

OPTIMIZATION AND HEURISTICS FOR COGNITIVE RADIO DESIGN

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Bharath Keshavamurthy

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2020

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL  
STATEMENT OF DISSERTATION APPROVAL**

Dr. Nicolò Michelusi, Chair

School of Electrical and Computer Engineering

Dr. Xiaojun Lin

School of Electrical and Computer Engineering

Dr. Shreyas Sundaram

School of Electrical and Computer Engineering

**Approved by:**

Dr. Dimitrios Peroulis

Head of the School of Electrical and Computer Engineering

## ACKNOWLEDGMENTS

This research has been funded in part by NSF, under grant CNS-1642982.

## TABLE OF CONTENTS

	Page
LIST OF FIGURES . . . . .	vi
ABSTRACT . . . . .	ix
1 INTRODUCTION . . . . .	1
2 THE DARPA SC2 RADIO DESIGN . . . . .	7
2.1 Cognitive Radio Subsystems . . . . .	8
2.1.1 Transmission Power Control (PHY) . . . . .	8
2.1.2 Network Control Data Broadcast . . . . .	10
2.1.3 The DFT-s-OFDM Data Link . . . . .	12
2.1.4 Modulation and Coding Scheme (MCS) Adaptation (PHY) . . .	14
2.1.5 Prioritized Flow Scheduling with ARQ Heuristic (DLL) . . . .	17
2.1.6 Channel and Bandwidth Allocation Heuristic (MAC) . . . .	20
2.1.7 Multi-hop Routing Heuristic (NET) . . . . .	23
2.2 The Colosseum Command and Control radio API (C2API) . . . . .	23
2.3 The CIRN Interface Language (CIL) . . . . .	24
2.4 Capabilities and Performance Evaluations from DARPA SC2 scenario emulations . . . . .	26
2.4.1 Alleys of Austin . . . . .	26
2.4.2 Wildfire . . . . .	32
2.5 Conclusion . . . . .	38
3 SPECTRUM SENSING AND ACCESS VIA APPROXIMATE POMDP . .	43
3.1 System Model . . . . .	44
3.1.1 Signal Model . . . . .	44
3.1.2 Occupancy Correlation Structure . . . . .	45
3.1.3 Channel Sensing Model . . . . .	48

	Page
3.1.4 POMDP Formulation . . . . .	49
3.2 Proposed Solution: The Algorithms . . . . .	53
3.2.1 Occupancy Correlation Structure Estimation . . . . .	53
3.2.2 The PERSEUS Algorithm . . . . .	54
3.3 Numerical Evaluations . . . . .	59
3.4 Conclusion . . . . .	68
4 FEASIBILITY ANALYSIS OF THE POMDP OPTIMAL POLICY ON ESP32 RADIOS . . . . .	70
4.1 Methodology . . . . .	70
4.2 Implementation Results . . . . .	72
4.3 Conclusion . . . . .	73
5 SUMMARY . . . . .	74
REFERENCES . . . . .	77

## LIST OF FIGURES

Figure	Page
2.1 The power spectrum observed by an SRN during an SC2 qualification run . . . . .	11
2.2 The power spectrum of the DFT-s-OFDM waveform . . . . .	13
2.3 The MCS adaptation scheme during a Payline scenario . . . . .	15
2.4 The block error rates (BLERs) for different modulation order and code rate pairs, over varying values of SNR (dB) . . . . .	16
2.5 The spectrum occupied by our network during an Alleys of Austin scenario emulation . . . . .	22
2.6 The offered and delivered traffic corresponding to various (selected) flows during the Alleys of Austin scenario emulation . . . . .	27
2.7 The estimated SNR on various (selected links) during the Alleys of Austin scenario emulation . . . . .	28
2.8 The MCS adaptation scheme at each (selected) link during the Alleys of Austin scenario emulation . . . . .	29
2.9 The scores attained by our network and other competing networks during the Alleys of Austin scenario emulation, in addition to the number of QoS mandates satisfied by our network in a given time snapshot . . . . .	30
2.10 The reported GPS locations of our SRNs, in addition to those of our competing SRNs, during the Alleys of Austin scenario emulation . . . . .	31
2.11 The description of various CIL message exchanges between our network and the collaboration server, in addition to those between our network and our peers, during the Alleys of Austin scenario emulation . . . . .	32
2.12 The PSD measurements received at our Gateway SRN from one particular SRN in our network (selected), in order to obtain location-specific spectrum occupancy information, during the Alleys of Austin scenario emulation . . . . .	33
2.13 The spectrum occupancy behavior of SRNs in our network (as guided by our Gateway SRN) during the Alleys of Austin scenario emulation . . . . .	34
2.14 The offered and delivered traffic corresponding to various (selected) links during the Wildfire scenario emulation (Stage 1) . . . . .	35

Figure	Page
2.15 The offered and delivered traffic corresponding to various (selected) links during the Wildfire scenario emulation (Stage 5) . . . . .	36
2.16 The estimated SNR on various (selected links) during the Wildfire scenario emulation . . . . .	37
2.17 The MCS adaptation scheme at each (selected) link during the Wildfire scenario emulation . . . . .	38
2.18 The scores attained by our network and other competing networks during the Wildfire scenario emulation, in addition to the number of QoS mandates satisfied by our network in a given time snapshot . . . . .	39
2.19 The reported GPS locations of our SRNs, in addition to those of our competing SRNs, during the Wildfire scenario emulation . . . . .	40
2.20 The description of various CIL message exchanges between our network and the collaboration server, in addition to those between our network and our peers, during the Wildfire scenario emulation . . . . .	41
2.21 The PSD measurements received at our Gateway SRN from one particular SRN in our network (selected), in order to obtain location-specific spectrum occupancy information, during the Wildfire scenario emulation .	41
2.22 The spectrum occupancy behavior of SRNs in our network (as guided by our Gateway SRN) during the Wildfire scenario emulation . . . . .	42
3.1 The incumbent spectrum occupancy heat-map assuming independence across both frequency and time . . . . .	45
3.2 The incumbent spectrum occupancy heat-map assuming a time-frequency correlation structure . . . . .	46
3.3 The visualization of the incumbent occupancy time-frequency correlation structure as two dependent Markov chains: one across time and the other across frequencies . . . . .	47
3.4 The evaluation of SU and PU network throughputs for different values of $\lambda$ , along with comparisons with the state-of-the-art . . . . .	61
3.5 The evaluation of the proposed solution, from an average utility per time-slot perspective, against a medley of approaches in the state-of-the-art . .	63
3.6 The evaluation of estimation accuracies for different channel sensing strategies, corresponding to a single frequency-correlation Markov chain Viterbi algorithm, in relation to variations in the amount of correlation . . . . .	64

Figure	Page
3.7 The convergence of the MSE of the HMM EM algorithm to estimate $\vec{\theta}$ , and the convergence of the loss of the fragmented PERSEUS algorithm with belief update simplification . . . . .	67
4.1 The channel access probability of the ESP32 SU radios per time-slot . . .	72

## ABSTRACT

Keshavamurthy, Bharath M.S., Purdue University, May 2020. Optimization and Heuristics for Cognitive Radio Design. Major Professor: Nicolò Michelusi.

Cognitive Radio technologies have been touted to be instrumental in solving resource-allocation problems in resource-constrained radio environments. The adaptive computational intelligence of these radios facilitates the dynamic allocation of network resources--particularly, the spectrum, a scarce physical asset. In addition to consumer-driven innovation that is governing the wireless communication ecosystem, its associated infrastructure is being increasingly viewed by governments around the world as critical national security interests--the US Military instituted the DARPA Spectrum Collaboration Challenge which requires competitors to design intelligent radios that leverage optimization, A.I., and game-theoretic strategies in order to efficiently access the RF spectrum in an environment wherein every other competitor is vying for the same limited resources. In this work, we detail the design of our radio, i.e., the design choices made in each layer of the network protocol stack, strategies rigorously derived from convex optimization, the collaboration API, and heuristics tailor-made to tackle the unique scenarios emulated in this DARPA Grand Challenge. We present performance evaluations of key components of our radio in a variety of military and disaster-relief deployment scenarios that mimic similar real-world situations. Furthermore, specifically focusing on channel access in the MAC, we formulate the spectrum sensing and access problem as a POMDP; derive an optimal policy using approximate value iteration methods; prove that our strategy outperforms the state-of-the-art, and facilitates means to control the trade-off between secondary network throughput and incumbent interference; and evaluate this policy on an ad-hoc

distributed wireless platform constituting ESP32 radios, in order to study its implementation feasibility.

## 1. INTRODUCTION

”Wireless Companies share the airwaves”--an article [1] published in the Wall Street Journal in December 2019 emphasises a few key points about Cognitive Radio (CR) technologies, also known as Dynamic Spectrum Access (DSA) or neXt-Generation (xG) technologies in the wireless communication landscape: firstly, it reiterates a critical fact that has long been known in the industry that the economics of spectrum licensing plays a vital role in driving innovation in Radio Access Technologies (RATs); secondly, the article reports that in September 2019, the Federal Communications Commission (FCC) allowed private players in the telephone, cable, and internet space to provide their services over Citizens Broadband Radio Service (CBRS) without having to buy a license, provided their transmissions do not interfere with the U.S. Navy and entities that pay for a license; thirdly, it details the three tiers under which the CBRS spectrum (3550MHz-3700MHz) has been categorized by the FCC—the U.S. military (particularly, naval radar operators and aircraft communications), ”Priority-Access” licensees constituting service providers that pay for access to this spectrum, and ”General Authorized Access” users comprising unlicensed users; and finally, the article reports on the administrative and bureaucratic red-tape that prevented this policy that was first brought-up in 2012 to be realized almost 8 years later, quoting Craig Moffett, an analyst at MoffettNathanson LLC—”The concept has been in place for a really long time, waiting for the pieces to fall in place”.

With fifth-generation (5G) mobile communication technologies in full-deployment mode around the world today, several countries--especially, the U.S. and China, are vying to dominate the space—with the U.S. being extra cautious, citing national security concerns. Many analysts have expressed the need for better spectrum auctioning and availability provisions at the federal level in order to facilitate efficient deployment of 5G services across the country [2]. The 5G ecosystem brings in an extraordinary

demand for these limited spectrum resources due to the promises of extremely high data rates; extremely low latencies; high reliability for critical infrastructure; massive Machine-Type Communication (MTC) enabling the embedded-IoT boom involving Wireless Sensor Networks (WSNs)-both, industrial and personal, Human-Computer interfaces, autonomous vehicles, and Vehicular Ad-Hoc Networks (VANs); and improved mobility [3, 4]. Although a significant number of research works exist in the state-of-the-art professing widespread adoption of cognitive radio technologies for the 5G landscape and trying to solve problems associated with shared access of spectrum resources [5–11], little progress has been made with respect to the real-world deployment of these technologies. Spectrum-sharing technologies have never been in the limelight more than they are today, especially among economists, academics, and national security analysts, with Holman Jenkins Jr. writing in the Journal, "...new spectrum-sharing technologies increasingly make the airwaves seem less scarce than once thought". He further goes on to state that efficient re-allocations of existing spectrum coupled with the widespread deployment spectrum-sharing technologies will have huge public benefits [12].

Exhibiting much-needed prescience, in 2016, a Grand Challenge was instituted by the U.S. Defense Advanced Research Projects Agency (DARPA), known as the Spectrum Collaboration Challenge (SC2) to understand the implications of spectrum sharing and to drive innovation in the space using Artificial Intelligence (A.I.). DARPA understood that the division of the spectrum into rigid, licensed bands is infeasible in the current wireless environment due to the massive demand [13]. The DARPA SC2 envisioned a fully collaborative radio environment in which competing radios exhibited collaborative spectrum access strategies to not only satisfy their individual Quality of Service (QoS) requirements, but to also view the problem altruistically, i.e., to allow for the entire ensemble to satisfy their QoS requirements. The DARPA SC2 involved several simulated scenarios that mimic similar situations these radios have to operate in, in the real-world, for example, troop-deployment scenarios in urban areas, high-priority audio and video communication among first

responders fighting a wildfire, everyday user communication in consumer centers such as shopping malls, and scenarios in which the radios have to work around jammers and licensed users (incumbents) [14]. Cognitive Radio technologies typically involve solving spectrum sensing and access problems in an independent setting wherein the cognitive radio node uses its passive sensing capabilities to deliver its traffic over licensed bands, subject to constraints on the amount of interference caused to military and licensed users. While we do discuss our solution to the independent spectrum sensing and access problem in this work using tools from Dynamic Programming (DP) and estimation theory, the DARPA SC2 featured a more collaborative radio environment in which the radios deployed in certain scenarios communicated with each other using a shared protocol (i.e., language) over a common communication link (air link/public internet/satellite) in order to exchange relevant performance metrics and their respective QoS requirements, which would then be used in solving a joint resource-allocation problem for mutual benefits [15]. Leveraging the capabilities of Software-Defined Radios (SDRs) and A.I., competitors from both the industry and academia competed in the challenge that lasted for 3 years (Dec 2016-Oct 2019). The Purdue BAM! Wireless team from the Department of Electrical and Computer Engineering (ECE) designed radios from the ground-up for operations in Collaborative Intelligent Radio Network (CIRN) environments emulated in the DARPA Massive CHannel EMulator (MCHEM) known as the Colosseum. In this work, we detail the design principles underlying the development of our radios for the SC2, while also discussing their crucial performance metrics and behavioral characteristics. Various advancements are now being built-upon the standards and strategies established as a result of this Grand Challenge, including the CIRN Interaction Language (CIL), the Colosseum test-bed for public use, adaptive spectrum use visualization technologies, and A.I. techniques in various layers of the radio network protocol stack [16–19]. Simplifying the problem by narrowing our focus on the design of optimal channel access strategies in a single cognitive radio node operating in a radio environment with multiple priority or licensed users, we introduce our POMDP formulation next.

From an independent cognitive radio perspective, our solution to the spectrum sensing and access problem in the Medium Access Control (MAC) layer of a cognitive radio node, referred to as a Secondary User (SU) from this perspective, sharing a discretized multi-channel AWGN radio environment with several licensed users, involves a Partially Observable Markov Decision Processes (POMDP) formulation [20]. As alluded to earlier, a cognitive radio facilitates efficient spectrum utilization by intelligently accessing unused spectrum holes across both time and frequency known as "spectrum white spaces", in order to deliver its network flows while limiting interference to the priority or licensed users (incumbents), referred to as Primary Users (PUs) from this perspective [21]. In order to intelligently access these white spaces, the SU needs to solve for a channel sensing and subsequent access policy based on the noisy observations of the occupancy behavior of the PUs in the network. However, critical design limitations prevent the SU from sensing all the channels in the discretized spectrum of interest. These sensing limitations are primarily driven by energy efficiency requirements, with some additional restrictions imposed by the need for minimal sensing times [22]. So, in view of these sensing limitations, the logical next step would be to develop algorithms that try to maximize the accuracy of incumbent occupancy behavior estimation, subject to upper bounds on the number of channels that can be sensed by the SU at the beginning of each time-slot: several works in the state-of-the-art [5–8] propose algorithms to solve this limited information spectrum sensing and access problem. However, almost all of these works [7–11, 23] fail to leverage the correlations exhibited in the occupancy behavior of the incumbents across both frequency and time [24], which as we will illustrate later in this work, lead to significant improvements in the estimation accuracy, which in turn leads to a greater number of white spaces accessed by the SU for delivering its network flows, thereby resulting in a higher SU network throughput with lower levels of interference caused to the PUs in the network. In the sections that follow, we detail solutions to learn these frequency-time correlations in PU occupancy behavior, and to concur-

rently utilize these learned statistics to solve for an optimal sensing and access policy using approximate POMDP value iteration methods.

As alluded to earlier, the existing state-of-the-art does tackle the spectrum sensing and access problem, albeit with some underlying assumptions---many of these assumptions when broken down or generalized will lead to a better solution, as discussed in this work. Firstly, [8] details a solution for distributed spectrum sensing employing SARSA with linear value function approximation. However, this work fails to capitalize on the correlated occupancy behavior of the PUs across channels. Additionally, the authors fail to provide a mechanism to manage the trade-off between secondary network throughput and incumbent interference, which we do. Unlike [8], although [5] considers frequency correlation, the assumed observation model is noise-free, which is not realistic. On the other hand, we in this work, present a Hidden Markov Model (HMM) system-level framework in which the true occupancy states of the incumbents in the channels are hidden behind noisy observations at the SU's spectrum sensor--HMMs call for the Viterbi algorithm (for state estimation), Baum-Welch algorithm (for model estimation), and POMDP formulations. In addition to the noise-free observation model assumptions in [5], our solution outperforms both the Minimum Entropy Merging (MEM) algorithms detailed in it, i.e., Markov Process Estimation coupled with Greedy Clustering (MPE-GC) and Markov Process Estimation (MPE) coupled with Minimum Entropy Increment Clustering (MPE-MEI). Additionally, among works that tackle this problem as an HMM framework [6] like we do, the Viterbi algorithm is featured as a potential solution for occupancy behavior estimation. As illustrated in the subsequent sections of this work, not only does our solution outperform the Viterbi algorithm (with the same channel sensing limitations), our solution also provides for an online transition model estimation algorithm that operates concurrently with the approximate POMDP value iteration algorithm, i.e., PERSEUS. In contrast, the proposals outlined in [6] and [5] determine the time-frequency incumbent occupancy correlation structure offline using pre-loaded databases, which is inefficient in non-stationary settings.

In this section, we have detailed the need for spectrum sharing from an administrative, economic, and scientific perspective, which serves as the motivation for our work. In view of this need, we have hinted at the prescience of DARPA in establishing the SC2, the design of our radios specifically for this competition, and the technologies/standards to be born out of this Grand Challenge. Furthermore, narrowing our focus down to a sub-problem in cognitive radio design, i.e., spectrum sensing and access in the MAC layer of the radio, we have introduced the design of our solution along with a comparison, both in terms of the capabilities and the system performance, with other similar works in the state-of-the-art. Additionally, the subsequent sections of this work will present illustrations and metrics regarding the implementation of the optimal POMDP channel sensing and access policy on an ad-hoc distributed test-bed consisting of ESP32 radios, which will establish the simplicity in the policy's implementation.

## 2. THE DARPA SC2 RADIO DESIGN

As discussed in Chapter 1, the ever-increasing demand for spectrum resources--fueled by the massive MTC (IoT devices, VANs, WSNs), high-capacity user applications (UHD real-time video streaming, cloud-based applications, consumer device diversity), and advancements in military technologies--has caused governments around the world to focus a lot of resources at scaling up, protecting, and maintaining the wireless communication infrastructure. In this regard, the DARPA Spectrum Collaboration Challenge (SC2) aims to bring in the enormous might of private industry, academic research organizations, and independent technology entrepreneurs to compete in a nation-wide competition that would result in the delivery of breakthrough technologies in the cognitive radio space--particularly promoting the exploitation of advancements in collaborative A.I. and the plethora of capabilities offered by software-defined radios.

In this chapter, we discuss the design of the software-defined radio node developed by the Purdue BAM! Wireless team which competed in the DARPA SC2. The BAM! Wireless software-defined radio operates as either a Standard Radio Node (SRN)--of which there are many in a given emulation scenario--the primary responsibilities for which include:

- Interacting with the DARPA SC2 emulation environment, i.e., the Colosseum, using the pre-determined and agreed-upon radio API,
- Delivering the assigned network flows within our network,
- Performing Power Spectral Density (PSD) measurements
- Following the various decisions made by the Gateway node in every layer of the protocol stack, and

- Sharing its PSD measurements, along with any other relevant metrics, with the Gateway node;

or a Gateway node--of which there is just one deployed for our network in a given emulation scenario--the primary responsibilities for which include:

- Communicating with other competitor networks using the peer-reviewed and standardised Collaborative Intelligent Radio Network (CIRN) Interaction Language (CIL) over a dedicated collaboration channel,
- Coordinating channel access decisions, bandwidth allocation decisions, incumbent interference monitoring activities, and scoring analysis activities, among others.

In Section 2.1 of this chapter, we discuss the various strategies employed in different layers of our radio's network protocol stack; in Section 2.2, we detail the radio API; in Section 2.3, we describe the CIRN Interface Language (CIL); in Section 2.4, we present the performance results of various components of our radio operating in various military and disaster-relief scenarios; and finally, in Section 2.5, we present our concluding arguments.

## 2.1 Cognitive Radio Subsystems

### 2.1.1 Transmission Power Control (PHY)

In the “Passive Incumbent Protect” SC2 scenario--emulated in the Colosseum--a licensed radio has prioritized access to the RF spectrum (20MHz bandwidth) and requires that the aggregate interference caused to its transmissions, in that bandwidth, by the radios of the competitors, should be below a specified threshold [25]. This threshold varies over time and varies based on the amounts of interference caused by the cognitive radio nodes, i.e., the threshold changes over time according to a pre-configured schedule and there are additional threshold tightening challenges wherein

if the aggregate interference caused by the competitors exceeds the nominal threshold, the threshold becomes more stringent and will stay there until the aggregate interference drops below this stricter threshold--additionally, the threshold will drop back to its nominal setting if the aggregate interference stays below the stricter threshold for a pre-determined duration. We incorporate transmission power control in our radio's design to specifically address the challenges brought on by this incumbent protection scenario. As a part of its design the Passive Incumbent, emulated in the Colosseum, performs periodic relative-power measurements in its specified 20MHz RF spectrum of interest, compares these measurements with the current aggregate interference threshold setting, and broadcasts warning or violation messages to the competitors over the collaboration network.

When the relative-power spectrum measurements at the Passive Incumbent exceed its current threshold, and this persists for a pre-defined duration, the Passive Incumbent broadcasts a “Violation” message to the competitors over the collaboration network. Upon receiving this violation message, the transmission power at every SRN within each competitor network should be lowered (collaboratively) until the aggregate interference level at the Passive Incumbent drops below the threshold. The competitor networks can also rely on the Passive Incumbent’s “Report” messages to determine if the SRN transmit power has to be lowered in anticipation of exceeding the allowed interference at the Passive Incumbent.

Now, specifically focusing on the transmission power control strategy at the SRNs in our network, deployed in a Passive Incumbent SC2 scenario: based on the CIL messages received from other competitor networks and knowing the occupancy characteristics of the SRNs in our network, our decision engine first determines the number of radio nodes in this deployment that are accessing portions of the spectrum under observation or under use by the Passive Incumbent, denoted by  $\Lambda$ , i.e., all the radios in all the competitor networks participating in this scenario that are occupying fragments of the 20MHz spectrum under observation by the Passive Incumbent--including our own--we reduce the transmission power of our SRNs by a factor proportional to the

difference between the aggregate interference power measured by the Passive Incumbent and its current threshold setting, divided by  $\Lambda$ , i.e.,

$$P'_T = P_T - \left( \frac{P_{obs} - \omega P_{th}}{\Lambda} \right), \quad (2.1)$$

where  $P_T$  denotes the transmit power of SRNs in our network (in dBFS),

$$P_{obs} = 10 \log_{10}(I^2 + Q^2) - 31.5 \quad [25], \quad (2.2)$$

denotes the relative-power measured at the Passive Incumbent (in dBFS),  $P_{th}$  denotes the current threshold of the Passive Incumbent (in dBFS), and  $\omega$  (slightly less than 1) is a multiplicative offset factor to account for synchronization inconsistencies. Here, for this formulation to work successfully and result in the aggregate interference observed at the Passive Incumbent to be

$$P_{obs} = \sum_{\nu} \sum_{\mu} (P'_T)^{\nu, \mu} = \sum_{\lambda=1}^{\Lambda} \left( P_T - \frac{\Lambda P_T}{\Lambda} + \frac{\omega P_{th}}{\Lambda} \right) = \omega P_{th} < P_{th}, \quad (2.3)$$

where  $\nu$  represents the competitor index,  $\mu$  represents the SRN index within a given competitor network--we ensure that all competitors make similar adjustments in the transmit power of their respective SRNs, by collaborating with them via CIL messages over the collaboration network.

### 2.1.2 Network Control Data Broadcast

There are two types of network control messages employed in our radio. The first being the so-called “short” control message that is transmitted using a non-coherent 8-FSK modulated link using a channel of 480kHz bandwidth and whose center frequency changes every second between the upper and lower band edges of the RF spectrum determined by the Colosseum environment; with Reed-Solomon error control coding; and a time-hopping synchronization technique. Every SRN broadcasts these short control messages to all the other SRNs in our network, and these messages contain information regarding the number of SRNs in our network, the current channel allocation of this SRN, and a timestamp. This FSK control

channel uses the following media access scheme: Considering a slot duration of 60ms, at the beginning of each time-slot, each SRN in our network transmits its short control message packet over the FSK control channel with a probability of  $\frac{1}{\tilde{\Lambda}}$ , where  $\tilde{\Lambda}$  denotes the number of SRNs in our network--additionally, packet collisions that occur between/among concurrent short control message transmissions of two or more SRNs result in decoding errors, and are hence discarded at the receiver. When an SRN receives a short control message from another SRN in our network, it updates its view of the network, and re-broadcasts this modified “observed network state” to other SRNs in the network. The short control messages are primarily used for network bootstrapping and for the node discovery phase during network setup. Additionally, this FSK control channel serves as a fallback mechanism for all control data in high-interference communication environments.

