

Enhancing Vehicular Networking Using Bumblebee Foraging Theory and Signals of Opportunity

Kuldeep Singh Gill



A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical and Computer Engineering
May 2022

APPROVED:

Professor Alexander M. Wyglinski
Primary Advisor
Worcester Polytechnic Institute

Professor Kaveh Pahlavan
Committee Member
Worcester Polytechnic Institute

Professor Seyed Zekavat
Committee Member
Worcester Polytechnic Institute

Abstract

In the future, connected vehicles will drastically reduce the number of road traffic accidents, leading to safer, more reliable transportation. Connected vehicle technology will enable cars to communicate with each other to share safety and infotainment information. However, there are some key challenges which must be addressed before large-scale deployments of this technology. First, the spectrum currently allocated for vehicular communication is insufficient to sustain high network traffic loads in congested urban environments. Second, GPS-based localization, which is critical for the operation of connected vehicles, is inadequate in urban environments. To address these challenges and improve vehicular networks, this dissertation presents two key contributions: (1) a novel bumblebee-based vehicular dynamic spectrum access (B-VDSA) algorithm as a promising solution for spectrum scarcity, and (2) signals-of-opportunity (SOP) based localization for GPS-denied environments.

The B-VDSA algorithm estimates optimal channels in a distributed and time-efficient manner by utilizing bumblebee intelligence. The channel selection strategy derives fundamental concepts from the bumblebee foraging model. The algorithm is integrated into the MAC layer of the DSRC and C-V2X protocol stacks to demonstrate its feasibility in Vehicle-to-Vehicle (V2V) communication. Numerical simulation results showed substantial gain in the probability of the best channel selection achieved relative to a uniform sampling allocation approach. Similarly, we observed an increase in the probability of successful reception when employing the bumblebee algorithm via a system-level simulator.

The SOP-based localization is a novel opportunistic approach of passive RF localization designed for detecting “phantom car” attacks, *i.e.*, vehicles intentionally reporting false information against the surrounding vehicles and communication networks. The feasibility of the proposed SOP-based localization approach was evaluated using a custom-built Python-based computer simulation platform. A hardware field experiment was also conducted for evaluating the performance of the proposed approach incorporating radio frequency (RF) localization, data fusion, and vehicle behavioral dynamics.

Acknowledgements

This research was supported through generous contribution of the National Science Foundation under Enhancing Access to the Radio Spectrum (EARS) program (award number 1547291) and Verizon. I want to thank Professor Kaveh Pahlavan and Professor Seyed Zekavat for serving on my committee and providing valuable edits and suggestions to enhance the quality of this work.

I want to express my deepest gratitude to my adviser Professor Alexander Wyglinski for his continuous guidance and support towards my doctoral studies. I am very grateful for the opportunity to work with him in the Wireless Innovation Laboratory (WILab) at Worcester Polytechnic Institute for the past 5 years. I would also like to take this opportunity to thank Adrian Kliks, Pawel Kryszkiewicz, and Pawel Sroka who helped me immensely in publishing quality work.

I would also like to thank my WILab team members Kyle, Nivetha, Le, and Renato for their immense support during my doctoral studies. I learnt a lot of exciting things from them, and discussions with them also helped me debug my projects. My parents pushed me to pursue higher studies, so without them nothing would have been possible.

Finally, I would like to thank my wife Sukhpreet and father-in-law Sukhwinder, without whose constant support I would never be able to finish my dissertation in time. My wife's time management skill and my father-in-law's constant encouragement came in handy during the final phase of my PhD.

Contents

| | |
|---|------------|
| List of Abbreviations | xiv |
| List of Figures | xv |
| List of Tables | xv |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 State-of-the-Art | 5 |
| 1.3 Research Contributions | 6 |
| 1.4 Dissertation Organization | 10 |
| 1.5 List of Related Peer-Reviewed Publications | 11 |
| 2 Bumblebee Behavioral Models and Dynamic Spectrum Access Fundamentals | 14 |
| 2.1 Bumblebee Foraging Theory | 14 |
| 2.1.1 Translation Between the Two Worlds | 17 |
| 2.2 Bumblebee-Based Channel Selection | 20 |
| 2.2.1 General Overview | 20 |
| 2.2.2 Channel Occupancy Characteristics and Theory | 22 |
| 2.3 Vehicular Dynamic Spectrum Access | 27 |
| 2.3.1 802.11p and C-V2X-based VANETs | 27 |
| 2.3.2 Platooning-based VDSA | 29 |
| 2.4 Chapter Summary | 30 |

| | | |
|----------|--|-----------|
| 3 | Proposed Bumblebee-based VDSA Framework | 31 |
| 3.1 | Bumblebee-Based Switching Decision | 31 |
| 3.2 | Capacity Bounds using Queuing Theory | 36 |
| 3.2.1 | Numerical Simulation Results | 38 |
| 3.3 | System-Level Simulation Results | 40 |
| 3.3.1 | GEMV ² and SUMO | 40 |
| 3.3.2 | Bumblebee-based Platooning VDSA | 43 |
| 3.4 | Hardware Validation | 49 |
| 3.4.1 | Adhoc Network Testbed | 52 |
| 3.5 | Chapter Summary | 56 |
| 4 | Memory-Enabled Bumblebee Algorithm for Efficient Channel Selection | 58 |
| 4.1 | MEBA for Platooning | 58 |
| 4.1.1 | Optimal Non-uniform Sampling Allocation | 59 |
| 4.1.2 | Analysis of Optimal Allocation and Proposal of Unequal Samples Allocation | 67 |
| 4.1.3 | Bumblebee Behavior-based Channel Sensing and Selection in Time-varying Environment with Channel Switching Cost | 75 |
| 4.1.4 | Memory-based Bumblebee Algorithm Description | 75 |
| 4.1.5 | Simulation Results | 77 |
| 4.2 | Evaluation in Platooning Scenario | 80 |
| 4.2.1 | Simulation Setup | 81 |
| 4.2.2 | Simulation Results for a Four Car Platoon | 83 |
| 4.2.3 | Simulation Results for a Ten Car Platoon | 85 |
| 4.3 | Hardware Validation | 88 |
| 4.3.1 | Bumblebee Model for C-V2X VDSA | 90 |
| 4.3.2 | OpenAirInterface-based Experimentation Testbed | 92 |
| 4.3.3 | Measurement and Results | 95 |
| 4.4 | Chapter Summary | 97 |
| 5 | Passive Opportunistic RF Localization of Connected Vehicles | 99 |

| | | |
|----------|---|------------|
| 5.1 | Overall Framework | 101 |
| 5.2 | EM Emissions and Sensing Module | 103 |
| 5.3 | Simulation Workflow | 105 |
| 5.3.1 | Use Case 1: Stationary Sensors and Moving Emitter | 108 |
| 5.3.2 | Use Case 2: Moving Sensors and Moving Emitter | 110 |
| 5.4 | Small-Scale Field Experimentation | 112 |
| 5.4.1 | Experimental Setup | 112 |
| 5.4.2 | Localization Results | 115 |
| 5.5 | Chapter Summary | 116 |
| 6 | Research Achievements and Future Work | 117 |
| 6.1 | Research Outcomes | 117 |
| 6.2 | Future Work | 119 |
| A | Appendix: Channel Busy Ratio Modeling | 120 |
| A.1 | binomial_based_channel_occupancy.py | 120 |
| A.2 | cbr_analytical_evaluation.m | 122 |
| | Bibliography | 130 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | C-V2X evolution toward 5G beginning from IEEE 802.11p which provided the foundation for 5G C-V2X [4]. | 2 |
| 1.2 | Federal Communication Commission (FCC) spectrum allocation chart for UNII-3 bands where upper 30 MHz is reserved for C-V2X direct operations and lower 45 MHz can be used for unlicensed operations [6]. | 3 |
| 1.3 | Artificial mixed floral environment used to investigate memory-based decision-making processes in bumblebees. Each color represents a species (channel) [23]. | 7 |
| 1.4 | Illustration of the proposed overall phantom car attack detection framework [24]. | 8 |
| 2.1 | The memory-based channel selection algorithm from bee (top) and vehicle (bottom) perspectives. Like each bumblebee, each vehicle is equipped with memory to store channel (floral) reward information, which is then used to select the channel (floral species) with the highest reward quality out of those available. Vehicles (bumblebees) alternate between sampling (T_{st}) and transmission (T_{tx}) periods to track changes in a time-varying noisy resource environment [23]. | 16 |
| 2.2 | Fixed time snapshot of power spectral density of PCS band shows the vacant and occupied channels in that band [23]. | 23 |
| 2.3 | Fixed Time Snapshot of Power spectral density of GSM band shows the vacant and occupied channels in that band [23]. | 24 |
| 2.4 | Fixed Time Snapshot of Power spectral density of lower LTE band shows the vacant channels in the band [23]. | 25 |

| | | |
|------|--|----|
| 2.5 | Percentage Occupancy of the PCS band during 30 minutes period from 1850–1990 MHz bandwidth [23]. | 26 |
| 2.6 | CBR β variation with cars arriving and departing the transmission range of the platoon vehicles. Different channels have different β values based on the channel utilization. Non-uniform channel sampling is performed by the platoon in order to find optimal channel from L channels with least β . There are total N sampling periods which are distributed across the channels to optimize the sampling resources [28]. | 28 |
| 3.1 | Flowchart describing the bumblebee based distributed optimization algorithm [25]. | 32 |
| 3.2 | Normalized squared magnitude of the channel impulse response. | 35 |
| 3.3 | Total vehicles supported with optimal communication between the vehicles [25]. | 38 |
| 3.4 | Comparison of Switching Cost in highway and urban environment. The high switching cost in the urban scenario implies the high amount of channel switching which results from highly time-variant channel [25]. | 39 |
| 3.5 | Average number of cars per square kilometer simulated in GEMV ² in Worcester, MA [25]. | 40 |
| 3.6 | The Probability (P_m) of all channel being busy observed by the vehicles during the discrete time-step emulating the real-world environment [25]. | 41 |
| 3.7 | The predicted Mean Response Time (T_R) for vehicles performing VDSA in a simulated real-time environment [25]. | 42 |
| 3.8 | Mean Channel Reward $r(\bar{t})$ comparison for bumblebee model. | 43 |
| 3.9 | Mean Channel Reward for memoryless bumblebee model. | 44 |
| 3.10 | Mean Channel Reward for highway and urban scenario for different memory size. | 45 |
| 3.11 | Probability of successful reception of leaders' packets [29]. | 46 |

| | | |
|------|--|----|
| 3.12 | Example of frequency switching for two platoons moving in opposite directions and different considered switching costs C . It can be observed that frequency band switching occurs when the platoons are close to each other in the time interval of 60-80 s. Frequency bands at 490 MHz and 522 MHz are rarely or never selected due to the presence of DVB-T transmissions [29]. | 47 |
| 3.13 | Cumulative distribution of SIR for the protected DTT receivers [29]. | 48 |
| 3.14 | Layout of the laboratory where the experiment is conducted. The layout shows the location of each individual radio node [26]. | 50 |
| 3.15 | The experimental test-bed consisting of six Pluto-SDRs forming an Ad-Hoc network. The experiment is conducted in a controlled laboratory environment [26]. | 50 |
| 3.16 | P_d vs SNR for threshold factor $K = 6, 8, 10, 12$. As we increase K , the sensing performance degrades due to large misdetections in the network [26]. | 54 |
| 3.17 | Packet Failure Rate in an Ad-hoc network as the nodes are increased from $N = 2$ to 6 for varying SNR [26]. | 55 |
| 3.18 | Bumblebee algorithm switching from one channel to another avoiding the interference node. During the entire experiment, radio running bumblebee algorithm stays on channel 2, reducing switching cost [26]. | 56 |
| 3.19 | PDR for bumblebee and random channel selection for different packet size. Due to frequent channel switchings and large packet collisions with the interference node, the random channel selection has a high packet loss. The variation of PDR for each packet size was around 1%-2% in both bumblebee and random channel selection [26]. | 57 |
| 4.1 | Probability of optimal channel selection versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 8$ [28]. | 68 |
| 4.2 | Cumulative number of sampling moments per channel (ch.) versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 8$ [28]. | 69 |

| | | |
|------|---|----|
| 4.3 | Probability of optimal channel selection versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 6$ [28]. | 71 |
| 4.4 | Cumulative number of sampling moments per channel versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 6$ [28]. | 72 |
| 4.5 | CDF of normalized number of iterations required to obtain probability of best channel selection equal 0.95 over 5200 random system configurations (varying N , L and β_i). Normalization over number of iterations required by equal allocation [28]. | 74 |
| 4.6 | Time-varying channel busy ratio generated using the simplified birth-death process [94] with $L = 4$ [28]. | 78 |
| 4.7 | Performance comparison for equal sample allocation and the proposed heuristic approach for $\gamma = -2$. It shows that heuristic approach outperform equal allocation with no memory. The time from 10 – 30 shows a very high channel utilization where heuristic needs to adapt to the variation but due to no memory the performance is low [28]. | 79 |
| 4.8 | Best Channel Selection gain (%) is evaluated for different forgetting factors and sliding window memory length. The plot demonstrate the optimal values of both EWMA and SWA memory schemes computed using simulation [28]. | 80 |
| 4.9 | Performance comparison for heuristic approach with $\gamma = -2$ for different memory lengths and equal weight tapering. Memory length $K = 4$ was found to be optimal in simulation. The right axis shows the performance gain over memoryless case [28]. | 81 |
| 4.10 | Performance comparison for the heuristic approach when $\gamma = -2$ with different exponentially weighted memory average tapering. $\alpha = 0.7$ appeared to be optimal in simulation. The axis on the right shows the performance gain over memoryless case [28]. | 82 |
| 4.11 | Selected channel (band) index versus time in the case of 4 vehicles platoon [28]. | 85 |

| | | |
|------|---|----|
| 4.12 | Estimates of the probability of successful reception of leader's packets versus car position in a 4-vehicles platoon [28]. | 86 |
| 4.13 | Example of time evolution of the successful reception ratio of leader's packets observed in a window of 10 s for car no. 2 in a 4-vehicles platoon [28]. | 86 |
| 4.14 | Selected channel (band) index versus time in the case of 10 vehicles platoon [28]. | 87 |
| 4.15 | Estimates of the probability of successful reception of leader's packets versus car position in a 10-vehicles platoon [28]. | 87 |
| 4.16 | Example of time evolution of the successful reception ratio of leader's packets observed in a window of 10 s for car no. 9 in a 10-vehicles platoon [28]. | 88 |
| 4.17 | C-V2X Protocol Stack implemented on OAI testbed. The PHY layer is modified to integrate the bumblebee-based VDSA algorithm while reusing the higher layer setup from OpenAirinterface. For the C-V2X Mode 4, random selection is employed on the PSSCH (Physical Sidelink Shared) at a given channel pool [28]. | 93 |
| 4.18 | C-V2X radio testbed consisting of two USRP X310. One X310 is equipped with one SBX daughterboard for LTE transceiver link, whereas another X310 has two SBX daughterboard for simultaneous C-V2X and spectrum sensing functionality. For the C-V2X link, a wired connection is used whereas the VDSA band for the sub-channels is monitored using the VERT2450 Antenna with 3 dBi gain. The Octoclock and GPSDO are used to provide the external 10 MHz frequency reference in order to achieve sidelink synchronization between the radios [28]. | 93 |
| 4.19 | The power spectral density (PSD) for DTV band from 470 MHz to 520 MHz across 5 minutes time domain. There are spectrum access opportunities across both time and frequency and due to the stationary nature of the band these opportunities are static [28]. | 95 |

| | | |
|------|---|-----|
| 4.20 | Monitoring the DTV channel 19 at 500 MHz using <i>Radio:1</i> while simultaneously performing C-V2X communication using <i>Radio:0</i> . In the experiment, sensing radio employing bumblebee algorithm monitors DTV channels whereas C-V2X radio keeps the communication active. Sensing radio discovers DTV band with better channel quality and perform the channel switching [28]. | 96 |
| 4.21 | Latency comparison for bumblebee-based VDSA algorithm for different memory block size. Two channel bandwidth 5 MHz and 10 MHz are employed for resource allocation [28]. | 97 |
| 5.1 | Illustration of the proposed phantom car attack detection framework, where Vehicles A, B, and C are passively sensing RF emissions coming from Vehicle D. These RF emissions can be any form of wireless signal, including 5G cellular communications, WiFi, Bluetooth, and tire pressure measurement sensor (TPMS) transmissions. It is important the surrounding vehicles are capable of detecting those signals although it is not necessary to decode transmissions. From these intercepted RF emissions, each vehicle extracts RF characteristics such as received signal strength (RSS) and arrival time values. These RF measurements are communicated to the network along with the precise position of Vehicles A, B, and C. The network for perform RF localization via data fusion using these measurements, and then compare with self-reported position information to determine whether the network is under attack [24]. | 100 |
| 5.2 | Input/output definition of vehicle electromagnetic (EM) emissions generation module [24]. | 105 |
| 5.3 | Multi-modality data-fusion employing RSS and TDoA for non-VZ subscriber localization [24]. | 107 |
| 5.4 | RSS-based localization using static sensors [24]. | 109 |
| 5.5 | TDoA-based localization using static sensors [24]. | 110 |

| | | |
|------|--|-----|
| 5.6 | Hybrid RSS-TDoA fusion-based localization using static sensors [24]. . . | 110 |
| 5.7 | RSS-based localization using moving sensors [24]. | 111 |
| 5.8 | TDoA-based localization using moving sensors [24]. | 111 |
| 5.9 | Hybrid RSS-TDoA fusion-based localization using moving sensors [24]. | 112 |
| 5.10 | Experiments performed in straight North/South direction for its capacity to drive three bikes and one emitter in a straight lane. Four different scenarios were evaluated using the proposed localization technique. . . | 113 |
| 5.11 | Raw estimated emitter trajectory, ground truth emitter trajectory, and trajectory of the three sensors. | 114 |
| 5.12 | Localization testbed consisting of ADALM Pluto and RTL-SDR software-defined radios along with smartphone which is equipped with GPS logger. | 115 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Several Definitions in Connected Vehicular Communications and Their Equivalent Definitions in Bumblebees [23] | 17 |
| 3.1 | Average number of frequency changes with bumblebee-based algorithm using different switching costs. | 47 |
| 4.1 | Set of simulated N and L configurations | 74 |
| 4.2 | Simulation parameters | 84 |
| 4.3 | Configuration parameters for bumblebee-based VDSA testbed. | 94 |
| 5.1 | Simulation Parameters for C-V2X Channel Model | 106 |
| 5.2 | Configuration parameters for emitter and sensor. | 113 |

List of Abbreviations

| | |
|---------|--|
| 3GPP | Third Generation Partnership Project |
| 4G | Fourth generation |
| 5G | Fifth generation |
| ABC | Artificial Bee Colony |
| ACIR | Adjacent Channel Interference Ratio |
| AWGN | Additive White Gaussian Noise |
| BPSK | Binary Phase Shift Keying |
| BSM | Basic Safety Messages |
| CACC | Cooperative Adaptive Cruise Control |
| CAM | Cooperative Awareness Messages |
| CBR | Channel Busy Ratio |
| CCH | Control Channel |
| COR | Channel Occupancy Ratio |
| CSA | Channel Switch Announcement |
| CSMA/CA | Carrier Sense Multiple Access with Collision Avoidance |
| CV2X | Cellular Vehicle to Everything |
| DNN | Deep Neural Network |
| DSA | Dynamic Spectrum Access |
| DSRC | Dedicated Short Range Communication |
| DTV | Digital Television |
| DTT | Digital Terrestrial Television |
| DVB-T | Digital Video Broadcasting Terrestrial |

| | |
|-------|--|
| ED | Energy Detection |
| FFT | Fast Fouier Transform |
| GA | Genetic Algorithms |
| GEMV2 | Geometry-Based Efficient Propagation Model Vehicles for V2V Communications |
| GPS | Global Positioning Systems |
| ITS | Intelligent Transportation Services |
| LoS | Line of Sight |
| LTE | Long-Term Evolution |
| MCS | Modulation and Coding Set |
| OFDM | Orthogonal Frequency Division Multiplexing |
| PER | Packet Error Ratio |
| PDF | Probability Distribution Function |
| PDR | Packet Delivery Ratio |
| ProSe | Proximity Services |
| PSO | Partial Swarm Optimization |
| PU | Primary User |
| QPSK | Quadrature Phase Shift Keying |
| REM | Radio Environment Map |
| RSSI | Received Signal Strength Indicator |
| SDR | Software-Defined Radio |
| SINR | Signal To Interference Plus Noise Ratio |
| SU | Secondary User |
| TVWS | Television White Space |
| VDSA | Vehicular Dynamic Spectrum Access |
| V2I | Vehicle-to-Infrastructure |
| V2V | Vehicle-to-Vehicle |
| V2X | Vehicle-to-everthing |
| VANET | Vehicular Ad-hoc Network |
| WAVE | Wireless Access in Vehicular Environment |

Chapter 1

Introduction

1.1 Motivation

Vehicle technology is gradually moving towards context aware systems where vehicles are cognizant of the environment for mobility decisions. Current vehicular systems heavily rely on direct line-of-sight (LOS) for sensing the environment. Connected vehicles can exchange driving information via Basic Safety Messages (BSM) within a transmission range of 500 meters [1]. The BSM can carry information regarding current position, speed of the vehicle, direction, *etc.*, and can provide critical support for vehicular technology infrastructure. Connecting vehicles via both wireless communication and networking solutions has been extensively studied, especially Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) architectures. The IEEE 802.11p Dedicated Short Range Communication (DSRC) standard [2] was the first framework designed to meet the demands of the Vehicular Network (VANET) architecture. However, despite several of its initial advantages, IEEE 802.11p also possesses shortcomings such as low reliability, hidden node problem, unbounded delay and sporadic V2I connectivity [3].

