of miss-categorization include false positives (normal operation events categorized as attacks), false negatives (i.e., attacks

Table 5.2: Threat module determination counts

Test scenario	Determination count				Accuracy	Error Rate
	Valid	CRC Corrupt	Replay	Random Guess		
Normal operation (Normal 2160 S) -cryptographic rollover period 2160S	394	24	0	2	93.8%	6.2%
Normal operation (Normal 216 S) -cryptographic rollover period 216S	391	28	0	22	88.6%	11.4%
Random Guess	0	30	0	408	93.1%	6.9%
Replay	0	0	463	0	100%	0%
CRC Corrupt	0	442	0	0	100%	0%

Error Rate: false positive / false negative / miss-categorization.

categorized as normal operation events), and miss-categorized attacks (i.e. events correctly classified as attacks but within a wrong category). In normal operations when the time between the cryptographic rollover period is 2160 S, the threat detection module has an accuracy of 93.8%. This accuracy is reduced to 88.6% when the cryptographic rollover period is reduced to 216 S. That is, a 10-fold reduction of the rollover period causes the system to increase its false alarm rate roughly 2-fold (i.e. from 6.2% to 11.4%). When the attacker plays a guessing attack, the threat detection module's accuracy is 93.1%. However, note that the detection rate (ability to detect an attack) is 100%, which means all attacks events are categorized as such, although in 6.9% of the time the random guess attacks are miss-categorized as replay attacks. In the CRC corruption attack and replay attack, the system has an accuracy of 100%.

The results obtained from experiments lead to the conclusion that cryptographic CE's accuracy is highly correlated to time synchronization. In the experiment, an accurate time source was not available and time synchronization was off on average by a few seconds. Decreasing the time segment intervals between cryptographic material increases the number of false positives. The experiment reducing the cryptographic utilization time from (2160 S

to 216 S), shown in Table 5.2 demonstrated a direct impact on accuracy. A reference time and an accurate timing source are needed for cryptographic key generation.

For all the test case the CPU utilization by the WIU application, which includes key generation and threat analysis was mostly zero, although there were spikes of 0.9%. Additionally, the replay test case shows the occasional CPU use of 1.8%. Fig. 5.35 shows the memory utilization of the WIU application with time. The memory utilization for all the test cases is around 0.6%. Normal operation and random guess attacks caused a slight increase in this number, where for all the other cases the memory utilization remained constant. More details about the memory and CPU utilization can be found in Table 5.3.

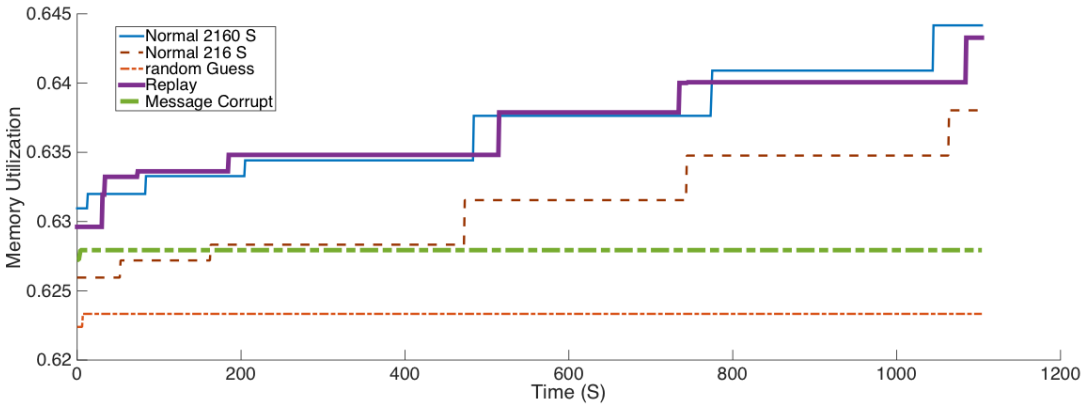

Figure 5.35: Memory Utilization variation

Table 5.3: CPU and memory utilization

	Memory utilization(%)		CPU utilization(%)	
	Mean	Standard Deviation	Mean	Standard Deviation
Normal operation (Normal 2160) - cryptographic rollover period 2160S	0.6605	0.0031	0.0171	0.1229
Normal operation (Normal 216) - cryptographic rollover period 216S	0.6325	0.0034	0.0171	0.1229
Random Guess	0.6300	5.9377e-05	0.0045	0.0635
Replay Attack	0.6369	0.0029	0.0225	0.1620
CRC corruption	0.6279	3.7909e-05	0.0018	0.0402