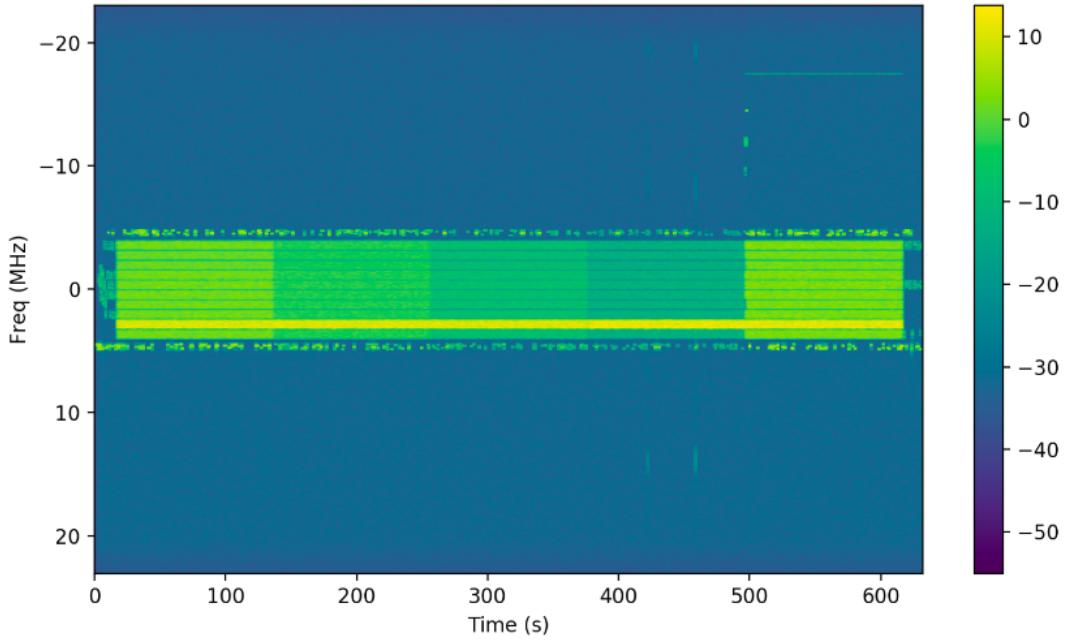


Fig. 2.1. The power spectrum observed by an SRN during an SC2 qualification run

The second type of control message, i.e., the “long” control message, contains information regarding traffic statistics, QoS metrics relevant for scoring (SC2 scoring

“rates” the performance of our network as a whole, by assigning points to successfully completed delivery of assigned flows at each source-destination pair in our network), the estimated noise-variance matrix, and DLL flow schedule updates, among others. Each SRN in our network broadcasts these long control messages every second, interleaved with the data traffic--transmitted over the DFT-s-OFDM high data rate link (discussed in 2.1.3) using the center frequency and bandwidth assigned to this SRN by the Gateway SRNs current channel allocation strategy. Fig. 2.1 illustrates the fact that “short” control transmissions employ the upper and lower band-edges of the spectrum, while “long” control transmissions are interleaved with data and sent over the four wideband channels.

### 2.1.3 The DFT-s-OFDM Data Link

As outlined in Section 2.1.2, both data traffic and “long” control message traffic are transmitted by the SRNs in our network over the high data rate Discrete Fourier Transform-spread-Orthogonal Frequency Division Multiplexing (DFT-s-OFDM) channel--using the center frequency and bandwidth assigned to the corresponding SRN, according to the current channel access action determined by the Gateway SRN. Each SRN, upon receiving the current channel and bandwidth allocations for all the other  $\tilde{\Lambda}-1$  SRNs in our network--over the FSK control link (i.e., “short” control messages) described in Section 2.1.2--tunes its  $\tilde{\Lambda}-1$  receive chains to these  $\tilde{\Lambda}-1$  channels (center frequency and bandwidth information shared by the other SRNs over the FSK control link), and is therefore able to receive transmissions from all the other SRNs in our network. Additionally, these receive chains perform Schmidl & Cox time synchronization and frequency offset compensation, in addition to channel noise variance estimation and frequency-domain equalization. All waveforms in this high data rate link constitute 128 sub-carriers: 108 are used for data symbols, 12 are used for pilot symbols, and the remaining 8 are used for null transmissions that are essential for

channel noise variance estimation (the estimates are used in the MCS adaptation technique described in Section 2.1.4).

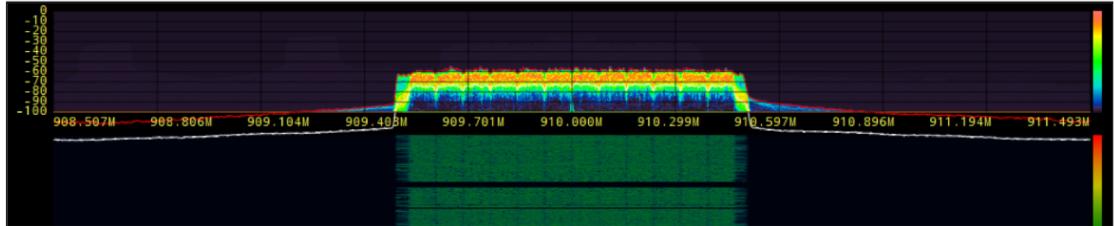


Fig. 2.2. The power spectrum of the DFT-s-OFDM waveform

A segment, which is the simplest data unit in the Data Link Layer (DLL) of our radio, can be an IPv4 segment carrying Colosseum data or an IPv4 segment carrying ARQ data or a control segment. These segments in the DLL constitute the payload in a frame, which is the simplest data unit in the PHY. A PHY frame consists of a frame header--which contains the source SRN ID, the destination SRN ID, the modulation order and code rate (MCS- $(\mathcal{M}, \mathcal{R})$  pair discussed in Section 2.1.4) used for the frame payload, a unique identifier for logging, a 32-bit Cyclic Redundancy Check (CRC), and the number of code-words (blocks) in the frame payload--with the frame header being modulated using QPSK and coded using a rate- $\frac{1}{2}$  Low Density Parity Check (LDPC) code; and the frame payload--which constitutes a sequence of code-words (blocks) populated from a sequence of DLL segments--with this frame payload being modulated and IEEE 802.11 Quasi-Cyclic LDPC (QC-LDPC) coded using an aptly chosen modulation order and code rate (MCS- $(\mathcal{M}, \mathcal{R})$  pair discussed in Section 2.1.4) which is additionally heralded in the frame header, and finally, mapped onto the sub-carriers of a sequence of OFDM symbols. It is important to note here that, as also discussed in Section 2.1.4, the modulation schemes used in our design are QPSK ( $\mathcal{M}=4$ ), QAM16 ( $\mathcal{M}=16$ ), QAM32 ( $\mathcal{M}=32$ ), and QAM64 ( $\mathcal{M}=64$ )--and the code rates ( $\mathcal{R}$ ) used in our design are  $\frac{1}{2}$ ,  $\frac{2}{3}$ ,  $\frac{3}{4}$ , and  $\frac{5}{6}$ . The power spectrum of the DFT-s-OFDM waveform is illustrated in Fig. 2.2.

### 2.1.4 Modulation and Coding Scheme (MCS) Adaptation (PHY)

The MCS adaptation formulation used in the PHY is a link throughput maximization problem, performed at each individual link  $l$  (i.e., a source-destination pair) in our network, subject to constraints on the minimum achieved throughput and maximum allowable bit error probability. This optimization problem involves maximizing the link throughput by finding the modulation order ( $=4, 16, 32, 64$ —corresponding to QPSK, QAM16, QAM32, and QAM64, respectively), denoted by  $\mathcal{M}$ , and the code rate ( $=\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}$ ), denoted by  $\mathcal{R}$ , having the smallest Euclidean distance between two circles centered about adjacent symbols in the constellation diagram, with the radius of each of these circles being proportional to the ratio of the estimated standard deviation of the additive channel noise (and the interference from other SRNs on this channel) to the asymptotic code gain. Additionally, in order to remove noise variance measurements with low probabilities of occurrence, we apply a sliding window median filter to the estimated noise variance. Furthermore, as a part of the constraint on the maximum allowable bit error probability, we employ a closed-form approximation for evaluating the bit error probability, denoted by  $\mathbb{P}_b^{(l)}$ , as a function of the noise variance estimate  $\hat{\sigma}^2$  and the minimum constellation distance  $d_{\min}$ , given by

$$\begin{cases} \mathbb{P}_b^{(l)} = Q\left(\frac{d_{\min}}{\sqrt{2}\hat{\sigma}^2}\right), & \text{if } \mathcal{M} = 4, \\ \mathbb{P}_b^{(l)} = \frac{4}{\log_2(\mathcal{M})}Q\left(\frac{d_{\min}}{\sqrt{2}\hat{\sigma}^2}\right), & \text{if } \mathcal{M} > 4, \end{cases} \quad (2.4)$$

where,

$$Q(x) = \frac{1}{2\pi} \int_x^\infty e^{-\frac{y^2}{2}}. \quad (2.5)$$

The throughput over this link  $l$ , denoted by  $\rho_l$ , is given by

$$\rho_l = \mathcal{R} \cdot BW_l \cdot \log_2(\mathcal{M}), \quad (2.6)$$

where  $BW_l$  refers to the bandwidth allocated by the Gateway SRN for this link (based on PSD measurements and collaboration network messages, as a part of the channel access strategy in the MAC).

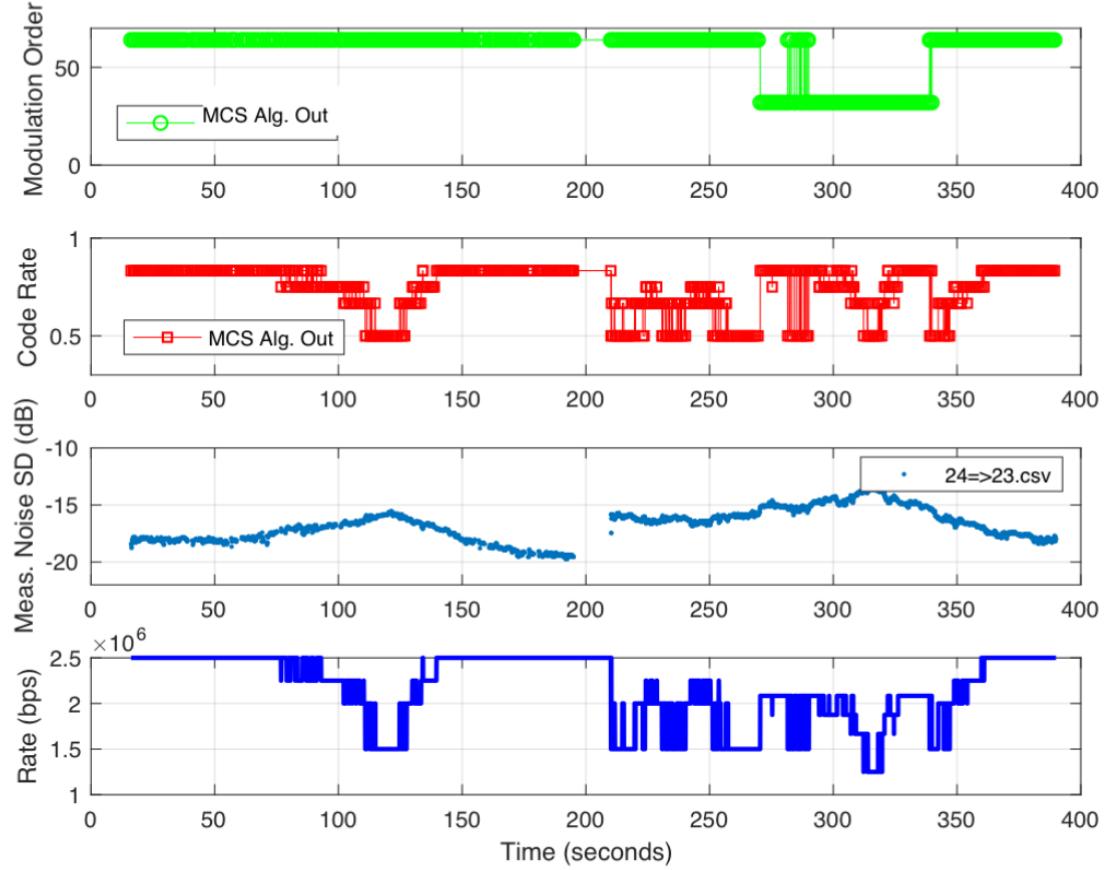


Fig. 2.3. The MCS adaptation scheme during a Payline scenario

So, at each link (source SRN-destination SRN), upon receiving a `NotificationEvent` (an inter-module memo in the C++ publish-subscribe software architecture) which indicates a change in the estimated noise variance of the channel used by this link, the MCS adaptation algorithm involves finding the  $(\mathcal{M}, \mathcal{R})$  pair that has the smallest Euclidean distance between two circles centered about adjacent constellation points, with the radius of each circle given by the ratio of the current estimate of the standard deviation of the channel noise (and the interference caused by other SRNs on this

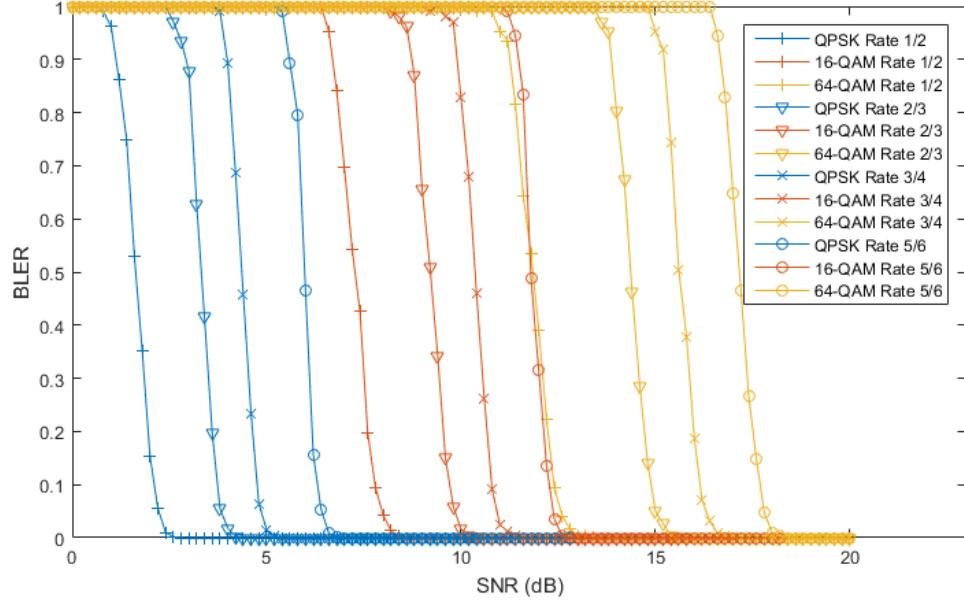


Fig. 2.4. The block error rates (BLERs) for different modulation order and code rate pairs, over varying values of SNR (dB)

channel) to the asymptotic code gain; subject to minimum required throughput and maximum allowable bit error probability requirements, i.e.,

$$\begin{aligned} & \min_{\mathcal{M}, \mathcal{R}} d_{\min}(\mathcal{M}, \mathcal{R}, \hat{\sigma}), \\ & \text{subject to} \\ & \rho_l \geq \rho_{l,\min}, \text{ and} \\ & \mathbb{P}_b^{(l)} \leq \mathbb{P}_b^{(l,\max)}, \end{aligned} \tag{2.7}$$

where  $\rho_{l,\min}$  refers to the QoS constraint of supporting a minimum required throughput,  $\mathbb{P}_b^{(l,\max)}$  refers to the maximum allowable bit error probability,  $\mathbb{P}_b^{(l)}$  is computed using (2.5), and  $\rho_l$  is computed using (2.6). Note here that a channel allocation update triggered by the Gateway SRN in our network, induces changes in the estimated (and filtered) noise variance map, which in turn triggers MCS adaptation. Consequently, MCS adaptation triggers an update to the flow schedule in the DLL, which is discussed in Section 2.1.5. Fig. 2.3 depicts the process of determining the

most appropriate modulation order and coding rate adaptive to the changes in the estimated additive channel noise standard deviation. Note here that the link data rate reduces as the channel noise variance increases. Fig 2.4 illustrates the Block Error Rate (BLER) curves for different combinations of the modulation order and the code rate, over different SNR values.

### 2.1.5 Prioritized Flow Scheduling with ARQ Heuristic (DLL)

The prioritized flow scheduling algorithm employed in the DLL of our radio involves an adaptive time-quanta based deficit round-robin scheduler with a value-per-resource heuristic and recursive revisit, in addition to Automatic Repeat reQuest (ARQ) for bursty User Datagram Protocol (UDP) flows (i.e., emulated file transfers). A DARPA SC2 emulation scenario in the Colosseum involves “mandates” to deliver a variety of flows with pre-specified priorities--quantified by their associated point-values. For instance, Voice Over Internet Protocol (VOIP) traffic is assigned 4 points per flow, the real-time camera feed from an Unmanned Aerial Vehicle (UAV) is assigned 9 points per flow, and the real-time continuous video stream from a water-bomber deployed to fight wildfires in California (emulated scenario) is assigned 15 points per flow. Additionally, a DARPA SC2 scenario also specifically categorizes flows into two groups: intermittent, bursty flows (file transfers) and all other non-bursty traffic.

Based on the available spectrum resources, i.e., channel bandwidth, and the amount of time remaining to complete an assigned flow, each flow is assigned time-quanta--with a non-zero time-quanta assigned to a flow only if it is determined to be likely successful with the given time-quanta. The underlying infrastructure of our scheduler involves individual flow-specific at each SRN in our network--additionally, this includes queues for file bursts and queues for ARQ packets that are supposed to be re-sent to the destination. At each SRN, we employ an ordered resource-fitting heuristic with recursive revisit wherein we first determine the upper and lower

bounds of the resource-block (which termed as a QuantumSchedule)--where the lower bound represents the smallest amount of time required to send a frame corresponding to all the flow-queues at this SRN, subject to the quality of the links; while the upper bound represents the smallest maximum allowable latency among all the flows at this SRN, indicating a temporal resource bound at which “mandates” start to fail. After determining the upper and lower bounds, based on its assigned flow metrics and quality of its links--each SRN ranks (in decreasing order of priority) its assigned flows according a value-per-resource heuristic, determines the percentage utilization of the QuantumSchedule block for each of these flows (i.e., the resource block with the quantified upper and lower bounds), and starts fitting these flows onto the resource block with the added optimization of minimizing the amount of unused resources in this QuantumSchedule block--a recursive revisit strategy helps the scheduler to recursively perform an ordered fit routine on flows that are yet to be fitted onto the resource block with the intention of completely utilizing the QuantumSchedule block.

In more detail, as and when a new flow is assigned to a particular SRN  $i$  in our network, the DLL creates a queue that holds the packets arriving at this SRN from the C2API (discussed in Section 2.2). When a schedule update is triggered by the arrival of new flows at the SRN, or when the channel and bandwidth allocation changes (as notified by the Gateway SRN via OFDMChannelUpdate NotificationEvents), or when the modulation order and code rate pair changes as a result of the MCS adaptation strategy in the PHY of this SRN (notified internally via MCSRequest NotificationEvents)--the scheduler evaluates the smallest amount of temporal resources needed to serve each flow (on a frame-by-frame basis), i.e.,

$$t_{\min}^{(f)} = \frac{f_{\text{ov}} + f_{\text{nseg}}(f_{\text{seg-ov}} + f_{\text{seg-bits}})}{\rho_{l_f}}, \quad (2.8)$$

where  $f_{\text{ov}}$  refers to the frame overhead of flow  $f$ ,  $f_{\text{nseg}}$  refers to the number of DLL segments per frame of flow  $f$ ,  $f_{\text{seg-ov}}$  denotes a per segment overhead per frame of flow  $f$ ,  $f_{\text{seg-bits}}$  is the number of bits per segment of flow  $f$ , and  $\rho_{l_f}$  indicates the link throughput--evaluated using the allocated channel bandwidth, the estimated channel noise variance (and interference from competitor SRNs), and the  $(\mathcal{M}, \mathcal{R})$  pair--with

$l_f$  referring to the link between the source SRN  $i$  and the destination SRN  $j$ , relevant to this flow  $f$ . The upper and lower bounds of the QuantumSchedule resource block, for this specific SRN  $i$ , at this time, are now determined as

$$\begin{aligned} \text{lb}_i &= \sum_{f \in \mathcal{F}_i} t_{\min}^{(f)}, \text{ and} \\ \text{ub}_i &= \min_{f \in \mathcal{F}_i} \delta_{\max}^{(f)}, \text{ respectively,} \end{aligned} \quad (2.9)$$

where  $\mathcal{F}_i$  refers to the set of all flows assigned to SRN  $i$ , and  $\delta_{\max}^{(f)}$  refers to the maximum allowable latency (QoS constraint) of flow  $f$  at SRN  $i$ . The scheduler then ranks the flows at this SRN according to a value-per-resource metric, i.e.,

Rank the flows  $f \in \mathcal{F}_i$  in the decreasing order of

$$\psi_f = \frac{V_f f_{\text{nseg}} (f_{\text{seg-bits}} - f_{\text{seg-payload-ov}})}{t_{\min}^{(f)} \rho_{l_f}}, \quad (2.10)$$

where  $V_f$  denotes the point value of the flow (discussed earlier in this section)  $f \in \mathcal{F}_i$  at SRN  $i$ , and  $f_{\text{seg-payload-ov}}$  denotes the bit overhead in the segment payload per frame corresponding to flow  $f$  at SRN  $i$ . Next, the scheduler “fits” these flows in a prioritized fashion corresponding to the value-per-resource ranked list of the flows at this SRN, denoted by  $\mathcal{S}_i$ , i.e., flows with a higher value-per-resource metric ( $\psi_f$ ) will be fitted first into the QuantumSchedule resource block--if all the flows can be fit into the resource block, we are done; else, the flow with the smallest value-per-resource metric will be removed from this list and added into a revisit list, denoted by  $\tilde{\mathcal{S}}$ , and the fitting operation is tried again on the set  $\mathcal{S} - \tilde{\mathcal{S}}$  in the same prioritized fashion. This procedure takes place until all the flows in the final list  $\mathcal{S}$  can be fit into the resource block, with flows (with comparatively lower value-per-resource) which could not be fit into the resource block will be present in  $\tilde{\mathcal{S}}$ . Next, if there exist no available resources in this QuantumSchedule resource block, the scheduler terminates and publishes the updated schedule to the queueing service controller, thereby leading to all the “high value-per-resource” flows (present in  $\mathcal{S}$ ) being perfectly scheduled in the available resource block (governed by link quality, allocated bandwidth, and flow-specific QoS constraints). On the other hand, if available resources exist in the

QuantumSchedule resource block, the list  $\tilde{\mathcal{S}}$  is recursively traversed, again in the decreasing order of the flow value-per-resource heuristic, and each of these value-per-resource ranked “recursively re-visited” flows are evaluated for their fit into the available/remaining portion of the resource block. These combination of heuristics account for a prioritized flow scheduling strategy in the DLLs of our SRNs--triggered by assigned traffic changes, QoS mandate changes, channel and bandwidth changes, and MCS choice updates; constituting a recursive-revisitation strategy to account for the maximum utilization of an elementary resource block, and a value-per-resource ranking and traversing heuristic to account for the need to prioritize flows that give our network a higher utility per unit resource consumed.

Additionally, ARQ is employed for file flows to ensure that the packets corresponding to these simulated file transfers are reliably sent across the network from the source SRN to the destination SRN. These simulated file transfers correspond to bursty UDP flows, and the packets within these bursts are sequenced by the DLL at the sender SRN. The receiver SRN provides positive acknowledgements (ACKs) for the sequence numbers within a burst which have been successfully received and decoded. A separate ARQ queue is employed in the deficit round-robin queueing infrastructure discussed earlier--based on the feedback from the ARQ mechanism, the scheduler (prioritized value-per-resource strategy with recursive revisit) dynamically reschedules these ARQ packets for re-transmission--the priority of these ARQ flows deteriorates with every re-transmission.