Alternatively, the Third Generation Partnership Project (3GPP) specified an approach to vehicular connectivity called Cellular Vehicle-to-Everything (C-V2X) in Release 14, where direct communication is supported between vehicles to accommodate the latency requirements for time-sensitive vehicular applications. C-V2X offers several features designed to

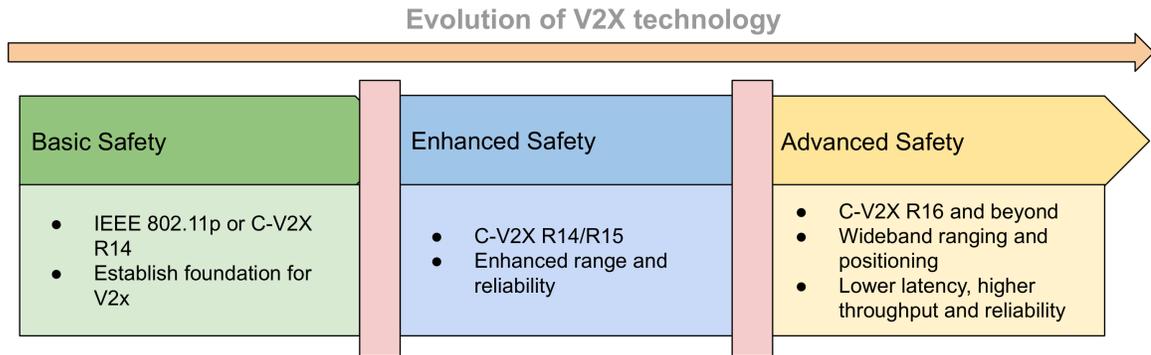


Figure 1.1: C-V2X evolution toward 5G beginning from IEEE 802.11p which provided the foundation for 5G C-V2X [4].

support ITS applications with respect to coverage, mobility support, reliability, and scalability. In particular, 3GPP added two new modes (Mode 3 and Mode 4) in order to supplement Proximity Services (ProSe) for V2V communications [5]. The primary driving force for C-V2X is the LTE backhaul network, which can directly be ported for vehicular communication without spending billions on setting up an entirely new infrastructure and large initial CAPEX requirement. Figure ?? describes the evolution of vehicular communication from 802.11p [4].

The major challenge in facilitating V2V communication is the spectrum requirement to meet the rapidly growing demand for connected cars. FCC had earlier allocated six channels in the 5.9 GHz Intelligent Transportation Systems (ITS) band, which was around 75 MHz. However, the channels were later reduced to only 30 MHz [6], which is clearly not enough to sustain the high demand of connected vehicles. Currently, the most optimistic solution to meet the spectrum scarcity demand is to leverage underutilized wireless spectrum without interfering with the primary users. In this dissertation, digital television (DTV) band from 470 MHz to 520 MHz was chosen to demonstrate the feasibility of dynamic spectrum access for vehicular communication. The primary user needs of the DTV band are relatively stable when compared to other wireless frequency bands [7]. Hence, a simple spectrum management system can achieve good throughput and low latency without requiring complex spectrum sensing techniques. The primary users (PU) are authorized for the spectrum use, and their performance cannot be compromised in any scenario. Hence, FCC has laid down

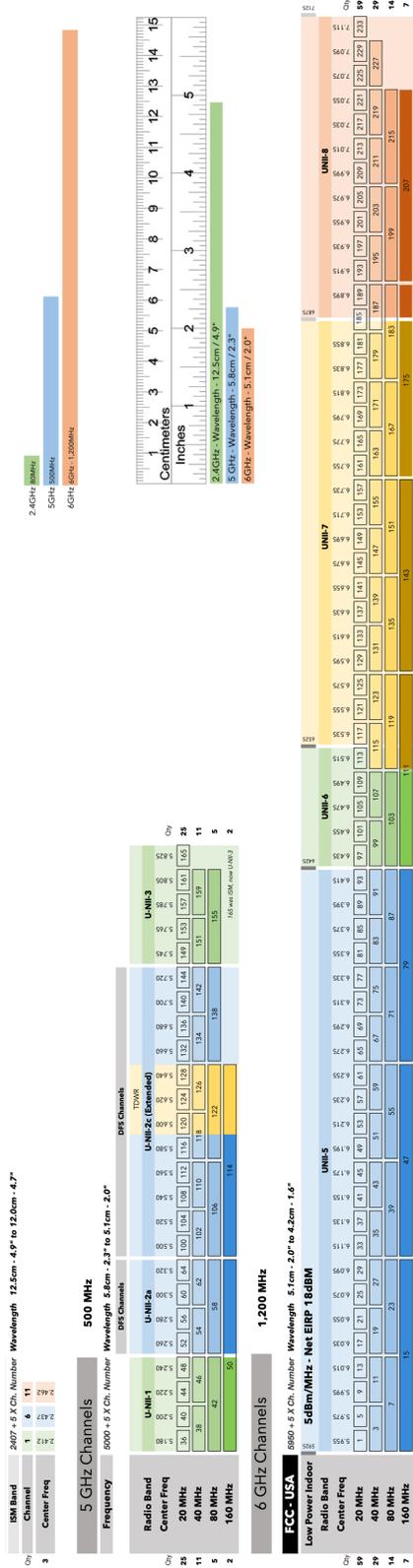


Figure 1.2: Federal Communication Commission (FCC) spectrum allocation chart for UNII-3 bands where upper 30 MHz is reserved for C-V2X direct operations and lower 45 MHz can be used for unlicensed operations [6].

some guidelines on efficiently utilizing the spectrum without causing problems for primary users (PU). This result can be achieved via spectrum sensing techniques such as energy detection [8], matched filter detection [9], and, cyclostationary feature detection [10]. In recent years, research has shifted to utilizing deep neural networks (DNN) or deep reinforcement learning (DL) for predicting the white space (spectrum opportunity) with better accuracy and low false alarms [11]. However, such channel sensing approaches possess two significant technical challenges that need to be accounted for in connected vehicles environments:

- Highly dynamic vehicular environments leading to network topology and channel conditions, which significantly impacts network reliability and efficiency. In particular, the time-varying propagation characteristics of a connected vehicles, environment, such as Doppler Effect, multipath fading channels, and transmission errors on the control messaging need to be considered [12, 13].
- The limited capability for information sharing, due to processing time or bandwidth capacity limitations, makes network organization challenging. These constraints may potentially make the adaptation to the current network conditions ineffective, although information sharing may increase environmental awareness.

In this dissertation, we propose a novel biologically-inspired vehicular dynamic spectrum access (VDSA) framework using bumblebee behavioral models to enable vehicular networks to reliably access frequency bandwidth resources. The adaptive behavioral responses of bumblebees are evolved to serve under similar complex and highly time-varying floral conditions wherein the insects try to maximize the nectar reward [14]. In the vehicular domain, the objective function's goal is to find the optimal channel which gives the best throughput and low latency performance. Second, the vehicular environment is analogous to bumblebee habitat, where the channel energy changes in a temporal manner as the nectar in foraging. The proposed channel selection algorithm developed in this work can rapidly and adaptively respond to changes in the multi-channel environment. The algorithm can be integrated in the MAC layer of both 802.11p and C-V2X technology, with their performance compared against the baseline random access.

1.2 State-of-the-Art

There have been several practical approaches proposed in the open literature that leverage distributed optimization techniques employed by natural model systems, such as ant colonies, honeybees, and other insects, which all perform swarm optimization of available resources [15]. In contrast to the individual-based bumblebee system, honeybees rely on the ‘scout-recruit’ method where one individual (scout) communicates resource quality information to many individuals (recruits) [16]. If the quality of the resource decreases, then the recruits are informed of a better food source by the scout when they return to the hive. The scouting process and the need for worker bees to return to the hive in order to be informed about a better food source is associated with time costs not present in the bumblebee system (where individuals sample resource alternatives and then specialize on the best one). Thus, the honeybee system is not an efficient resource exploitation mechanism when resources vary rapidly and are unpredictable over time and space. Similarly, ant colony behavior is based on tracking the pheromones that primer ants leave behind [17]. Although ant colonies are very efficient for routing scheduling and organization, this mechanism also cannot deal with the highly time-varying vehicular networking environment [18]. Reinforcement learning [19,20] mechanisms have been presented in the existing literature as an alternative to colony behavior. Genetic algorithms provide a reliable optimization technique, but at the expense of a large computation latency with respect to converging to the optimum value [21]. Partial swarm optimization is a very fast optimization technique since it jointly solves the fitness function based on a multiobjective formulation [22]. However, it is highly dependent on the initial information about the swarm structure, which is not realistic for connected vehicle networks. The aforementioned techniques require that each node within the network is dependent on the social interaction with all other nodes within the network, which is not the case in applications such as connected vehicle networks.

We propose the bumblebee as a more suitable social insect model for studying distribution optimization of channel resources. Unlike ants and honeybees, individual bumblebee foragers acquire information on their own and independently solve optimization problems within the distributed network. Thus, bumblebees do not depend on a centralized infor-

mation system, which can be highly ineffective and unreliable in environments that rapidly change over time and space. For vehicular networks, this description corresponds to rapidly changing environments, where centralized information may be inaccurate or too slow to reflect local changing conditions. Furthermore, vehicles may lose connectivity to a centralized database or other neighboring vehicles under some conditions (*e.g.*, highway, rural area). In such a scenario, any optimization mechanism relying on centralized communication is highly inefficient and poses a major safety concern. Bumblebee foraging behavior utilizes individual decision mechanisms, which can include information about the behavior of others but does not depend on it. Since there is no need to access any centralized system or wait for information from others, the decision and adaptation to change can occur as rapidly as their highly efficient neural processing system allows. Bumblebee framework provides an opportunistic channel access scheme for enabling VDSA-based vehicular communication. In congested environments, this framework improves throughput and reduces latency.

Connected vehicles need to accurately localize other vehicles for smooth traffic flow. Human-operated vehicles will be expected to co-exist along with connected vehicles. GPS will not be sufficient in urban environments either due to unintentional effects like non-line of sight (NLOS) and multipath propagation, or due to intentional GPS spoofing by malicious users. In order to facilitate V2V and get accurate locations of vehicles, we have also proposed a novel opportunistic approach of passive RF localization designed for detecting “phantom car” attacks. The radio emissions from the vehicles such as Bluetooth, TPMS, or cellular signals are exploited to accurately localize the vehicles with a proposed hybrid RSS and TDoA localization scheme. The proposed scheme is also evaluated using a custom-built Python simulator which incorporates a traffic flow model, realistic channel model, trajectory estimation, and hybrid localization.

1.3 Research Contributions

This PhD dissertation presents three novel contributions for enhancing vehicular networking using bumblebee-based dynamic spectrum access and signals of opportunity.

The first research contribution provides a framework for opportunistic channel selec-

tion in white spaces for V2V communication. The vehicular dynamic spectrum access (VDSA) framework leverages under-utilized spectrum elsewhere, *e.g.*, DTV bands, for improving connectivity in congested environments. Figure 1.3 illustrates the wide array of floral species which bumblebees need to sample for nectar reward estimation. In vehicular communication, these flowers are represented as channels, and the nectar reward defines the vehicle channel quality over time. The algorithm is enhanced by adding simple memory model which helps in reducing the sampling overhead and also minimizes frequent channel switching which drastically affects the throughput. The algorithm is able to adapt to time-varying interference environment and converge to optimal channels in the window of coherence time.



Figure 1.3: Artificial mixed floral environment used to investigate memory-based decision-making processes in bumblebees. Each color represents a species (channel) [23].

The second research contribution details the platooning-based VDSA framework. The initial bumblebee algorithm is modified for improving data connectivity in platooning au-

tomotive operations. Platooning requires seamless connectivity between master and slave vehicles as the inter-vehicle distance is short. Loss of basic safety messages due to collisions or high latency incurred due to radio environment map (REM)-based centralized schemes can lead to traffic accidents. In this dissertation, we have proposed a reliable Bumblebee-based VDSA framework which leverages a non-uniform sampling heuristic to support optimal channel selections in a platoon operating environment.

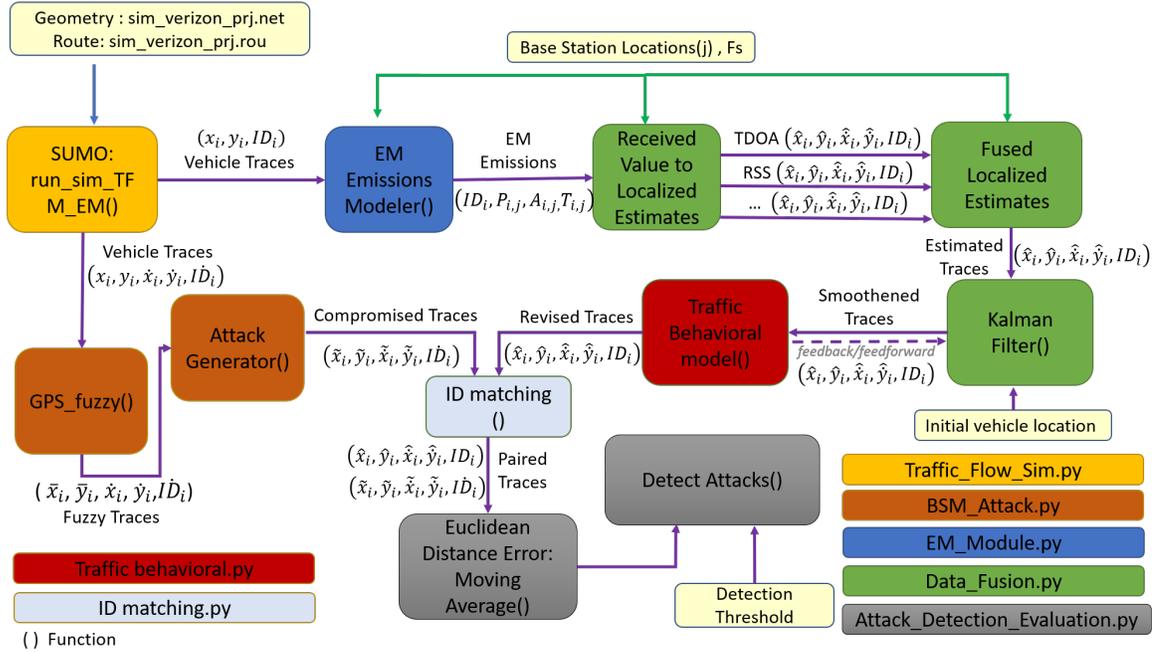


Figure 1.4: Illustration of the proposed overall phantom car attack detection framework [24].

The third research contribution describes a novel localization framework (Figure 1.4) which leverages wireless emissions from other vehicles in an opportunistic manner to exploit their positions. This project was a large collaborative effort, with the overall framework consisting of traffic flow modeling, realistic RF emissions, and smoothing of trajectory estimation using Kalman filtering. My contribution was integrating a realistic 3GPP extended vehicular A (EVA) channel model in the overall custom-built Python framework. The channel model was employed in order to create a non-ideal multipath for evaluating the hybrid RSS-TDoA localization scheme.

The detailed contributions of the research are as follows:

- **Bumblebee Algorithm-based DSA (Chapter 3):** The proposed bumblebee-inspired algorithm with decision mechanism within a vehicular dynamic spectrum access framework is explored. Channel reward levels stored in memory are weighed against switch costs to decide whether to stay on the current channel or move to a different channel. Channel reward information is frequently updated in memory through periodic sampling, which provide vehicles with a more accurate estimate of the degree to which channels differ in their quality for a given environment. Results show that a large increase in channel selection performance was obtained for sparse highway and urban traffic by utilizing our bumblebee-based algorithm enabled with memory. Two simple memory structures using block sampling of the environment followed by averaging energy values or selecting the maximum value, were designed for channel selection [23, 25–27].
- **Memory-Enabled Bumblebee Algorithm (Chapter 4):** The framework for a memory enabled bumblebee foraging algorithm is presented and applied to vehicular platoon communications. An optimized unequal sampling allocation heuristic is proposed to estimate the CBR with sufficiently high accuracy. The results showed that unequal sampling instant allocation approach outperforms the equal sampling allocation scheme with the proposed sub-optimal allocation heuristic. We also integrated two memory models into the bumblebee foraging algorithm to leverage available memory, which boosts the probability of the best channel selection. Sliding window average and exponentially weighted moving average memory schemes are employed, and their performance is compared against the memoryless model. Different memory lengths and forgetting factors are used in SWA and EWMA schemes, respectively. [28, 29].
- **Phantom Car Attack Detection (Chapter 5):** A comprehensive Python-based simulator framework is implemented in order to evaluate and test custom localization methods and communication protocols. A hybrid RSS-TDoA localization algorithm is also proposed to exploit signals of opportunity from non-subscriber vehicles. Radio frequency emissions from those vehicles can be detected by subscriber vehicles whose positions are known perfectly. The emissions captured by different subscriber vehicles

can be sent to a fusion center for accurate location and trajectory estimation. The proposed approach was also validated using a hardware implementation. A small-scale field experiment was conducted using RTL-SDR and Pluto SDR to validate the computer simulation results [24] in WPI Gateway park.

1.4 Dissertation Organization

The dissertation is organized as follows. Chapter 2 describes the bumblebee behavioral model, which is based on foraging theory, and gives background on dynamic spectrum access. In Chapter 3, a novel bumblebee-based dynamic spectrum access algorithm is proposed for vehicular communication. Firstly, a performance bound for the proposed algorithm is evaluated using queuing theory. Next, the algorithm is evaluated using a V2V system-level simulator called GEMV² and compared against a baseline model. Finally, the algorithm is implemented on a software-defined radio (SDR) platform known as Pluto [30] in an *ad-hoc* manner to demonstrate the performance.

In Chapter 4, the bumblebee algorithm is enhanced further by utilizing memory to avoid frequent channel switchings. The channel sampling instants are allocated using a heuristic in a non-uniform manner to optimize the sampling resources. Two different types of memory models are employed and are compared against the memoryless model. In the first part, focus is on a proposed heuristic to accurately estimate the channel busy ratio for opportunistic channels. Next, the two memory models employed are described in detailed. They are evaluated using an analytical simulation framework and compared against the memoryless baseline algorithm. Finally, the full memory-based bumblebee algorithm with optimal channel sampling heuristic is evaluated using a *C++* system-level simulator for platoon communication [28, 29].

Chapter 5 discusses the identification and localization of adversarial vehicles using opportunistic radio frequency (RF) emissions. First, a custom simulation framework is described in detail, which is implemented using the traces from *sumo* for vehicular trajectory and *3GPP* extended vehicular channel model. The hardware test-bed was also developed using inexpensive RTL-SDR radios acting as the receivers, and Pluto SDR was employed as source

emitter. The localization estimates were computed during post-processing of the raw I/Q samples [24].

Finally, the dissertation is concluded with Chapter 6. The main results are summarized, and several avenues for future work are discussed.

1.5 List of Related Peer-Reviewed Publications

The following publications, which resulted from this dissertation’s work, are published in high-impact peer-reviewed venues.

List of Related Peer-Reviewed Journal Papers

1. **Kuldeep S. Gill**, Kevin N. Heath, Robert J. Gegeer, Elizabeth F. Ryder and Alexander M. Wyglinski, “Memory Matters: Bumblebee Behavioural Models for Vehicle to Vehicle Communications”, in *IEEE Access*, vol. 6, pp. 25437-25447, 2018.
2. **Kuldeep S. Gill**, Pawel Kryszkiewicz, Pawel Sroka, Adrian Kliks, Alexander M. Wyglinski, “Memory Enabled Bumblebee-based Dynamic Spectrum Access for Platooning Environments”, *IEEE Transactions on Vehicular Technology*, under review, Oct 1st, 2021.
3. Alexander M. Wyglinski, Thanuka Wickramaratne, Danjue Chen, Nicholas J. Kirsch, **Kuldeep S. Gill**, Taru Jain, Varun Garg, Tienan Li, Shuva Paul, and Xi Zhang, “Phantom Car Attack Detection Via Passive Opportunistic RF Localization”, in progress.

List of Related Peer-Reviewed Conference Papers

1. **Kuldeep S. Gill**, Kevin N. Heath, Robert J. Gegeer, Elizabeth F. Ryder and Alexander M. Wyglinski, “On The Capacity Bounds For Bumblebee-Inspired Connected Vehicles Networks Via Queuing Theory” in *IEEE 87th Vehicular Technology Conference (VTC Spring)*, Porto, Portugal, 2018, pp. 1-6.

2. **Kuldeep S. Gill**, Kevin N. Heath, Robert J. Gegeer, Elizabeth F. Ryder and Alexander M. Wyglinski, “Experimental Test-Bed For Bumblebee-Inspired Channel Selection in an Ad-hoc Network”, in IEEE 88th Vehicular Technology Conference (VTC Fall), Chicago, USA, 2018, pp. 1-5.
3. **Kuldeep S. Gill**, Kevin N. Heath, Robert J. Gegeer, Elizabeth F. Ryder and Alexander M. Wyglinski, “Bumblebee-Inspired C-V2X Dynamic Spectrum Access Testbed Using OpenAirInterface”, in IEEE 91st Vehicular Technology Conference (VTC Spring), Atwerp, Belgium, 2020, pp. 1-5.
4. Pawe Sroka, Pawe Kryszkiewicz, Micha Sybis, Adrian Kliks, **Kuldeep S. Gill**, Alexander Wyglinski, “Distributed Vehicular Dynamic Spectrum Access for Platooning Environments”, in IEEE 91st Vehicular Technology Conference (VTC Spring), Atwerp, Belgium, 2020, pp. 1-5.

List of Other Peer-Reviewed Publications and Patents

Some miscellaneous publications and patents are listed here. They are not related to the primary work, but were still important for this dissertation as they enhanced my knowledge for the subject matter.

1. Wenxun Qiu, VA Vutha, **Kuldeep Gill**, Boon Loong Ng, Choi Junsu, Sungchul Park, “Radar based pattern code identification”, US Patent US20200319301A1, filed June 2020.
2. Wenxun Qiu, Abhishek Sehgal, **Kuldeep Gill**, Va Vutha, Boon Loong Ng, “Systems and methods for radar based face authentication anti-spoofing”, US Patent US20200319301A1, filed August 2019.
3. Nivetha Kanthasamy, Ruixiang Du, **Kuldeep S. Gill**, Alexander Wyglinski, Raghendra V. Cowlagi, “Assessment of Positioning Errors on V2V Networks Employing Dual Beamforming”, in IEEE 88th Vehicular Technology Conference (VTC Fall), Chicago, USA, 2018, pp. 1-5.

-
4. J. Legaspi, C. I. Canfield, **Kuldeep S. Gill**, A. M. Wyglinski and S. V. Bhadai, “Integrated Agent-Based Model for Broadband Resource Allocation Analysis,” 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5.