Chapter 6: Conclusions

6.1 Summary of my work

PTC relies on having a reliable and secure communication radio network to provide safe and secure services. The primary objective of this dissertation is to design an intelligent radio network to improve the security and reliability of PTC communication. The main contributions of my thesis can be identified as follows.

1. **Analyzing the PTC communication to identify potential spectrum vulnerabilities and bandwidth constraints:** I did so by studying existing documents [20,21] and discussing with PTC experts. Main findings of this study are presented in Chapter 3. My analysis showed that PTC communication becomes inefficient when the train density is high due to bandwidth congestion. To reduce this, I have developed an algorithm to allocate frequency channels to trains dynamically based on train priority. The primary goal I achieved here was to reduce the number of stopping attempts for high-speed trains

2. **Designing a frequency channel allocation architecture:** This is intended to allocate frequency channels between the signaling network and the WIU network to reduce spectrum congestion. It uses unique properties of PTC communication to plan the spectrum with a limited number of channels. This architecture is described in Chapter 3.

3. **Designing SIRT:** SIRT is designed to overcome the vulnerabilities identified in the frequency analysis. SIRT has two tiers. The upper tier has MCE which communicates with other SIRT nodes and decides how the train should navigate with lower risk for

travel. To do so, the MCE communicates with cognitive engines at the lower tier of SIRT CCE (that provide cryptographic security and threat detection) and SCE (that uses spectrum monitoring, frequency hopping, and adaptive modulation to ensure the reliability of the radio communication medium).

4. **Prototype development and testing:** I have developed a prototype of SIRT using Software Defined Radios. SIRT network monitors the spectrum continuously, changes the modulation based on channel conditions (switch to a lower order modulation when the channel quality is low and change back to a higher order modulation when the channel recovers) and hop the frequency when the channel is jammed. The switching delay is less than or closer to 1 (S) in most scenarios.

5. **Provide a testbed to test different wireless protocols:** In addition to the main contributions that lists above the prototype that I implemented using SDRs can be utilized as a test bed to test different wireless protocols. The components of SIRT are implemented as separate processes running in parallel that only exchange messages through 'REDIS' [50] . Therefore this radios can be extended for other protocols, and I am planning to expand my research to other applications shortly.

6.2 Migration to real PTC Radio

In my opinion, the most significant and non-trivial additions are required to incorporate messages related to potential security issues in the radio spectrum area. My design as envisioned in this preliminary design, envision informing the back office and PTC controllers spectrum related attacks such as (selective or wide-band) jamming, noise, etc. Potential reactions to them may vary widely and will take processing power from the train resident PTC controller and back-office. The train controller or the SIRT module itself may require extra logging, or reaction to these messages, which in-turn require re-engineering the vital real-time nature of the PTC controller. My ongoing work attempts to explore the non-trivial questions that beg answers. It shows the non-triviality and the significance of considering

effects of security on a vital real-time control system.

6.3 Future Work

In the future, I plan to focus on expanding my current research to different applications such as Intelligent Vehicles, Power systems, and Internet of Things (IoT). One line of research I plan to pursue is to experiment using modulation and frequency changing in IoT applications and other vehicular applications. Another is to build a radio level intrusion detection system for different vehicular applications. This intrusion detection system will use radio propagation characteristics and use multilateration to detect the location of the vehicle and correlate the location with the expected position obtained using vehicles equation of motion to detect whether the vehicle is an authorized vehicle or a malicious attacker trying to imitate a vehicle.