### **2.1.6 Channel and Bandwidth Allocation Heuristic (MAC)**

Based on the given initial network conditions (by the Command and Control radio API (C2API) discussed in Section 2.2), the information gleaned during network discovery over the FSK control channel, and the offered traffic statistics reported by the SRNs over the high data rate DFT-s-OFDM link, the channel and bandwidth allocation algorithm in the Gateway SRN (fusion/data aggregation center) coordi-

nates a center frequency and bandwidth assignment to all the back-logged SRNs in our network. An environment update--such as a change to the offered traffic at certain SRNs, or a change in the minimum required throughput and maximum allowable latency QoS constraints for certain traffic flows, or a change in the overall available RF spectrum bandwidth in the scenario, among others; or a performance notification from our peers, or incumbents (see Section 2.1.1), or our network coordinator (i.e., self: the Gateway SRN)--indicating that the ensemble is performing poorly due to the over-exploitation of spectrum resources by our network, or that the ensemble transmissions are interfering with the incumbent resulting in the aggregate interference observed at the incumbent being higher than the current threshold, or that our network is failing to satisfy the QoS requirements for a significant number of traffic flows for the past pre-specified duration of time (typically, 10 seconds), respectively--triggers a change in the channel and bandwidth allocation.

We decompose this problem into two sub-problems: bandwidth allocation and subsequent center-frequency assignment. Based on the emulation scenario bandwidth, the Gateway SRN's bandwidth allocation strategy assigns bandwidths to each SRN in the network based on the amount of traffic offered to the SRN, the QoS requirements imposed on each of these traffic flows, and the quality of the links.

Post-bandwidth assignment, the channel allocation sub-problem is taken up by the Gateway SRN: it should determine the center frequencies that minimize interference from (and in turn interference to) the other competitors' SRNs in the emulation scenario, while achieving the mandated minimum required throughput and maximum allowable latency QoS constraints imposed on each traffic flow (additional QoS constraints can be imposed on file transfer flows--typically, the time window within which the file transfer needs to be completed from the source SRN to the destination SRN within our network). In order to obtain a complete picture about the spectrum occupancy at a specific SRN in our network (i.e., at a particular location), we exploit two kinds of data: CIL messages from the other competitor networks in the emulation scenario--specifically, the SpectrumUsage messages and the LocationUp-

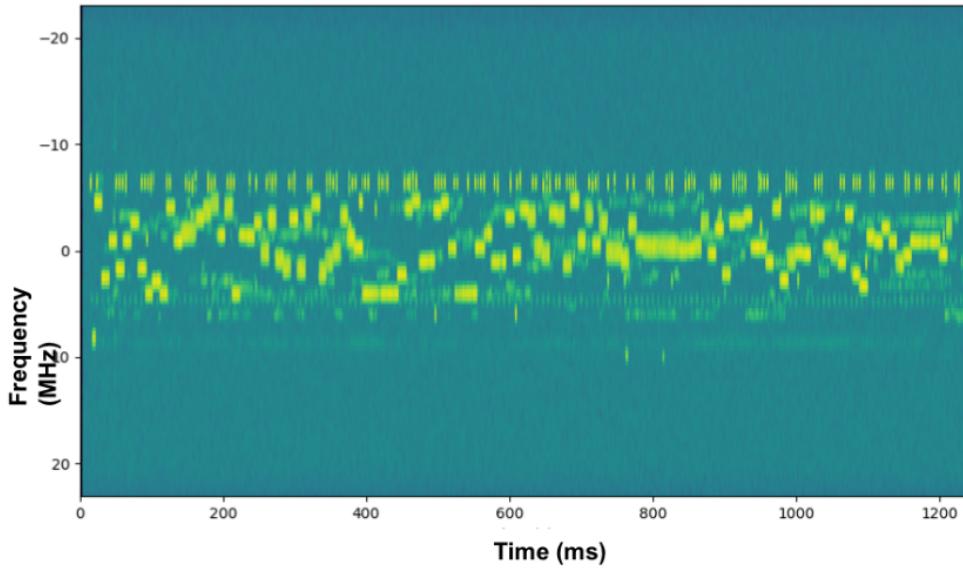


Fig. 2.5. The spectrum occupied by our network during an Alleys of Austin scenario emulation

date messages; and the PSD measurements made at the SRN. The Gateway SRN's channel allocation heuristic employs the data gleaned from the SpectrumUsage and LocationUpdate CIL messages broadcasted over the collaboration network by the other competitors in the emulation scenario, and the collated location-specific PSD measurements in the following fashion: first, the channel gain between a pair of SRNs in our network is estimated based on their location information (which is typically obtained from the C2API, if not, it is obtained from the periodic GPS Notification-Events published by individual SRNs and sent to the Gateway SRN over the high data rate DFT-s-OFDM control link (interleaved with data)) and an empirical path loss exponent; second, based on this estimated channel gain--and a weighted combination of collated PSD measurements and competitor CIL messages--the Gateway SRN determines the amount of interference experienced by our SRNs in each of these channels, and performs a heuristic search to determine the center frequencies that minimize the total interference caused at our SRNs. Fig.2.5 illustrates the spectrum

occupancy of SRNs in our network (as seen by one SRN--collated using PSD measurements and internal occupancy notifications from the Gateway SRN).

### 2.1.7 Multi-hop Routing Heuristic (NET)

When the “short” control messages over the FSK control channel cannot be received by SRN  $i$  from SRN  $j$  for a pre-determined amount of time (typically, 10 seconds), the link  $l_{ji}$  is reported to be blocked/down by SRN  $i$  (similar behavior by all the other SRNs in our network). This routinely occurs in any given DARPA SC2 emulation scenario due to the presence of emulated physical obstacles (hills, buildings, etc.), or the presence of jammers in the network, or the emulation of node mobility characteristics, or high noise and interference in the spectrum as a whole resulting in failed decoding of the short control message packets sent over the FSK control channel (which uses the band-edges of the spectrum). On the other hand, a successful/working link is reported by SRN  $i$  with respect to link  $l_{ji}$ , if short control message packets from SRN  $j$  can be successfully decoded. Hence, a binary vector can be constructed at each individual SRN indicating the status of its incoming links--which is then shared with all the other SRNs in the network over the FSK control channel, in order to build a routing table at each SRN, which is periodically updated throughout the emulation.

During data transmission, assuming a correct and current routing table exists at each SRN (which is true because of the periodic updates sent from all the other SRNs over the robust FSK control link), in the Network layer (NET) of each SRN, Dijkstra’s algorithm is applied to this routing table, to find the route to the destination SRN using the smallest number of hops.

## 2.2 The Colosseum Command and Control radio API (C2API)

The Colosseum Command and Control API (C2API), also referred to as the Radio C2API, is an interface of four shell scripts--namely, start.sh, stop.sh, status.sh, and

statistics.sh; which should be supported by every SRN in the deployment, and the combinations of which are used by the scenario emulator, i.e., the Colosseum, in any given deployment, to orchestrate the following activities [26]:

- Notify all the SRNs in the deployment of changes to the radio environment, i.e., changes to the available scenario bandwidth, Passive Incumbent center frequency and threshold changes (discussed in Section 2.1.1), and scenario stage resets;
- Notify all the SRNs in the deployment of their respective newly assigned traffic flows and the QoS constraints (termed “Individual Mandates (IMs)”) for each of them--in addition to performance mandates for a competitor network as a whole (termed “Network Mandated Outcomes”);
- Provide static Colosseum configuration information such as the channel emulator’s operating frequency, in addition to collaboration network parameters; and
- Administration and Management of individual SRNs in individual competitor networks in the deployment, throughout the scenario execution process.

### 2.3 The CIRN Interface Language (CIL)

As discussed earlier, the DARPA SC2 Grand Challenge involves a collaboration channel (outside the RF environment) that facilitates the cooperative exchange of crucial operational and performance messages among competing CIRNs--in order to not only ensure optimal performance by individual competitor networks, but also by the ensemble, as a whole. By design, every competitor network should have a designated gateway node that communicates with other competitors in the deployment, over the collaboration channel. The gateway SRN corresponding to every competitor network in the deployment will connect to a collaboration server over a wired IP link (ip\_addr:access\_interface), with this collaboration server acting as a publisher-

subscriber--tracking competitor SRNs in the deployment, and re-directing peer-to-peer collaboration traffic from the sender gateway to the destination gateway [15].

The collaboration API involves two types of messages [27]:

- Client-Server: a set of procedures and semantics to be followed by competitor networks in the deployment, while interacting with the collaboration server in order to “discover” the addresses of other competitors in the scenario emulation; and
- Peer-to-Peer: a set of procedures and semantics that forms the crux of the CIRN Interface Language (CIL) specifications, and is used to describe the collaborative information exchange between two competitor networks in the deployment.

The format/structure/semantics of these messages are described in the now standardised CIL protocol specifications. We list a few important collaboration client-server and peer-to-peer messages below:

- TalkToServerMessage: All client-server messages, i.e., the messages from a gateway SRN to the collaboration server, should be formatted according to this top-level wrapper--derived instances include Register (a gateway SRN wishes to register its network with the collaboration server), KeepAlive (serves as a heartbeat message from the gateway, informing the collaboration server that it is still active), and Leave (a gateway SRN informs the collaboration server that its network wishes to leave the collaboration channel);
- TalkToClientMessage: All messages from the collaboration server to the client are instances of this parent structure--encapsulates Inform (response to Register from the client: the collaboration server sends the unique client ID, maximum keep-alive count, and a list of registered competitors in the collaboration network (i.e., their client IDs and IP addresses); and Notify (the collaboration server broadcasts to all registered clients on the collaboration network that either a new competitor has successfully registered with it, or an existing competitor has left the collaboration channel);

- CilMessage: A top-level wrapper encapsulating all peer-to-peer messages in the collaboration network--specifically, IncumbentNotify and IncumbentPassiveInfo (Passive Incumbent messages: discussed in Section 2.1), LocationUpdate (GPS location information for all SRNs in a competitor network), SpectrumUsage (the exact spectrum voxels, i.e., time-frequency resources, being accessed by SRNs in a competitor network, along with their perceived importance to satisfying the network's assigned mandates), and DetailedPerformance (total number of QoS constraints assigned, total number of QoS constraints satisfied, total score achieved, score threshold for ensemble utility optimization, etc.).

## 2.4 Capabilities and Performance Evaluations from DARPA SC2 scenario emulations

In this section, we present plots illustrating the operational capabilities and the performance of our SRNs (and our network, as a whole) in a military deployment scenario, i.e., Alleys of Austin, and a disaster relief deployment scenario, i.e., Wildfire.

### 2.4.1 Alleys of Austin

This emulated military deployment scenario in the Colosseum, involves a 45-member platoon (with an Unmanned Aerial Vehicle (UAV)) from the Texas Army National Guard practicing urban maneuvers and communications in Austin, Texas. The platoon is divided into 5 squads, with 9 members in each squad, and is moving through the streets of Austin in three stages--involved in basic voice communications in Stage 1, Stage 2 involves the exchange of voice, imagery, and video, and Stage 3 involves a significant increase in the amount of traffic (voice, imagery, and video) exchanged among the squad members. Each competitor CIRN represents a squad--thereby resulting in a 5 team, 50 node ( $5 \cdot 9 = 45$  SRNs +  $5 \cdot 1 = 5$  gateway SRNs, with an emulated UAV), large-scale, small packet, military deployment scenario, em-

ploying a single-tap propagation model [14]. Each team must achieve 50% of the QoS mandates assigned to it, during the scenario emulation--in a given time snapshot, teams get scores only if every competitor network achieves their corresponding desired performance in that time snapshot, thereby each competitor is incentivized to work with the others in the network in achieving the required mandates.

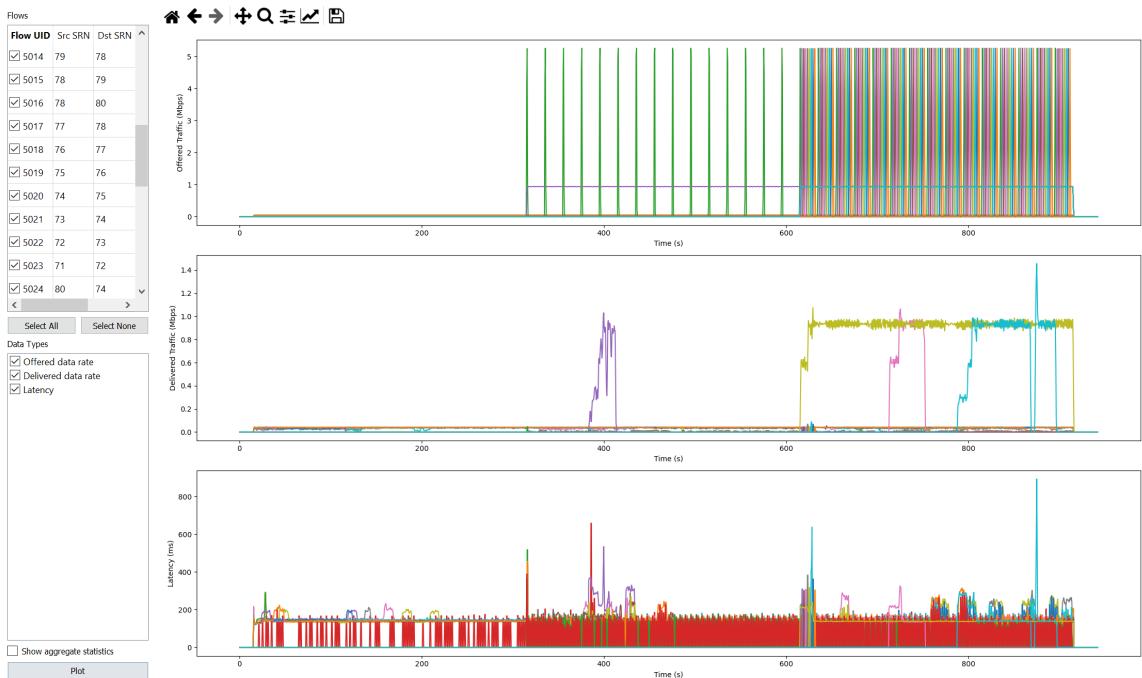


Fig. 2.6. The offered and delivered traffic corresponding to various (selected) flows during the Alleys of Austin scenario emulation

Fig. 2.6 illustrates the offered traffic corresponding to the individual selected flows, their corresponding delivered traffic, and the latency experienced by these flows as they are being scheduled/delivered by our network. Note here that our network is able to deliver almost all of these flows with minimal latency. Also, note the stages in the scenario emulation--as described earlier, Stage 1 constitutes basic voice traffic, i.e., the initial flows have a low offered traffic data rate of 40kbps; Stage 2 constitutes voice, imagery, and video traffic, i.e., the flows in this Stage have low offered traffic data rate, but imagery and video flows carry higher point values (flow priorities); and

finally Stage 3 involves a significant increase in the amount of offered traffic for the voice, imagery, and video flows. From Fig. 2.6, we observe that our network is able to deliver most of the flows in Stages 1 and 2--however, in Stage 3, as is usually expected, due to the extremely high offered traffic, our network is not able to satisfy all the assigned flows.

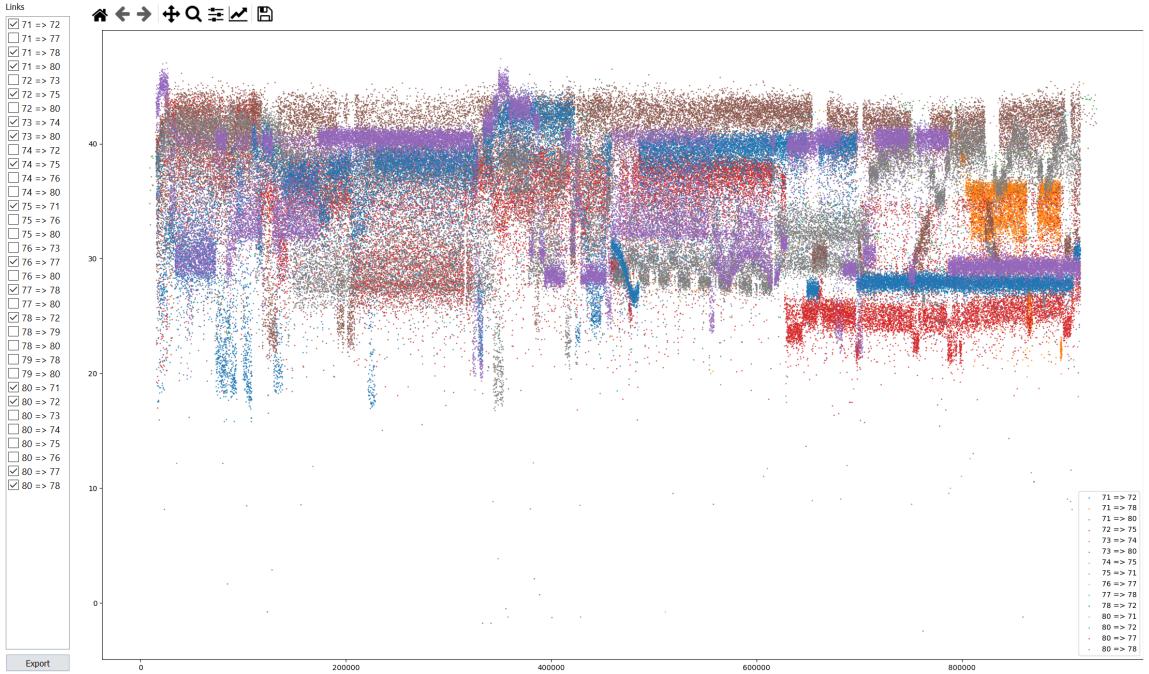


Fig. 2.7. The estimated SNR on various (selected links) during the Alleys of Austin scenario emulation

Fig. 2.7 depicts the estimated SNR on various selected links (source SRN - destination SRN), during the progression of the Alleys of Austin scenario--these SNR estimates at each link are, as discussed earlier, employed in the MCS adaptation scheme, the prioritized flow scheduler, and the bandwidth allocation algorithm. Consequently, Fig. 2.8 depicts the MCS adaptation scheme at each of these selected links, adapting to the changing estimated SNR with respect to the corresponding channels used by these links--the Y-axis of the MCS adaptation plot in Fig. 2.8 refers to the C++ enumeration value corresponding to the  $(\mathcal{M}, \mathcal{R})$  pair, i.e., for example, 43 in

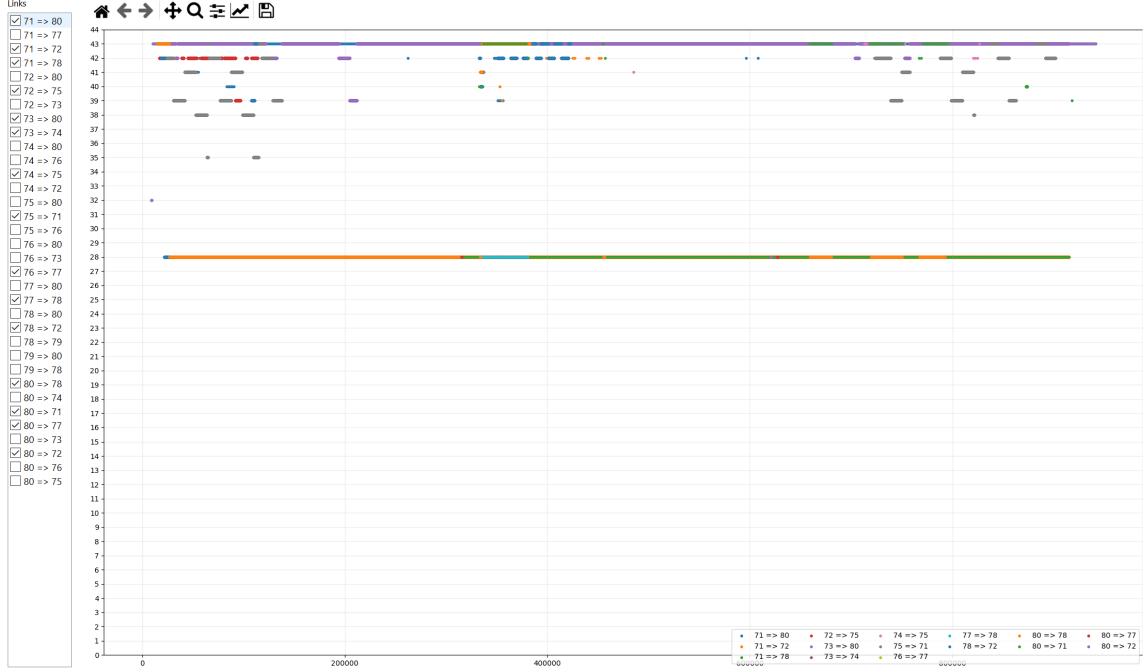


Fig. 2.8. The MCS adaptation scheme at each (selected) link during the Alleys of Austin scenario emulation

the plot refers to QAM64 modulation with a code rate of  $\frac{5}{6}$ , while 28 refers to QPSK modulation with  $\frac{1}{2}$  code rate--in other words, the highest modulation order used in this Alleys of Austin run is  $\mathcal{M}=64$  (i.e., QAM64), while the lowest modulation order used in  $\mathcal{M}=4$  (i.e., QPSK).

Fig. 2.9 illustrates the score obtained by our network (indicated by our Gateway SRN's IP address) along with the scores obtained by our peers (indicated by the IP addresses of their corresponding gateway SRNs) in the scenario emulation. The dotted lines in the first sub-plot indicate the score thresholds that need to be achieved by each network, in a Stage. As is evident from the figure, our network satisfies/exceeds the score threshold in Stages 1 and 2, but fails in Stage 3--however, our network does have the best performance among all the teams in the scenario emulation. It is important to note here that even though there were 5 teams in the emulation of this Alleys of Austin scenario, only the scores of 3 of them are visualized in this figure--2 teams did

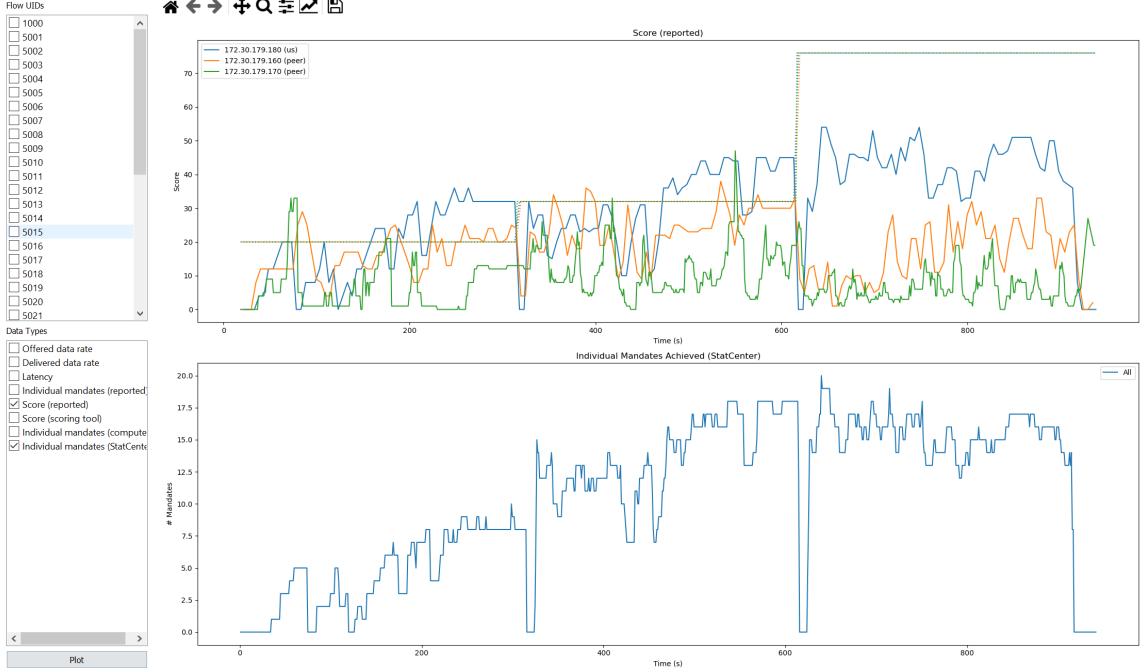


Fig. 2.9. The scores attained by our network and other competing networks during the Alleys of Austin scenario emulation, in addition to the number of QoS mandates satisfied by our network in a given time snapshot

not report their scores over the collaboration network, and hence their scores were not visualized in this illustration. The second sub-plot illustrates the number of flow-specific QoS mandates that were achieved by our network in a given time snapshot.