Chapter 2

Bumblebee Behavioral Models and Dynamic Spectrum Access Fundamentals

This chapter provides background knowledge on bumblebee-based distributed optimization for channel selection and principles of Dynamic Spectrum Access (DSA) systems. The chapter starts with the introduction of the bumblebee algorithm, and provides in-depth information on the channel selection metrics employed using the pseudo-code. Foraging theory, which is the foundation for bumblebee-based optimal channel selection, is also discussed to explain the motivation behind the behavioral model. Finally, dynamic spectrum access is explained with its application to platooning.

2.1 Bumblebee Foraging Theory

Bumblebee foraging behavior utilizes individual decision mechanisms which can include information about the behavior of other insects but does not depend on it. This section describes bumblebee foraging theory in detail, which forms the foundation of the bumblebee behavioral model as applied to vehicular communication. Bumblebees provide a robust biological framework for building and implementing cognitive algorithms for DSA in vehicular

networks.

Bumblebees are social insects that form colonies comprised of a single queen and up to several hundred workers. A small subset of workers called “foragers” have the sole task of finding and collecting food for the colony in the form of floral nectar and pollen rewards. Foragers routinely encounter a wide array of flowers with reward levels that rapidly change over time and space (see Figure 1.3). Foragers are not pre-programmed with information on the reward level associated with different flowers. Rather, they learn and remember the reward level and sensory cues (color, odor, shape) associated with each flower species and then decide which ones to visit. Importantly, bumblebee foragers do not depend on “scout” bees such as honeybees or pheromone trails left by other insects such as ants. Consequently, each individual has the capacity to learn, remember, and track changes in floral rewards on its own. This system has evolved to enable maximal reward intake to the colony across complex and highly variable floral conditions.

While searching for flowers containing the greatest reward, foragers implement a number of adaptive behavioral processes [31,32] that are comparable to those processes needed for vehicles to function independently and effectively in a connected network environment. First, foragers evaluate the available flower species and then select the type that yields the greatest reward [33]. Second, foragers track and respond to changes in floral reward levels in a flexible manner. Finally, foragers make floral decisions that maximize the rate of nectar delivery to the colony [34]. For example, the decision on whether or not to switch to a new flower species is based on a trade-off between the rewards gained by visiting a new type and the time costs incurred when switching to that type. Although bumblebees primarily use their personal experiences to make floral decisions, they can also enhance their knowledge of floral environments by gaining information from other foragers. For example, individuals can passively acquire information about reward quality from cuticular hydrocarbon “footprints” left on flowers by previous foragers. Low hydrocarbon levels signal high likelihood of reward, and high hydrocarbon levels signal low likelihood of reward [35]. In this way, individual bumblebee foragers can use the experiences of others to increase their efficiency of flower selection by minimizing the amount of time spent on empty flowers. By incorporating this agent-based approach learned from empirical studies of forager behavior, we

greatly accelerate the subsequent development and implementation of cognitive algorithms for optimal channel selection by vehicles in connected network environments.

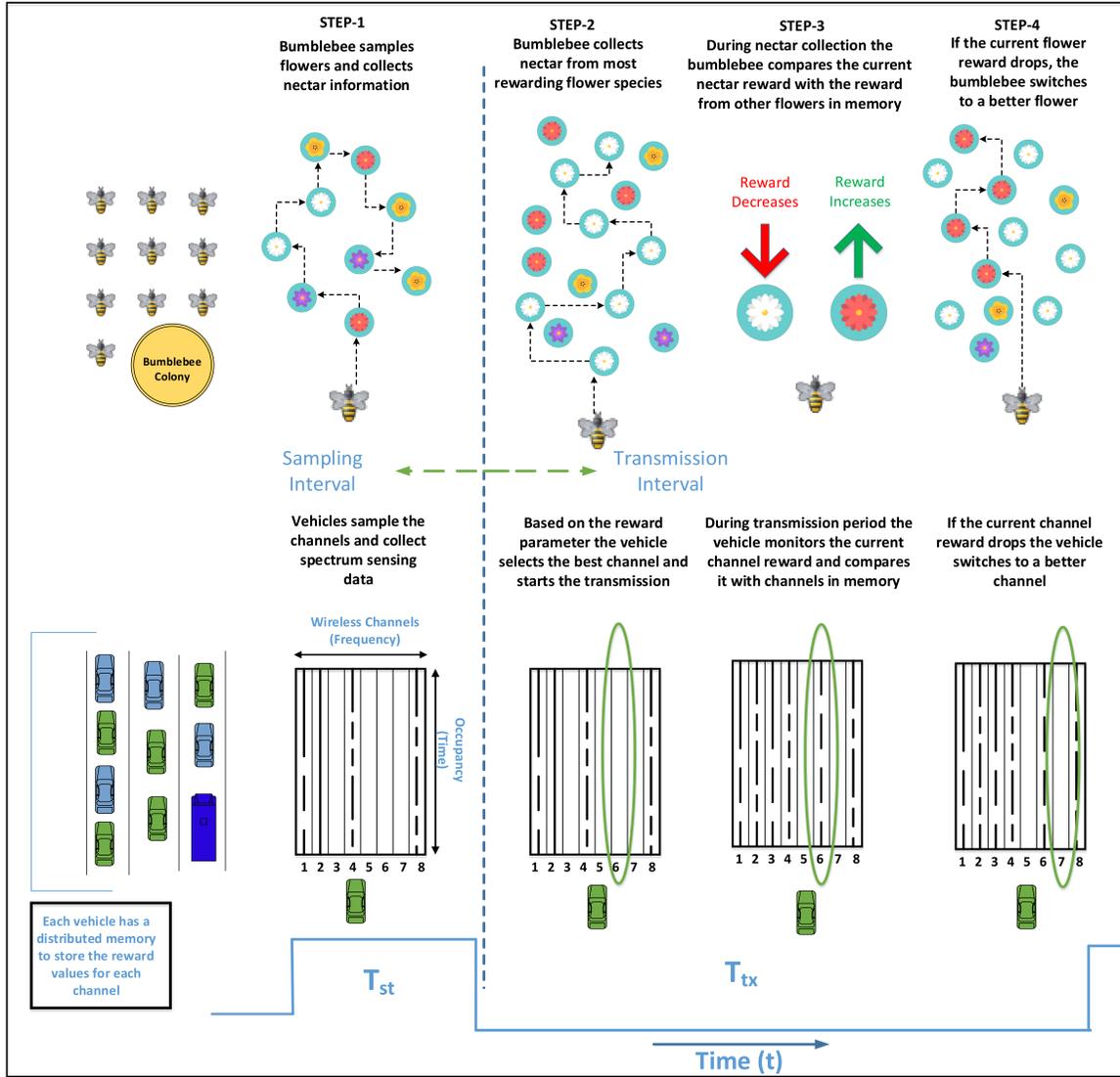


Figure 2.1: The memory-based channel selection algorithm from bee (top) and vehicle (bottom) perspectives. Like each bumblebee, each vehicle is equipped with memory to store channel (floral) reward information, which is then used to select the channel (floral species) with the highest reward quality out of those available. Vehicles (bumblebees) alternate between sampling (T_{st}) and transmission (T_{tx}) periods to track changes in a time-varying noisy resource environment [23].

2.1.1 Translation Between the Two Worlds

Matching the terminology between bumblebee foraging and vehicular communications is the first step in applying bumblebee behavior to the vehicular optimization problem (Table 2.1). In-band interference is an unwanted phenomenon in the channel bandwidth. The equivalent of this phenomenon for bees is the presence of other bees as competitors on a particular species of flower. The presence of hydrocarbon secreted by bees correlates with reduction in nectar reward due to the foraging of other bees. Inter-channel interference is a form of interference produced by channels that is similar to the bee competitors' effects on nectar level estimations of alternative floral species. Channel energy levels are a key feature with respect to channel access and is similar to nectar levels of the flowers for the foraging bees. However, there is an inverse relationship between these two features. In vehicular communications, it is desired to access the channel with as low energy level as possible since low energy level means there is no other user in the channel, which also means low noise levels and interference effects. On the other hand, bumblebees aim to access the flower with the highest nectar (energy) levels since it means there are few other bees competing for the same floral species, thereby increasing their rate of nectar return to the colony.

Table 2.1: Several Definitions in Connected Vehicular Communications and Their Equivalent Definitions in Bumblebees [23]

| Vehicles | Bumblebees |
|---------------------------------------|---|
| In-band Interference | Bees foraging on the same floral species |
| Out-of-Band Interference | Bees foraging on alternative floral species |
| Minimum Channel Energy Level | Maximum Nectar Level Per Floral Species |
| Computation/Process Time | Handling/Searching Time |
| Latency vs. Reliability | Sampling Frequency vs. Choice Accuracy |
| Switching Cost/ Time between channels | Switching Cost/Time between Floral Species |
| Channel activity over time | Floral Species occupancy over time |
| Channel-user distribution | Bee distribution across floral species |

Computation/process time of the algorithms used by connected vehicles corresponds

to flower handling and search time of bees. Many algorithms have been proposed for connected vehicles that provide an optimal channel access scheme [36, 37]. However, if an algorithm gives the result after a longer time interval than the coherence time, the environment conditions change and the output of the algorithm is no longer appropriate. Focusing on vehicular ad-hoc networks (VANETs), there has been extensive research on the application of optimization techniques based on natural models. For example, in [38], artificial bee colony (ABC) optimization algorithm has been applied to VANETs in order to overcome connectivity and signal fading issues. The authors in [39] employed modified GA algorithms to optimize white space utilization in VANETs. The optimization of routing protocol performance based on both particle swarm optimization (PSO) and ant-colony optimization (ACO) in VANET is discussed in [40]. However, quantitative analyses on the bounds of these approaches have been noticeably absent. Most of the research into bio-inspired optimization techniques has been presented with a problem-specific approach. To the best of the authors knowledge, no one has yet investigated quantitative performance bounds at the vehicular level.

We leverage the concept of queuing theory in order to conduct a performance bound analysis of bumblebee-inspired distributed optimization operation in vehicle-to-vehicle (V2V) environments. The work is based on the techniques presented in [7], where the authors assume the channel is quasi-static whereas in this work we conduct the performance analysis on a time-variant channel and further explore correlated effects between channels. Unlike ants and honeybees, individual bumblebee foragers can acquire information on their own and independently solve optimization problems within a distributed network. Consequently, bumblebees do not need to depend on a centralized information system, which can be highly ineffective and unreliable in environments that rapidly change over time and space [41]. For vehicular networks, this description corresponds to rapidly changing urban environments, where centralized information may be inaccurate or too slow to reflect local changing conditions. Furthermore, a vehicle in a rural area may lose connectivity with a centralized database or other neighboring vehicles. In such a scenario, any optimization mechanism relying on this form of communication may potentially not work properly. In a honeybee colony, the scout bee finds a food source, and then returns to the hive to perform a dance

to communicate information on the food source to recruit bees [42]. If the nectar level decreases at the food source, the worker bees are informed of a better food source when they all return to the hive and decode the dance. The scouting process and the need for worker bees to return to the hive in order to be informed about a better food source can incur a significant delay penalty with respect to accessing a better food source. Thus, the honeybee food selection process would be inefficient when the quality of each flower type changes rapidly and unpredictably over time. Similarly, ant colony behavior is based on the tracking of pheromones that primer ants have deposited [43]. Although ant colonies are very efficient with respect to routing, scheduling, and organization, this mechanism also cannot deal with highly time-varying environments.

As an alternative to colony behavior, reinforcement learning (RL) mechanisms have been presented in the existing literature [44, 45]. Genetic algorithms provide a reliable optimization technique but at the expense of a large computational latency with respect to the convergence to the optimum value [46]. Partial swarm optimization is a very fast optimization technique since it jointly solves the fitness function based on a multi-objective formulation [47]. However, it is highly dependent on the initial information about the swarm structure, which is not realistic for connected vehicle networks. Bumblebee foraging behavior is mainly based on individual decision mechanisms, hence it is well-suited for applications where decision-making is performed independently. Since there is no need to access any centralized system or wait for information from others, the decision and adaptation to change can occur as rapidly as their highly efficient neural processing system allows.

Similarly, if bumblebees sample available species in a varying floral environment too infrequently, they may be delayed in switching to a more rewarding floral species should it become available. In other words, the tradeoff between latency versus reliability is mirrored with respect to the bumblebees in terms of sampling frequency versus choice accuracy. Switching cost/time between channels should be considered although switching operations provide the access to the channel with higher quality. Similarly, bumblebees switch to the floral species with the highest nectar returns in order to gain more energy. However, they can also incur a significant time cost when switching from one floral species to an-

other. Channel activity over time helps to understand channel behavior as well as design a prediction mechanism. Similarly, bees alter foraging decisions based on the number of bees within and across floral species, often showing an *Ideal Free Distribution* [48]. We detail specific components of the bumblebee system leveraged to create a vehicular channel selection algorithm below.

2.2 Bumblebee-Based Channel Selection

The technical challenges resulting from the severe dynamic characteristics of the vehicular networking environment makes it difficult to employ DSA via conventional approaches [49]. This is especially true when channel sensing parameters such as the noise floor, propagation fading, and interference are time-varying. We explore how a DSA framework for distributed Vehicle-to-Vehicle (V2V) networks can be based on adaptive behavioral responses of animals that must survive under similar complex and highly varying resource conditions in their natural habitat. In particular, we focus on bumblebee foragers since they have evolved cognitive abilities that enable them to make adaptive behavioral decisions under such conditions based on individually acquired information. Using the bumblebee model, an efficient channel sensing and selection system has been developed that can rapidly and adaptively respond to changes in multichannel environments. The key component of this system is *channel memory*, which enables more accurate estimates of available channel quality to determine the optimum point to switch to better quality channel. Mapping of stored information on channel quality using “Mean” strategy (*i.e.*, past information on each channel gained through sampling is averaged and then used to make the decision on the channel with the best quality).

2.2.1 General Overview

Section 2.2 describes the bumblebee-based optimal channel selection in details. We start by giving the general overview of the bumblebee algorithm and discuss how bees sample the flora species and optimizes the nectar reward. Figure 2.1 describes the mapping of bumblebee foraging on different floral species and vehicles sampling different channels. The

bumblebee algorithm operates in two modes.

1. *Sampling*: During the sampling interval, the radio collects stats like signal to noise interference ratio (SINR) and channel busy ratio (CBR), which are fed to the channel switching decision system.
2. *Transmission*: During the transmission interval, the vehicle initiates channel switch announcement (CSA) and starts transmitting packets, while simultaneously monitoring the current channel. If the SINR drops below a certain threshold, it switches to a new channel based on the quality.

To leverage the potential of bumblebee foraging behavior [50] in connected vehicle environments, we translated evolutionarily optimized [51] memory-mediated bumblebee foraging strategies to a DSA decisionmaking algorithm for connected vehicle networks. One of the major challenges faced by vehicles in a connected network environment is that they must accurately estimate channel quality from power levels that significantly vary over both time and space. The incorporation of an individual memory component into the algorithm design overcomes this challenge by enabling individual vehicles to derive estimates of local channel quality, which could then be shared throughout the vehicular network. Equipping vehicles with an unlimited memory capacity would provide the most accurate estimate of channel quality. However, unlimited memory would also generate additional costs, *e.g.*, information processing speed and time lag in reacting to environmental changes. Thus, determination of an optimal decision-making strategy requires consideration of memory capacity, dynamics, and associated costs. Bumblebees face identical constraints in choosing the optimal foraging strategy in variable floral environments. They first start by sampling different floral species and collect nectar reward information. The nectar reward values per floral species are stored in their memory during, sampling interval, and the flower type with highest nectar reward is selected. The bumblebees then start collecting nectar and take it back to the hive. If the current floral species nectar reward drops below some threshold, they switch to a new floral species. The threshold is determined by the next best floral species and the switching cost, since switching from one floral species to another can incur search and handle time penalties.

We have modeled this bumblebee behavior to select optimal channels for V2V communication. We initialize the bumblebee algorithm by defining a set of opportunistic channels in the dedicated memory of the vehicle [23]. During the initialization phase channel quality is not available, so a random channel is selected for V2V. During the sampling interval, the vehicles sample the channels and collect spectrum sensing data. The energy values collected are then converted to the channel rewards and stored in the memory. In the transmission interval, each vehicle selects the channel with the best reward gain and starts the transmission. During transmission, the vehicle keeps monitoring the channel reward level. If the level drops below a certain range, the vehicle switches to another channel provided the switching cost is not too high. After the transmission interval, the vehicle initiates another sampling interval where new sampling values are stored in the memory, and if a better channel is available the vehicle switches to the new channel.

2.2.2 Channel Occupancy Characteristics and Theory

The channel occupancy ratio (COR) quantifies the channel usage across time. For example, channel A has a COR of 90%, it means the channel was occupied most of the time and can be avoided for spectrum access. If channels with high COR are selected, they can lead to interference and packet collisions, leading to low throughput and worse latency performance. In Figures 2.2, 2.3, and 2.4, we see the channel spectral map with respect to time. Figure 2.2 describes the broadband personal communications service (PCS) and it is a FCC-licensed frequency band near 1.9 GHz. This band currently support frequency division duplexing (FDD) and is used by cellular phone service providers to enhance LTE coverage in the congested regions. In Figure 2.3, a channel characterization of GSM band was performed in order to assess its performance for DSA. These bands will be reallocated for narrowband operations or NB-IOT, which was introduced in LTE Release 13 and can support low throughput internet of things (IoT) devices.

Channel characteristics change stochastically with interference from incumbent or other secondary users. Our memory-based bumblebee algorithm can help improve channel selection performance by making switching decisions using past energy samples from the channels. We conducted the channel characterization study for Personal Communications

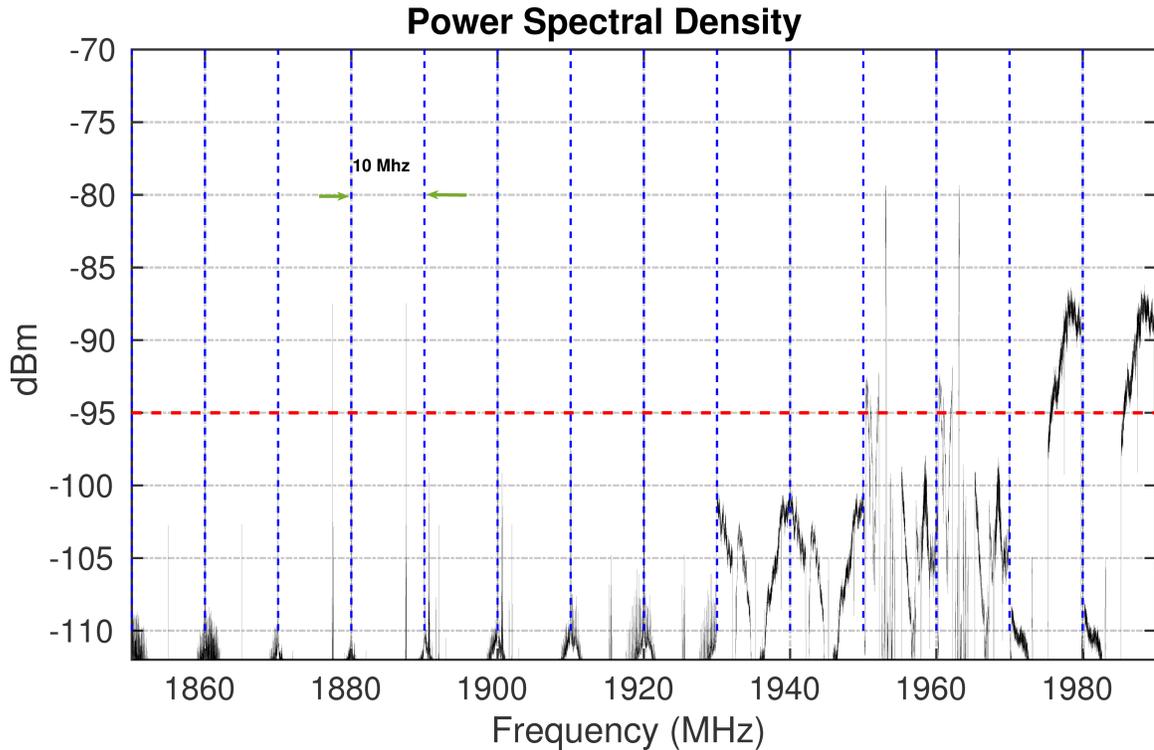


Figure 2.2: Fixed time snapshot of power spectral density of PCS band shows the vacant and occupied channels in that band [23].

Service (PCS) [52], Global System for Mobile Communications (GSM) [53], and lower Long Term Evolution (LTE) [54] bands using USRP N210 [55] software-defined radio in an indoor laboratory environment. The Figure 2.5, 2.3, 2.4 show the snapshot of the power spectral density of PCS, GSM and lower LTE band respectively.