I am also planning to work on extending my work on physical and datalink radio security. I recently started to explore on using OFDM techniques for radio propagation. I am planning to implement different modulation and coding techniques with OFDM and change them dynamically as a security measure. Another is to develop a steganography like scheme by hiding data using a lower order modulation inside a higher order modulation.

6.4 Peer Reviewed Work

1. Damindra Bandara, Satish Kolli, Duminda Wijesekara, Secure Intelligent Radio for Trains(SIRT), Accepted for publication in ASME-IEEE Joint Rail Conference, April 2017

2. Damindra Bandara, Tony Melargno, Duminda Wijesekara, Paulo Costa, A Case Study of Cognitive Radio Networks: Secure Spectrum Management for Positive Train Control Operations, in Spectrum Access and Management for Cognitive Radio Networks, M. A. Matin, Ed. Springer Singapore, 17 September 2016, pp. 121152, doi:10.1007/978 981 10 2254 8 5.

3. Tony Melargno, Damindra Bandara, Ajay Fewell, Duminda Wijesekara, Rail Radio Intrusion Detection System (RRIDS) for Communication Based Train Control (CBTC), IEEE International Conference on Intelligent Rail Transportation (ICIRT), August 2016, doi: 10.1109/ICIRT.2016.7588548

4. Damindra Bandara, Tony Melargno, Duminda Wijesekara, Paulo Costa, Multi-Tiered Cognitive Radio Network for Positive Train Control Operations, Paper No. JRC2016-5784, ASME-IEEE Joint Rail Conference, April 2016, doi:10.1115/JRC2016- 5784

5. Damindra Bandara, Andre Abadie, Duminda Wijesekara, Cell planning for high-speed train operations in USA, Paper No. JRC2015-5805, ASME-IEEE Joint Rail Conference, April 2015, doi:10.1115/ JRC2015-5805

6. Damindra Bandara, Andre Abadie, Tony Melaragno, Duminda Wijesekara, Providing Wireless Bandwidth for High-speed Rail Operations, Conference on ENTERprise Information Systems (CENTERIS 2014), November 2014, doi:10.1016/j.protcy.2014.10

7. Andre Abadie, Damindra Bandara, Duminda Wijesekera, Risk engine design as a key security enhancement to the standard architecture for cognitive radio, IGI Global Disseminator of Knowledge, August 2014, doi: 10.4018/978-1-4666-6571- 2.ch030

8. Andre Abadie, Damindra Bandara, Duminda Wijesekera, Instituting a Risk Engine as Cognitive Radio Technologies, National Wireless Research Collaborative Symposium, May 14-16, 2014

9. Andre Abadie, Damindra Bandara, Duminda Wijesekera, A Composite Risk Model for Railroad Operations Utilizing Positive Train Control (PTC), Paper No. JRC2014-3730, ASME-IEEE Joint Rail Conference, April 2014, 7 pages, doi:10.1115/JRC2014-3730

10. Anthony Patrick Melaragno, Damindra Bandara, Duminda Wijesekera, James Bret Michael, Securing the ZigBee Protocol in the Smart Grid, IEEE Computer Society, 05 April 2012, (vol. 45 no. 4) pp. 92-94

11. Andre Bondi, Damindra Bandara, Michael Smith, Rajni Goel and Duminda Wijesekera, Timely delivery of messages in positive train control, in Critical Infrastructure Protection VII, J. Butts and S. Shenoi, Eds. Springer Berlin Heidelberg, 2013, pp. 139152.

12. Damindra S. Bandara, Andr B. Bondi,Rajni Goel, Nalin Pilapitiya, Duminda Wijesekera,Developing a Framework to Address Performance and Security Protocol Concerns in Identity Management for Interoperable Positive Train Control Systems, ASME-IEEE Joint Rail Conference, April 2012, Paper No. JRC2012- 74113, pp. 389-396; 8 pages, doi:10.1115/JRC2012-74113

Bibliography

Bibliography

[1] Metercomm LLC, "ITCR 1.1 base radio installation and field service guide," version 1.0.