The reported GPS locations (latitude, longitude) of the SRNs in our CIRN, and the SRNs in our competitors' CIRNs, over time, are depicted in a 3-dimensional time plot shown in Fig. 2.10. The location information of these SRNs--particularly, in relation to physical obstacles (emulated) and competitor SRNs, is employed in the channel allocation algorithm in our Gateway SRN. Fig. 2.11 illustrates the various timestamped CIL messages (both Client-Server messages and Peer-to-Peer messages) received by our Gateway SRN over the collaboration network--these messages are exploited by our Gateway SRN, which along with the collated location-specific PSD measurements: one such PSD measurement in a given time snapshot observed at

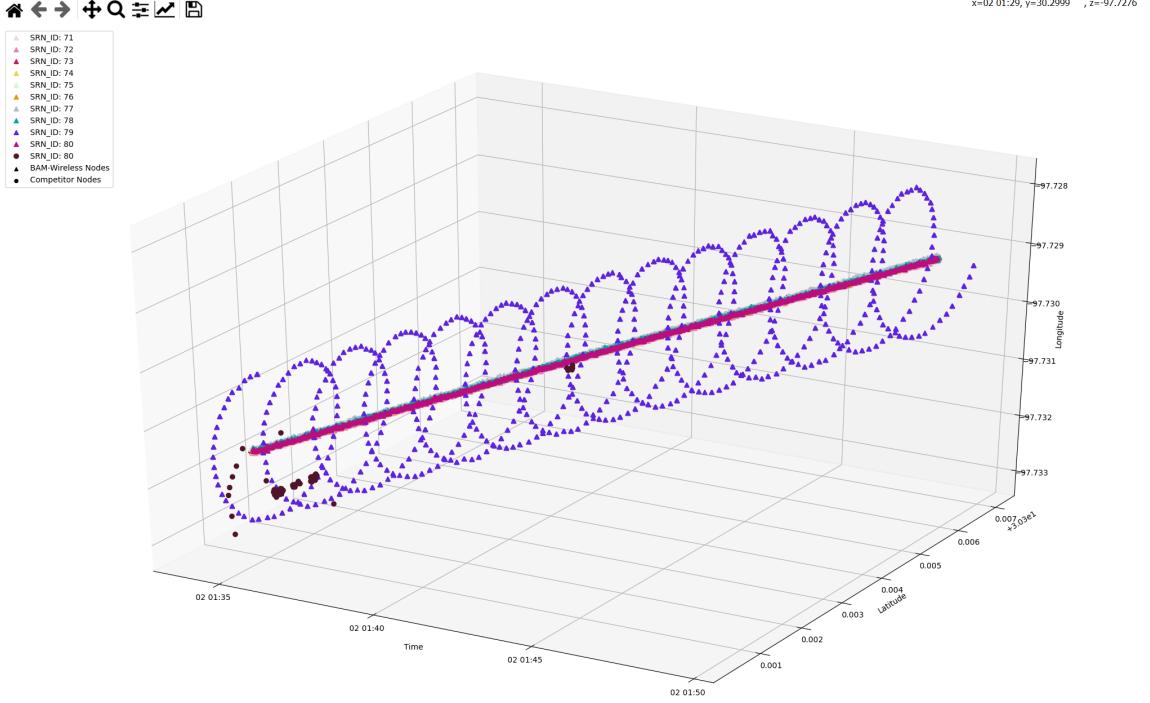


Fig. 2.10. The reported GPS locations of our SRNs, in addition to those of our competing SRNs, during the Alleys of Austin scenario emulation

SRN with ID-72 and sent to our Gateway, is depicted in Fig. 2.12--determines a channel allocation (center frequencies) that minimizes the total interference caused at our SRNs by our competitors. Consequently, from an overall scenario duration perspective, the channel allocations of our network are collated into one “big-picture” illustration, shown in Fig. 2.13. It is evident from Fig. 2.13 that in Stage 1, there are frequent channel re-allocations and bandwidth adjustments, as evidenced by the breaks in the beginning of the plot--this is due to the fact that all the networks, including ours, are trying to find an “equilibrium” allocation that not only helps them achieve their mandates, but also ensures that the ensemble, as whole, achieves its score threshold; Stage 2 has a stable channel and bandwidth allocation, as evidenced by the lower number of breaks in the middle of the plot, due to “equilibrium” being achieved by our network and still a lower amount of traffic, similar in scale to Stage

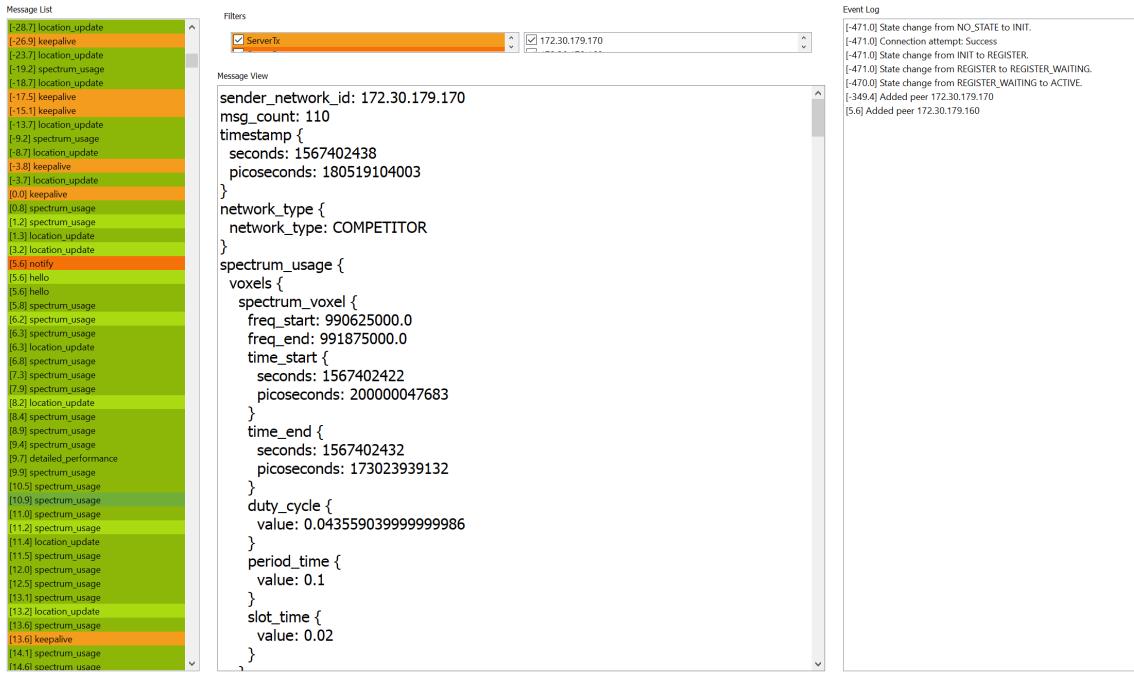


Fig. 2.11. The description of various CIL message exchanges between our network and the collaboration server, in addition to those between our network and our peers, during the Alleys of Austin scenario emulation

1; and finally, in Stage 3, the significant increase in the offered traffic of all three flows (voice, imagery, and video) places an enormous strain on the limited available RF spectrum allotted for this scenario emulation, and since all teams are vying for the same spectrum resources all the time, due to the increased offered traffic, we see significant breaks towards the end of the plot, indicating frequent bandwidth adjustments and channel re-allocations--this causes poor performance in Stage 3, as seen in Fig. 2.6 and Fig. 2.9, but our network still outperforms our competitors in this Stage.

#### 2.4.2 Wildfire

This scenario emulates a disaster-relief deployment--5 National Guard units have been deployed to the Lake Sutherland area in Washington in order to fight a wildfire

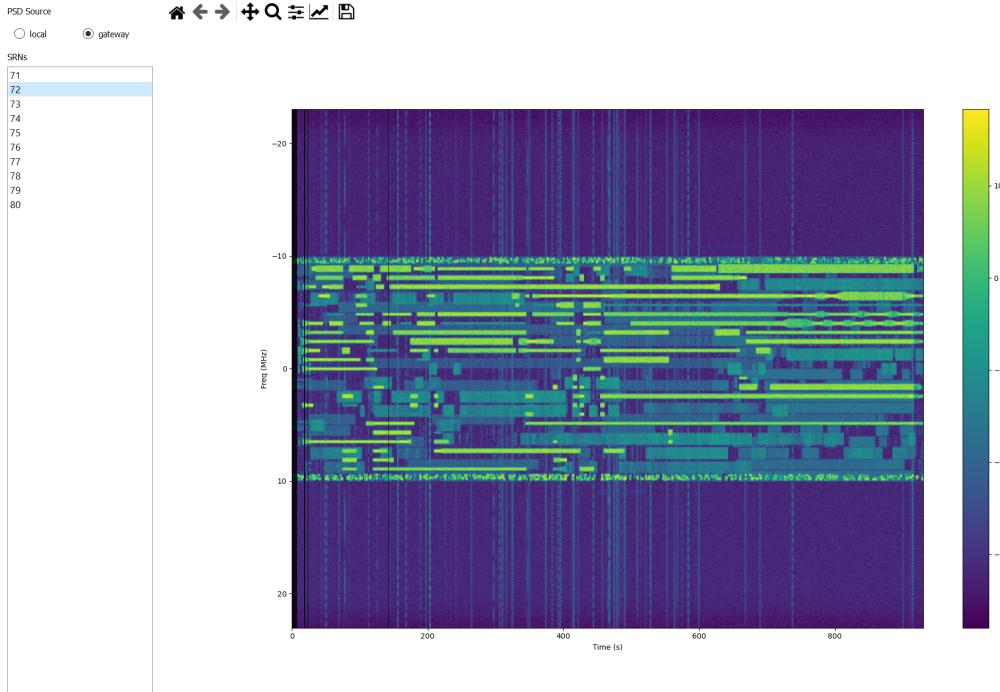


Fig. 2.12. The PSD measurements received at our Gateway SRN from one particular SRN in our network (selected), in order to obtain location-specific spectrum occupancy information, during the Alleys of Austin scenario emulation

that is raging there. Each of these 5 National Guard teams are equipped with an Aerostat to assist in communications, and 4 water-bombers (unmanned water tankers) to put out the blaze. This scenario emulation involves 6 stages: the water-bombers get into position in Stage 1; and in Stages 2 – 6, 5 water-bombers from 2 teams are sent to drop their load onto the blaze, during which these 5 water-bombers stream high-definition video back (Video\_Bombing\_Run flows–15 points per flow) to the Aerostat (i.e., the unit command and control center), while the remaining loitering 15 water-bombers send out basic pings to the Aerostat--after dropping their load onto the fire, these 5 water-bombers come back to their initial position to reload, the next 5 go out to drop their load, and the process continues until the end of this scenario emulation. Each competitor CIRN represents a National Guard unit--thereby resulting in a 5 team, 50 node, large-scale, small packet, disaster-relief deployment scenario,

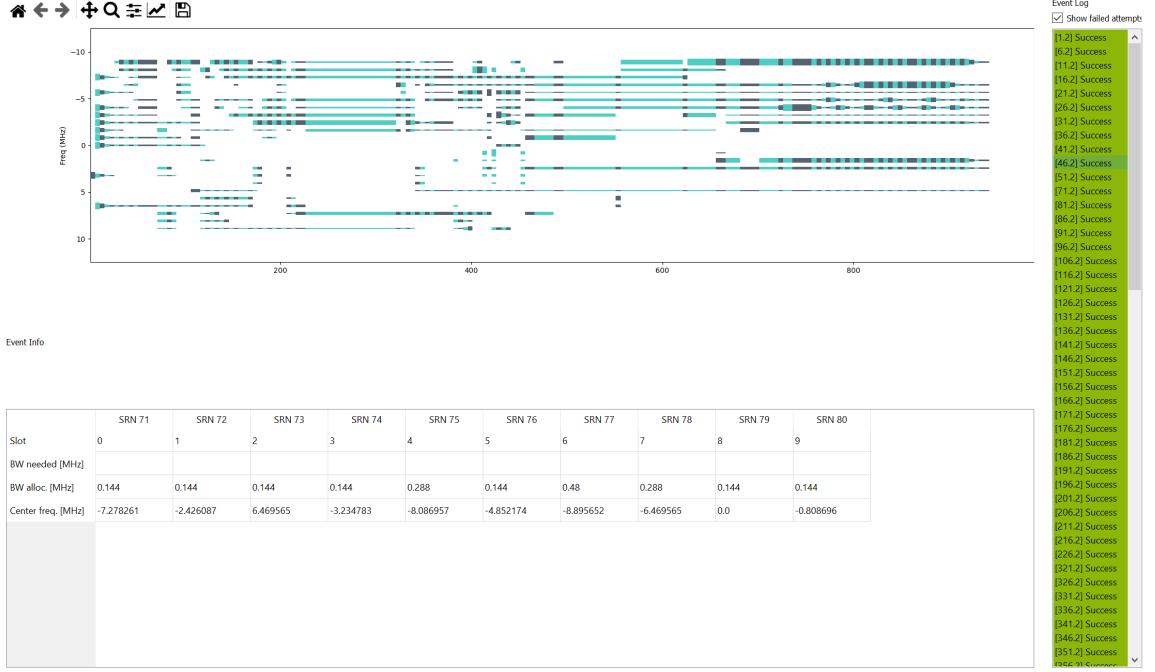


Fig. 2.13. The spectrum occupancy behavior of SRNs in our network (as guided by our Gateway SRN) during the Alleys of Austin scenario emulation

employing a single-tap propagation model [14]. Each team must achieve 50% of the QoS mandates assigned to it, during the scenario emulation--in a given time snapshot, teams get scores only if every competitor network achieves their corresponding desired performance in that time snapshot, thereby each competitor is incentivized to work with the others in the network in achieving the required mandates. Note here that the main challenge in this scenario emulation is to collaborate with our competitors in this deployment in order to ensure that the “active” SRNs, i.e., the water-bombers that are sent out to drop their load get most of the spectrum because of the high-definition video flows communicated by these “active” water-bombers to the Aerostat, are emulated/modeled to need ~20% of the total allocated scenario bandwidth--therefore, teams must work together to facilitate efficient allocation of resources to these “active” SRNs in any given time snapshot [14].

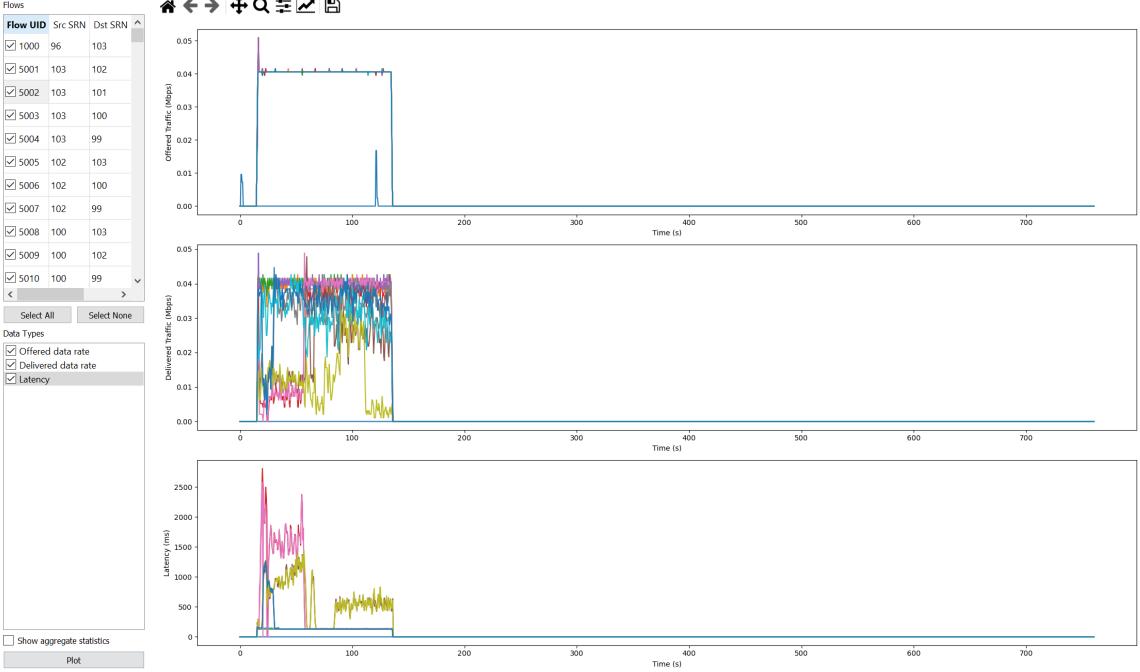


Fig. 2.14. The offered and delivered traffic corresponding to various (selected) links during the Wildfire scenario emulation (Stage 1)

Figs. 2.14 and Fig. 2.15 illustrate the offered traffic corresponding to the individual selected flows, their corresponding delivered traffic, and the latency experienced by these flows as they are being scheduled/delivered by our network, with respect to Stage 1 and Stage 5 of the scenario emulation, respectively. Note here that our network is able to deliver almost all of these flows with minimal latency.

Fig. 2.16 depicts the estimated SNR on various selected links (source SRN - destination SRN), during the progression of the Wildfire scenario--these SNR estimates at each link are, as discussed earlier, employed in the MCS adaptation scheme, the prioritized flow scheduler, and the bandwidth allocation algorithm. Consequently, Fig. 2.17 depicts the MCS adaptation scheme at each of these selected links, adapting to the changing estimated SNR with respect to the corresponding channels used by these links--the Y-axis of the MCS adaptation plot in Fig. 2.17 refers to the C++ enumeration value corresponding to the  $(\mathcal{M}, \mathcal{R})$  pair, i.e., for example, 43 in

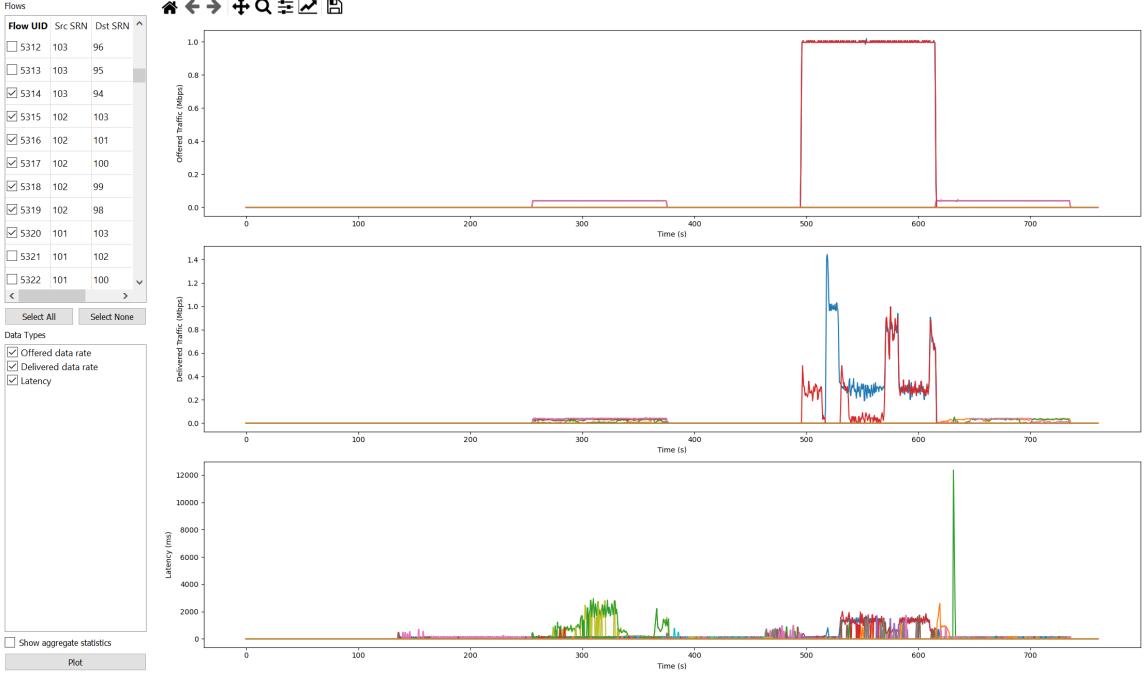


Fig. 2.15. The offered and delivered traffic corresponding to various (selected) links during the Wildfire scenario emulation (Stage 5)

the plot refers to QAM64 modulation with a code rate of  $\frac{5}{6}$ , while 28 refers to QPSK modulation with  $\frac{1}{2}$  code rate--in other words, the highest modulation order used in this Wildfire run is  $\mathcal{M}=64$  (i.e., QAM64), while the lowest modulation order used in  $\mathcal{M}=4$  (i.e., QPSK).

Fig. 2.18 illustrates the score obtained by our network (indicated by our Gateway SRN's IP address) along with the scores obtained by our peers (indicated by the IP addresses of their corresponding gateway SRNs) in the scenario emulation. The dotted lines in the first sub-plot indicate the score thresholds that need to be achieved by each network, in a Stage. The second sub-plot illustrates the number of flow-specific QoS mandates that were achieved by our network in a given time snapshot. As evident from the figure, our peer with IP address identifier 172.30.202.183 performs better than our network, while the other peer with IP address identifier 172.30.202.193 performs poorly throughout the scenario emulation. Note here that, our network

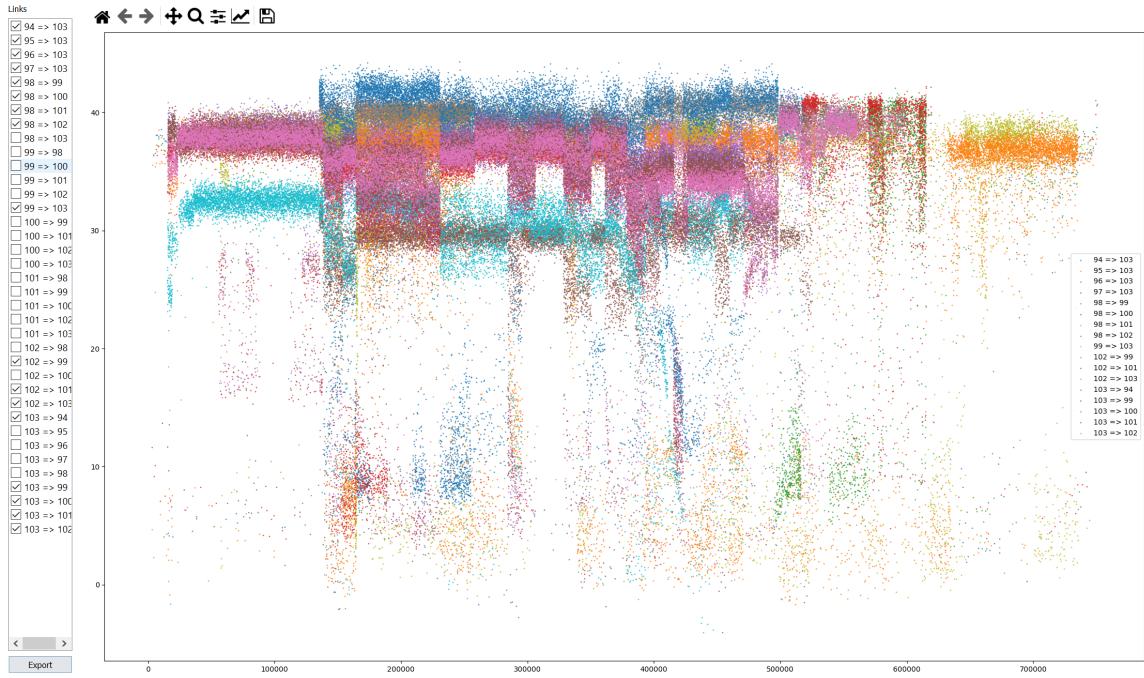


Fig. 2.16. The estimated SNR on various (selected links) during the Wildfire scenario emulation

performs quite well in Stages 1, 2, and 3, with our scores above the performance threshold most of time during these initial 3 stages--however, in Stages 4, 5, and 6, frequent MCS pair switching (adaptation and re-adaptation) as seen in Fig. 2.17, possibly triggered by poor link quality at a significant number of SRNs (possibly due to aggressive interference from competitor SRNs--specifically, the SRNs from CIRN 172.30.202.183), causes our performance to deteriorate significantly--leading to scores below the threshold in these stages.