Figure 2.5 shows the occupancy of PCS band through a 30 minutes period. The bands from 1950 – 1980 have very high occupancy around 100%, while the other bands are very underutilized or completely vacant. These bands can be used for vehicular communication using DSA, and their occupancy can be stored in the memory to help avoid the occupied channels during busy intervals (when channels are being used). Without loss of generality, we will employ digital television (DTV) spectrum for this DSA-based vehicular architecture, known as *vehicular dynamic spectrum access* or VDSA, since the primary users of this band are relatively stable [7] when compared to other wireless frequency bands. Vehicles within the vicinity individually detect available channels for unlicensed users. The spectrum is

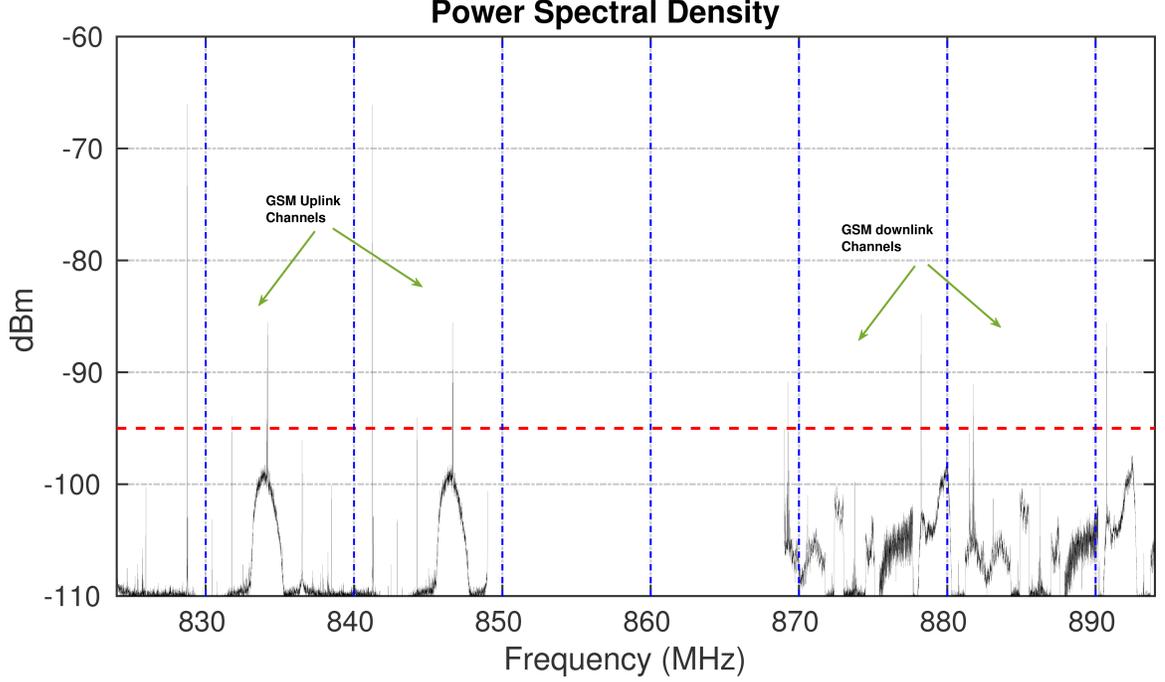


Figure 2.3: Fixed Time Snapshot of Power spectral density of GSM band shows the vacant and occupied channels in that band [23].

sensed based on a mechanism that detects energy levels for each channel. The channel model considers all entities specific to a vehicular environment such as multipath fading, Doppler shift, and, scattering, which can be mathematically expressed as [56]:

$$h(\tau, t) = \sum_{k=0}^{P-1} h_k(t) e^{-j2\pi f_c \tau_k(t)} \delta[\tau - \tau_k(t)], \quad (2.1)$$

where τ is the path delay, t is time variable, $h_k(t)$ is channel envelope, and f_c is the carrier frequency. Using the channel impulse response, the detection problems can be formulated as an M -ary hypothesis test; in this case the spectrum sensing performs a binary hypothesis test as follows:

$$\begin{cases} \mathbf{H}_0 : y(t) = n_r, \\ \mathbf{H}_1 : y(t) = \int_{-\infty}^{\infty} h(\tau, t) x(t - \tau) dx + n_r \end{cases} \quad (2.2)$$

Once the SU vehicles occupy the available channel, \mathbf{H}_0 , they need to periodically check whether they may switch to a better channel. The key parameter associated with the channel switching decision is the switching cost, which determines whether the vehicles

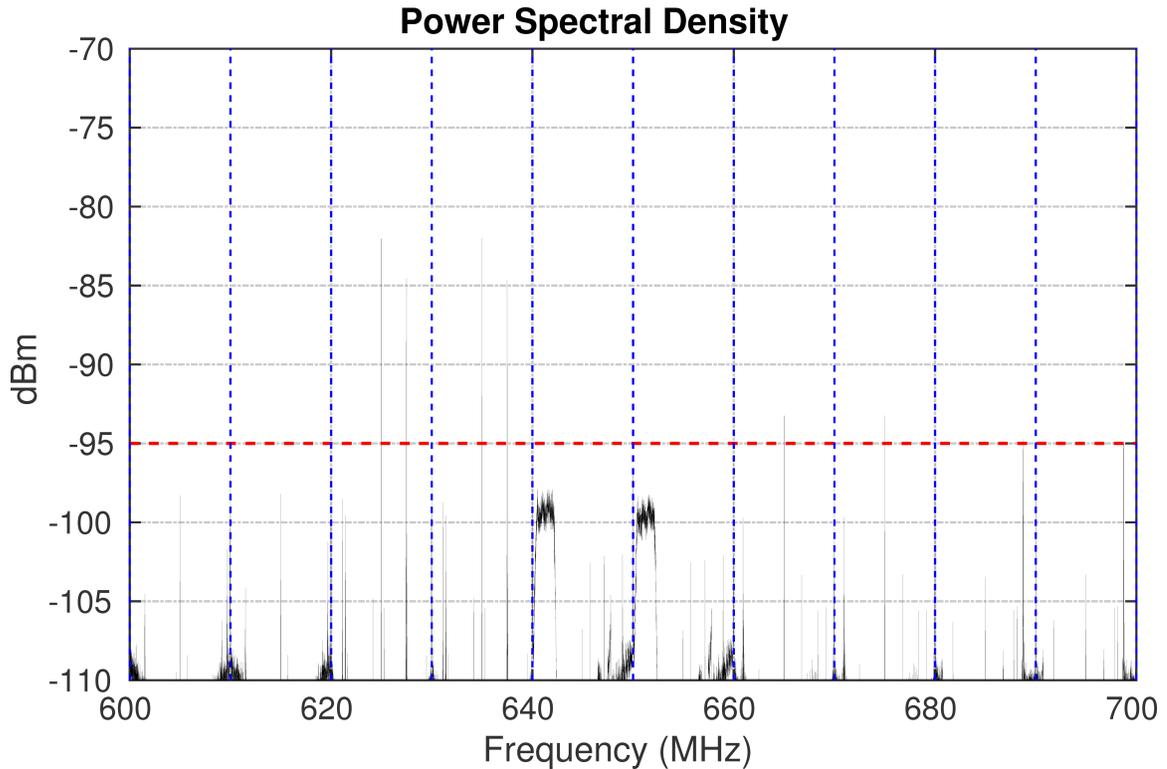


Figure 2.4: Fixed Time Snapshot of Power spectral density of lower LTE band shows the vacant channels in the band [23].

should continue to use the same channel or search for another. For example, if a vehicle switches from channel A to channel B for better reward but does not take into consideration the switching cost, it may actually have less throughput because, due to the lag in switching, the vehicle didn't use any channel for the switching duration. Also the channel it switched to might only be slightly better than status quo. To counter this issue, we take switching cost into consideration when maximizing the channel reward for vehicles. We cannot use a fixed value for switching cost, because a fixed value does not work for a highly dynamic connected vehicle environment. For example, the noise level may be low while the vehicle drives across a highway during one time step, but then be high when the vehicle enters an urban area during the next time step. In this example, switching to another channel may not be the best decision since all of the channels could potentially be noisy.

We can overcome this issue by employing our bumblebee based algorithm which is suitable for a highly time-variant channel environment. For the bumblebee algorithm we first

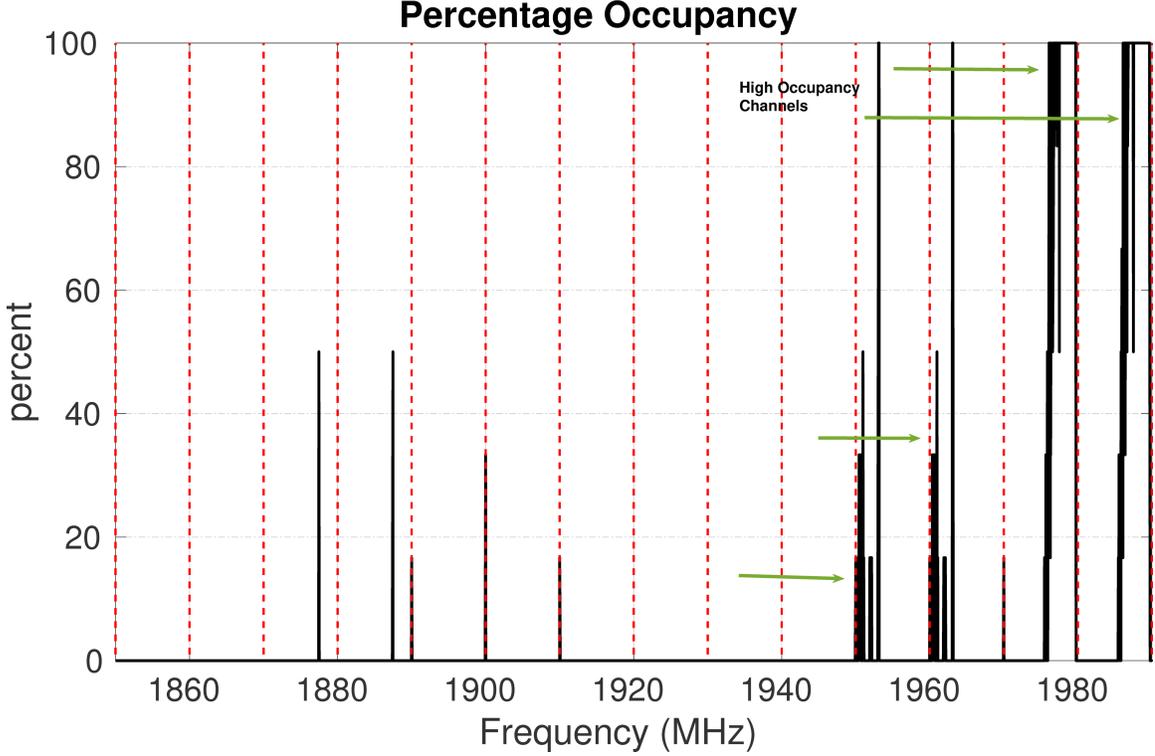


Figure 2.5: Percentage Occupancy of the PCS band during 30 minutes period from 1850–1990 MHz bandwidth [23].

derive the channel reward function $r(t)$ using the energy values of the channels. The channel reward function $r(t)$ is given by:

$$r(t) = |\min(\hat{E})| - |\hat{E}|, \quad (2.3)$$

where $|\min(\hat{E})|$ is the noise floor of the vehicular radio and $|\hat{E}|$ is the energy value of the channels used in DSA. Thus, the higher the channel reward better the channel quality. The channel reward function can be made more sophisticated to depend on the radio characteristics and environment. Eq. (2.4) describe the memoryless bumblebee algorithm:

$$\text{Switching Decision} = \begin{cases} r_c \leq (r_n - s_n), & \text{“Switch”} \\ \text{else,} & \text{“Stay”} \end{cases} \quad (2.4)$$

where r_c is the current channel reward, r_n is the new channel reward and s is the switch cost for the new channel. The switching cost s in this work is described in terms of reward

(% of reward value r_c) to reduce the complexity. The switching cost will vary depending on the cognitive radio characteristics and channel environment.

2.3 Vehicular Dynamic Spectrum Access

Focusing on vehicular ad-hoc networks (VANETs), there has been extensive research on the application of channel selection techniques based on natural models. For example, in [38], artificial bee colony (ABC) optimization algorithm has been applied to VANETs in order to overcome connectivity and signal fading issues. The authors in [39] employed modified GA algorithms to optimize white space utilization in VANETs. The optimization of routing protocol performance based on both particle swarm optimization (PSO) and ant-colony optimization (ACO) in VANETs is discussed in [40]. However, quantitative analyses on the bounds of these approaches have been noticeably absent. Most of the research into bio-inspired optimization techniques has been presented with a problem-specific approach. To the best of the authors' knowledge, no one has yet investigated quantitative performance bounds at the vehicular level. In this dissertation, we leverage the concept of queuing theory in order to conduct a performance bound analysis of bumblebee-inspired distributed optimization operation in vehicle-to-vehicle (V2V) environments.

2.3.1 802.11p and C-V2X-based VANETs

Connecting vehicles by leveraging wireless communication and networking solutions has been studied intensively in the past, especially with respect to Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure [57]. IEEE 802.11p Dedicated Short-Range Communications (DSRC)/Wireless Access in Vehicular Environments (WAVE) [58] standard was the first framework designed to meet demands of Vehicular Network (VANET). Although 802.11p was a good starting point, the standard has some obvious shortcomings, including low reliability, hidden node problem, unbounded delay, and sporadic Vehicle-to-Infrastructure (V2I) connectivity [59, 60]. 3GPP Long Term Evolution (LTE) has been proposed to mitigate some of the drawbacks of 802.11p with LTE Release 15, whose main focus is on low latency for vehicular communication [61]. C-V2X, which uses LTE infrastructure for vehicular com-

munication, is now seen as a mainstream contender for future connected cars. The primary driving force for C-V2X is the LTE back-haul network, which can directly be ported for vehicular communication without spending billions on setting up the entirely new infrastructure. Studies are now being conducted to evaluate the feasibility of LTE for vehicular networks.

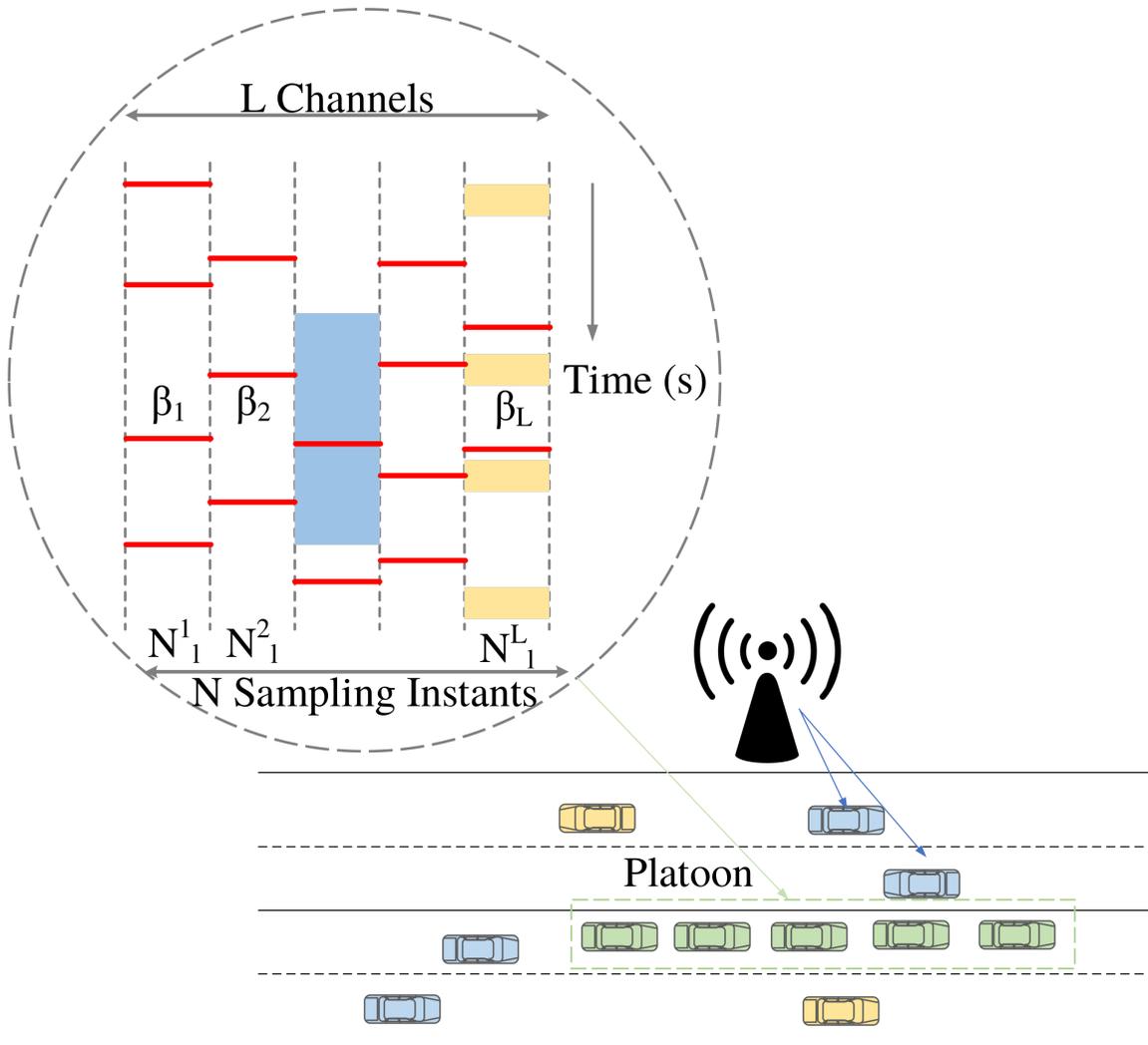


Figure 2.6: CBR β variation with cars arriving and departing the transmission range of the platoon vehicles. Different channels have different β values based on the channel utilization. Non-uniform channel sampling is performed by the platoon in order to find optimal channel from L channels with least β . There are total N sampling periods which are distributed across the channels to optimize the sampling resources [28].

2.3.2 Platooning-based VDSA

Coordinated movement of a vehicle formation, known as *platooning*, is one prospective application of emerging autonomous driving technology, in which a group of self-driving cars and/or trucks forms a convoy, lead by a single vehicle [62]. Figure 2.6 describes the platoon-based V2V communication system (in green). The vehicle in the front acts a master vehicle (an *AP*) and selects the operating channel or primary channel, while other vehicles follow the master as *STA*. For operations of autonomous driving, *e.g.*, using Cooperative Adaptive Cruise Control (CACC) [63], wireless communications can be used to exchange control information within the platoon. Data exchange within the vehicular network, such as within a platoon formation, can be realized with using short-range wireless communication schemes, such as Dedicated Short-Range Communications (DSRC). However, various studies have show that solutions based on the IEEE 802.11p and Wireless Access in Vehicular Environment (WAVE) standards are susceptible to medium congestion when the number of communicating cars is large [64]. An alternative approach to remedy this issue is to offload traffic to other frequency bands, such as underutilized television channels (known as TV White Spaces (TVWS) [65, 66]); this concept is called Vehicular Dynamic Spectrum Access (VDSA) [67]. Fixed locations and relatively stable transmission parameters of Digital Terrestrial Television (DTT) transmitters provide attractive spectrum opportunities in TVWS that can be reused for vehicular communications. In order to employ the VDSA framework for Vehicle-to-Vehicle (V2V) networks, the allocation of channels to the primary users (*i.e.*, DTT transmitters) and the associated power levels for the areas of interest should be known. Optimal allocation be achieved using infrastructure support such as dedicated databases, although in some areas these may be unavailable.

An alternative approach is to apply VDSA in a distributed manner, where each vehicle or group of vehicles (*e.g.*, platoon) selects a transmission channel based on spectrum sensing. This approach has been proposed in [68], which relies on the behavioral model of bumblebee foragers to provide efficient channel sensing and selection. We investigate the idea of partial intra-platoon traffic offloading from the Control Channel (CCH) in congested 5.9 GHz band to TVWS using the VDSA framework. Furthermore, we adopt the bumblebee-based

algorithm from [68] to dynamically select the transmission channel in TVWS in a distributed manner. We assume that all platoon vehicles perform spectrum sensing and share the results with the platoon leader, which is responsible for the selection of the transmission band. Furthermore, we evaluate the proposed distributed VDSA framework using computer simulations aided with realistic DTT signal power obtained from measurements described in [69].

2.4 Chapter Summary

In this chapter, we discussed bumblebee behavioral models and vehicular dynamic spectrum access in detail. Foraging theory, which forms the foundation for bumblebee-based channel selection, also was discussed in detail. We also covered VDSA and the current challenges faced by spectrum sensing technologies, and how the proposed bumblebee algorithm can help in overcoming these issues. We also covered vehicular access technologies such as IEEE 802.11p and C-V2X. In the next chapter, we will examine the proposed bumblebee-based channel access scheme in detail and present the results comparing the scheme against baseline performance.

Chapter 3

Proposed Bumblebee-based VDSA Framework

This chapter presents a novel bumblebee VDSA algorithm which employs signal to interference noise ratio (SINR) to make optimal channel switching decisions. The algorithm employs a SINR-based objective function to compare the current channel performance and, based on the output, either switches to a new channel or stays on the home channel [23]. The framework's performance is evaluated using queuing theory in order to derive the upper and lower bounds [25]. Afterwards, the algorithm is implemented on Pluto SDR for realistic hardware performance statistics [26]. The bumblebee framework is also compared against a centralized radio environment map (REM) by integrating it, within the medium access control layer (MAC) of a system-level C++ V2V simulator in a platooning scenario [29].

3.1 Bumblebee-Based Switching Decision

Our proposed mechanism includes an individual memory structure to store the energy levels of the channel during each energy detection period. In an interference-prone time-varying environment, channel quality varies drastically across the spatial and temporal domains. The proposed approach exploits memory to store the energy values of channels across time and selects optimal channel based on past samples. Without memory, vehi-

cles will frequently change channels based on environmental cues, leading to high switching costs. The time a radio spends switching could have been utilized in transmission, thus lowering throughput and channel utilization. With the help of memory, we can reduce the computation load and eliminate inefficiencies of overzealous switching. Figure 3.1 gives an overview of the proposed approach in a system-level block diagram. It illustrates the bumblebee-inspired distributed optimization algorithm and the steps involved in the process. In the first step, the bumblebee discovers a flower of a specific type (channel), and depending on whether the flower type has been visited before or not the following decision is made. If the flower type is new, we calculate its nectar amount and store it in memory. Otherwise, we update the previous memory regarding the flower-type if the nectar level has changed. Subsequently we compare the nectar level of the current flower-type with the existing nectar rewards in the memory and based on the nectar value we harvest the nectar and exit algorithm or we move onto new flower type. A time-varying stochastic channel also poses the same problems in vehicular environment as a nectar distribution in flowers where bumblebees forage. Because of the strong analogy between the two systems, we expect utilizing bumblebee-inspired algorithms for autonomous vehicle channel selection to be highly efficient. However, before utilizing the bumblebee model for optimization, we need to first establish the bounds expected for efficiency via queuing theory.

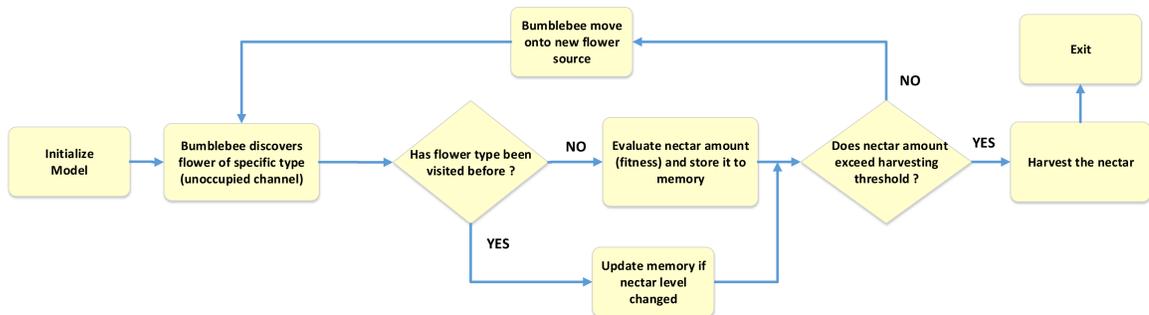


Figure 3.1: Flowchart describing the bumblebee based distributed optimization algorithm [25].