[2] Wikipedia, "Pulse code cab signaling," 2015. [Online]. Available: $https://en.wikipedia.org/wiki/Pulse\_code\_cab\_signaling$

[3] Moxa Railroads, "A robust communication network for mainline signaling and train control applications," 2012. [Online]. Available: www.moxa.com/solutions/railway/Solution/Wayside_Backbone.htm

[4] Darmstadt University of Technology, "Ofdm basics for wireless communications." [Online]. Available: $http://faculty.kfupm.edu.sa/COE/ashraf/RichFilesTeaching/COE082\_543/chap2\_OFDM\_basics.pdf$

[5] Federal Railroad Administration, "Office of safety analysis reports on rail accidents." [Online]. Available: http:/http://safetydata.fra.dot.gov/OfficeofSafety/default.aspx

[6] Wikipedia, "2008 chatsworth train collision," 2015. [Online]. Available: https://wikipedia.org/wiki/$2008\_Chatsworth\_train\_collision$

[7] L. Lamport, "Password Authentication with Insecure Communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981. [Online]. Available: http://doi.acm.org/10.1145/358790.358797

[8] S. Dietzel, R. van der Heijden, H. Decke, and F. Kargl, "A flexible, subjective logic-based framework for misbehavior detection in V2v networks." IEEE, Jun. 2014, pp. 1–6.

[9] W. Bamberger, Schlittenlacher, Jesef, and Diepold, Klaus, "A Trust Model for Inter-vehicular Communication Based on Belief Theory." IEEE International Conference on Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust, Aug. 2010.

[10] Supriya and U. Shenoy, "Analysis of BPSK, QPSK, 16-QAM and 64- QAM over Wireless Networks using NS-2," *International Journal on Applications in Electrical and Electronics Engineering*, vol. 1, no. 6, pp. 13–16, Jun. 2015.

[11] C. Christopher and R. M. Noory, "Wireless channel optimization in VANET by using adaptive modulation," in *2012 12th International Conference on ITS Telecommunications (ITST)*, Nov. 2012, pp. 295–299.

[12] G. Sharma, B. Suman, A. Verma, and S. Tej, "Security in Wireless Sensor Networks using Frequency Hopping," *International Journal of Computer Applications*, Dec. 2010.

[13] M. Tabassum, M. A. Razzaque, M. M. Hassan, A. Almogren, and A. Alamri, "Interference-aware high-throughput channel allocation mechanism for CR-VANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, Dec. 2016. [Online]. Available: http://jwcn.eurasipjournals.com/content/2016/1/2

[14] D. Bandara, A. Abadie, T. Melaragno, and D. Wijesekara, "Providing wireless bandwidth for high-speed rail operations," *Procedia Technology*, vol. 16, pp. 186–191, 2014.

[15] K. R. D. S. Bandara, A. Abadie, and D. Wijesekara, "Cell Planning for High-Speed Train Operations in USA." ASME, Mar. 2015, p. V001T03A007. [Online]. Available: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/JRC2015-5805

[16] A. Abadie, D. Bandara, and D. Wijesekera, "A Composite Risk Model for Railroad Operations Utilizing Positive Train Control (PTC)." ASME, Apr. 2014, p. V001T06A004. [Online]. Available: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/JRC2014-3730

[17] K. R. D. S. Bandara, A. P. Melaragno, D. Wijesekara, and P. Costa, "Multi-Tiered Cognitive Radio Network for Positive Train Control Operations." Columbia, South Carolina, USA: ASME Joint Rail Conference, Apr. 2016. [Online]. Available: http://dx.doi.org/10.1115/JRC2016-5784

[18] K. R. D. S. Bandara, A. Melaragno, D. Wijesekera, and P. Costa, "A Case Study of Cognitive Radio Networks: Secure Spectrum Management for Positive Train Control Operations," in *Spectrum Access and Management for Cognitive Radio Networks*, ser. Signals and Communication Technology. Springer Singapore, 2017, pp. 121–152, dOI: 10.1007/978-981-10-2254-8_5.