The reported GPS locations (latitude, longitude) of the SRNs in our CIRN, and the SRNs in our competitors' CIRNs, over time, are depicted in a 3-dimensional time plot shown in Fig. 2.19. The location information of these SRNs--particularly, in relation to physical obstacles (emulated) and competitor SRNs, is employed in the channel allocation algorithm in our Gateway SRN. Fig. 2.20 illustrates the various timestamped CIL messages (both Client-Server messages and Peer-to-Peer messages) received by our Gateway SRN over the collaboration network--these messages are

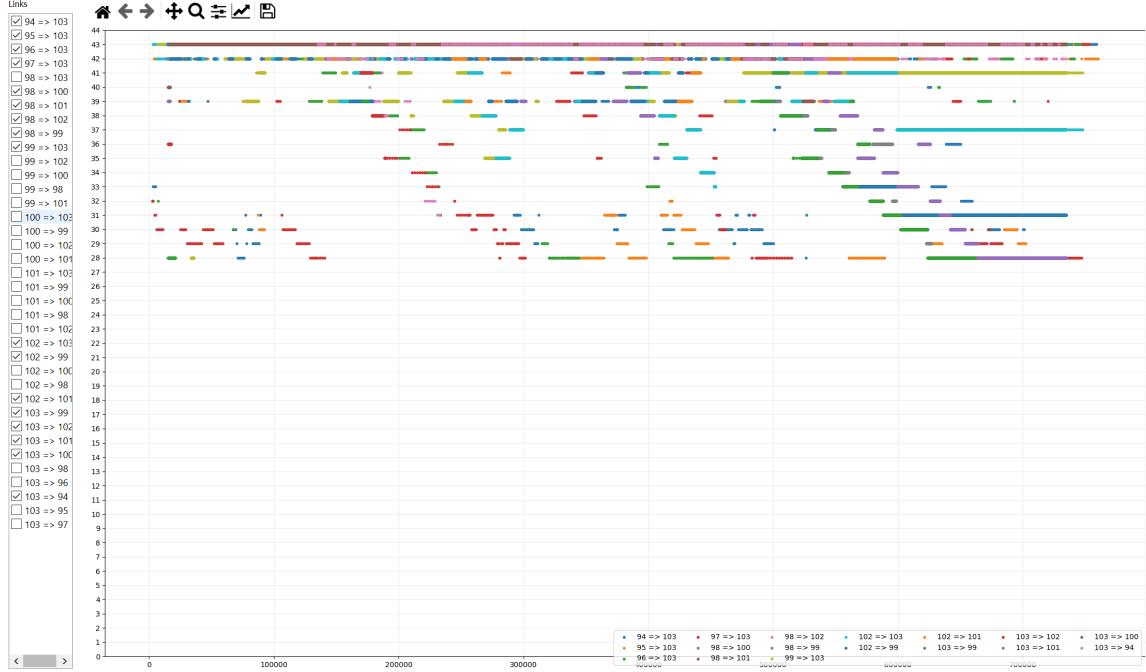


Fig. 2.17. The MCS adaptation scheme at each (selected) link during the Wildfire scenario emulation

exploited by our Gateway SRN, which along with the collated location-specific PSD measurements: one such PSD measurement in a given time snapshot observed at SRN with ID-95 and sent to our Gateway, is depicted in Fig. 2.21--determines a channel allocation (center frequencies) that minimizes the total interference caused at our SRNs by our competitors. Consequently, from an overall scenario duration perspective, the channel allocations of our network are collated into one “big-picture” illustration, shown in Fig. 2.22--it is evident from this figure (the plot and the table underneath it) that the channel and bandwidth allocation strategy of our network is pretty consistent and stable throughout the scenario emulation.

## 2.5 Conclusion

In this chapter, we discussed the design principles of our cognitive radio developed for the DARPA SC2 Grand Challenge--the transmission power control algorithm

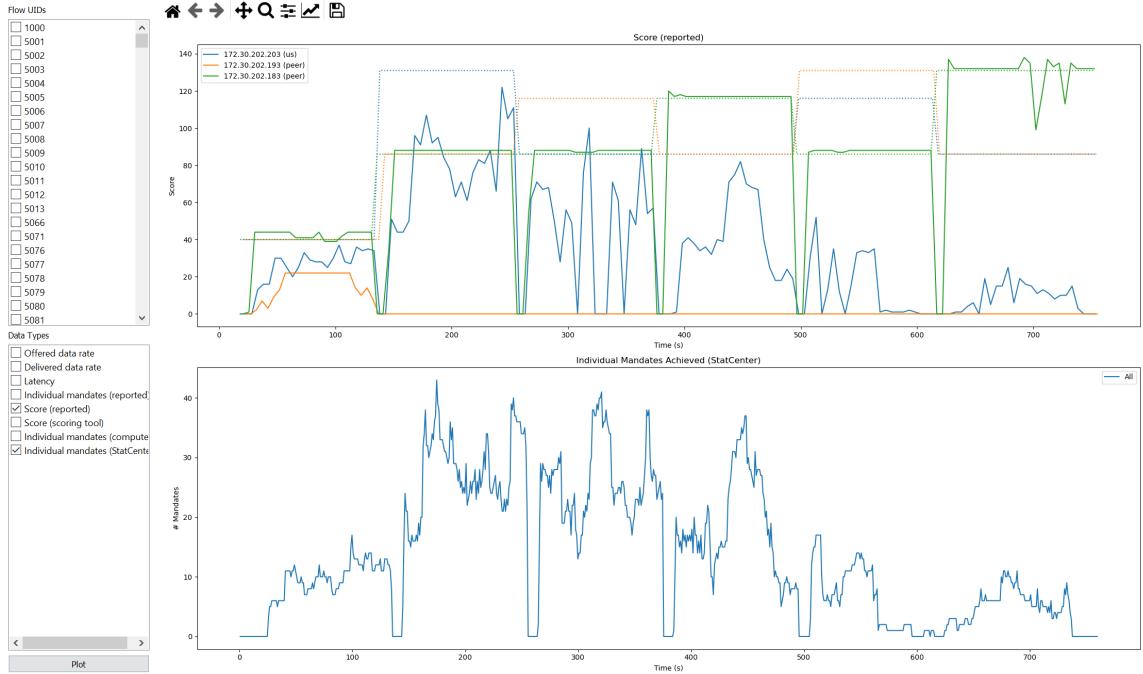


Fig. 2.18. The scores attained by our network and other competing networks during the Wildfire scenario emulation, in addition to the number of QoS mandates satisfied by our network in a given time snapshot

in the PHY for incumbent protection, the robust FSK control channel design, the high data rate DFT spread OFDM link for data traffic and “long” control messages, the MCS adaptation algorithm in the PHY, the prioritized value-per-resource flow scheduling heuristic in the DLL with recursive revisit, the channel and bandwidth allocation heuristic in the MAC, and the Dijkstra algorithm based multi-hop routing heuristic in the NET; the Radio Command and Control API (C2API) of the Colosseum that facilitates network management and orchestration; and the CIRN Interface Language (CIL) specifications, along with details about the collaboration network. Additionally, we presented component-wise plots illustrating the operational capabilities of our radios--driven by the underlying design principles mentioned earlier, and evaluation of our network’s performance against that of our competitors (other teams in this Grand Challenge: Northeastern University, Vanderbilt Univer-

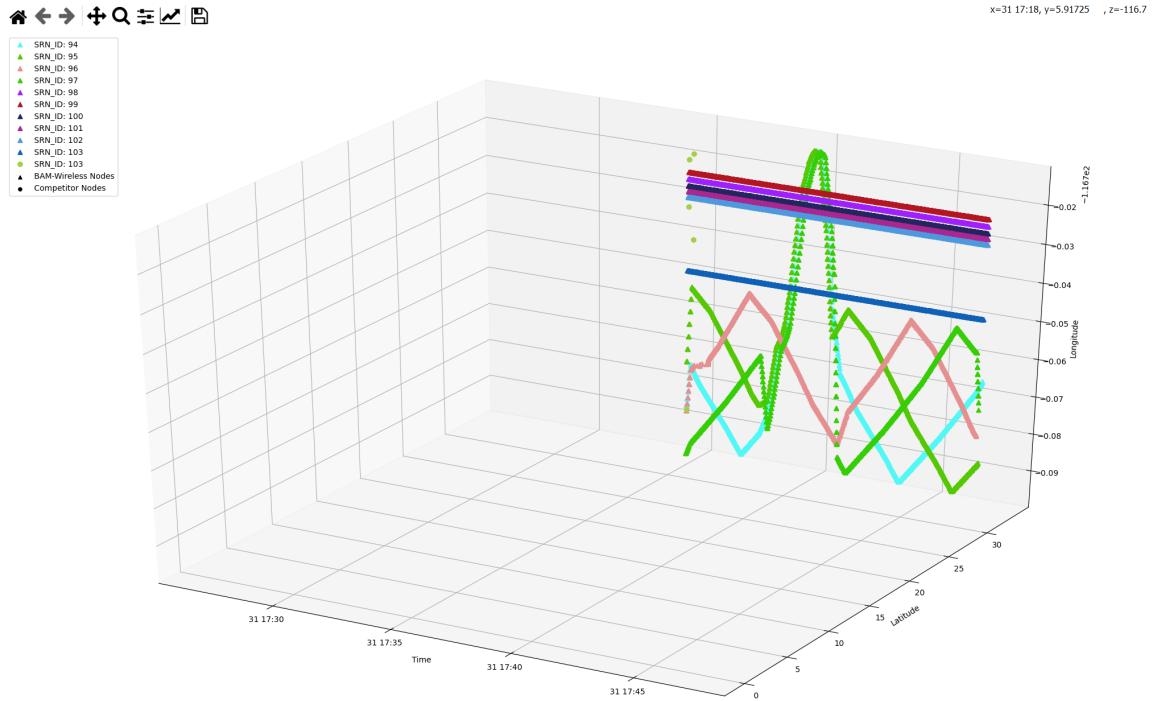


Fig. 2.19. The reported GPS locations of our SRNs, in addition to those of our competing SRNs, during the Wildfire scenario emulation

sity, University of Florida, Virginia Tech (with Lockheed Martin), and Rutgers University, to name a few) in a military deployment scenario, i.e., Alleys of Austin, and a disaster-relief deployment scenario, i.e., Wildfire. Having presented the design of a cognitive radio node (and its deployment in a network of peers) and analyzed the operational capabilities and performance of our radios in quasi-collaborative scenario emulations, we now narrow our focus down to a specific problem--the spectrum sensing and access problem in the MAC layer, and instead of proposing a complex heuristic like we did in this chapter, in the next chapter, we detail a highly rigorous approach to solve for the optimal spectrum sensing and access policy using an POMDP formulation.

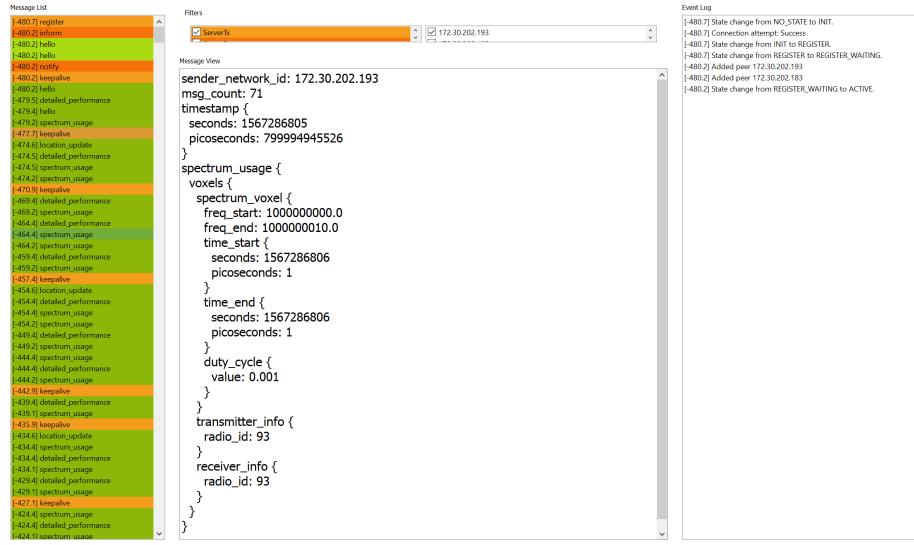


Fig. 2.20. The description of various CIL message exchanges between our network and the collaboration server, in addition to those between our network and our peers, during the Wildfire scenario emulation

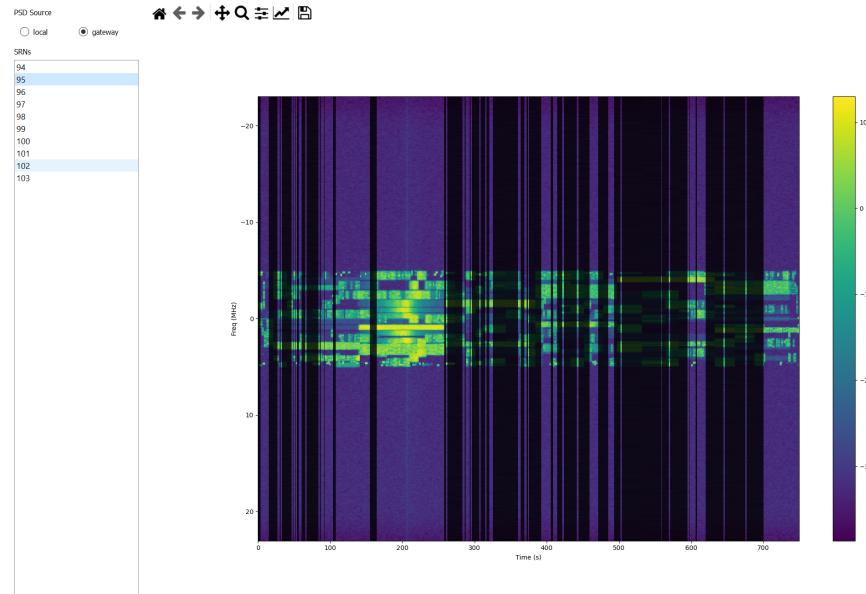


Fig. 2.21. The PSD measurements received at our Gateway SRN from one particular SRN in our network (selected), in order to obtain location-specific spectrum occupancy information, during the Wildfire scenario emulation

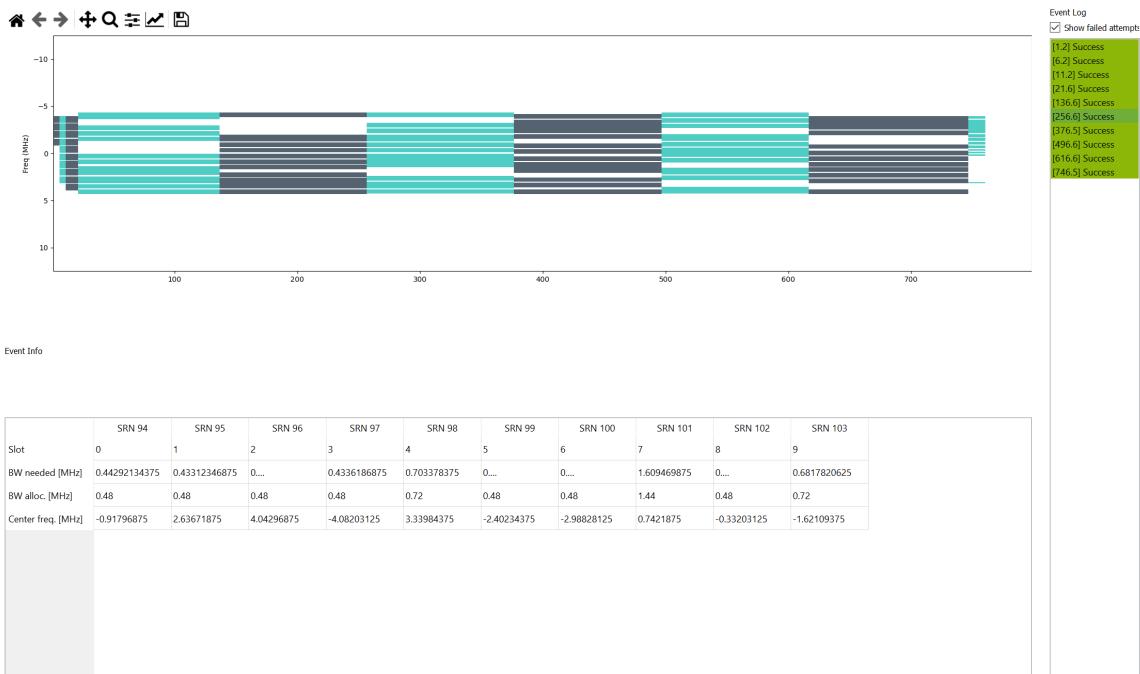


Fig. 2.22. The spectrum occupancy behavior of SRNs in our network (as guided by our Gateway SRN) during the Wildfire scenario emulation

### 3. SPECTRUM SENSING AND ACCESS VIA APPROXIMATE POMDP

In this research, as mentioned in Chapter 1, we exclusively tackle the spectrum sensing and access in cognitive radios by formulating the problem within an approximate POMDP framework with additional embeddings and heuristics, which will be detailed in the subsequent sections of this chapter--leveraging the time-frequency correlation structure in the occupancy behavior of the PUs in the network, estimating this correlation structure online using Hidden Markov Model (HMM) specific Maximum Likelihood Estimation (MLE) methods, and embedding this estimation into a randomized point-based POMDP value iteration method known as the PERSEUS algorithm to find the optimal sensing and access policy.

This chapter is organized as follows: Section 3.1 details the system model: the signal model, the incumbent occupancy time-frequency correlation structure, the channel sensing model, and the POMDP formulation; in Section 3.2, we describe the algorithms employed to solve for the optimal sensing and access policy: an estimator to determine the incumbent occupancy correlation parameters, and the PERSEUS algorithm; in Section 3.3, we present our numerical evaluations and performance comparisons with the state-of-the-art; and finally, in Section 3.4 outlines our concluding remarks.

At the time of writing this thesis, the research detailed in this chapter had been submitted as a journal article [20] for publication in the IEEE Wireless Communication Letters, and was under review.

### 3.1 System Model

#### 3.1.1 Signal Model

A cognitive radio, referred to hereafter in this chapter as the Secondary User (SU), constitutes a spectrum-sensing is tasked with the objective of maximizing its throughput (to satisfy the imposed QoS requirements) while limiting interference with the priority or licensed users in the network, hereafter referred to in this chapter as the Primary Users (PUs). As discussed in Chapter 3.1, these PUs (also known as incumbents) can either be military applications or commercial service providers who buy licenses from the FCC for access to the spectrum. We study a wireless radio environment in which the spectrum of interest has been discretized into  $K$  channels of equal bandwidth  $W$ , with  $J$  PUs and an SU trying to exploit portions of the spectrum left unused by these PUs, across time-slots and across frequencies. The discretized wide-band signal received at the SU's spectrum sensor in time-slot  $i$  can be described in the frequency domain as

$$Y_k(i) = \sum_{j=1}^J H_{j,k}(i) X_{j,k}(i) + V_k(i), \quad (3.1)$$

where  $X_{j,k}(i)$  represents the frequency domain signal of PU  $j \in \{1, 2, \dots, J\}$  in channel  $k \in \{1, 2, \dots, K\}$ , with  $X_{j,k}(i)=0$ , if PU  $j$  is not transmitting over channel  $k$  in time-slot  $i$ ;  $H_{j,k}(i)$  denotes the frequency domain channel between the SU and PU  $j$ ; and  $V_k(i) \sim \mathcal{CN}(0, \sigma_V^2)$  constitutes the zero-mean circularly symmetric additive complex Gaussian noise with variance  $\sigma_V^2$ , i.i.d across frequency and time, and independent of the channel  $H$  and the PU signal  $X$ . Assuming an Orthogonal Frequency Division Multiple Access (OFDMA) strategy among the PUs with respect to the channels in this discretized spectrum, and letting  $X_k(i) \triangleq X_{j_{k,i},k}(i)$  and  $H_k(i) \triangleq H_{j_{k,i},k}(i)$ , where subscript  $j_{k,i}$  denotes the index of the PU that occupies channel  $k$  in time-slot  $i$ , we can rewrite (3.1) as

$$Y_k(i) = H_k(i) X_k(i) + V_k(i), \quad (3.2)$$

where  $X_k(i)=0$ , if channel  $k$  is not occupied by any PU in time-slot  $i$ , and  $H_{j,k}(i) \sim \mathcal{CN}(0, \sigma_H^2)$  constitutes the zero-mean circularly symmetric complex Gaussian random variable with variance  $\sigma_H^2$ , modeling the Rayleigh fading channel, i.i.d across frequency and time.

### 3.1.2 Occupancy Correlation Structure



Fig. 3.1. The incumbent spectrum occupancy heat-map assuming independence across both frequency and time

The frequency domain signal of the PU occupying channel  $k$  in time-slot  $i$  is modeled as

$$X_k(i) = \sqrt{P_T} B_k(i) S_k(i), \quad (3.3)$$

where  $P_T$  denotes the transmission power of the occupant PU (we assume that all the PUs exhibit some kind of power control mechanisms in their PHY, and therefore have the same transmission power  $P_T$ ),  $B_k(i)$  represents the binary channel occupancy variable, i.e.,  $B_k(i)=1$ , if channel  $k$  is occupied by a PU in time-slot  $i$ ,  $B_k(i)=0$ , otherwise, and  $S_k(i)$  denotes the transmitted symbol.  $S_k(i)$  can be modeled as a constant amplitude signal, i.e.,  $|S_k(i)|=1$ , i.i.d across frequency and time; if however,  $S_k(i)$  is not a constant amplitude signal, we write  $H_k(i)X_k(i)$  as  $\sqrt{P_T}B_k(i)H_k(i)S_k(i)$ , where

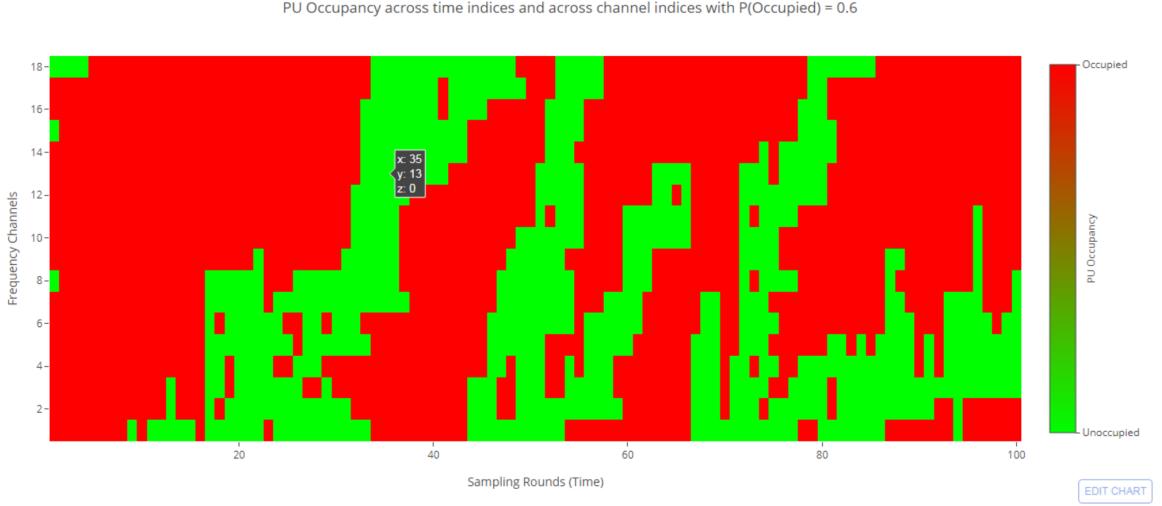


Fig. 3.2. The incumbent spectrum occupancy heat-map assuming a time-frequency correlation structure

$H_k(i)S_k(i)$  can be approximated as a zero-mean complex Gaussian random variable with variance  $\sigma_H^2 \mathbb{E}[|S_k|^2]$ , without any modifications to the subsequent analysis [20]. The spectrum occupancy state in time-slot  $i$  can be written as

$$\vec{B}(i) = [B_1(i), B_2(i), B_3(i), \dots, B_K(i)]^\top, \quad (3.4)$$

where  $\vec{B}(i) \in \{0, 1\}^K$ . There exists a time-frequency correlation structure in the occupancy behavior of the PUs in the network because a PU usually occupies adjacent channels for prolonged periods of time, i.e., the incumbents usually restrict their transmissions to certain parts of the spectrum, occupying a set of adjacent bands, and exhibit temporal patterns in their occupancy of these bands, with the temporal patterns governed by the operational periodicity of military users or by the prolonged usage by licensed service providers [24]. We decompose this time-frequency correlation structure as follows [20]: we model the temporal correlation in incumbent occupancy behavior as a Markov process described by

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(j), \forall j \leq i) = \mathbb{P}(\vec{B}(i+1)|\vec{B}(i)), \quad (3.5)$$

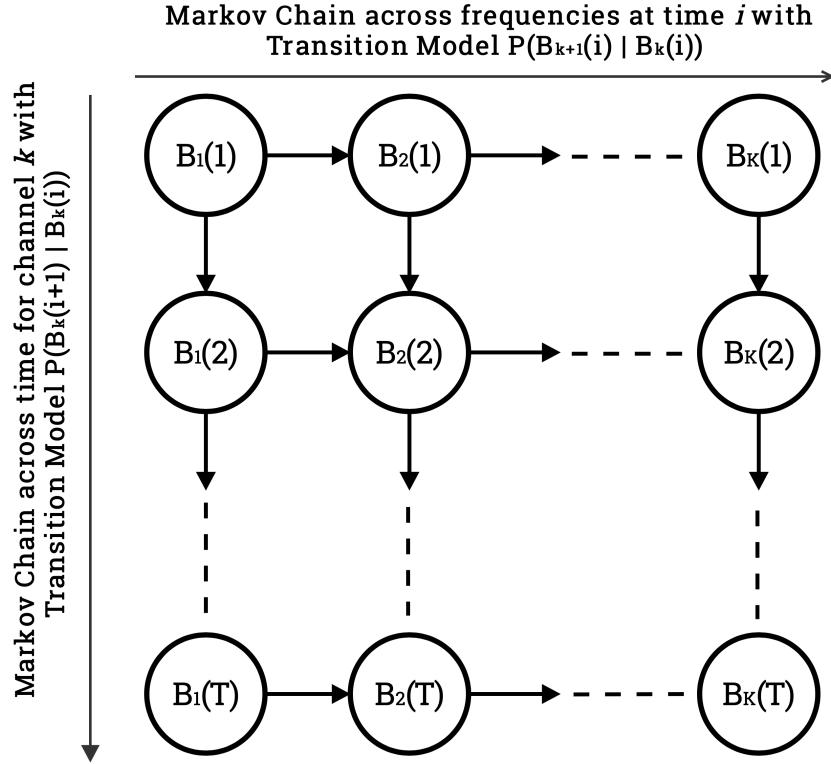


Fig. 3.3. The visualization of the incumbent occupancy time-frequency correlation structure as two dependent Markov chains: one across time and the other across frequencies

and we couple this model with the another Markov chain across the frequency bands to capture the frequency correlation in incumbent occupancy behavior, to get the final correlation structure as

$$\mathbb{P}(\vec{B}(i+1) | \vec{B}(i)) = \mathbb{P}(B_1(i+1) | B_1(i)) \prod_{k=2}^K \mathbb{P}(B_k(i+1) | B_{k-1}(i+1), B_k(i)). \quad (3.6)$$

In other words, we can describe this Markovian time-frequency correlation as follows: the occupancy of frequency band  $k$  in time-slot  $i+1$  depends on the occupancy of the adjacent frequency band  $k-1$  in the same time-slot  $i+1$ , and the occupancy of the

same frequency band  $k$  in the previous time-slot  $i$ . We parameterize this two-chain Markovian correlation structure by

$$\vec{\theta} = [\vec{p} \ \vec{q}]^\top, \text{ where}$$

$$\vec{p} = [p_{uv} = \mathbb{P}(B_k(i+1) = 1 | B_{k-1}(i+1) = u, B_k(i) = v) : u, v \in \{0, 1\}]^\top, \text{ and} \quad (3.7)$$

$$\vec{q} = [q_w = \mathbb{P}(B_1(i+1) = 1 | B_1(i) = w) : w \in \{0, 1\}]^\top.$$

This vector  $\vec{\theta}$ , parameterizes the transition model of our POMDP formulation described in Section 3.1.4, and is estimated by an HMM-specific Expectation Maximization (EM) algorithm, i.e., the Baum-Welch algorithm, which is the solution to the framed MLE problem, all of which is detailed in Section 3.2.1.