In the framework, we first initialize the model by assigning a random channel to the vehicle. The vehicle (bumblebee) starts sampling the channels (floral species) from white space DSA list and store the channel values in the memory. The channel with the high-

est nectar reward (low interference) is selected and packet transmission is started (nectar collection). In the meantime, the vehicle keeps monitoring the interference level of current channel and if the quality drops it switches to another channel given the switching cost.

The algorithm 1 describes the memory based bumblebee algorithm in detail from implementation perspective in the vehicular communication framework. The parameter M defines the memory length, which depends on the sampling rate of the cognitive radio; N is the total number of samples collected in sampling interval t_s . In this work, we have kept the sampling time $t_s = 200 \text{ ms}$ fixed in all scenarios. It will be an interesting problem to see how varying the sampling and transmission time affects the channel rewards of our bumblebee algorithm, but it is out of the scope of the dissertation. For the simulation we have set the memory length to $5N$, $10N$, $15N$ and $20N$; T is the total simulation time, r is the reward values of the channel, C is the set consisting of all channels used in the simulation, s gives the switching cost and finally V is the total number of vehicles. We initialize our bumblebee model by assigning random channels to the vehicles and setting the memory length to l . We then perform the computation for each time step. Discrete time steps approximate the continuous real-time variation of the channel energies. As explained in Section 2.2, there are two modes in the algorithm: sampling and, transmission. We start each iteration with a sampling interval and we compute the energy values, map it to reward r , and select best channel for each vehicle v_i . In the transmission interval, we start using the channel while simultaneously monitor the current channel reward values. If the current channel reward value drops lower than those in the memory, we switch to the new channel if the switching cost is low enough or else stay on the same channel. The parameter r_{new} is the reward value for the new channel, we consider all the channels in the memory when making the switching decision. For each time step iteration we have sampling and transmission intervals. During the sampling interval we sample all the channels and add new values to the memory. Depending on the memory length, we flush out the old values and keep inserting the new values in First-in-First-Out (FIFO) fashion.

Methods such as genetic algorithm (GA) [70] and particle swarm optimization (PSO) [71] are able to converge to the optimal solution without relying on social interaction. However, the convergence time is much longer than the coherence time. Hence, these algorithms

Algorithm 1 Memory Based Bumblebee Algorithm

```

1: procedure BUMBLEBEEALGORITHM( $M, r, T, C, V, s$ )
2:   Initialize:
       $v_i \in rand\{C\}$ 
       $M = l$ 
3:   for  $t = 1$  to  $T$  do
4:     Sampling Interval :
5:     Compute Energy values  $E \in \{C\}$ 
6:     Map to reward  $r \in \{C\}$ 
7:     Select best  $\{C\}$  at  $t$  for  $v_i$ 
8:     Transmission Interval:
9:     Start the packet transmission
10:    Monitor  $v_i \in \{C\}$  for  $v_i \in V$ 
11:    if  $r_i < (r_{new} - s)$  then
12:      Switch to the new Channel
13:    else
14:      Stay on the same channel
15:    end if
16:  end for
17: end procedure

18: procedure CHANNELREWARD( $E, C, T, V$ )
19:   for  $t = 1$  to  $T$  do
20:     while  $c = 1$  to  $C$  do
21:        $r(t, c) = |\min(\hat{E})| - |\hat{E}|$ 
22:     end while
23:      $V \leftarrow r(t)$ 
24:   end for
25: end procedure

```

cannot be implemented for realistic V2V scenarios. We have analyzed the performance of the proposed bumblebee framework using the adaptive behavioral response mechanism in the GEMV² Vehicle-to-Vehicle (V2V) propagation simulator via MATLAB [72]. GEMV² is a computationally efficient propagation model for V2V communications, which accounts for the surrounding objects. The model considers different V2V link types (LOS, non-LOS due to static objects, non-LOS due to vehicles) depending on the LOS conditions between the transmitter and the receiver in order to deterministically calculate large-scale signal variations [73, 74]. Additionally, GEMV² determines small-scale signal variations stochastically using a simple geometry-based model that takes into account the surrounding static and mobile objects (specifically, their number and size). We use Simulation of Urban Mobility (SUMO) [75] to generate the traffic data since it allows generation of different scenarios such as different environments (*e.g.*, urban, suburban, highway) and traffic densities (*e.g.*, high-density, low-density, changing density). The channel sensing algorithm is performed for DTV frequency band at 700 MHz.

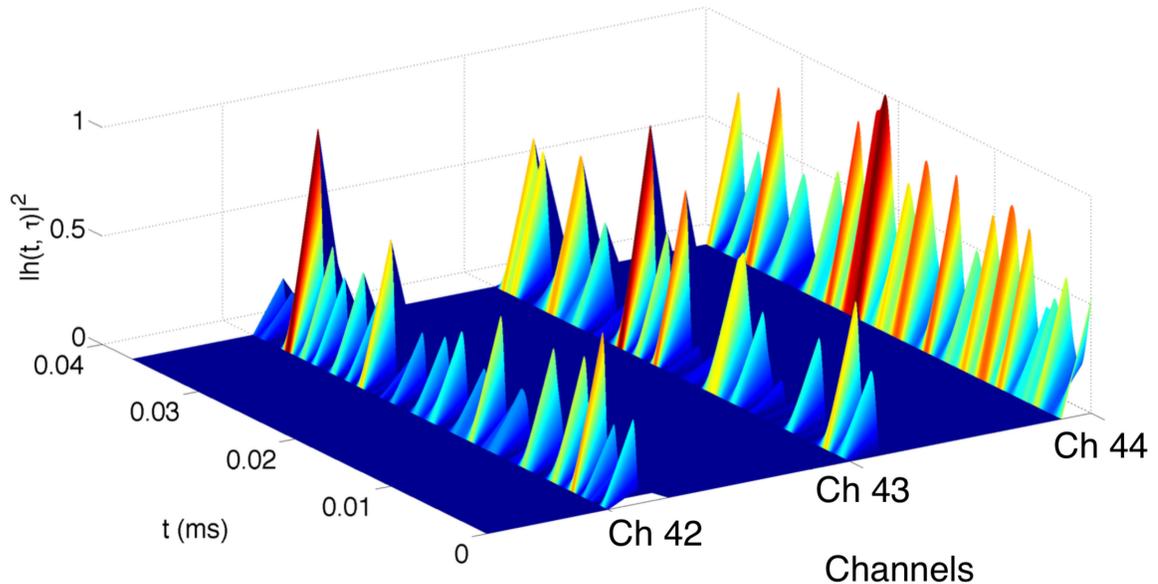


Figure 3.2: Normalized squared magnitude of the channel impulse response: t refers to the time variation on a channel. Three representative channels are visualized to indicate the environment changes over time [23].

The resulting channel characteristic is shown in Figure 3.2. Vehicles switch between

channels in order to find the most powerful available channel of a given time instant. The adaptive behavioral response mechanism is needed to decide whether the channel is worth switching to despite the switching cost. The individual memory helps provide a solid decision on channel switching. For example, a vehicle chooses to be on Channel 42, if available, by using its individual memory since Channel 42 is more powerful based on its long term behavior. Therefore, unwanted switching cost, which is caused by instantaneous decisions, will be avoided.

3.2 Capacity Bounds using Queuing Theory

Queuing theory has often been used to model multiple access schemes or transmission delay in communication systems, and for determining the performance bounds for vehicular communication system [7, 76]. Previous work has been conducted on the application of queuing analysis to cognitive radio systems [77–79]. The authors in [76] proposed a priority virtual queue interface at each unlicensed user in order to abstract the multimedia user’s interactions and proposed a channel selection strategy. The queuing model introduced in [77] analyzes the performance of a cognitive radio link subject to recurrent failures and interruptions. The authors also incorporated network models with single or multiple channels. From the perspective of unlicensed users, the activity of incumbent users can be regarded as a server breakdown. Queuing models with server interruptions have been widely studied in the literature. In [7] queuing theory was used to analyze the performance of VDSA in vacant UHF TV channels from 470 – 698 MHz (channels 14 through 51) where they carried out a measurement campaign along I-90 highway in Massachusetts to collect data for analysis.

For the scenario being addressed in this dissertation, namely high- speed ground vehicles performing dynamic spectrum access across stochastic time-varying vacant TV channels, we propose a queuing model consisting of multiple server and same priority class . This scenario can be formulated as an opportunistic multiple access problem [80]. We assume a transmission range of 500 m for each consumer vehicle, the same as the transmission range defined in the DSRC standard [81]. The vehicular channel connections are treated

as an M/M/S queuing system due to the time-varying nature of the environment. The arrival process is assumed as Poisson with arrival rate λ , and the inter-arrival times are exponentially distributed with mean $1/\lambda$. The service times of the servers are also assumed to be exponentially distributed with mean $1/\mu$. The wireless channels/servers S are assumed to be providing service independently of each other. The first-in first-out (FIFO) queuing discipline is adopted to maintain the same priority for all the vehicles. The traffic intensity ρ which depends on the number of vehicles N , λ , μ , and ,number of servers S is given by:

$$\rho = \frac{N\lambda}{S\mu}. \quad (3.1)$$

The probability that there are no jobs in service in such a system, is given by [80]:

$$P_0 = \left[\sum_{k=0}^{S-1} \frac{(S\rho)^k}{k!} + \frac{(S\rho)^S}{S!(1-\rho)} \right]^{-1}, \quad (3.2)$$

while the probability that all servers are busy, also referred to as Erlang-C P_m is given by:

$$P_m = \frac{(S\rho)^S}{S!(1-\rho)} P_0. \quad (3.3)$$

Thus, the probability of having at least one vacant UHF-TV channel is given by $1 - P_m$. For M/M/S queuing model, the mean waiting time T_W is given by [82]

$$T_W = \frac{\rho}{\lambda(1-\rho)} P_m. \quad (3.4)$$

The expected mean response time T_R , which describes the mean amount of time spent by each job in the queue, is expressed as:

$$T_R = \frac{1}{\mu} + T_W = \frac{1}{\mu} \left(1 + \frac{P_m}{S(1-\rho)} \right). \quad (3.5)$$

For S channels/servers the optimal number of vehicles which can sustain the communication without breaking the connection, with each vehicle requesting data with a packet-rate λ and $1-\lambda$ of the requested packets will be served within time t is given by [83]:

$$N < \frac{\mu t S - \log\left(\frac{P_m}{\gamma}\right)}{\lambda t}. \quad (3.6)$$

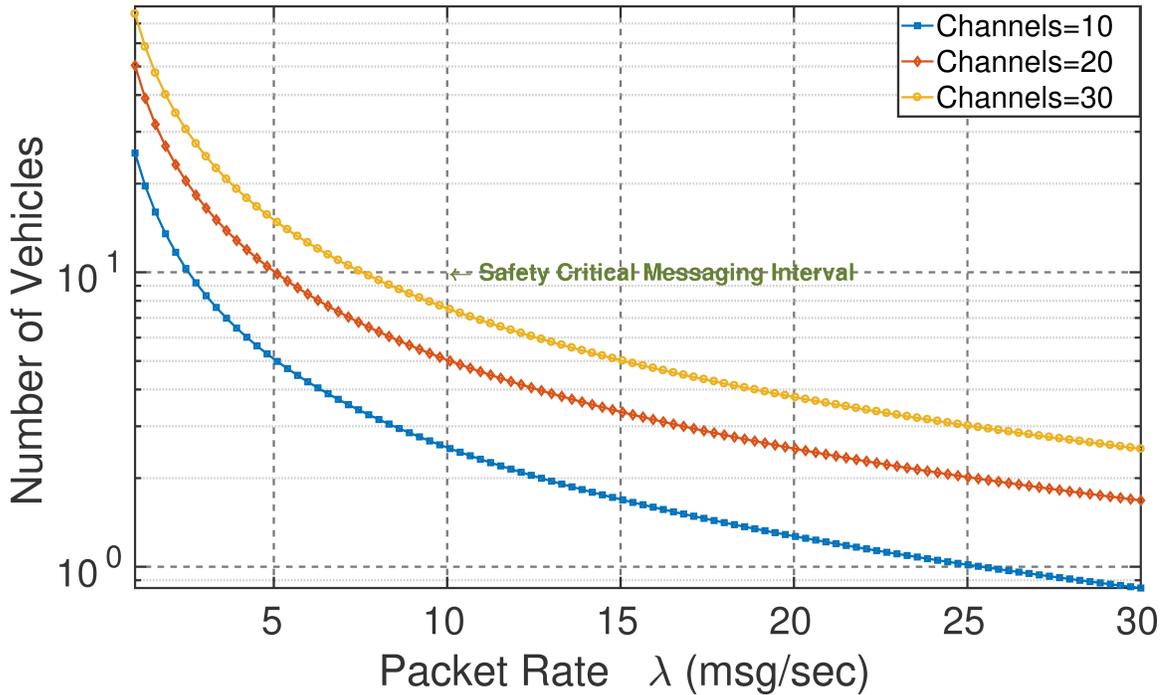


Figure 3.3: Total vehicles supported with optimal communication between the vehicles [25].

Eq. (3.6) shows that the number of vehicles N should be less than the bound to achieve optimum performance. The Figure 3.3 shows the relationship between the number of vehicles and packet rate with fixed number of channels.

3.2.1 Numerical Simulation Results

In Figure 3.4 we compare the switching cost between highway and urban environments. The highway scenario assumes sparse traffic conditions ($12 \text{ vehicles}/\text{km}^2$), while the urban scenario assumes dense traffic conditions ($150 \text{ vehicles}/\text{km}^2$). We assume a random channel switching for both scenarios where vehicles decides to switch to another channel during each time-step without any switching decision. The urban scenario has a very high switching cost due to the highly stochastic nature of the environment. This highly time-variant urban scenario is similar to the bumblebee foraging environment, where the nectar reward (noise in the channel) varies rapidly, requiring the bumblebee to respond quickly. Thus, we expect that applying the bumblebee-based distributed optimization algorithm will greatly reduce switching cost by selecting the channel based on past experiences.

Figure 3.5 shows the average number of cars per square kilometer simulated in GEMV 2 using SUMO. To estimate the bounds in a dense urban environment an average of 100 vehicles are used entering and leaving the downtown Worcester area. The figure shows how the vehicles change each time-step around the area and later gradually exit it. In Figure 3.6, the probability P_m of all servers being busy is shown for the Worcester area. The figure is generated using 10 servers and two different packet rates λ are used. For $\lambda = 10$, we have excellent P_m depending on the number of vehicles. But as the packet rate is increased, the system's performance begins to degrade. Finally, the mean response time is provided in Figure 3.7 for both packet rates, and as expected we see the increase in mean response time for higher packet rate.

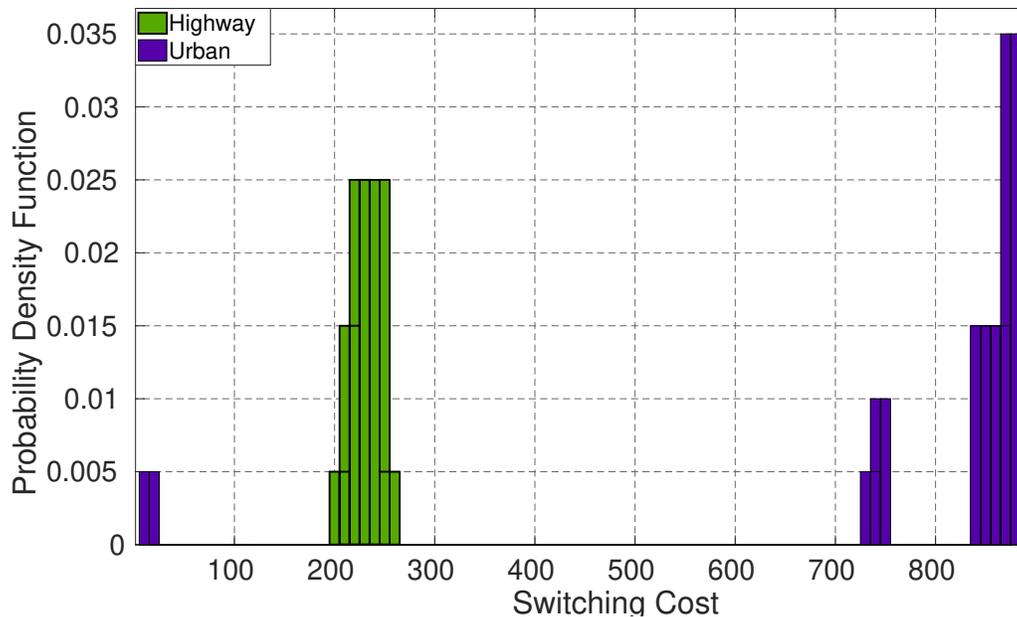


Figure 3.4: Comparison of Switching Cost in highway and urban environment. The high switching cost in the urban scenario implies the high amount of channel switching which results from highly time-variant channel [25].

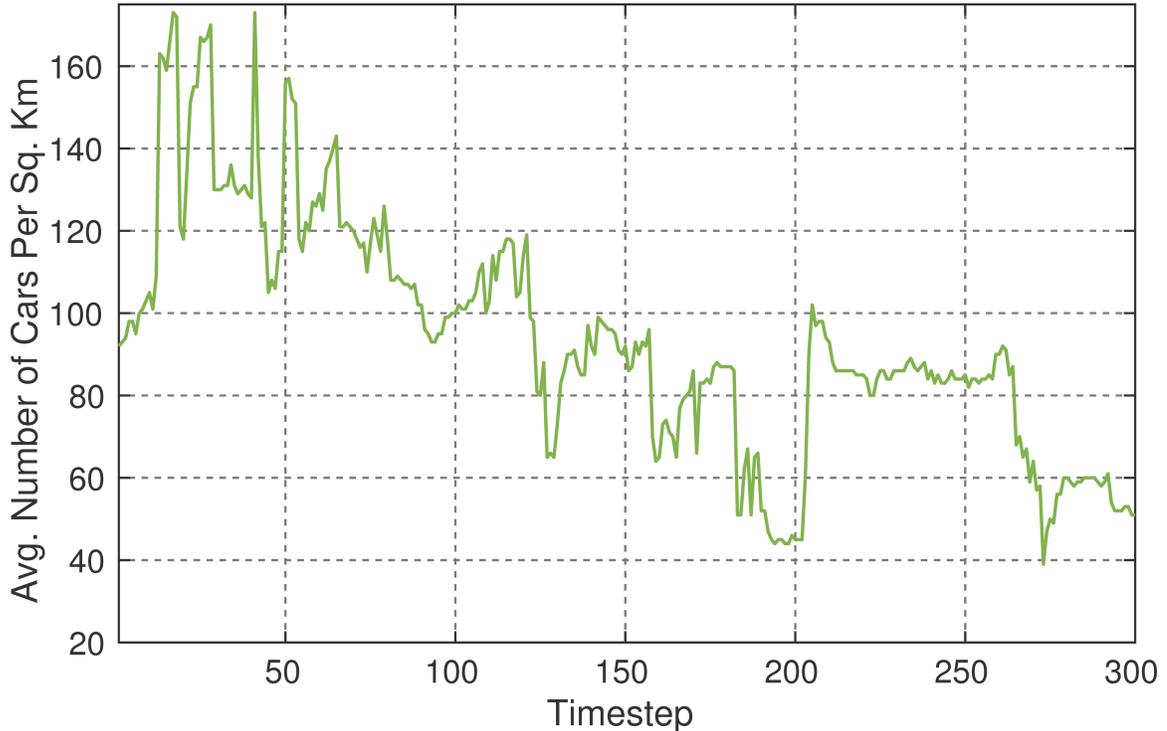


Figure 3.5: Average number of cars per square kilometer simulated in GEMV² in Worcester, MA [25].

3.3 System-Level Simulation Results

We have analyzed the performance of a DSA-based VANET using the adaptive behavioral response mechanism in the GEMV² Vehicle-to-Vehicle (V2V) propagation simulator via MATLAB [84]. The channel sensing algorithm is performed for DTV frequency band at 700 MHz.

3.3.1 GEMV² and SUMO

In Figure 3.8 we compare the memoryless bumblebee algorithm with random channel selection. In random channel selection vehicles select random channels at each time step and stay on the channel until the connection is lost. The bumblebee framework compares the current channel reward with other channels and switches if it is beneficial after taking into account the switching cost. For the highway scenario, sparse traffic conditions, 12 *vehicles/km*² is simulated, whereas for urban traffic conditions we consider

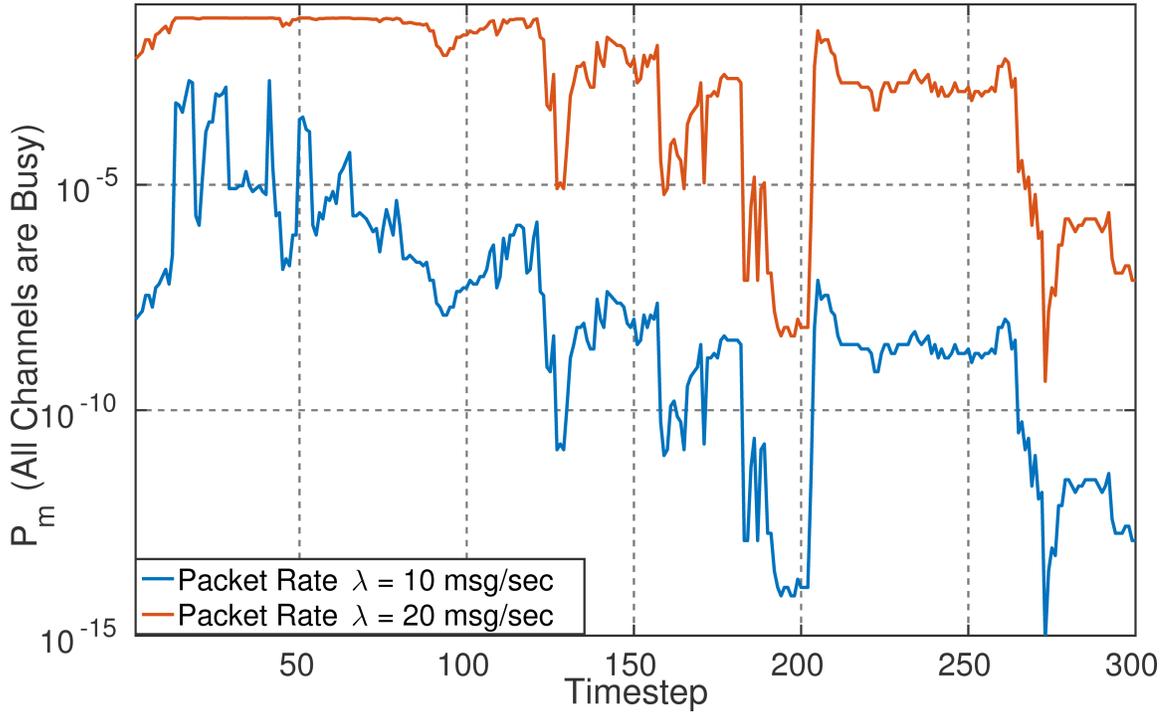


Figure 3.6: The Probability (P_m) of all channel being busy observed by the vehicles during the discrete time-step emulating the real-world environment [25].