[19] K. R. D. S. Bandara, S. Kolli, and D. Wijesekara, "Secure-Intelligent Radio for Trains (SIRT)." Accepted for ASME Joint Rail Conference 2017.

[20] American Association for Railroads, "Interoperable train control Wayside Interface Unit Requirements Railway Electronics S9202," Jan. 2010.

[21] ——, "Office to Locomotive ICD version draft 2.6 S 9352A," Dec. 2010.

[22] Joint council on transit wireless communications, "Positive Train Control-White Paper," May 2012.

[23] M. Hartong, "Secure communications based train control (CBTC) operations," Ph.D. dissertation, George Mason University, Fairfax, Virginia, Sep. 2009.

[24] A. Abadie, "Combining Operational and Spectrum Characteristics to Form a Risk Model for Positive Train Control Communications," Ph.D. dissertation, George Mason University, 2014.

[25] K.-D. Lin and J.-F. Chang, "Communications and entertainment onboard a high-speed public transport system," *IEEE Wireless Communications*, vol. 9, no. 1, pp. 84–89, Feb. 2002.

[26] A. Kanafan, H. Benou, B. Chiou, J.-L. Ygnac, K. Yamad, and A. Dankbe, "California trains connected," California PATH Research Repo, Tech. Rep. ISSN 1055-1425 UCB-ITS-PRR-2006-, Apr. 2006, ISSN 1055-1425.

[27] G. Hui, W. Hao, and Z. Yushu, "GSM-R network planning for high speed railway." IET, 2010, pp. 10–13.

[28] C. M. Alexandresc and L.-M. Nemto, "Considerations regarding a radio planning procedure for GSM-R network covering the bucuresti- Constant railway corridor," *U.P.B. Sci. Bull.*, vol. 73, no. 3, 2011.

[29] J. Mithola, "Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio," Ph.D. dissertation, Royal Institute of Technology(KTH), Sweden, May 2000.

[30] A. Crohas, "Practical Implementation of Cognitive Radio System for Dynamic Spectrum Access."

[31] S. Chen, "Vehicular Dynamic Spectrum Access: Using Cognitive Radio for Automobile Networks," Ph.D. dissertation, Worcester Polytechnic Institute, 2012.

[32] K. Singh, P. Rawat, and J.-M. Bonnin, "Cognitive radio for vehicular ad hoc networks (CR-VANETs): approaches and challenges," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 49, 2014.

[33] M. Jalil Piran, Y. Cho, J. Yun, A. Ali, and D. Y. Suh, "Cognitive Radio-Based Vehicular Ad Hoc and Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 154193, Aug. 2014. [Online]. Available: http://journals.sagepub.com/doi/10.1155/2014/154193

[34] A. Amanna, M. Gadhiok, M. J. Price, J. H. Reed, W. P. Siriwongpairat, and T. K. Himsoon, "Rail-CR: Cognitive Radio for Enhanced Railway Communication." ASME, 2010, pp. 467–473.

[35] A. MacKenzie, J. Reed, P. Athanas, C. Bostian, R. Buehrer, L. DaSilva, S. Ellingson, Y. Hou, M. Hsiao, Jung-Min Park, C. Patterson, S. Raman, and C. da Silva, "Cognitive Radio and Networking Research at Virginia Tech," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 660–688, Apr. 2009.

[36] T. Keller and L. Hanzo, "Adaptive modulation techniques for duplex OFDM transmission," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1893–1906, Sep. 2000.

[37] F. Harris, "Lets Assume the System Is Synchronized," in *Globalization of Mobile and Wireless Communications*, R. Prasad, S. Dixit, R. van Nee, and T. Ojanpera, Eds. Dordrecht: Springer Netherlands, 2011, pp. 311–325. [Online]. Available: http://www.springerlink.com/index/10.1007/978-94-007-0107-6_20

[38] Peterson, L L and Davie B S, *Computer Networks:a systems approach.* Morgan Kaufmann.