Fig. 3.1 illustrates the spectrum occupancy heat-map, assuming independence in occupancy behavior across both frequency and time, while Fig. 3.2 depicts the spectrum occupancy heat-map, assuming a time-frequency correlation structure in incumbent occupancy behavior, with  $\vec{p} = [p_{00}=0.1, p_{01}=0.3, p_{10}=0.3, p_{11}=0.7]^\top$  and  $\vec{q} = [q_0=0.3, q_1=0.8]^\top$ . The time-frequency correlation structure underlying the occupancy behavior of PUs in the network, as described by (3.6), can be illustrated as two dependent Markov chains: one across time and the other across frequencies, as shown in Fig. 3.3.

### 3.1.3 Channel Sensing Model

Equipped with a spectrum sensor, the SU detects white spaces and accesses them to deliver its network flows. Owing to physical design limitations--specifically, the restriction on the number of channels that can sensed by the SU's spectrum sensor in any given time-slot, primarily due to concerns about energy-efficiency and sensing/data aggregation times [22], the SU can sense a maximum of  $\kappa$  spectrum bands in a time-slot, with  $1 \leq \kappa \leq K$ . In a specific time-slot  $i$ , the SU senses all the channels in the set  $\mathcal{K}_i \subseteq \{1, 2, \dots, K\}$ , with the sensing limitation imposed as  $|\mathcal{K}_i| \leq \kappa$ . The solution to the spectrum sensing problem is to determine an optimal set of channels sensed by the SU in any time-slot  $i$ , and this optimal set is dictated by the optimal

policy derived from the POMDP formulation via the PERSEUS algorithm detailed in Section 3.2.2. The solution to the access problem hinges on the optimal sensing policy, i.e., based on the “best-possible” spectrum occupancy picture painted by the optimal sensing action in a time-slot  $i$ , the SU accesses all the channels it deems to be idle--more details on this access strategy are discussed in Section 3.1.4. After sensing the channels listed in  $\mathcal{K}_i$ , governed by the sensing policy, the obtained observation vector  $\vec{Y}(i) = [Y_k(i)]_{k \in \mathcal{K}_i}$ , where  $Y_k(i)$  is described in (3.2) [20].

In statistics, Hidden Markov Models (HMMs) are used to describe systems modeled by Markov processes, with the actual system states “hidden” behind the observed noisy measurements of these states. Along these lines, constructing an HMM for our problem, the linear, additive, Gaussian noise in the observation model described in Section 3.1.1, introduces uncertainty into the sensing process--the true occupancy states of the frequency bands in time-slot  $i$ , i.e.,  $\vec{B}(i)$ , represent the actual states of the model, while the observations at the SU’s spectrum sensor, i.e.,  $\vec{Y}(i)$ , represent the noisy observations of these true occupancy states. The observation vector in time-slot  $i$ ,  $\vec{Y}(i)$ , given the occupancy vector in that time-slot,  $\vec{B}(i)$ , has a Probability Density Function (PDF) described by

$$f(\vec{Y}(i)|\vec{B}(i), \mathcal{K}_i) = \prod_{k=1}^K f(Y_k(i)|B_k(i)), \quad (3.8)$$

due to the i.i.d assumptions of the noise,  $V_k(i)$ , the transmitted symbols,  $S_k(i)$ , and the Rayleigh fading variables  $H_k(i)$ , across channels, given the occupancy state vector, as discussed in Section 3.1.1. Additionally, we can infer from (3.2) that

$$Y_k(i)|B_k(i) \sim \mathcal{CN}(0, \sigma_H^2 P_T B_k(i) + \sigma_V^2). \quad (3.9)$$

### 3.1.4 POMDP Formulation

Partially Observable Markov Decision Processes (POMDPs) are employed in modeling the repeated, sequential interactions of an agent--tasked with maximizing its reward, subject to the problem at hand--with a stochastic environment, wherein the

limited observational capacity of the agent and/or the observation noise, creates partial observability vis-à-vis the underlying states of the environment. Modeling the spectrum sensing process at the SU as a POMDP, we find that, as expected, the partial observability caused by the sensing restrictions and a noisy observation model, result in an increased level of uncertainty regarding the exact effect of executing a spectrum access action on the radio environment [20]. Our POMDP formulation, represented by the 5-tuple  $(\mathcal{B}, \mathcal{A}, \mathcal{Y}, \mathbf{A}, \mathbf{M})$ , features the state space of the underlying MDP, denoted by  $\mathcal{B} \equiv \{0, 1\}^K$ , which is given by all possible realizations of the occupancy vector  $\vec{B}$ ; the action space of the SU, denoted by  $\mathcal{A}$ , which is described by all possible combinations in which  $1 \leq \kappa \leq K$  channels are chosen to be sensed in a time-slot (discussed in Section 3.1.3); the observation space, denoted by  $\mathcal{Y}$ , which is discussed in Section 3.1.1; the transition model of the underlying MDP, denoted by  $\mathbf{A}$ , which is discussed in Section 3.1.2; and the observation model (also known as the emission model), denoted by  $\mathbf{Y}$ , which is described by (3.8) and (3.9).

Prior to gathering the occupancy information in time-slot  $i$ , based on the measurements obtained by the SU's spectrum sensor up to, but not including, time-slot  $i$ , the POMDP state is described by the prior belief, denoted by  $\beta_i$ , which describes the probability distribution of the underlying MDP state, i.e.,  $\vec{B}(i)$ . Given this prior belief  $\beta_i$ , based on the SU's sensing policy, the SU chooses a sensing action, i.e.,  $\pi(\beta_i) = \mathcal{K}_i \in \mathcal{A}$ , wherein as detailed in Section 3.1.3, the SU senses the frequency bands corresponding to the channel indices in the set  $\mathcal{K}(i)$ , and this observation vector  $[Y_k(i)]_{k \in \mathcal{K}_i} \in \mathcal{Y}$ , and updates its belief of the underlying MDP state  $\vec{B}(i)$  to obtain its posterior belief, which is written as

$$\begin{aligned}\hat{\beta}_i(\vec{B}') &= \mathbb{P}(\vec{B}(i) = \vec{B}' | \beta_i, \mathcal{K}_i, [Y_k(i)]_{k \in \mathcal{K}_i}) \\ &= \frac{\mathbb{P}([Y_k(i)]_{k \in \mathcal{K}_i} | \vec{B}', \mathcal{K}_i) \beta_i(\vec{B}')}{\sum_{\vec{B}'' \in \{0,1\}^K} \mathbb{P}([Y_k(i)]_{k \in \mathcal{K}_i} | \vec{B}'', \mathcal{K}_i) \beta_i(\vec{B}'')}. \end{aligned}\quad (3.10)$$

After channel sensing is performed by the SU's spectrum sensor, according to the sensing policy, using the posterior belief described in (3.10), channel access decisions have to be made: the underlying MDP state  $\vec{B}(i)$  is estimated as

$$\vec{\phi}(\hat{\beta}_i) = \arg \max_{\vec{B} \in \mathcal{B}} \hat{\beta}_i(\vec{B}), \quad (3.11)$$

following which, if  $\phi_k(i)=1$ , which implies that the SU estimated channel  $k$  to be occupied by a PU in this time-slot  $i$ , and hence leaves it untouched, while if  $\phi_k(i)=0$ , the SU accesses this “estimated idle” channel  $k$  in time-slot  $i$  to deliver its network flows. Upon executing an access decision based on the Maximum-A-Posteriori (MAP) estimation procedure laid down in (3.11), i.e., in time-slot  $i$ , after accessing all the channels with  $\phi_k(i)=0$ , the SU receives a reward from the radio environment, denoted by  $R(\vec{B}(i), \hat{\beta}_i)$ , dictated by the number of truly idle frequency bands accessed by the SU, which accounts for the SU throughput maximization aspect of its objective, and a penalty ( $\lambda > 0$ ) for the number of incumbent occupied channels that were incorrectly accessed by the SU, which accounts for the aspect of its objective that involves minimizing the interference caused to PUs in the network [20]. Mathematically, this reward metric is described as

$$R(\vec{B}(i), \hat{\beta}_i) = \sum_{k=1}^K (1 - B_k(i))(1 - \phi_k(i)) - \lambda B_k(i)(1 - \phi_k(i)). \quad (3.12)$$

Ensuing the determination of the reward for its access decision from the radio environment, the SU computes the prior belief for the next time-slot  $i + 1$  as

$$\beta_{i+1}(\vec{B}'') = \sum_{\vec{B}'} \mathbb{P}(\vec{B}(i+1) = \vec{B}'' | \vec{B}(i) = \vec{B}') \hat{\beta}(\vec{B}'). \quad (3.13)$$

Let

$$\hat{\beta}_i = \hat{\mathbb{B}}(\beta_i, \mathcal{K}_i, \vec{Y}(i)) \quad (3.14)$$

denote the function that maps the prior belief  $\beta_i$  to the posterior belief  $\hat{\beta}_i$  in time-slot  $i$ , and let

$$\beta_{i+1} = \mathbb{B}(\hat{\beta}_i) \quad (3.15)$$

denote the function that maps the posterior belief  $\hat{\beta}_i$  in time-slot  $i$  to the prior belief  $\beta_{i+1}$  in time-slot  $i + 1$  [20]. The objective of the SU is to determine the optimal spectrum sensing policy (based on which the access decisions are made in the corresponding time-slots) to maximize its infinite-horizon discounted reward, i.e.,

$$\pi^* = \arg \max_{\pi} V^\pi(\beta), \quad (3.16)$$

where

$$V^\pi(\beta) = \mathbb{E}_\pi \left[ \sum_{i=1}^{\infty} \gamma^i R(\vec{B}(i), \hat{\beta}_i) \middle| \beta_0 = \beta \right], \quad (3.17)$$

where  $0 < \gamma < 1$  is the discount factor,  $\beta_0 = \beta$  is the initial belief such that the value function  $V^\pi(\beta)$  is evaluated from this starting belief, and  $\hat{\beta}_i$  is the posterior belief induced by the policy  $\mathcal{K}_i = \pi(\beta_i)$  and the observation vector  $[Y_k(i)]_{k \in \mathcal{K}_i}$  via the formulation  $\hat{\beta}_i = \hat{\mathbb{B}}(\beta_i, \mathcal{K}_i = \pi(\beta_i), [Y_k(i)]_{k \in \mathcal{K}_i})$  [20]. The Bellman operator, denoted by  $\mathcal{H}$ , employed in the Bellman optimality equation,  $V^* = \mathcal{H}(V^*)$ , is defined at iteration  $t + 1$  as,  $\forall \beta$

$$\begin{aligned} V_{t+1} &= \mathcal{H}(V_t) \\ &= \max_{\mathcal{K} \in \mathcal{A}} \sum_{\vec{B} \in \mathcal{B}} \beta(\vec{B}) \mathbb{E}_{[Y_k]_{k \in \mathcal{K}} | \vec{B}, \mathcal{K}} \left[ R(\vec{B}, \hat{\mathbb{B}}(\beta, \mathcal{K}, [Y_k]_{k \in \mathcal{K}})) + \gamma V_t(\hat{\mathbb{B}}(\beta, \mathcal{K}, [Y_k]_{k \in \mathcal{K}})) \right]. \end{aligned} \quad (3.18)$$

By employing value iteration algorithms, (3.18) can be solved iteratively until convergence to a fixed point that corresponds to the optimal sensing policy. However, this direct approach results in complications associated with the lack of prior knowledge about the incumbent occupancy time-frequency correlation structure that defines the transition model of the underlying MDP, and the computational infeasibility of the approach--as the number of channels in the discretized spectrum of interest increases, the number of states of the underlying MDP scales exponentially, resulting in a high-dimensional belief space, which makes the approach intractable [20].

We solve the problem of intractability of the POMDP for large state and action spaces by employing a randomized, point-based, approximate value iteration algorithm known as PERSEUS [28] to solve for the optimal sensing policy, while an

online parameter estimator embedded into PERSEUS allows us to solve the problem of transition model ignorance. Both these algorithms are detailed in Section 3.2.

## 3.2 Proposed Solution: The Algorithms

### 3.2.1 Occupancy Correlation Structure Estimation

Practical implementations of the MAC layer of the cognitive radio’s network protocol stack involve solving for the optimal sensing and access policy, without having any prior information about the time-frequency correlation structure underlying the occupancy behavior of the incumbents in the network [20]. This correlation structure, as discussed earlier, can be leveraged to improve the occupancy state estimation accuracy, which in turn facilitates higher SU network throughput with lower PU interference. In this regard, in this section, we propose a parameter estimator algorithm that learns this correlation structure over time--with the learned correlation structure iteratively fed into the POMDP optimal policy solver (i.e., PERSEUS), in order to enable a concurrent framework that minimizes the amount of computational resources (time, memory, processing power) required to obtain the optimal policy, which is especially crucial in non-stationary settings.

Let  $\tau$  refer to the learning period of the parameter estimation algorithm: this can be equal to the entire duration of the SU’s interaction with the radio environment while solving for the optimal policy, implying concurrent model learning facilitated by a publisher-subscriber software architecture and multi-threading features--in a time-slot, the diverse, sparse observations made by the SU as a part of the PERSEUS thread’s exploration period are concatenated into a complete observation vector over repeated iterations (we assume the dynamics of the PU occupancy over the time-slots are slower than the time needed for these observations and their subsequent concatenation) and injected into the EM thread, which estimates the transition probabilities in that iteration (which is synchronized with the PERSEUS thread’s time-slot dynamics in order to have these two threads operate on the same time-scale) and publishes

them, with the PERSEUS thread using the most current published estimates in its operation; or it can be equal to an initial learning period that has been set aside exclusively for the SU to estimate the underlying MDP's transition model--after which the PERSEUS algorithm is initiated, employing these final estimated (converged) transition probabilities. Defining  $\mathbf{B}=[\vec{B}(i)]_{i=1}^{\tau}$  as the sequence of states encountered by the SU in time-slots  $i=1$  to  $i=\tau$ , and  $\mathbf{Y}=[\vec{Y}(i)]_{i=1}^{\tau}$  as the sequence of observations made at the SU's spectrum sensor  $i=1$  to  $i=\tau$ , having a one-to-one correspondence with the elements of  $[\vec{B}]_{i=1}^{\tau}$ , we formulate the Maximum Likelihood Estimation (MLE) problem to estimate the vector  $\vec{\theta}$  that parameterizes the PU occupancy time-frequency correlation structure (detailed in Section 3.1.2) as follows:

$$\vec{\theta}^* = \arg \max_{\vec{\theta}} \log \left( \sum_{\mathbf{B}} \mathbb{P}(\mathbf{B}, \mathbf{Y} | \vec{\theta}) \right). \quad (3.19)$$

Solving this MLE formulation using the Expectation-Maximization algorithm [29] for HMMs, i.e., the Baum-Welch algorithm, the algorithm boils down to two-steps--the E-step constitutes

$$Q(\vec{\theta} | \vec{\theta}^{(t)}) = \mathbb{E}_{\mathbf{B} | \mathbf{Y}, \vec{\theta}^{(t)}} \left[ \log (\mathbb{P}(\mathbf{B}, \mathbf{Y} | \vec{\theta}^{(t)})) \right], \quad (3.20)$$

where  $Q(\vec{\theta} | \vec{\theta}^{(t)})$  is computed using the Forward-Backward algorithm [29]; and the M-step constitutes

$$\vec{\theta}^{(t+1)} = \arg \max_{\vec{\theta}} Q(\vec{\theta} | \vec{\theta}^{(t)}), \quad (3.21)$$

which involves the re-estimation of  $\vec{\theta}$  by employing the statistics  $Q(\vec{\theta} | \vec{\theta}^{(t)})$  obtained from the Forward-Backward algorithm [20].

### 3.2.2 The PERSEUS Algorithm

In our proposed solution, we solve for the optimal spectrum sensing (and access, based on the MAP estimation detailed in Section 3.1.4) policy, in parallel with the parameter estimation algorithm--employing its published iterative transition model estimates--until both the EM algorithm and the POMDP policy solver algorithms converge [20].

As alluded to in Section 3.1.4, in order to solve the computational infeasibility precipitated by the exponential increase in the number of states of the underlying MDP, induced by an increase in the number of frequency bands in the discretized spectrum of interest, we employ approximate POMDP value iteration methods to ensure that the formulations and the algorithms scale well to a large number of relevant channels in the radio environment in which the SU operates. Consequently, we choose the PERSEUS algorithm [28] to solve for the optimal policy, primarily motivated by the following: the exact value iteration strategies proposed in [30], namely the Exhaustive Enumeration algorithm and the Witness algorithm are untenable for large belief spaces, because these techniques involve performing the backup procedure, i.e., determining the optimal action (or hyperplane in a Piece-Wise Linear Convex (PWLC) context) for every belief point in the belief space; and a fellow contemporary approximate value iteration algorithm known as the Point-Based Value Iteration (PBVI) algorithm proposed in [31], although involves performing the backup operation over a reduced set of beliefs known as the “reachable beliefs”, unlike the strategies in [31], is computationally expensive due to the task of computing the distances between all the belief points in the set of reachable beliefs in addition to the subsequent backup operation on all these belief points. The PERSEUS algorithm, on the other hand, does not involve performing the backup operation for every point in the belief space, unlike the Exhaustive Enumeration and Witness algorithms detailed in [30]; and unlike the PBVI algorithm [31] does not involve computing the distances between all the belief points in the set of reachable beliefs, and furthermore, does not involve performing backups on all the reachable belief points--instead, PERSEUS involves “backing-up” only on a subset of this set of reachable beliefs, while ensuring that the computed solution is effective for all the points in the reachable belief set.

PERSEUS--a randomized, point-based, approximate POMDP value iteration algorithm--involves an initial phase of exploration, wherein the set of “reachable-beliefs”, denoted by  $\tilde{\mathcal{B}}$ , is determined by allowing the SU to randomly interact with the radio environment. As referenced earlier, one simplifying (or approximating) fea-

ture of PERSEUS is to improve the value of all the belief points in the set  $\tilde{\mathcal{B}}$ , by computing the value of only a subset of these belief points--which are chosen iteratively at random [20]. For finite-horizon POMDP formulations, the optimal value function  $V^*$  described by (3.18), can be approximated by a Piece-Wise Linear Convex (PWLC) function [28]--in other words, the value function at iteration  $t$  is parameterized by a set of hyperplanes, denoted by  $\{\vec{\alpha}_t^u\}_{u \in \{1, 2, \dots, |\tilde{\mathcal{B}}|\}}$ , wherein each hyperplane represents a region of the belief space for which the action corresponding to this hyperplane, denoted by  $\mathcal{K}_t^u$ , is the maximizer. Ergo, the value function of belief  $\beta$  in a given iteration  $t$  is approximated as

$$V_t(\beta) \approx \beta \cdot \vec{\alpha}_t^{u^*}, \quad (3.22)$$

where,

$$u^* = \arg \max_{u \in \{1, 2, \dots, |\tilde{\mathcal{B}}|\}} \beta \cdot \vec{\alpha}_t^u, \quad (3.23)$$

with

$$\beta \cdot \vec{\alpha} = \sum_{\vec{B}} \beta(\vec{B}) \vec{\alpha}(\vec{B}) \quad (3.24)$$

describing the inner product, and  $\mathcal{K}_t^{u^*}$  representing the optimal spectrum sensing action.

We define a set of unimproved belief points, denoted as  $\tilde{\mathcal{U}}$ , which initially corresponds to the set of reachable beliefs  $\tilde{\mathcal{B}}$  obtained by the random exploration procedure detailed earlier. Pick a belief  $\beta_u$  from this set  $\tilde{\mathcal{U}}$ , and perform the backup operation on this chosen belief point--which, as discussed earlier, involves associating a new hyperplane and its corresponding spectrum sensing action with this belief  $\beta_u$ . In iteration  $t + 1$ , defining  $\mathcal{K}_{t+1}^u$  as the action associated with hyperplane  $\vec{\alpha}_{t+1}^u$ , corresponding to belief  $\beta_u \in \tilde{\mathcal{U}}$ , we can describe the backup procedure mathematically as

$$\begin{aligned} \vec{\alpha}_{t+1} &= \xi_{\mathcal{K}_{t+1}^u}^u, \\ \mathcal{K}_{t+1}^u &= \arg \max_{\mathcal{K} \in \mathcal{A}} \beta_u \cdot \xi_{\mathcal{K}}^u, \end{aligned} \quad (3.25)$$

where  $\xi_{\mathcal{K}}^u$  is the hyperplane corresponding to the one-step look-ahead under action  $\mathcal{K} \in \mathcal{A}$  and belief  $\beta_u$ , i.e.,

$$\xi_{\mathcal{K}}^u(\vec{B}) = \mathbb{E}_{\vec{Y}|\vec{B},\mathcal{K}} \left[ R(\vec{B}, \hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y})) + \gamma \sum_{\vec{B}'} \mathbb{P}(\vec{B}(i+1) = \vec{B}' | \vec{B}(i) = \vec{B}) \xi_{\mathcal{K},\vec{Y}}^u(\vec{B}') \right], \quad (3.26)$$

where  $\xi_{\mathcal{K},\vec{Y}}^u$  refers to the hyperplane corresponding to the future value function computed from the previous set of hyperplanes from the new belief  $\mathbb{B}(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y}))$  obtained from  $\beta_u$  by executing action  $\mathcal{K}$  and observing  $\vec{Y}$ , as

$$\xi_{\mathcal{K},\vec{Y}}^u = \arg \max_{\vec{\alpha}_t^{u'}, u' \in \{1, 2, \dots, |\tilde{\mathcal{B}}|\}} \mathbb{B}(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y})) \cdot \vec{\alpha}_t^{u'}. \quad (3.27)$$

After determining the hyperplane  $\vec{\alpha}_{t+1}^u$  associated with this chosen belief point  $\beta_u$  using the backup procedure detailed above, we now know that  $V_{t+1}(\beta_u) = \beta_u \cdot \vec{\alpha}_{t+1}^u$  is its approximate value function. The most crucial aspect of PERSEUS that approximates the optimization of a randomly chosen belief point to the entire set  $\tilde{\mathcal{U}}$  is as follows: if the approximate value function for belief point  $\beta_u \in \tilde{\mathcal{U}}$  is improved by the aforementioned backup iteration, i.e.,  $V_{t+1}(\beta_u) \geq V_t(\beta_u)$ , the belief point  $\beta_u$  is removed from the set  $\tilde{\mathcal{U}}$ --and now, we check if this hyperplane  $\vec{\alpha}_{t+1}^u$  improves the approximate value functions of the other beliefs in  $\tilde{\mathcal{U}}$ , i.e.,  $\forall \beta' \in \tilde{\mathcal{U}} - \{\beta_u\}$ , if  $\beta' \cdot \vec{\alpha}_{t+1}^u \geq V_t(\beta')$ , this new hyperplane generates an improved approximate value function, and these respective belief points for which this hyperplane improves their approximate value functions, are removed from the set  $\tilde{\mathcal{U}}$ . In other words,

$$\begin{aligned} \tilde{\mathcal{U}} &\leftarrow \tilde{\mathcal{U}} - \{\beta_u\}, \text{ if } \beta_u \cdot \vec{\alpha}_{t+1}^u \geq V_t(\beta_u), \text{ and subsequently} \\ \tilde{\mathcal{U}} &\leftarrow \tilde{\mathcal{U}} - \{\beta' \in \tilde{\mathcal{U}} : \beta' \cdot \vec{\alpha}_{t+1}^u \geq V_t(\beta')\}. \end{aligned} \quad (3.28)$$

On the other hand, if this hyperplane  $\vec{\alpha}_{t+1}^u$  worsens the approximate value function of  $\beta_u$ , i.e.,  $\beta_u \cdot \vec{\alpha}_{t+1}^u < V_t(\beta_u)$ , the old hyperplane and its associated sensing action persist for  $\beta_u$ --mathematically,  $\vec{\alpha}_{t+1}^u := \vec{\alpha}_t^u$  and  $\mathcal{K}_{t+1}^u := \mathcal{K}_t^u$ ; but we still check for improvements with respect to the other belief points in  $\tilde{\mathcal{U}}$ , and remove all those belief points  $\beta' \in \tilde{\mathcal{U}}$  for which  $\beta' \cdot \vec{\alpha}_{t+1}^u \geq V_t(\beta')$ .