150 *vehicles/km*². The number of randomly moving vehicles increases at each time step from 0 to 800 vehicles for urban scenario, whereas for the highway scenario vehicles increase from 0 to 180. We see a huge increase in the mean channel reward for both urban and highway scenario at each time step.

For highly dynamic channel environments, the energy will vary instantaneously, and without memory, vehicles cannot efficiently make a switching decision. For a memoryless model, any channel switching is based on current time step data, and for highly time-variant channel environment, it does not perform efficiently. In Figure 3.9 we compare the channel reward for various switching cost under memoryless bumblebee model. It is evident from the plots that as the penalty increases, the channel reward starts decreasing. As we discussed earlier, the switching cost depends on the environment and cognitive radio characteristics, and the channel reward will vary with the switching cost. For example, a better cognitive radio will be able to switch to a new channel faster, and hence it will have low switching cost.

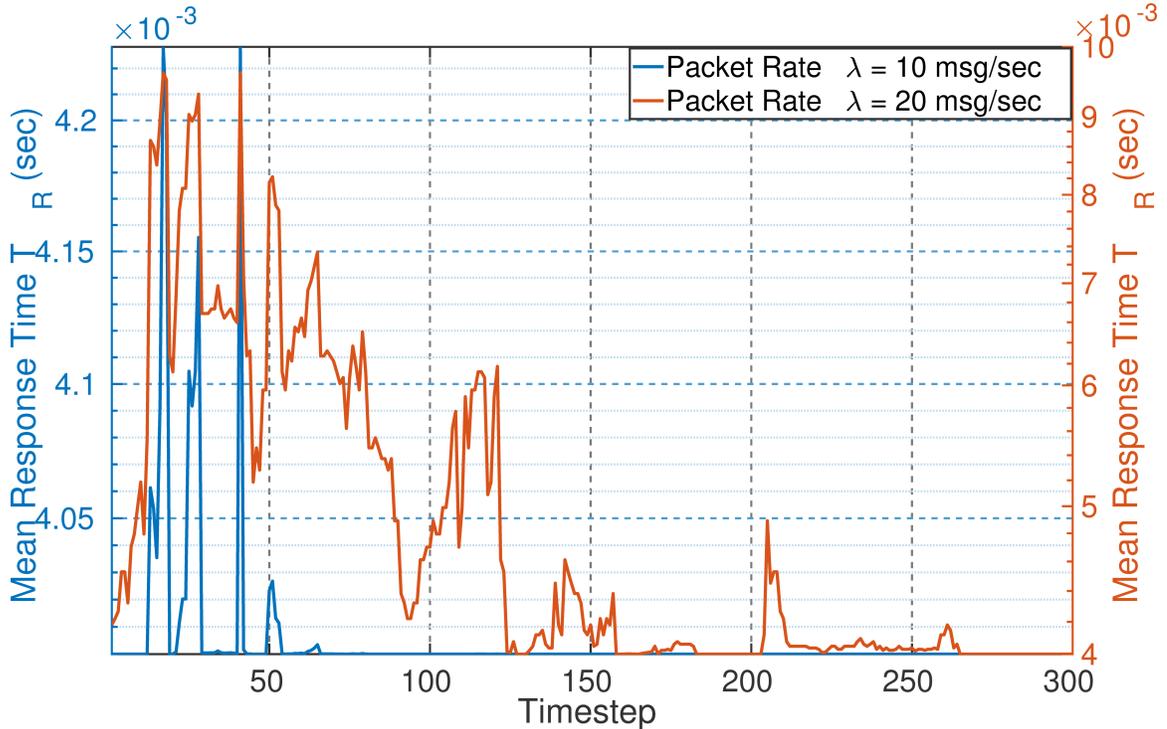


Figure 3.7: The predicted Mean Response Time (T_R) for vehicles performing VDSA in a simulated real-time environment [25].

In Figure 3.10 we compare our memory-based bumblebee algorithm using two different memory strategies under both urban and highway scenarios. In the "Max" memory strategy we select the best channel reward from the past samples in the memory and compare it with the current channel reward to make our switching decision. If the new channel has better reward after taking switching cost into consideration then we switch to the new channel. Using the "Max" strategy we see a 40% improvement in urban environment as we increase the memory length from $M = 0$ (memoryless) to $M = 20N$. In "Mean" strategy we take the average of all the channel rewards in memory and compare our current channel reward with the mean values. If the reward is larger after subtracting the switching cost s , then we switch to the channel; otherwise we stay on the same channel. Since by averaging out the channel rewards we can better estimate the channel quality over time, the "Mean" strategy outperforms "Max" strategy (by approximately 50% for highway scenario and 9% for urban). If we increase the memory length from $M = 0$ to $M = 20N$ using the "Mean" scheme, we

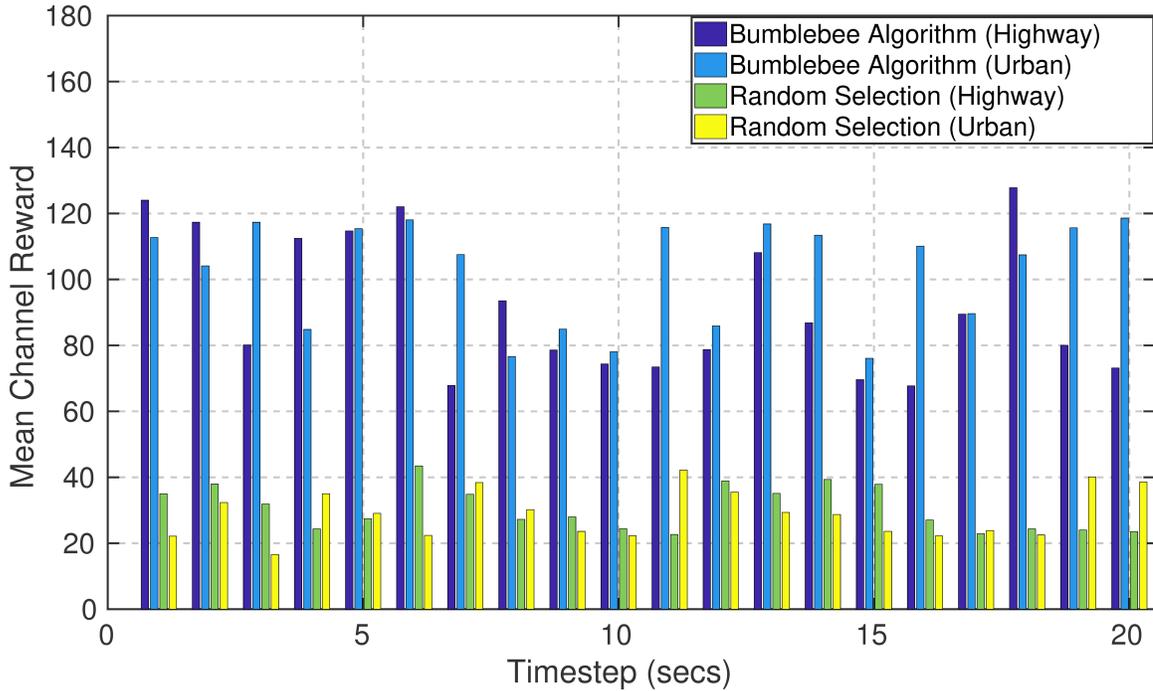


Figure 3.8: Mean Channel Reward $\bar{r}(t)$ comparison for bumblebee model and random channel selection in urban and highway scenario. Vehicles employing the bumblebee framework tends to choose the channel with best reward and hence maximizes the overall channel reward [23].

see an overall increase of 52% for urban and 37% for highway scenario. These results show by utilizing memory we can improve the channel selection performance drastically and use the channels efficiently.

3.3.2 Bumblebee-based Platooning VDSA

The performance of VDSA in TVWS frequencies for intra-platoon communications has been evaluated using system-level simulations of the considered scenario. We used a simulation tool developed in C++. The duration of a single simulation represented a platoon traveling a distance of 5 km over a time period of 140 s. The distributed VDSA framework was applied with 5 adjacent DTT channels having center frequencies located at 490 MHz, 498 MHz, 506 MHz, 514 MHz, and 522 MHz. Since the IEEE 802.11p standard uses 10 MHz bandwidth and the center frequencies for VDSA transmission are assumed to be equal to

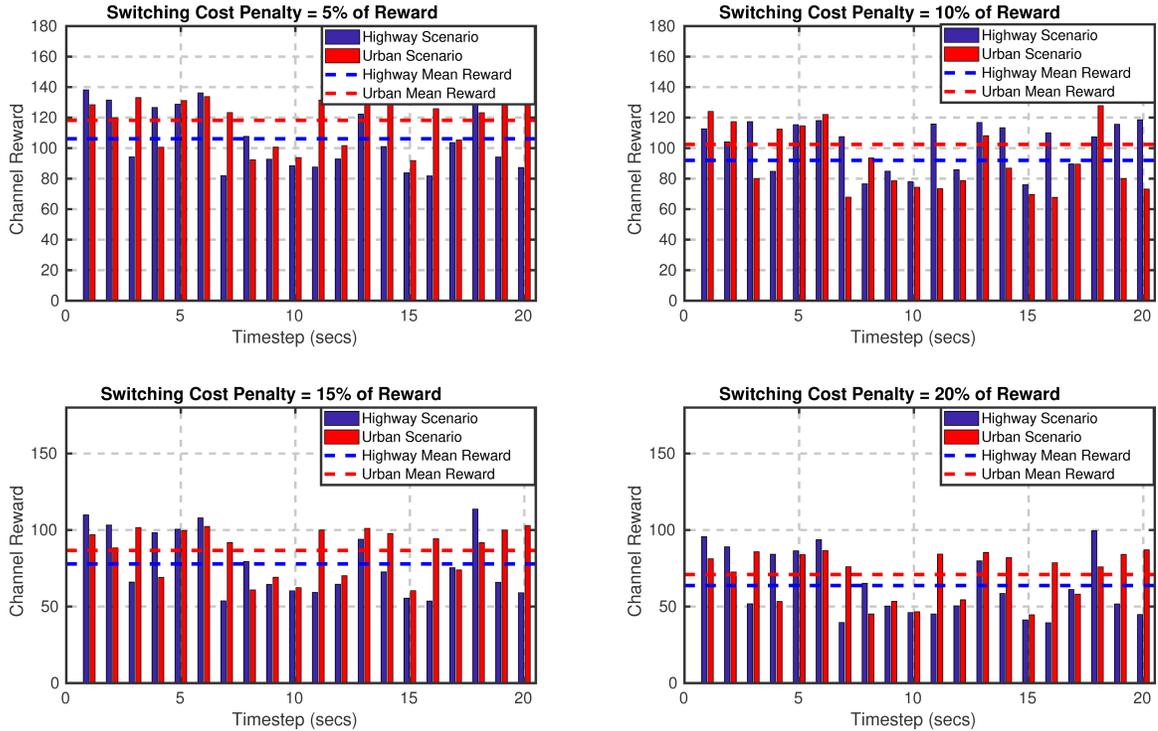


Figure 3.9: Mean Channel Reward for memoryless bumblebee model for different values of switching cost. The switching cost is computed in terms of channel reward to make it more generalize and independent of cognitive radio characteristics [23].

the DTT channel frequencies, the signals from neighbouring bands may partially overlap. Moreover, the two edge bands, *i.e.*, at 490 MHz and 522 MHz, are used by DTT transmitters, hence strong inter-system interference is expected. We compared the performance of a bumblebee-based VDSA framework with the reference system that used transmissions operating only in the 5.9 GHz band or in TVWS frequencies but without channel switching (a fixed channel at 506 MHz was used). For the bumblebee-based approaches, three switching cost values were considered: 0 dB (no cost), 3 dB, and 6 dB. One should also note that for the TVWS band, the default IEEE 802.11p CSMA sensitivity threshold was increased by 10 dB in order to reduce the impact of transceiver blocking by high DTT power levels.

One important performance indicator for intra-platoon communications is transmission reliability, which can be represented by the ratio of successful receptions to the total number of messages transmitted. It is assumed this indicator should be kept over 99% to enable safe

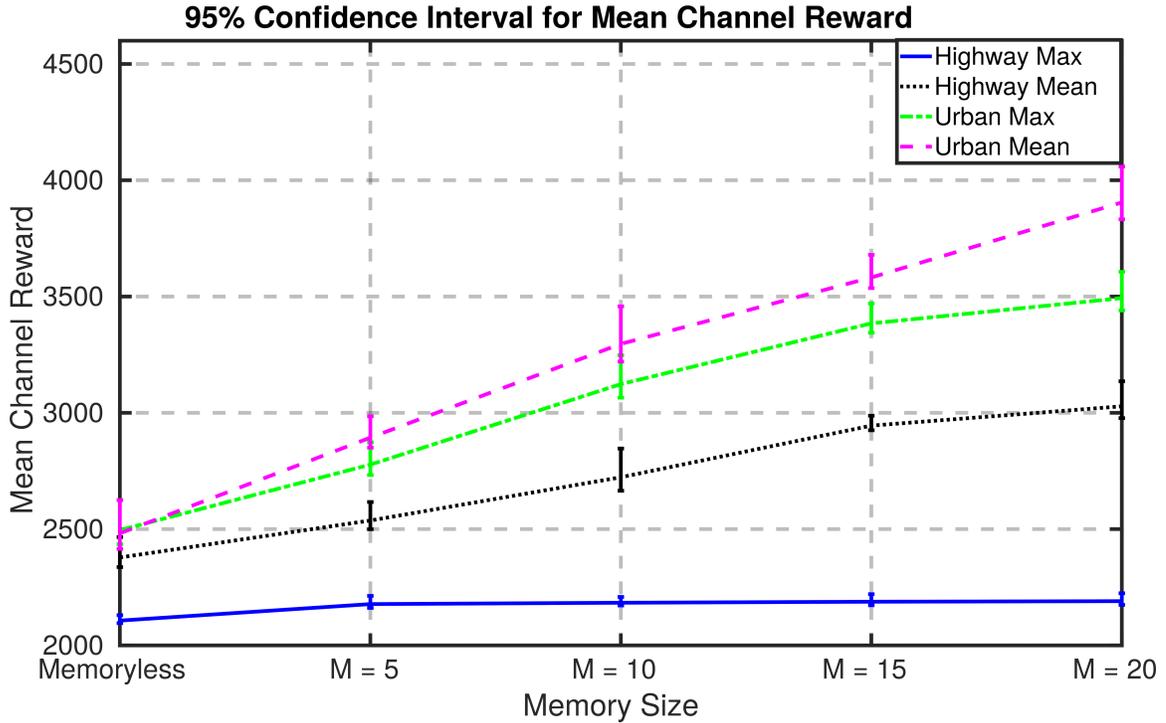


Figure 3.10: Mean Channel Reward for highway and urban scenario for different memory size with 95% Confidence Intervals. The channel reward increases by using higher memory size for both urban and highway scenarios for the “Mean” strategy. Using the “Max” method the performance increase is only significant for the urban scenario. In comparison to memoryless system we see an increase of about 52% in $M = 20N$ memory size for urban scenario for “Mean” strategy [23].

autonomous platooning. Fulfilling this requirement, especially in the case of high density traffic on motorways, might not be possible when only CCH is used. This conclusion is supported by the results shown in Figure 3.11, which presents the successful reception rate of packets transmitted by the platoon leader versus the vehicle position in the platoon. Employing intra-platoon communications at TVWS frequencies with additional CACC packets improves the reception rate, especially for vehicles at the tail of the platoon, which is mainly due to the lower messaging rate of CCH and the higher transmission range for TVWS. However, the reception probability is almost the same for all transmission configurations using TVWS. The application of the bumblebee-based algorithm does slightly improve the recep-

tion rate for the vehicles at the tail of the platoon, but the observed difference is marginal. Surprisingly, the best reception rate is observed with the additional cost of 6 dB applied in switching procedure. Hence, one may conclude that high sensitivity to changes in the observed energy results in the selection of poor frequency bands.

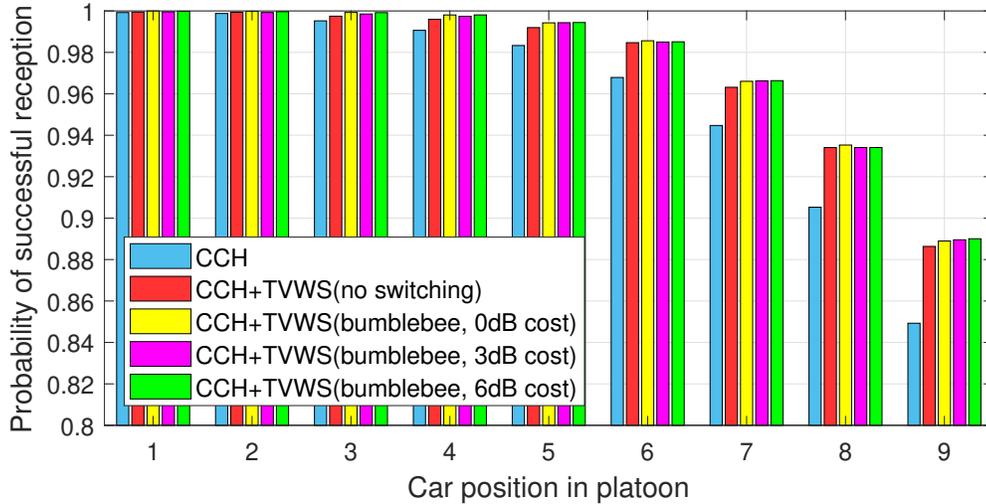


Figure 3.11: Probability of successful reception of leaders' packets [29].

Besides the reliability requirement, another factor affecting the performance of VDSA is the introduced cost of changing the operating frequency for the entire platoon, which requires additional level of dissemination and coordination effort. Such cost depends on the number of frequency switches performed by each platoon. We considered three configurations with different expected rates of switching depending on the selected value of the C parameter in the bumblebee-based algorithm; an example of selected frequencies for different configurations shown in Fig. 3.12. When the platoons are sufficiently separated, they both select the middle frequency band (*i.e.*, 506 MHz) since the interference from the DTT is the lowest. However, when the platoons are closing, their transmission in the same band starts to affect the channel selection procedure and switching starts. Since no coordination between platoons is applied, it is observed that sometimes both platoons select the same band (the one with the lowest energy), which does not improve the performance. An additional factor (*e.g.*, randomness) could be introduced to improve the channel selection procedure.

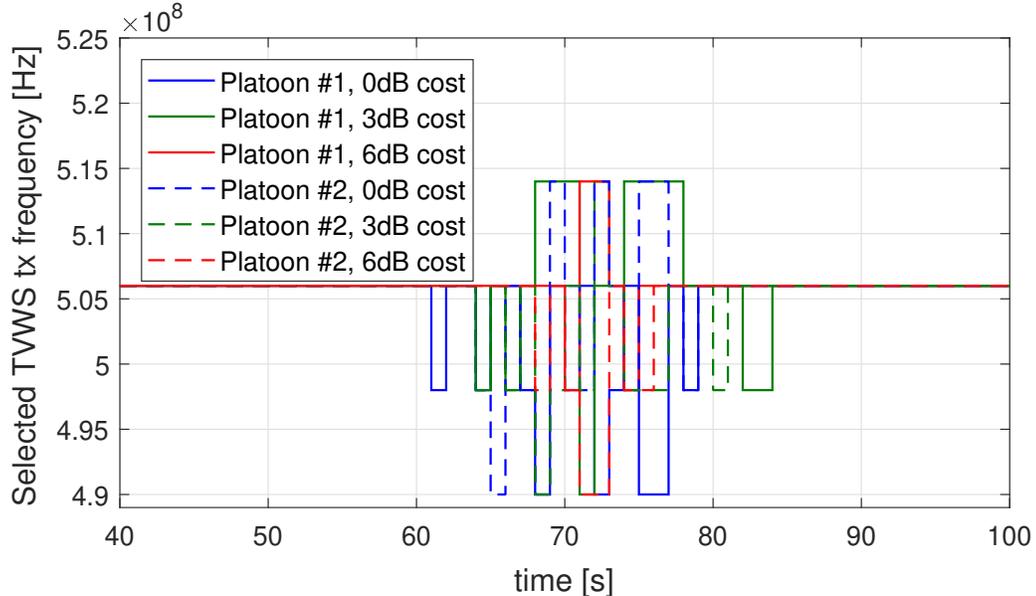


Figure 3.12: Example of frequency switching for two platoons moving in opposite directions and different considered switching costs C . It can be observed that frequency band switching occurs when the platoons are close to each other in the time interval of 60-80 s. Frequency bands at 490 MHz and 522 MHz are rarely or never selected due to the presence of DVB-T transmissions [29].

The average number of switches, representing the expected additional cost of VDSA, is given in Table 3.1. The highest rate of changes is observed with the 3 dB cost, although the difference with the case for no switching cost is small and may be caused by chance variations, *e.g.*, use of shadowing for DTT. With 6 dB cost, the switching rate is lower with only significant changes in sensed energy having impact on new channel selection.