[39] R. Hartly, "Transmission of Information." Bell System Technical Journel.

[40] B. Vincze and G. Tarmi, "Development and analysis of train brake curve calculation methods with complex simulation," Zilna, Slovika, May 2006.

[41] S. Loyka and A. Kouki, "Using two ray multipath model for microwave link budget analysis," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 5, pp. 31–36, Oct. 2001, 00015.

[42] M. Willis, "Propagation tutorial - introduction," 2007. [Online]. Available: http://www.mike-willis.com/Tutorial/PF1.htm

[43] T. S. Rappaport, *Wireless communications: principles and practice*, 2nd ed., ser. Prentice Hall communications engineering and emerging technologies series. Upper Saddle River, N.J: Prentice Hall PTR, 2002, 00015.

[44] L. Klozar and J. Prokopec, "Propagation path loss models for mobile communication." IEEE, Apr. 2011, pp. 1–4, 00006.

[45] A. Sharma and P. Jain, "Effects of rain on radio propagation in GSM," *International Journal of Advanced Engineering & Applications*, pp. 83–86, Jan. 2010.

[46] A. Lagrone, "Forecasting television service fields," *Proceedings of the IRE*, vol. 48, no. 6, pp. 1009–1015, Jun. 1960, 00018.

[47] Terra Wave and Wireless Solutions, "2.4 GHz 12 dBi 180 degree sector panel antenna with n-style jack connector:." [Online]. Available: http://www.terra-wave.com/shop/24-ghz-12-dbi-180-degree-sector-panel-antenna-with-nstyle-jack-connector-p-811.html

[48] A. Perrig and T. J.D., *Secure Broadcast Communication in Wired and Wireless Networks.* Assinippi Park, Norwell Massachusetts 020601 USA: Kluwer Academic Publishers, 2002, vol. 1.

[49] T. melaragno, K. R. D. S. Bandara, A. Fewel, and D. Wijesekara, "Rail radio intrusion detection system (rrids) for communication based train control (cbtc)," 2016.

[50] Wikipedia, "Redis," 2015. [Online]. Available: https://en.wikipedia.org/wiki/Redis

[51] J. Corgan, "Welcome to GNU Radio," 2015. [Online]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki

[52] T. Rondeau, "Gnu radio new tutorial 7," 2014. [Online]. Available: $http://gnuradio.org/redmine/projects/gnuradio/wiki/Guided\_Tutorial\_PSK\_Demodulation/5$

[53] GNURadio, "Gnu radio manual and c++ api reference-tagged streams," 2016. [Online]. Available: $http://gnuradio.org/doc/doxygen/page_tagged_stream_blocks.html$

[54] M. Ettus, T. W. Rondeau, and R. McGwier, "Ofdm implementation in gnu radio," 2007. [Online]. Available: $gnuradio.org/redmine/attachments/download/751/gr\_ofdm.ppt$

[55] National Instruments, "Ofdm and multi-channel communication systems." [Online]. Available: http://www.ni.com/white-paper/3740/en/

[56] K. Bansal and V. Tripathi, "Ofdm tranmission and reception of packets using gnu-radio and usrp - communications lab project." [Online]. Available: https://www.ee.iitb.ac.in/student/~vishrant/ofdm-tranmission-reception.pdf

[57] S. Pathak and H. Sharma, "Channel Estimation in OFDM Systems," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 3, Mar. 2013.

[58] Wikipedia, "Universal software radio peripheral," 2014. [Online]. Available: https://en.wikipedia.org/wiki/Universal_Software_Radio_Peripheral

Curriculum Vitae

Damindra Bandara grew up in Matale, Sri Lanka. She received her Bachelor's degree in Electrical and Electronic Engineering from University of Peradeniya, Sri Lanka and Master degree in Information Security and Assurance from George Mason University. Previously she has worked as a Wireless Quality Assurance intern at Time Warner Cable, Herndon Virginia and as a lecturer in Department of Electrical and Electronic Engineering, University of Peradeniya, Sri Lanka.