In general, if a hyperplane determined from the backup procedure improves a belief point in the set of unimproved belief points  $\tilde{\mathcal{U}}$ , this news hyperplane (and its associated sensing action) becomes the relevant hyperplane (and the relevant sensing action) for this belief point, and the belief point will be removed from the set of unimproved belief points  $\tilde{\mathcal{U}}$ . These sequence of operations (random choice from  $\tilde{\mathcal{U}}$  → backup → check for improvement and removal) are performed iteratively until the set  $\tilde{\mathcal{U}}$  is empty--this constitutes a single PERSEUS iteration [20]. These PERSEUS iterations are executed until the specified value iteration termination condition is satisfied, i.e.,

$$|V_{t+1}(\beta) - V_t(\beta)| < \epsilon, \quad \forall \beta \in \tilde{\mathcal{B}}, \quad (3.29)$$

where  $\epsilon > 0$  (a very small value), is the value iteration difference threshold.

The PERSEUS algorithm--although is an approximate POMDP method which eliminates the computational overhead associated with the exhaustive belief space and reachable space optimization techniques [30, 31] by approximating the optimization of a randomly chosen belief point to the entire set of unimproved, reachable belief points--still possesses computational intractability challenges because it involves iterations over all possible combinations of the occupancy state vector, i.e.,  $\vec{B} \in \{0, 1\}^K$ --the computational cost scales exponentially with the number of states in the underlying MDP, which is induced by the number of channels  $K$  in the discretized spectrum of interest. In order to solve this computational tractability problem, we introduce two simplifying heuristics into the PERSEUS algorithm. Firstly, we avoid iterating over all possible occupancy states by considering only those state transitions that involve a Hamming distance of  $d \in \{1, 2, \dots, K\}$  between two consecutive state vectors,  $\vec{B}(i)$  and  $\vec{B}(i + 1)$ --this is practical because the temporal dynamics of the occupancy of the radio environment, dictated by the behavior of the PUs in the network, are typically slower than the processing dynamics of the POMDP agent, i.e., the SU. Secondly, we fragment the discretized spectrum into smaller, independent sets of correlated channels (for example, an 18 channel radio environment with 3 PUs is fragmented into 3 independent fragments, each comprising 6 channels correlated by the occupancy behavior of the corresponding PU) [20]; run PERSEUS on these

fragments concurrently by employing multi-threading capabilities in software frameworks; and finally, combine the results from each of these fragmented, parallel runs to get a full picture about the performance of our POMDP agent--this is practical because in a radio environment with multiple PUs, each PU is typically restricted to a portion (a set of adjacent frequency bands) of the spectrum--either by design or by bureaucracy.

### 3.3 Numerical Evaluations

Our simulations evaluate the operational capabilities of the proposed framework and compare it against the state-of-the-art. The simulated radio environment constitutes  $J=3$  incumbents, i.e., PUs, accessing a 2.88MHz spectrum, discretized into  $K=18$  channels--each having a bandwidth of  $W=160\text{kHz}$ . The 3 PUs access these 18 channels according to a time-frequency Markovian correlation structure parameterized by

$$\vec{\theta} = \begin{bmatrix} \vec{p} \\ \vec{q} \end{bmatrix}$$

as described in Section 3.1.2, where

$$\vec{p} = \begin{bmatrix} p_{00} = 0.1 \\ p_{01} = 0.3 \\ p_{10} = 0.3 \\ p_{11} = 0.7 \end{bmatrix},$$

and

$$\vec{q} = \begin{bmatrix} q_0 = 0.3 \\ q_1 = 0.8 \end{bmatrix}.$$

Regarding the channel sensing limitations induced by a need to minimize the amount of time and energy spent sensing the spectrum [22], we model our simulation framework on  $\kappa=6$ , i.e., in any given time-slot  $i$ , the SU can sense a maximum of 6 channels out of the 18 in the discretized spectrum of interest.

Regarding the expected Signal to Interference Noise Ratios (SINR) at the PUs and the SU, subject to fading, and conditioned on the PU and SU access decisions, we model our simulation framework based off the following numbers:

$$\text{SINR}_{\text{SU}}(k, i) = 0, \text{ if the SU does not access channel } k \text{ in time-slot } i,$$

$$\text{SINR}_{\text{SU}}(k, i) = 11 \text{dB, if the SU accesses a truly idle channel } k \text{ in time-slot } i,$$

$$\text{SINR}_{\text{SU}}(k, i) = -6 \text{dB, if SU accesses an incumbent-occupied channel } k \text{ in slot } i,$$

$$\text{SINR}_{\text{PU}_j}(k, i) = 0, \text{ if the PU } j \text{ does not access channel } k \text{ in time-slot } i,$$

$$\text{SINR}_{\text{PU}_j}(k, i) = 17 \text{dB, if PU } j \text{ occupies channel } k \text{ in slot } i \text{ without SU interference,}$$

$$\text{SINR}_{\text{PU}_j}(k, i) = 6 \text{dB, if PU } j \text{ occupies channel } k \text{ in slot } i \text{ with SU interference.}$$

As described in Section 3.1, the only objective of the SU is to maximize its throughput subject to a constraint on the amount of interference its transmission can cause to incumbents in the network. To this end, assuming an always back-logged SU, i.e., the SU always has network flows to deliver, the optimal POMDP sensing policy dictates which channels should be sensed by the SU's spectrum sensor in a given time-slot--based off of the learned correlated occupancy dynamics of the PUs--in order to obtain an optimal picture about the occupancy of the channels in this time-slot, and then access all the channels deemed to be idle by the MAP estimation procedure detailed in Section 3.1.4. The average throughput attained by the SU over  $T$  time-slots is given by

$$C^{\text{SU}} = \frac{1}{T} \sum_{i=1}^T \sum_{k=1}^K R_{\text{SU}} \mathcal{I} \left\{ \text{SINR}_{\text{SU}}(k, i) \geq 2^{\frac{R_{\text{SU}}}{W}} - 1 \right\}, \quad (3.30)$$

where  $R_{\text{SU}}=0.6 \text{Mbps}$  is the transmission rate of the SU on each channel, and  $\mathcal{I}$  is an indicator variable; and the throughput attained by the PUs in the network over the same  $T$  time-slots, normalized over time and the number of transmissions (normalization is necessary here because of the temporally intermittent transmissions of the PUs--the PUs are not always back-logged, unlike the SU) is given by

$$C^{\text{PUs}} = \frac{\sum_{i=1}^T \sum_{k=1}^K R_{\text{PU}} B_k(i) \mathcal{I} \left\{ \text{SINR}_{\text{PU}}(k, i) \geq 2^{\frac{R_{\text{PU}}}{W}} - 1 \right\}}{\sum_{i=1}^T \sum_{k=1}^K B_k(i)}, \quad (3.31)$$

where  $R_{\text{PU}}=0.9\text{Mbps}$  is the transmission rate of the PUs on each channel,  $\mathcal{J}$  is an indicator variable,  $\text{SINR}_{\text{PU}}(k, i)=\text{SINR}_{\text{PU}_j}(k, i)$ -- $j \in \{1, 2, \dots, J\}$  being the index of the PU occupying channel  $k$  in time-slot  $i$ , and  $B_k(i)=1$  if channel  $k$  is occupied by an incumbent in time-slot  $i$  (note here that PUs do not interfere with each other because of clearly laid out administrative guidelines about licensed frequency use for incumbents, so only one PU accesses a frequency band and that band would have been specifically licensed for that PU).

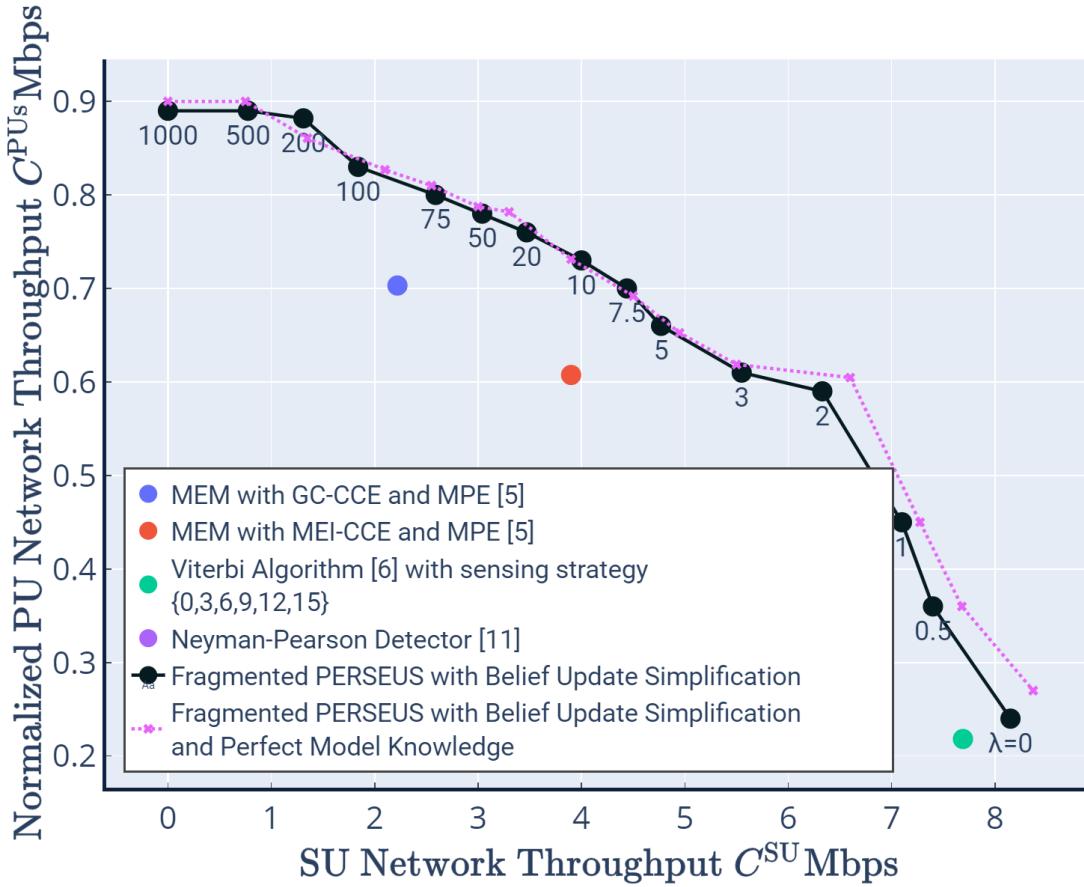


Fig. 3.4. The evaluation of SU and PU network throughputs for different values of  $\lambda$ , along with comparisons with the state-of-the-art

Incorporating a concurrent parameter estimator embedded into the fragmented PERSEUS algorithm (with belief update simplification--the Hamming distance state filter), with an iterative publisher-subscriber routine described in Section 3.2.1, we

find that our framework outperforms the state-of-the-art algorithms that also tackle the spectrum sensing and access problem in a correlated incumbent occupancy behavior setting. Specifically, evaluating the performance of our framework against the Minimum Entropy Merging (MEM) algorithm with Greedy Clustering based Channel Correlation Estimation (GC-CCE) and Markov Process Estimation (MPE) [5], we find that with a correlation threshold of  $\rho_{th}=0.77$  in the MEM with GC-CCE and MPE solution, our framework achieves a 104% improvement in the throughput attained by the SU, for the same level of interference to the incumbents in the network. Similarly, we find that our solution achieves a 38% improvement in the SU throughput, for the same level of PU interference, over the Minimum Entropy Merging (MEM) algorithm with Minimum Entropy Increment (MEI) Clustering based Channel Correlation Estimation (CCE) and Markov Process Estimation (MPE) with a correlation threshold of  $\rho_{th}=0.77$  [5]. Additionally, our solution attains a 25% increase in SU network throughput, for the same level of interference caused to the PUs in the network, over a Neyman-Pearson Detector that assumes independence among the channels across both frequency and time, has no channel sensing restrictions, involves an AND fusion rule across 300 different samplings, and whose threshold is determined based off of a specified false alarm probability of 0.3 [11, 20]. Finally, comparing the performance of our POMDP framework against well-known HMM state estimators--specifically, the Viterbi algorithm that solves the MAP state estimation problem for the system described in this simulation setup consisting of a two chain Markovian correlation structure (one across time and the other across frequency) [6], with the same channel sensing restriction as ours, i.e., 6, but that which knows the exact underlying MDP transition model **A** as an apriori, we note that our solution offers a 6% increase in the attained SU network throughput, for the same amount of incumbent interference. Sticking to the fact that our framework does not know the underlying MDP transition model--which is governed by the correlated PU occupancy behavior--ahead of time, but instead learns this correlation structure as it is interacting with the radio environment and solving for the optimal policy, we

evaluated the accuracy of our optimal policy's behavior against a similar PERSEUS agent which knew the transition model beforehand: we find that in the worst-case with respect to the difference in performance between the two, i.e., when  $\lambda=0$ , for the same level of incumbent interference, knowing the correlation model ahead of time only provided a 3.75% increase in the attained SU network throughput--which is a testament to the accuracy of our embedded parameter estimator and the iterative publish-subscribe to-and-fro between the EM thread and the PERSEUS thread. These evaluations are illustrated in Fig. 3.4.

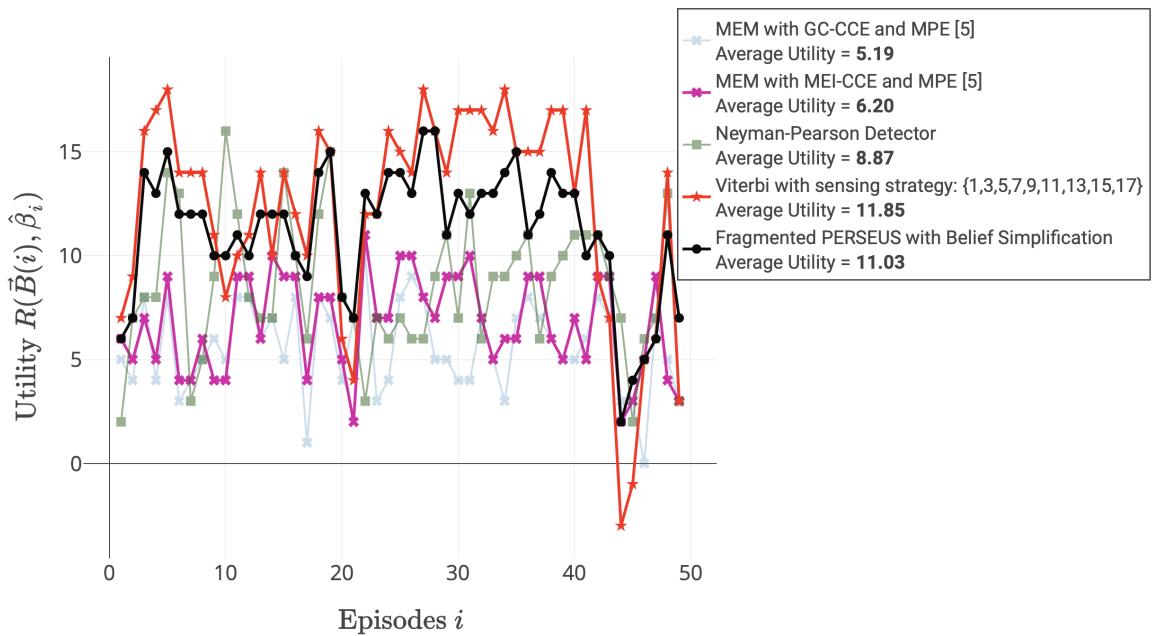


Fig. 3.5. The evaluation of the proposed solution, from an average utility per time-slot perspective, against a medley of approaches in the state-of-the-art

Analyzing the performance of our POMDP solution from a different perspective, we find that, as illustrated in Fig. 3.5, our framework obtains an average utility, i.e.,  $R(\vec{B}(i), \hat{\beta}_i)$  described in Section 3.1.4, of 11.03 per time-step  $i$ -- 112% higher than that achieved by the MEM with GC-CCE and MPE algorithm from [5], 80% higher than that achieved by the MEM with MEI-CCE and MPE algorithm from [5], and 25% more than that attained by the Neyman-Pearson Detector detailed above

[11]. Furthermore, in order to understand how our framework performs against a standard HMM state estimation solution like the Viterbi algorithm described earlier [6]--especially, one with apriori transition model information and one that senses a maximum of 9 channels as opposed to our 6 (in order to contrast this with the previous comparison with the Viterbi algorithm in which both have the same channel sensing restrictions)--we compare our solution with this Viterbi agent, and find that in spite of this Viterbi agent having more occupancy information per time-step (owing to it sensing a maximum of 3 additional channels), the average utility obtained by the Viterbi agent ( $=11.85$ ) is only 7% higher than ours ( $=11.03$ ).

Detection Accuracy v/s  $P(\text{Occupied} \mid \text{Idle})$  for 18 channels at  $P(X_i = 1) = 0.6$  with varying uniform channel sensing strategies

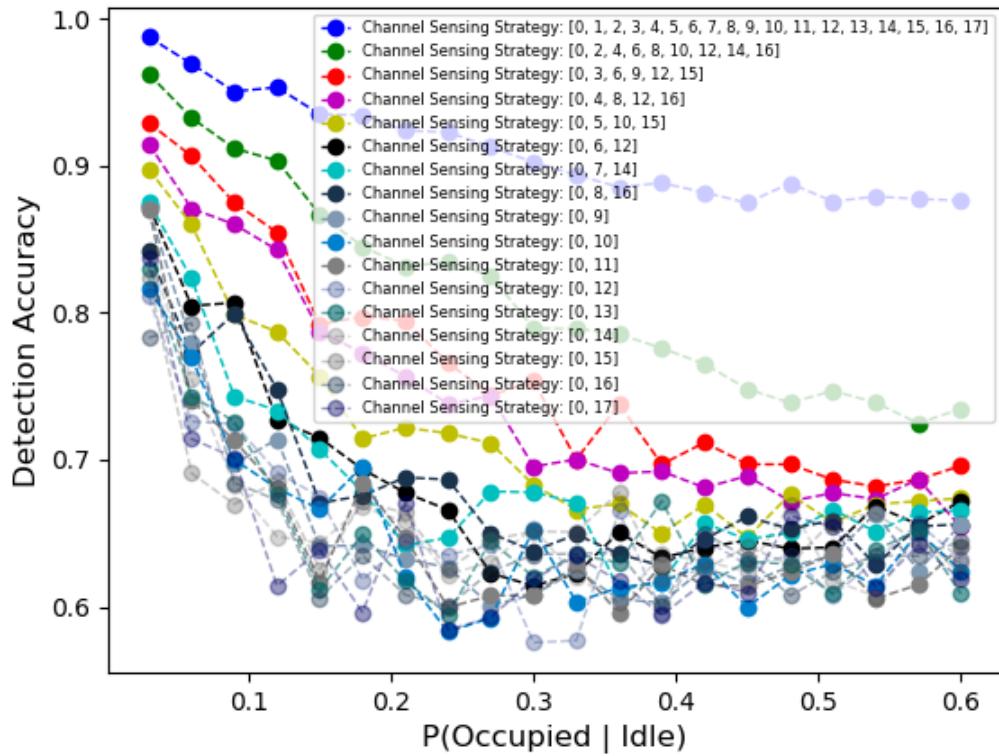


Fig. 3.6. The evaluation of estimation accuracies for different channel sensing strategies, corresponding to a single frequency-correlation Markov chain Viterbi algorithm, in relation to variations in the amount of correlation

As opposed to the double-chain Viterbi algorithm described above, now we separately simulate a single-chain Viterbi algorithm to address the advantage of having more sensing information and to prove a few results about the importance of leveraging the correlations in incumbent occupancy behavior across frequencies, which many works in the state-of-the-art fail to do. Addressing the advantage of having more sensing information, Fig. 3.6 drives home the point that sensing more channels improves the accuracy of the HMM state estimator, i.e., the Viterbi algorithm discussed here, which in turn gives the SU a better occupancy picture, and hence, a higher average utility per time-step for the Viterbi algorithm over ours, as seen in Fig. 3.5. As we go down the plot in Fig. 3.6, i.e., as the number of channels sensed per time-step decreases, the estimation accuracy/detection accuracy, which refers to the number of channels whose states (0 or 1) were correctly estimated by the Viterbi algorithm--mathematically described as  $\sum_{k=1}^K \mathcal{L} \left\{ B_k(i)=\hat{B}_k(i) \right\}$ , where  $\mathcal{L}$  is an indicator variable,  $B_k(i) \in \{0, 1\}$  is the true occupancy state of the channel in time-slot  $i$ , and  $\hat{B}_k(i) \in \{0, 1\}$  is the occupancy state of the channel estimated by the Viterbi algorithm--averaged over 300 sampling rounds, decreases consistently.

Two additional points about the effects of correlation in incumbent occupancy behavior across frequency on the accuracy of state estimation can be made by analyzing Fig. 3.6 from a different perspective--the first being that as the incumbent occupancy behavior becomes more and more correlated across frequency, as denoted by the X-axis, i.e., as  $\mathbb{P}(B_{k+1}(i)=1|B_k(i)=0), \forall 1 \leq k \leq K$  moves from a highly correlated model ( $\mathbb{P}(B_{k+1}(i)=1|B_k(i)=0)=0.1 > 0.6 = \mathbb{P}(B_{k+1}(i)=1), \forall 1 \leq k \leq K$ ) to an independence model ( $\mathbb{P}(B_{k+1}(i)=1|B_k(i)=0)=0.6 = \mathbb{P}(B_{k+1}(i)=1), \forall 1 \leq k \leq K$ ), the estimation accuracy/detection accuracy decreases; and the second point being that as the gap between the consecutive channels sensed by the agent increases, the estimation/detection accuracy worsens--as evident from the final few plot lines in Fig. 3.6--for instance, the estimation accuracy for channel sensing strategy [0, 9] is higher than that for strategy [0, 17], for the same amount of PU occupancy correlation across frequency ( $\mathbb{P}(B_{k+1}(i)=1|B_k(i)=0)=0.1, \forall 1 \leq k \leq K$ ). Therefore, this evaluation of the

single-chain Viterbi algorithm in a separate, hypothetical simulation model proves that we can achieve an improved estimation of incumbent occupancy behavior by sensing more channels per time-step and by leveraging the correlations in PU occupancy behavior across frequencies. However, as already noted, the number of channels that can be simultaneously sensed by the SU in a given time-step is restricted by design limitation [22]--hence, fixing  $\kappa=6$  in our POMDP solution, we resort to leveraging the correlation in incumbent occupancy behavior across frequency (and time) in order to attain better performance. Additionally, we find that adapting the spectrum sensing decision based on the system state (true or perceived) [8, 20], in contrast to a fixed sensing strategy [5, 6], adds to the performance gains attained by exploiting the PU occupancy correlation.

As already discussed, our proposed framework can be decomposed into two components: the parameter estimator and PERSEUS. Next, we take up each of these two individually and analyze their performance.