Table 3.1: Average number of frequency changes with bumblebee-based algorithm using different switching costs.

| Platoon | 0 dB (no cost) | 3 dB cost | 6 dB cost |
|---------|----------------|-----------|-----------|
| 1 | 3.9 | 4.1 | 3.4 |
| 2 | 4.4 | 4.5 | 3.6 |

An important consideration of using the TVWS as a secondary system is not to cause the

degradation of the DTT service. Therefore, we investigated the Signal-to-Interference (SIR) levels of the DVB-T transmission observed at the locations of the respective DTT receivers as the platoons move across the motorway. According to [85], the SIR value of 39.5 dB should be kept in order to provide the required QoS of DTT. Fig. 3.13 presents the empirical cumulative distribution of the SIR values of the primary system obtained in simulations for the protected locations of DVB-T receivers. The results indicate that for every considered strategy, the SIR levels are similar with a slightly better performance achieved using the bumblebee-based algorithm assuming 6 dB cost. However, for both observed DTT bands, a large number of collected SIR samples is below the required threshold. This indicates that the bumblebee-based algorithm should be enhanced with the measures to protect the primary users, *e.g.*, by modifying the channel switching metric. Furthermore, the available spectrum is probably also significantly narrow to effectively mitigate the interference to DTT without any power control applied.

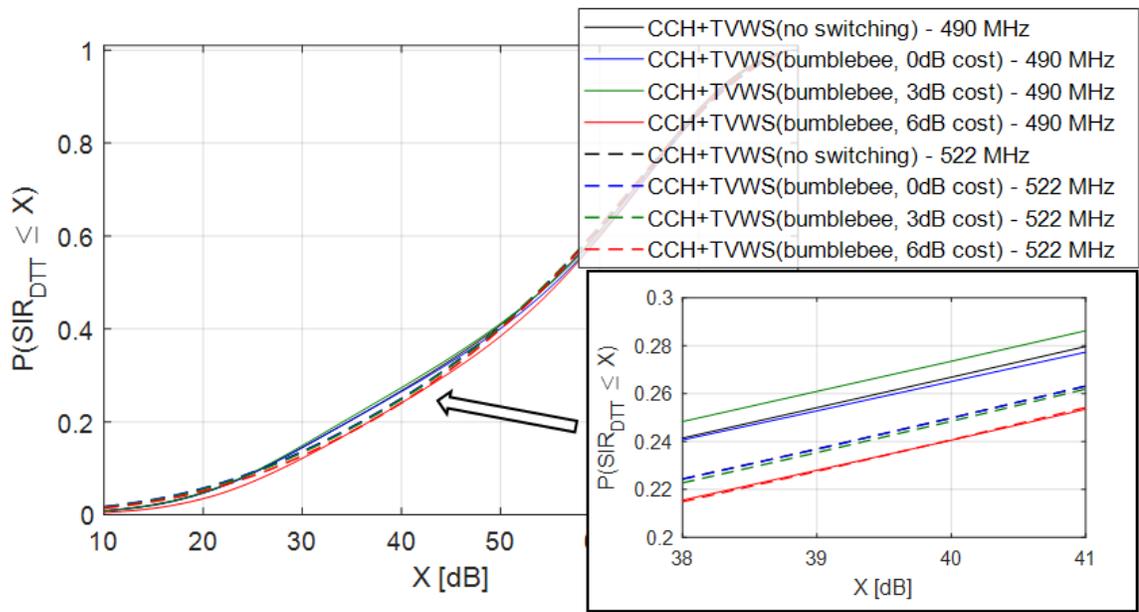


Figure 3.13: Cumulative distribution of SIR for the protected DTT receivers [29].

3.4 Hardware Validation

Bumblebee foraging behavior is mainly based on individual decision mechanisms, making it well-suited for applications where decision-making is performed independently [86]. Since there is no need to access any centralized system or wait for information from others, decision-making and adaptation to change can occur as rapidly as the local information processing system allows. A time-varying stochastic channel also poses the same challenge in a wireless environment as a nectar distribution in flowers where bumblebees forage. Due to the similarities between the two systems, we found that bumblebee-inspired algorithms for optimal channel selection in a time-varying noisy environment, were observed to be highly efficient [87].

Algorithm 2 Memoryless Bumblebee Algorithm

```

1: procedure BUMBLEBEEALGORITHM( $r, T, C, N$ )
2:   for  $t = 1$  to  $T$  do
3:     Sampling Interval :
4:     Compute Energy values  $E \in \{C\}$ 
5:     Map to reward  $r \in \{C\}$ 
6:     Select best  $\{C\}$  at  $t$  for  $n_i$ 
7:     Transmission Interval:
8:     Start the packet transmission
9:   end for
10: end procedure

11: procedure CHANNELREWARD( $E, C, T, N$ )
12:   for  $t = 1$  to  $T$  do
13:     while  $c = 1$  to  $C$  do
14:        $r(t, c) = \min\{\hat{E}\}$ 
15:     end while
16:      $N \leftarrow r(t)$ 
17:   end for
18: end procedure

```

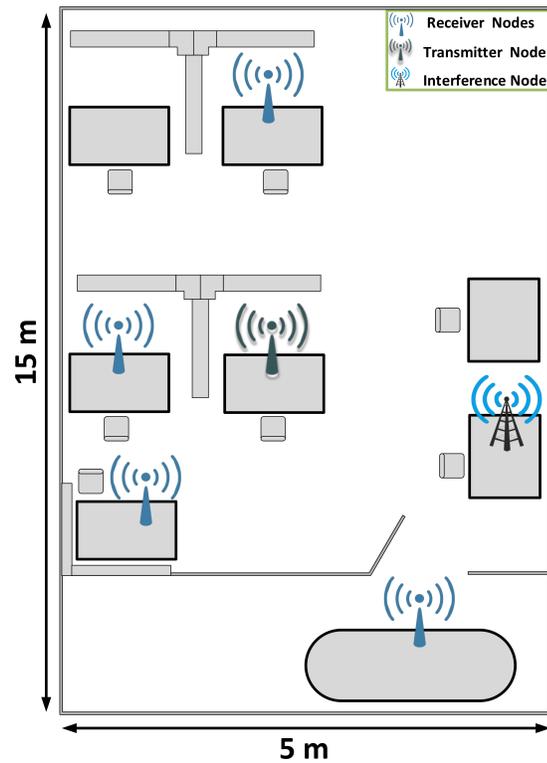


Figure 3.14: Layout of the laboratory where the experiment is conducted. The layout shows the location of each individual radio node [26].

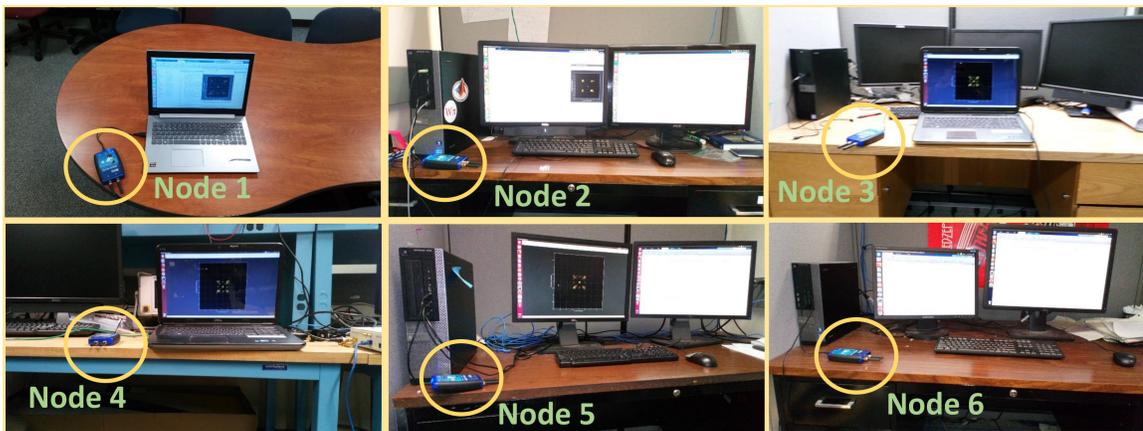


Figure 3.15: The experimental test-bed consisting of six Pluto-SDRs forming an Ad-Hoc network. The experiment is conducted in a controlled laboratory environment [26].

Algorithm 2 describes the bumblebee channel selection algorithm in detail. The bumblebee framework is initiated by sampling the channels in the list and then mapping the

energy values to the reward $r(t)$, which is a linear function of the energy value E . The best channel is selected based on the reward $r(t)$ and it is assigned to the node N . The channel reward function $r(t)$ is given by:

$$r(t) = \min\{\hat{E}\}, \quad (3.7)$$

where $\min\{\hat{E}\}$ is the minimum energy value of the channels used in DSA at that time interval. A more complex channel reward function can be used based on the radio characteristics and the channel environment. However, we have used a simple channel reward mapping to reduce the processing delay caused during the channel selection algorithm. Here T is the total simulation time, C is the set consisting of all channels used in the ad-hoc network test-bed, s is the the switching cost, and N is the total number of Pluto SDR units. The test-bed is initiated by sampling the spectrum space for t_s duration and then assigning the channel to our radios based on the channel selection strategy. In this work, we have kept the sampling time $t_s = 300 \text{ ms}$ fixed for both bumblebee and random channel selection. For the bumblebee-based channel selection, the radio nodes select the best channel out of available options for each cycle using channel energy values. In the case of the random channel selection algorithm, the nodes randomly select a channel and start the packet transmission. Both schemes are employed on similar radio networks for time duration T , and their packet delivery ratio performance is compared. Eq. (4.25) describes the memoryless bumblebee algorithm:

$$\begin{array}{l} \text{Switching} \\ \text{Decision} \end{array} = \begin{cases} r_c \leq (r_n - s_n), & \text{“Switch”}, \\ \text{otherwise}, & \text{“Stay”} \end{cases} \quad (3.8)$$

where r_c is the current channel reward, r_n is the new channel reward, and s_n is the switch cost for the new channel during the sampling time interval. The switching cost s in this work is assumed in terms of channel energy values in order to reduce the test-bed implementation complexity. The switching cost will vary depending on the cognitive radio characteristics and channel environment. In our prototyping test-bed, the wireless spectrum is sensed based on a mechanism that detects energy levels for each channel. The channel model considers all entities specific to a vehicular environment such as multipath fading, doppler

shift, and scattering, which can be mathematically expressed as [56]:

$$h(\tau, t) = \sum_{k=0}^{P-1} h_k(t) e^{-j2\pi f_c \tau_k(t)} \delta[\tau - \tau_k(t)], \quad (3.9)$$

where τ is the path delay, P is total number of paths, t is time variable, $h_k(t)$ is channel envelope, δ is the channel impulse response, and f_c is the carrier frequency. Using the channel impulse response, the detection problems can be formulated as an M -ary hypothesis test. In this case, the spectrum sensing performs the following a binary hypothesis test:

$$\begin{cases} \mathbf{H}_0 : y(t) = n_r, \\ \mathbf{H}_1 : y(t) = \int_{-\infty}^{\infty} h(\tau, t) x(t - \tau) dx + n_r \end{cases} \quad (3.10)$$

where $y(t)$ is the received signal and n_r is the noise. Once the vehicles occupy a channel that is available, such as \mathbf{H}_0 , they need to periodically check to see whether they can potentially switch to a better channel.

3.4.1 Adhoc Network Testbed

The experimental test-bed was implemented using the ADALM-Pluto SDR units with each radio connected to a desktop computer running Ubuntu 16.04. The configuration and measurements for the ad-hoc network were performed using MATLAB [88]. Figure 3.14 describes the layout of the laboratory where the experiment was conducted. In this experiment, we utilized six Pluto SDRs to form a wireless ad-hoc network where four radios were configured as receivers with each listening to a different channel. One node was used as a transmitter, which employed the bumblebee framework to select the optimal channel and used it for packet transmission. We configured another radio to be an interference node whose role was to select channels randomly. The interference node was added in order to test the performance of the bumblebee framework in a interference-prone environment.

Figure 3.15 shows the experimental setup we used in our work to evaluate the performance of bumblebee algorithm. In order to experimentally determine the channel selection performance of the bumblebee framework, we conducted measurements up to ten minutes for different packet sizes. Each trial was conducted three times in order to get an average

estimate of packet delivery ratio (PDR) for each packet size. For the bumblebee framework, we first start by sampling the channels in the list and select the channel with the lowest energy at that time instant. The radio starts the packet transmission on the selected channel for 60 seconds, whereupon it re-samples the channels and switches channels based on the switching cost. Switching channels for every sampling interval based on channel information may sound rewarding with an ideal radio. Without any channel switching delay, as well as timing or frequency correction delay, selecting a channel with least energy is the optimal choice. However, in the real-world we need to take the aforementioned delays into consideration, which can severely affect the radio performance during frequent channel switching. For example, using the ADALM-Pluto SDR takes about 300 *ms* to switch channels resulting in the loss of approximately 10 packets of the size 800 bytes. Additionally, it takes about 400–500 *ms* for the Phase-Locked Loop (PLL) implemented in MATLAB to lock onto the signal to achieve accurate timing and frequency correction. Our bumblebee channel selection algorithm takes this switching cost into consideration before making a decision to switch to another channel. In this work, we implemented the switching cost in a very simplistic manner in order to reduce the processing time and overhead delay for making the switching decision.

We ran the same experiment for random channel selection wherein the transmitter selects a random channel from a list and starts packet transmission. Random channel selection does not sample the spectral environment, which results in a high frequency of collisions with the interference node and incurring a high packet loss. In Figure 3.19, we can observe a large difference in the PDR whilst comparing the two schemes. Severe packet loss is caused due to the interference node and frequent channel switching. To evaluate the performance of our channel selection algorithm, we first computed the channel sensing performance of the energy detection. In this work, we used an adaptive threshold for the energy detection scheme in order to efficiently determine the channel utilization. A fixed threshold can be used in a static environment for accurate primary user (PU) detection but in a time-varying noise environment it can lead to a high rate of false alarm and missed detection probabilities. We compute the mean energy of all the channels in each time interval and then add a random factor K to compute the final threshold. Consequently, the threshold changes dynamically

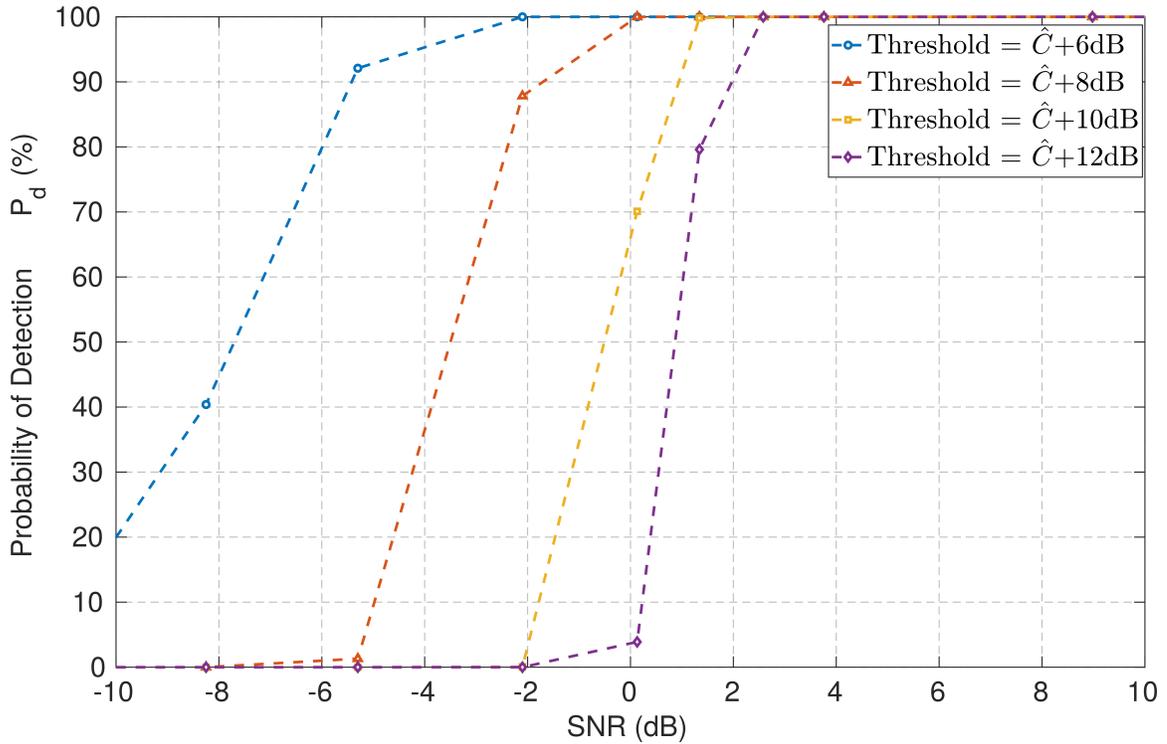


Figure 3.16: P_d vs SNR for threshold factor $K = 6, 8, 10, 12$. As we increase K , the sensing performance degrades due to large misdetections in the network [26].

with the environment and we can get an accurate estimate of the primary user. Figure 3.16 shows the probability of detection (P_d) vs SNR for the ED scheme for different K values. In this work, we set the K value to 6 dB as that lead to the best detection probabilities. Lower K values increase detection, but they also leads to an increase in false alarm probabilities.

We also tested the interference caused due to an increased number of nodes and how it affects the packet failure rate. Two radios were configured in order to transmit and receive packets continuously on single channel. The transmit power was kept fixed at 8 dBm, while the receiver gain was changed to generate different Signal-to-Noise Ratio (SNR) data. Initially, only the transmitter and receiver nodes were used to get the benchmark packet failure rate (PFR) performance in the absence of any interference. An experiment was then performed with four and six nodes; we observed that as the number of nodes increased, the network performance degraded. The effect is more severe at low SNR values, where a large amount of data packets are lost. Figure 3.17 shows the plot for PFR vs SNR as the

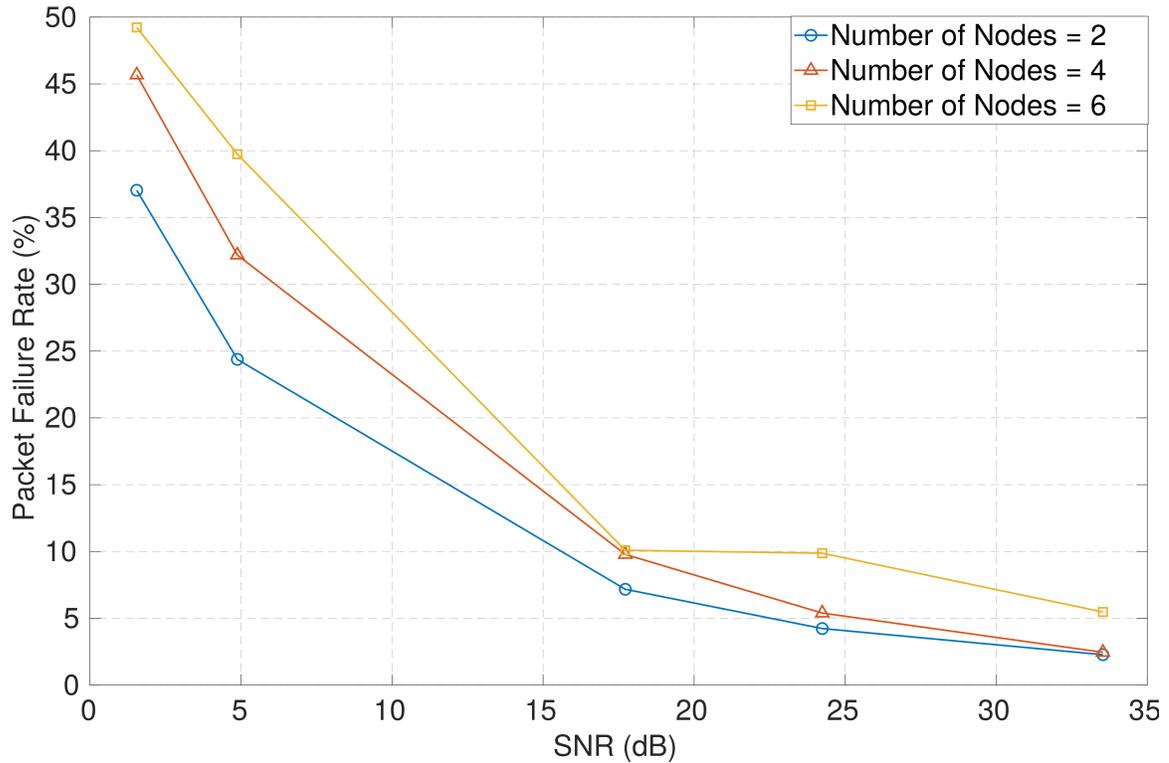


Figure 3.17: Packet Failure Rate in an Ad-hoc network as the nodes are increased from $N = 2$ to 6 for varying SNR [26].

number of nodes are increased from $N = 2$ to 6. Figure 3.18 describes the behavior of the bumblebee framework in the presence of the interference node. The radio node avoids the interference node to avoid the packet collisions based on the switching cost. If the cost is high for the channel switching, then the bumblebee framework will stay on the same channel. During the experimental run, channel 5 was being continuously used by the ISM band users, which means it was never utilized by the radio node. Channel 2 was the most rewarding during our simulation run, which was verified with the spectrum analyzer and is evident by the bumblebee VDSA framework's channel selection. Finally, we computed the PDR for the ad-hoc network using the bumblebee and random channel selection scheme. For each packet size, we performed the experiment for 10 minutes and ran it three times to get the mean PDR performance. As the packet size was increased from 800 bytes to 1600 bytes we saw a decrease in PDR for both bumblebee and random channel selection. The performance of random channel selection is severely affected by the interference node and

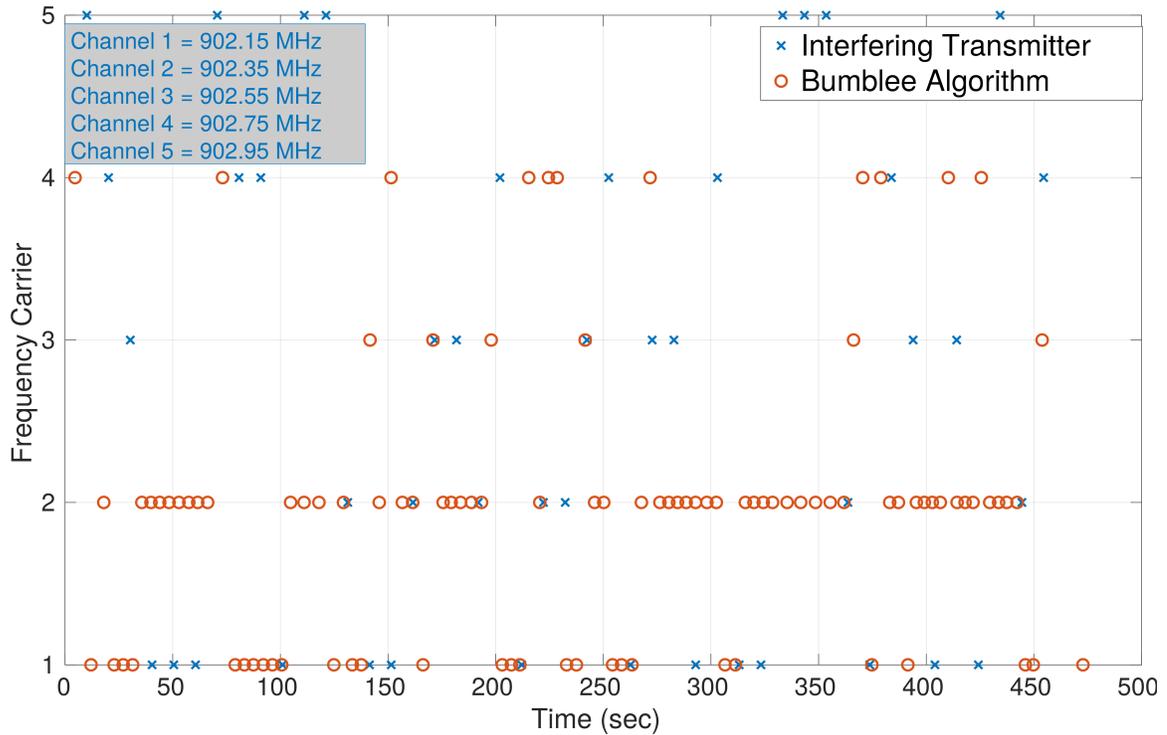


Figure 3.18: Bumblebee algorithm switching from one channel to another avoiding the interference node. During the entire experiment, radio running bumblebee algorithm stays on channel 2, reducing switching cost [26].

frequent channel switching which causes additional packet loss. The bumblebee framework on the other hand takes switching cost into consideration and avoid switching channels if the reward is not worth it. Due to the relatively stationary laboratory environment, the bumblebee framework used 902.55 MHz about 60% of the time. This reduced the need for frequency switching and lead to a significant gain in bumblebee performance.