Specifically discussing the performance of the parameter estimation algorithm, i.e., the EM algorithm, we find that, with initial estimates of 0.5, i.e.,  $p_{uv}=0.5, \forall u, v \in \{0, 1\}$  and  $q_w=0.5, w \in \{0, 1\}$ , the estimator converges to the true parameter vector  $\vec{\theta}$  with an error/delta of  $\eta=10^{-8}$  in 45,000 iterations--this corresponds to an observation and estimation period of 135s, considering a typical time-slot duration of 3ms [20]. We illustrate this convergence via the Mean Square Error (MSE) plot depicted in Fig. 3.7, in which the MSE in iteration  $t$  given by,

$$\|\vec{\theta} - \hat{\vec{\theta}}^{(t)}\|_2^2 = \sum_{\theta \in \vec{\theta}} \mathbb{E}[(\theta - \hat{\theta}^{(t)})^2] \quad (3.32)$$

is decreased iteratively, as the estimation process goes through the E-step and the M-step in each iteration  $t$  until  $\mathbb{E}[\theta - \hat{\theta}^{(t)}] \leq 10^{-8}, \forall \theta \in \vec{\theta}$ .

On the same time-scale as the parameter estimation algorithm, focusing on the loss convergence of the PERSEUS algorithm with a discount factor of  $\gamma=0.9$  and a termination threshold of  $\epsilon=10^{-5}$ , wherein we define the expected loss as the difference between the utility obtained by the proposed PERSEUS framework, denoted

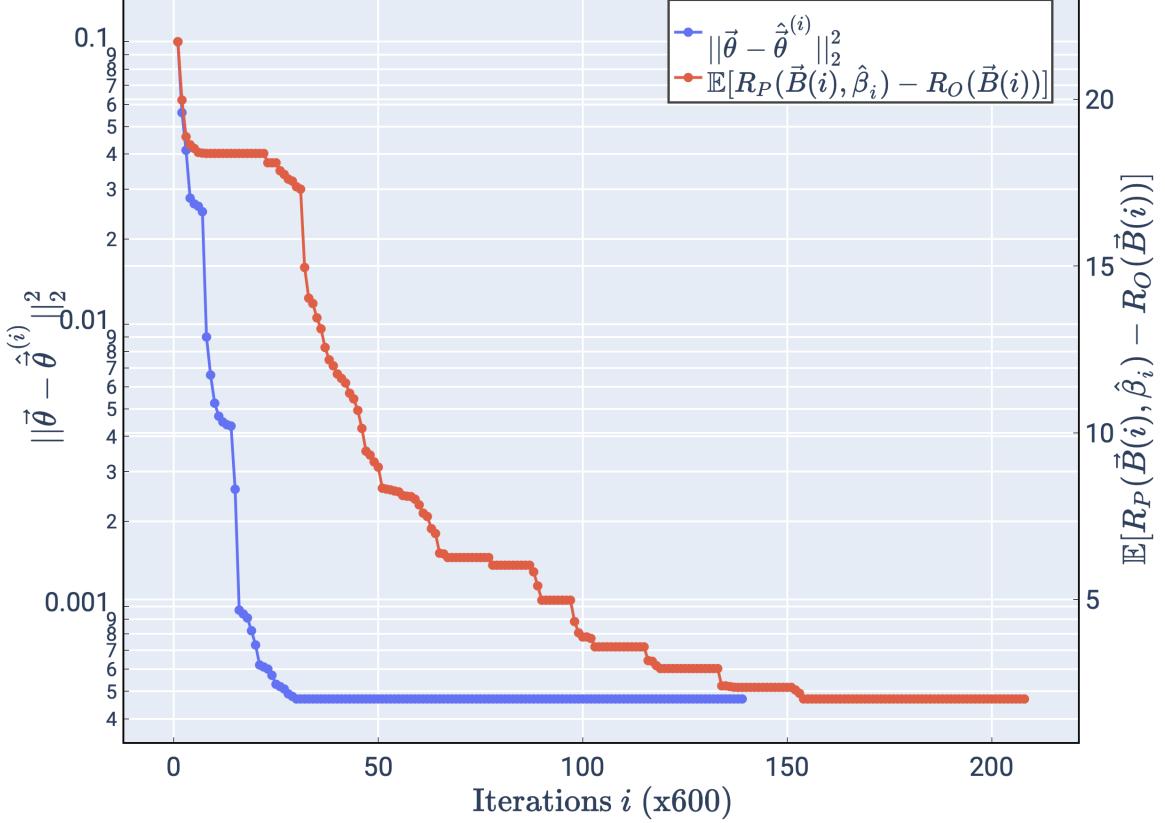


Fig. 3.7. The convergence of the MSE of the HMM EM algorithm to estimate  $\vec{\theta}$ , and the convergence of the loss of the fragmented PERSEUS algorithm with belief update simplification

by  $R_P(\vec{B}(i))$  (discussed in Section 3.1.4), and that obtained by an Oracle, which knows the exact occupancy behavior of the incumbents in the network, denoted by  $R_O(\vec{B}(i))$ --we find that, as depicted in Fig. 3.7, the loss convergence of PERSEUS is relatively slower while the parameter estimator is learning the transition model; as opposed to after the convergence of the parameter estimator, when we see a more consistent gradient towards the optimality.

Finally, inspecting Fig. 3.4 in a new light, we see that our POMDP agent limits channel access when the penalty ( $\lambda$ ) is high--leading to lower SU throughput and lower PU interference--and conversely, follows a more lenient channel access strategy when the penalty is low--resulting in higher SU throughput and higher PU inter-

ference. Generally speaking, Fig. 3.4 depicts a trend of increasing SU throughput and increasing incumbent interference, as the penalty for missed detections, i.e.,  $\lambda$  is lowered [20]. Therefore, our framework provides a crucial practical tool in cognitive radio MAC design—the ability to tune the trade-off between the throughput obtained by the cognitive radio and the interference caused by it to incumbent transmissions in the network.

### 3.4 Conclusion

In this chapter, we specifically tackle the spectrum sensing and access problem in the MAC layer of a cognitive radio that is intelligently trying to access white-spaces in the spectrum left-unused by multiple licensed users--whose occupancy behavior is correlated across both frequency and time, in a Markovian fashion. In a linear AWGN observation model, a Rayleigh fading channel model, wherein the time-frequency correlation structure dictating the occupancy behavior of the incumbents in the network is unknown to the cognitive radio node--which, by design, possesses channel sensing restrictions; we formulate the spectrum sensing problem as a POMDP (the access action depends on the observations and/or beliefs obtained from the sensing action) and propose the use of an online, concurrent framework consisting of a parameter estimator based on the HMM EM algorithm, embedded with an approximate POMDP method, i.e., the PERSEUS algorithm, in order to obtain the optimal sensing (and access) policy. By numerical evaluations, we prove that our framework outperforms the state-of-the-art--a higher SU throughput for the same level of PU interference, over the MEM with GC-CCE and MPE, and the MEM with MEI-CCE and MPE techniques in [5]; has superior performance compared to standard HMM state estimators, i.e., the Viterbi algorithm [6], proving that adapting the spectrum sensing decision in each time-slot to optimize the utility provides better performance over fixed sensing strategies; and finally, outperforms a Neyman-Pearson Detector [11] thereby showing that leveraging the incumbent occupancy correlation structure helps boost perfor-

mance. Additionally, our solution provides a means to regulate the trade-off between SU throughput and PU interference, by adjusting the parameter  $\lambda$ --this capability is particularly important in scenarios wherein different interference constraints may be imposed on the incumbents, over time.

## 4. FEASIBILITY ANALYSIS OF THE POMDP OPTIMAL POLICY ON ESP32 RADIOS

In this chapter, we detail the implementation of the POMDP optimal spectrum sensing and access policy (from Chapter 3) on ad-hoc distributed network consisting of ESP32 radios, in order to prove the feasibility of our formulation. Section 4.1 details the methodology followed in implementing the POMDP optimal sensing and access policy; Section 4.2 describes the results obtained from this implementation; and Section 4.3 involves concluding arguments.

### 4.1 Methodology

We employ 8 ESP32 radios [32], with each one embedded in a GCTronic e-puck2 robot [33], categorized into a network of 3 PUs (and their 3 corresponding sinks) occupying 6 channels in the discretized spectrum of interest according to a Markovian time-frequency correlation structure (described by (3.6))--and 2 independent SUs, with each having the capability of sensing only one channel at a time, intelligently trying to exploit the white-spaces in the spectrum. The detailed methodology of this implementation is provided below:

1. Considering a network with  $J=3$  PUs and one SU with a channel sensing restriction of  $\kappa=2$  out of  $K=6$  channels in the discretized spectrum of interest--and assuming a linear AWGN observation model, with a Rayleigh channel fading model (discussed in Section 3.1.1), we simulate the occupancy behavior of the PUs according to a Markovian time-frequency correlation structure parameterized by  $\vec{\theta}=[\vec{p}, \vec{q}]^\top$ , where  $\vec{p}=[p_{00}=0.1, p_{01}=0.3, p_{10}=0.3, p_{11}=0.7]^\top$  and  $\vec{q}=[q_0=0.3, q_1=0.8]^\top$ ; and solve for the optimal spectrum sensing and access policy using PERSEUS, embedded with a concurrent parameter estimation algo-

rithm learning the parameter vector  $\vec{\theta}$ --by mimicking the observational capabilities of the actual ESP32 radios. Note this step is performed on a PC.

2. The simulated PU occupancy behavior--Markovian correlated according to (3.6) and parameterized by  $\vec{\theta}$ , and the time-slot specific optimal channel access decisions (derived off of the POMDP optimal sensing policy and the simulated PU occupancy behavior), are stored in databases (for export onto the ESP32 network).
3. Peer-to-Peer communication links are established between a PU ESP32 radio and its sink, using the 3 ESP32 radios designated as PUs--in other words, 3 wireless communication links are established: one for each ESP32 PU pair (a source and a sink), over WiFi (2.4GHz) and using a channel according to the occupancy information detailed in the exported PU occupancy database, in time-slot  $i$ .
4. Note here that in this ESP32 PU network implementation, in time-slot  $i$ , while establishing a wireless communication link between a ESP32 PU  $j \in \{1, 2, 3\}$  and its respective sink  $i \in \{1, 2, 3\}$  s.t.  $i$  is the designated sink for PU  $j$ , i.e., while forming link  $l_{ij}$  over channel  $k_{l_{ij}} = k \in \{1, 2, \dots, 6\}$  (as determined by the exported PU occupancy database which contains simulated PU occupancy behavior according to the Markovian time-frequency correlation structure described above) such that  $k_{l_{ij}} \neq k_{l_{i',j'}}$ ,  $\forall i, i' \in \{1, 2, 3\}, j, j' \in \{1, 2, 3\}$ --PU  $j$  serves as an Access Point (AP) accepting transmission requests from PU  $i$ , which is designates as a STAtion (STA). In the next synchronized time-slot  $i + 1$ , this link  $l_{ij}$  moves to channel  $k' \in \{1, 2, \dots, 6\}$ , as detailed in the exported PU occupancy database. This same procedure takes place for the other two incumbent communication links in every time-slot until the end of the implementation evaluation period.
5. Although the PC-based POMDP solver employs an SU which can access 2 channels at a time in order to deliver its flows (see the access part of the POMDP formulation in Section 3.1.4), we employ 2 ESP32 SU radios in the network--with

the channel access work synchronously and evenly split between the two--due to the actual physical design limitation of the ESP32 radio that it can only access one channel at a time, forcing us to be creative: split the optimal 2 channel access decision in time-slot  $i$ , as determined by the time-slot specific optimal POMDP channel access database, into a single-channel access action at each ESP32 SU radio. Next, based on whether the channel access at the 2 ESP32 SU radios was successful, we compute the success rate.

## 4.2 Implementation Results

The channel access success rate metric defined as

$$\text{Channel Access Success Probability} = \frac{\sum_{j=1}^2 \mathcal{I}\{B_{k_{SU_j}}(i) = 0\}}{2}, \quad (4.1)$$

where  $\mathcal{I}$  corresponding to  $\mathcal{I}\{B_{k_{SU_j}}(i) = 0\}$  is an indicator variable whose value is 1 if the channel accessed by the ESP32 SU  $j \in \{1, 2\}$  in time-slot  $i$  is not occupied by an incumbent PU ESP32 radio, and  $B_{k_{SU_j}} \in \{0, 1\}$  is the occupancy variable of the channel accessed by the ESP32 SU  $j$  in time-slot  $i$ --is evaluated per time-slot  $i$ , and the resultant metrics are plotted against time, which is illustrated in Fig. 4.1.

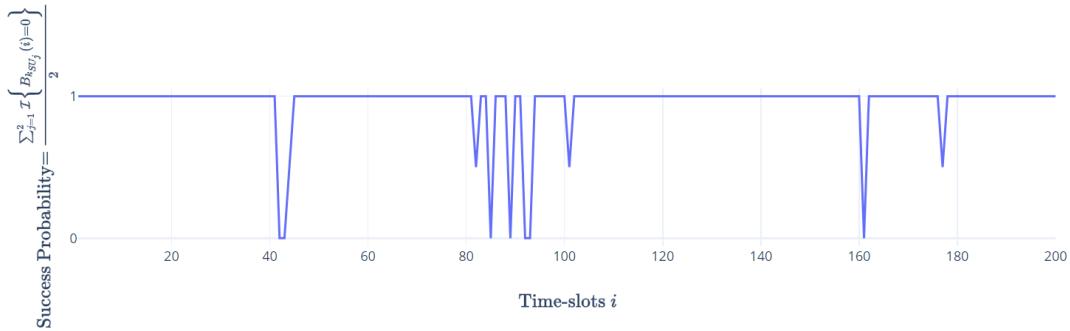


Fig. 4.1. The channel access probability of the ESP32 SU radios per time-slot

### 4.3 Conclusion

In this chapter, we simulated an ad-hoc distributed peer-to-peer network with incumbent radios occupying the spectrum according to a Markovian time-frequency correlation structure, and an SU constituting a parameter estimator and a POMDP agent solving for the optimal channel sensing and access policy--the simulated time-slotted incumbent occupancy behavior is exported onto a peer-to-peer network formed by actual ESP32 radios serving as PUs; and the time-slotted optimal channel access decisions are exported onto a peer-to-peer network formed by actual ESP32 radios serving as SUs (we use 2 SUs in the actual implementation due to design limitations, with the access work split between the two, and merged upon completion). Upon merging the channel access results from both the ESP32 SU radios, we plot the SU network's success probability (given by (4.1))--we find that the channel access decisions made by our SU network are accurate for a significant portion of the implementation evaluation period. The primary intention behind this analysis is to show that the optimal policy obtained by the POMDP solver with respect to an ad-hoc emulated network can be transferred (i.e., exported) onto the actual physical network with ease, by leveraging synchronization and data aggregation techniques.

## 5. SUMMARY

Comprehending the need for spectrum sharing from an economic and national security point of view, in this work, we describe the various design principles involved in the development of cognitive radios, while specifically focusing on certain aspects of the design, proving their superiority to the state-of-the-art, and finally, their implementation feasibility. Detailing further, in this work, we describe the design principles underlying the operation of our cognitive radio in the DARPA SC2 Colosseum--importantly, the CIRN architecture under which our networks were deployed in any given scenario, the OFDM PHY, the MCS adaptation algorithm driven by packet error rates, prioritized flow-scheduling involving a recursive-revisitation value-per-resource heuristic, the channel access algorithm driven by PSD measurements and collaboration messages, the CIL protocol, and multi-hop routing. Moreover, we present actual performance plots of these algorithms in DARPA SC2-emulated scenarios, in which the performance of our network is compared in real-time with that of other competitor networks deployed in the same scenario, in addition to an ensemble-view of the performance of all the networks in the scenario, to prove the need for collaboration among networks.

While acknowledging the possibility and potential of improvements in the other aspects of our radio's design, we focused our subsequent research particularly on the spectrum sensing and access algorithms in the MAC layer of the radio's network protocol stack and sought to improve it by adopting a rigorous mathematical approach as opposed to a more heuristic one incorporated in our DARPA SC2 radio. In this regard, in an AWGN observation model and a Raleigh fading channel model, we developed a POMDP formulation for channel sensing and access in a radio environment involving multiple incumbents exhibiting a time-frequency correlation structure in their occupancy behavior. In this POMDP formulation, in order to solve for the opti-

mal channel sensing and access policy to be employed in the MAC layer of our radio, we designed an online parameter estimation algorithm leveraging tools from HMMs and MLE, and embedded it directly into a randomized point-based approximate value iteration method known as the PERSEUS algorithm with additional customizations such as fragmentation and belief update simplification. To prove the superiority of our POMDP framework for spectrum sensing and access, we have presented numerical evaluations of our algorithm against the state-of-the-art--in doing so, we have proved that, for the same amount of deterioration in the throughput of the PUs in the network, our solution obtains a 37.5% improvement in SU network throughput, compared to the MEM with MPE-MEI from [5]; a 25% improvement over a Neyman-Pearson Detector with no sensing restrictions from [11]; and 6% improvement over the Viterbi algorithm from [6]. Critically, accounting for the need to handle deployment scenarios in which different interference constraints may be imposed at different times and in different geographical regions, our framework facilitates the trade-off between the SU network throughput and PU interference, by tuning the penalty parameter  $\lambda$ . Additionally, with this formulation, we drive home three crucial results: leveraging the time-frequency correlation structure exhibited in the occupancy behavior of the incumbents in the network leads to significant improvements in estimation accuracy, while allowing the radio to make wise decisions with limited information (due to channel sensing restrictions); adapting the spectrum sensing decisions according to past actions and their corresponding rewards leads to more white spaces being found for use by the cognitive radio; and an online EM algorithm for HMMs (known as the Baum-Welch algorithm) can be used to estimate the time-frequency occupancy correlation structure in tandem with a POMDP policy optimization algorithm, in non-stationary settings.

Finally, the performance metrics and illustrations presented from the POMDP optimal policy implementation experiment on an ad-hoc distributed network of ESP32 radios embedded on off-the-shelf e-puck2 robots, prove the simplicity in adapting the

algorithms to different network setups and to different radio terminals having varying degrees of computational capabilities.

## REFERENCES

## REFERENCES

- [1] D. FitzGerald, "Wireless companies share the airwaves," *The Wall Street Journal*, 2019. [Online]. Available: <https://www.wsj.com/articles/wireless-companies-share-the-airwaves-11577538001>
- [2] D. Goldin, "Keep 5g safe from chinese domination," *The Wall Street Journal*, 2020. [Online]. Available: <https://www.wsj.com/articles/keep-5g-safe-from-chinese-domination-11580342112>
- [3] C. Pradhan, K. Sankhe, S. Kumar, and G. R. Murthy, "Revamp of enodeb for 5g networks: Detracting spectrum scarcity," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2015, pp. 862–868.
- [4] Ericsson, "5g use cases—explore how 5g will revolutionize 5 key industries including: Tv and media; manufacturing; healthcare; telecommunications; and transportation and infrastructure." *Ericsson*, 2019. [Online]. Available: <https://www.ericsson.com/en/5g/use-cases>
- [5] M. Gao, X. Yan, Y. Zhang, C. Liu, Y. Zhang, and Z. Feng, "Fast spectrum sensing: A combination of channel correlation and markov model," in *2014 IEEE Military Communications Conference*, Oct 2014, pp. 405–410.
- [6] C. Park, S. Kim, S. Lim, and M. Song, "Hmm based channel status predictor for cognitive radio," in *2007 Asia-Pacific Microwave Conference*, Dec 2007, pp. 1–4.
- [7] K. Cohen, Q. Zhao, and A. Scaglione, "Restless multi-armed bandits under time-varying activation constraints for dynamic spectrum access," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Nov 2014, pp. 1575–1578.
- [8] J. Lundén, S. R. Kulkarni, V. Koivunen, and H. V. Poor, "Multiagent reinforcement learning based spectrum sensing policies for cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 858–868, Oct 2013.
- [9] N. Michelusi and U. Mitra, "Cross-layer estimation and control for cognitive radio: Exploiting sparse network dynamics," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 1, pp. 128–145, March 2015.
- [10] N. Michelusi, M. Nokleby, U. Mitra, and R. Calderbank, "Multi-Scale Spectrum Sensing in Dense Multi-Cell Cognitive Networks," *IEEE Transactions on Communications*, vol. 67, no. 4, pp. 2673–2688, April 2019.
- [11] S. Mosleh, A. A. Tadaion, and M. Derakhtian, "Performance analysis of the neyman-pearsong fusion center for spectrum sensing in a cognitive radio network," in *IEEE EUROCON 2009*, May 2009, pp. 1420–1425.

- [12] H. W. Jenkins Jr., "How government can get brave about spectrum," *The Wall Street Journal*, 2019. [Online]. Available: <https://www.wsj.com/articles/how-government-can-get-brave-about-spectrum-11560552078>
- [13] M. Rosker, "Spectrum collaboration challenge (sc2)," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: <https://www.darpa.mil/program/spectrum-collaboration-challenge>
- [14] DARPA, "Scenarios summary list," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2019. [Online]. Available: <https://sc2colosseum.freshdesk.com/support/solutions/articles/22000236679-scenarios-summary-list-phase-3>
- [15] DARPA, "Collaboration network," *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: <https://sc2colosseum.freshdesk.com/support/solutions/articles/22000220450-collaboration-network>
- [16] F. A. P. d. Figueiredo, D. Stojadinovic, P. Maddala, R. Mennes, I. Jabandžić, X. Jiao, and I. Moerman, "Scatter phy: A physical layer for the darpa spectrum collaboration challenge," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.
- [17] R. J. Baxley and R. S. Thompson, "Team zylinium darpa spectrum collaboration challenge radio design and implementation," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.
- [18] D. Stojadinovic, F. A. P. de Figueiredo, P. Maddala, I. Seskar, and W. Trappe, "Sc2 cil: Evaluating the spectrum voxel announcement benefits," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.
- [19] S. Giannoulis, C. Donato, R. Mennes, F. A. P. de Figueiredo, I. Jabandžić, Y. De Bock, M. Camelo, J. Struye, P. Maddala, M. Mehari, A. Shahid, D. Stojadinovic, M. Claeys, F. Mahfoudhi, W. Liu, I. Seskar, S. Latre, and I. Moerman, "Dynamic and collaborative spectrum sharing: The scatter approach," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–6.
- [20] B. Keshavamurthy and N. Michelusi, "Spectrum Sensing in Cognitive Radio Networks via Approximate POMDP," 2020, Under review at IEEE Wireless Communication Letters.
- [21] F. Xu, L. Zhang, Z. Zhou, and Y. Ye, "Architecture for next-generation reconfigurable wireless networks using cognitive radio," in *2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, May 2008, pp. 1–5.
- [22] S. Maleki, S. P. Chepuri, and G. Leus, "Energy and throughput efficient strategies for cooperative spectrum sensing in cognitive radios," in *2011 IEEE 12th International Workshop on Signal Processing Advances in Wireless Communications*, June 2011, pp. 71–75.

- [23] L. Ferrari, Q. Zhao, and A. Scaglione, “Utility maximizing sequential sensing over a finite horizon,” *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3430–3445, July 2017.
- [24] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, “Mining spectrum usage data: A large-scale spectrum measurement study,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1033–1046, June 2012.
- [25] DARPA, “Incumbent-passive,” *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: <https://sc2colosseum.freshdesk.com/support/solutions/articles/22000223822-incumbent-passive>
- [26] DARPA, “Radio command and control api,” *Defense Advanced Research Projects Agency (DARPA) Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: <https://sc2colosseum.freshdesk.com/support/solutions/articles/22000220460-radio-command-and-control-c2-api>
- [27] DARPA, “Collaboration network protocol specification,” *DARPA Spectrum Collaboration Challenge (SC2)*, 2018. [Online]. Available: <https://sc2colosseum.freshdesk.com/support/solutions/articles/22000220468-collaboration-protocol-specification>
- [28] M. T. J. Spaan and N. A. Vlassis, “Perseus: Randomized point-based value iteration for pomdps,” *CoRR*, vol. abs/1109.2145, 2011. [Online]. Available: <http://arxiv.org/abs/1109.2145>
- [29] W. Turin, “Map decoding using the em algorithm,” in *1999 IEEE 49th Vehicular Technology Conference (Cat. No.99CH36363)*, vol. 3, May 1999, pp. 1866–1870 vol.3.
- [30] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artif. Intell.*, vol. 101, no. 1–2, p. 99–134, May 1998.
- [31] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for pomdps,” in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, ser. IJCAI’03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, p. 1025–1030.
- [32] Espressif Systems (Shanghai) Co. Ltd., “Espressif esp32: A different iot power and performance,” *Espressif*, 2019. [Online]. Available: <https://www.espressif.com/en/products/hardware/esp32/overview>
- [33] GCTronic, “e-puck2 specifications and general wiki,” *GCTronic e-puck2 online wiki*, 2020. [Online]. Available: <https://www.gctronic.com/doc/index.php/e-puck2>