3.5 Chapter Summary

In this chapter, we presented the proposed bumblebee-based VDSA and compared it against the random-access baseline. The performance bounds were evaluated using queuing theory, consisting of multiple server and same priority class. The algorithm was also implemented using Pluto software-defined radio and showed performance gain compared to the random-access scheme, where RF parameters were kept same for both. Finally,

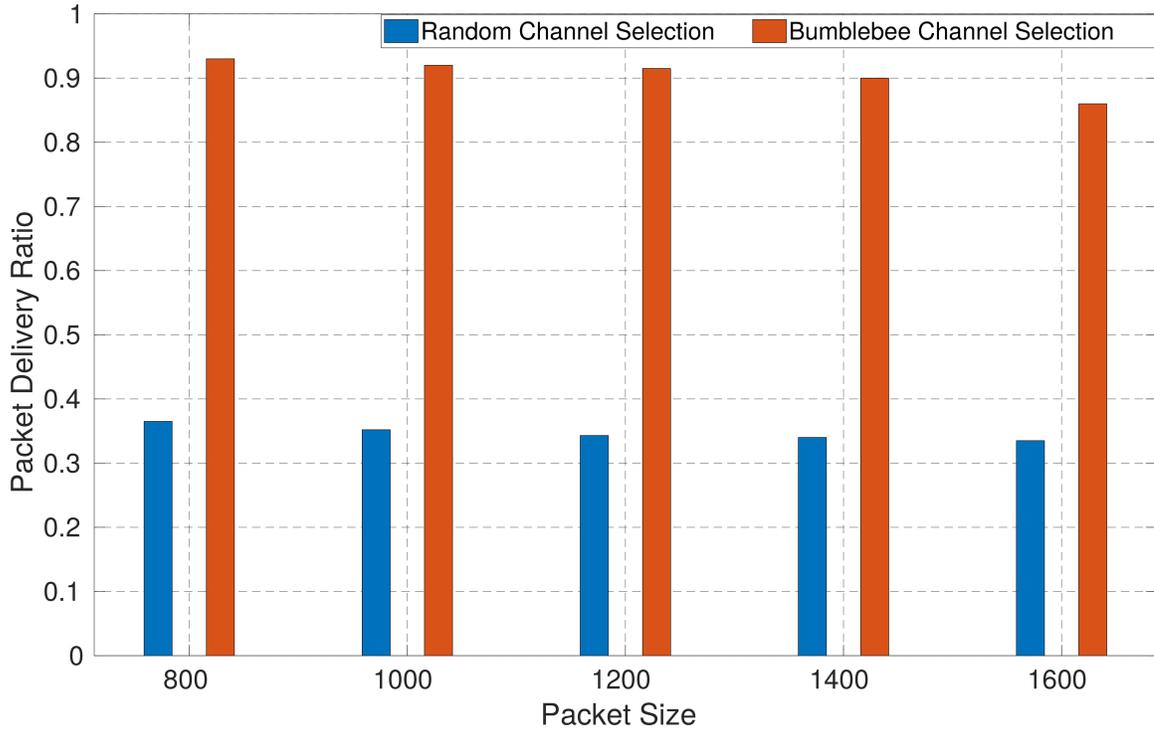


Figure 3.19: PDR for bumblebee and random channel selection for different packet size. Due to frequent channel switchings and large packet collisions with the interference node, the random channel selection has a high packet loss. The variation of PDR for each packet size was around 1%–2% in both bumblebee and random channel selection [26].

the distributed bumblebee algorithm was also compared against a centralized REM-based architecture, where bumblebee algorithm achieved considerable gain in latency.

Chapter 4

Memory-Enabled Bumblebee Algorithm for Efficient Channel Selection

In this chapter, we present the memory-enabled bumblebee algorithm with optimal channel sampling allocations for efficient channel selection. First, we introduce a heuristic to utilize non-uniform sampling instants for optimally allocating resources. Instead of allocating sampling instances to all channels equally, the heuristic distributes sampling instances to each channel based on the Channel Occupancy Ratio (COR). The performance is evaluated using a C++ simulator in a platooning environment. Lastly, we consider sub-channel selection in cellular V2X, where semi-persistent scheduling is modified to enable spectrum access in DTV band.

4.1 MEBA for Platooning

A commonly used method for spectrum sensing is energy detection [8], which is simple to implement and fast, and both of which are critical for V2V communications. While the typical problem of energy detection is to obtain sufficient sensitivity in the presence of noise, this is not the only important consideration for a V2V communications environment,

where the primary challenge is bursty interference. Persistent interference can be assessed via a single sweep of an energy detection process at each geographical location. However, in case of bursty interference, a single sweep of an energy detection scheme can result in the channel being detected as completely vacant or occupied at a single instant in time, but this may not be the case at another time instant. The IEEE 802.11p standard for V2V communications, which is considered in this work, employs bursty transmission that may generate bursty interference for other users. Once spectrum sensing has been performed, the platoon should find the optimal frequency channel in which to operate. The optimal frequency channel is the one occupied for the lowest percentage of time, *i.e.*, characterized with the lowest Channel Busy Ratio (CBR). The CBR is also referred to by some authors as Channel Occupancy Ratio or Duty Cycle. It is probability of an active transmission in a given channel at a given time. Alternatively, the CBR can be defined as the ratio of time when the channel is busy versus the total observation time [89]. The challenge is how to reliably find the channel possessing the lowest CBR out of the considered set of channels using a limited number of samples during a rapidly varying environment resulting from vehicles movement. CBR estimation procedures are rarely discussed in the literature. In [90], the authors focus on an optimization of the CBR estimation for weak signals that are close to the noise floor. In particular, they consider the non-zero probabilities of false alarm and miss detection. Unfortunately, the discussion was limited to a single channel case and is not applicable to the multi-channel scenario considered in our research. Thus, we concentrate on multi-channel CBR estimation while applying non-uniform allocation of sampling instants among the channels. Additionally, we consider the effects of time-varying CBR, which occurs in realistic V2V environment where vehicles employ VDSA.

4.1.1 Optimal Non-uniform Sampling Allocation

Consider a high-speed road consisting of four lanes, with two lanes devoted to each direction. The cars exchange BSM messages via the IEEE 802.11p communication scheme operating in the DTV band, as it is assumed that the nominal 5.9 GHz band is congested. All the cars communicate over L frequency channels indexed by $l = 1, \dots, L$. The traffic generated by each car is – from the perspective of the platoon – random and of various

intensity and duration. Therefore, each channel is characterized by Channel Busy Ratio (CBR) denoted as $\beta_l \in \langle 0, 1 \rangle$ for the l -th channel, which can be interpreted as the ratio of time when the channel is busy to the total time of observation. In order to detect the presence of other ongoing transmission, non-uniform channel sampling is performed; and the sampling moments are depicted as red lines. The platoon aims to find the channel with the lowest CBR (denoted as \hat{l}) that allows for the highest reliability of intra-platoon transmission:

$$\hat{l} = \arg \min_l \beta_l. \quad (4.1)$$

It is possible that more than one channel will have the minimal value of β_l . We denote this set of *optimal* channel indices as $\mathbb{O} = \{\hat{l} | \beta_{\hat{l}} = \min_l \beta_l\}$. The complementary set of *wrong* channel indices is denoted as \mathbb{W} such that $\mathbb{W} \cup \mathbb{O} = \{1, \dots, L\}$. While β_l is not known by the vehicles, it has to be estimated using sensing. The cars perform periodic spectrum sensing to find the most suitable channel to transmit. In our investigation, we chose an energy detection with power threshold based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme used in 802.11p [91] to detect signals that can block intra-platoon communications. Various V2V communication protocols are able to operate under significant interference, *e.g.*, the required SINR level for the 802.11p transmission is 2 dB for the most robust modulation and coding scheme (MCS) which is MCS 0. Therefore, the platoon has to detect only the interference that is capable of blocking its transmission, *i.e.*, significantly above the signal noise floor. As such, we can safely ignore the non-zero probability of false alarms, which is important for weak signal detection [90]. In the simplest approach, the platoon, which is based on the available N_l sensing decisions for the l th channel, estimates the β_l value. The Maximal Likelihood (ML) estimator of β_l is given by:

$$\hat{\beta}_l = \frac{k_l}{N_l}, \quad (4.2)$$

where $k_l \in \{0, \dots, N_l\}$ is the number of times the l th channel has been sensed as busy over N_l sampling moments. While this estimator is consistent, *i.e.*, converges to real β_l as N_l increases to infinity, in practice its accuracy is limited. This can be both a result of finite number of sensing opportunities before the channel is selected, and the limited

number of sensing results within a given time period due to the technical limitations of the applied sensors (*e.g.*, a sensor cannot sample more frequency channels as a result of high frequency-switching latency).

Let us extend the above model to account for the increasing amount of platoon knowledge. This knowledge is improved during each consecutive iteration, *i.e.*, a selected time period allowing for intra-platoon communications, performing of spectrum sensing and selection of a new frequency channel to be used for communications. Each iteration, indexed with (i) , consists of limited total number of N sensing periods that can be carried over all L frequency channels. At i -th iteration the platoon has to decide on the number of sensing moments $N_l^{(i)}$ assigned to each channel $l \in \{1, \dots, L\}$ so that the total limit of N is reached, *i.e.*, $\sum_{l=1}^L N_l^{(i)} = N$. Observe that we assume the total number of sensing samples N is fixed, *e.g.*, as a result of the total number of cars in the platoon and the fixed iteration duration. Without loss of generality, it can be assumed the first iteration is $i = 1$. Based on the number of times channel l is detected as busy at iteration i , *i.e.*, $k_l^{(i)}$, and the prior knowledge obtained as a result of sensing in previous iterations, the estimate of β_l is obtained. Assuming a stationary β_l , the ML estimator is specified as:

$$\hat{\beta}_l^{(i)} = \frac{\sum_{j \leq i} k_l^{(j)}}{\sum_{j \leq i} N_l^{(j)}} = \frac{\tilde{k}_l^{(i)}}{\tilde{N}_l^{(i)}}, \quad (4.3)$$

where $\tilde{k}_l^{(i)}$ denotes the cumulative number of times the signal was detected, and $\tilde{N}_l^{(i)}$ is the cumulative number of samplings done over all past and the current i th iteration.

The platoon switches its carrier frequency to the one of the lowest expected occupancy:

$$\hat{l} = \arg \min_l \hat{\beta}_l^{(i)}. \quad (4.4)$$

The question is how to distribute N sampling moments among L channels in such a way that probability of the optimal channel selection is maximized in each iteration. The discussion on the optimal channel selection in the case of varying CBR values will be continued in Section 4.3.1.

Samples-to-Channel Allocation Strategies:

The simplest approach is to assign an equal number of sensing moments (referred from now as *equal allocation*) to each channel:

$$N_l^{(i)} \approx \frac{N}{L}. \quad (4.5)$$

The approximation sign is used as the number $N_l^{(i)}$ has to be natural while the division of N over L can result with a non-integer value. In such a case each channel will have assigned $\lfloor \frac{N}{L} \rfloor$ samples and the remaining $N - L \lfloor \frac{N}{L} \rfloor$ ¹ samples will be distributed randomly among all L channels. However, after a few iterations if the estimated β_l for some channels is relatively low and very high for some others, the equal allocation could be not optimal. There is no point in increasing the accuracy of CBR estimates for very busy channels, as these have low probability of being selected as the optimal for V2V communications. The sampling should then be increased for the channels with relatively low CBR value, as these are the ones that may contain the optimal channel(s) for data transmission (*i.e.*, the channels of indices belonging to the set \mathbb{O}).

Optimal Channel Sampling for *A Priori* Known CBR Values

This subsection focuses on the optimal allocation of samples among channels. The upper bound of the probability of the optimal channel selection is calculated while adjusting samples allocation, assuming the real β_l values are known. In order to do so, first an analytical formula for a tight Upper Bound (UB) and Lower Bound (LB) on the probability of success will be derived. The probability of success, *i.e.*, the selection of an optimal channel, can be calculated for a given number of samples allocated till the i th iteration to channel l , denoted as $\tilde{N}_l^{(i)}$. Success is a random event that can be formally defined as \hat{l} chosen using (4.4) belonging to \mathbb{O} . The randomness is a result of $\hat{\beta}_l^{(i)}$ being a random variable that depends on the random values of $k_l^{(j)}$ and allocated $N_l^{(j)}$ for $j \leq i$ according to (4.3). The success will for sure happen every time when the minimal estimated CBR value of any optimal channel, *i.e.*, $\hat{\beta}_l^{(i)}$ for $l \in \mathbb{O}$, will be smaller than minimal estimated

¹The operation $\lfloor \cdot \rfloor$ represents the floor function, which outputs the greatest integer not greater than the input argument.

CBR for any wrong channel, *i.e.*, $\hat{\beta}_l^{(i)}$ for $l \in \mathbb{W}$. This can be defined as:

$$\Pr \left\{ \min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} < \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)} \right\} \leq \Pr \left\{ \hat{l} \in \mathbb{O} \right\}, \quad (4.6)$$

where the left hand side constitutes a LB, and the right hand side forms the probability of success. The above defined probability provides a lower bound for the total probability of success, as it considers that for all the events when $\min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} = \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)}$ the wrong channel is selected. On the other hand, the upper bound of the probability of success can be derived by counting all the events when $\min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} = \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)}$ as success. This results in:

$$\Pr \left\{ \hat{l} \in \mathbb{O} \right\} \leq \Pr \left\{ \min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} \leq \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)} \right\}. \quad (4.7)$$

It is observed the right-hand side of the above formula can be partitioned into a sum of probabilities, where the minimum estimated CBR for optimal channels will be lower than the minimum for the wrong channels, and that both these minimums will be equal:

$$\begin{aligned} \Pr \left\{ \min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} \leq \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)} \right\} = \\ \Pr \left\{ \min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} < \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)} \right\} + \Pr \left\{ \min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)} = \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)} \right\}. \end{aligned} \quad (4.8)$$

Let us define new random variables $b^{(i)} = \min_{l \in \mathbb{O}} \hat{\beta}_l^{(i)}$ and $c^{(i)} = \min_{l \in \mathbb{W}} \hat{\beta}_l^{(i)}$. The Cumulative Density Functions (CDFs) of $b^{(i)}$ and $c^{(i)}$ can be calculated assuming that $\hat{\beta}_l^{(i)}$ values are independent among channels. This is a reasonable assumption, as for a given $\hat{\beta}_l^{(i)}$ and $N_l^{(i)}$ the number of times the signal is detected in channel $k_l^{(i)}$ should be independent from $k_m^{(i)}$ for $m \neq l$ (as a result of adjacent channels not overlapping in frequency). The CDF of $b^{(i)}$ is denoted as $F_{b^{(i)}}(x) = \Pr \{b^{(i)} \leq x\}$. It is known that $1 - \Pr \{b^{(i)} \leq x\}$ equals to $\Pr \{b^{(i)} > x\}$, *i.e.*, probability that all $\forall_{l \in \mathbb{O}} \hat{\beta}_l^{(i)}$ are greater than x . Utilizing the independence of the $\hat{\beta}_l^{(i)}$ random variables, with CDF denoted as $F_{\hat{\beta}_l^{(i)}}(x)$ we obtain:

$$\begin{aligned} F_{b^{(i)}}(x) &= 1 - \Pr \left\{ b^{(i)} > x \right\} = 1 - \prod_{l \in \mathbb{O}} \Pr \left\{ \hat{\beta}_l^{(i)} > x \right\} \\ &= 1 - \prod_{l \in \mathbb{O}} \left(1 - F_{\hat{\beta}_l^{(i)}}(x) \right). \end{aligned} \quad (4.9)$$

Following the same reasoning, the CDF for $c^{(i)}$ can be derived, yielding the expression:

$$F_{c^{(i)}}(x) = 1 - \prod_{l \in \mathbb{W}} \left(1 - F_{\hat{\beta}_l^{(i)}}(x)\right). \quad (4.10)$$

As the random variable $\hat{\beta}_l^{(i)}$ is, according to (4.3), a sum of binomial distributed random variables $k_l^{(j)}$ scaled by a given value $\tilde{N}_l^{(i)}$, its CDF can be defined as:

$$F_{\hat{\beta}_l^{(i)}}(x) = \sum_{q=0}^{\lfloor \min(1,x)\tilde{N}_l^{(i)} \rfloor} \binom{\tilde{N}_l^{(i)}}{q} \beta_l^q (1 - \beta_l)^{\tilde{N}_l^{(i)} - q}. \quad (4.11)$$

Recall that $F_{\hat{\beta}_l^{(i)}}(x)$ is 0 for $x < 0$ and 1 for $x > 1$.

Knowing $F_{c^{(i)}}(x)$ and $F_{b^{(i)}}(x)$, the lower bound defined in (4.6) can be calculated. It is the probability of all events when the random variable $c^{(i)}$ is greater than $b^{(i)}$. The probability that a given random variable is greater than a given value is defined by a complementary cumulative density function, *i.e.*, $1 - F_{c^{(i)}}(x)$ for $c^{(i)}$. By the law of the total probability, the probability of $c^{(i)} > x$ have to be added for all possible x values being values of a random variable $b^{(i)}$ (for $c^{(i)} > b^{(i)}$), weighted by probabilities of $b^{(i)}$:

$$\Pr \left\{ b^{(i)} < c^{(i)} \right\} = \int_0^1 f_{b^{(i)}}(x) (1 - F_{c^{(i)}}(x)) dx, \quad (4.12)$$

where $f_{b^{(i)}}(x)$ is the Probability Density Function (PDF) of $b^{(i)}$. It is calculated using differentiation operation as:

$$f_{b^{(i)}}(x) = \frac{dF_{b^{(i)}}(x)}{dx}. \quad (4.13)$$

Observe that both $b^{(i)}$ and $c^{(i)}$ are specified in range $\langle 0, 1 \rangle$ that is reflected by the range of integral. This probability can be calculated accurately using software tools as all these distributions are for discrete variables (*e.g.*, integration can be replaced with summation). While the above formula gives the lower bound of the probability of success, the upper bound requires additionally the probability that minimums of $\hat{\beta}_l^{(i)}$ for both the optimal and wrong channel sets are equal, according to (4.8). This can be obtained by using PDFs of $b^{(i)}$ and $c^{(i)}$, and, thanks to these events independence, integrate over all their possible values as:

$$\Pr \left\{ b^{(i)} = c^{(i)} \right\} = \int_0^1 f_{b^{(i)}}(x) f_{c^{(i)}}(x) dx. \quad (4.14)$$

The PDF $f_{c^{(i)}}(x)$ is calculated from $F_{c^{(i)}}(x)$ similarly as for the variable $b^{(i)}$ in (4.13).

The upper bound considers as a success all the events when: 1) the minimal estimated CBR for the optimal channel is less than the minimal estimated CBR for a wrong channel, and 2) the minimal estimated beta for both the optimal and wrong channel sets are equal. To tighten the upper bound we focus on event 2). If this happens, the number of optimal channels having the minimal estimated CBR value ranges from 1 to $|\mathbb{O}|$, where $|\cdot|$ denotes the cardinality of a set. This number can be denoted as a random variable $q_O \in \{1, \dots, |\mathbb{O}|\}$. Under the same condition, the number of wrong channels having the minimal estimated CBR value ranges from 1 to $|\mathbb{W}|$ and is denoted as a random variable q_W . We can assume that selection of each of these $q_W + q_O$ channels is equally probable while using (4.4). As such, the conditional probability of success, *i.e.*, considering the event 2) happens, equals $q_O/(q_O+q_W)$. Unfortunately, the probability derived in (4.14) does not differentiate between various values of q_O and q_W . For each of these $|\mathbb{W}||\mathbb{O}|$ events possibly different probability is expected. However, the upper bound can be defined assuming the maximal value of $q_O/(q_O + q_W)$ factor for a non-empty set \mathbb{W} , *i.e.*, $|\mathbb{O}|/(|\mathbb{O}|+1)$. The tightened upper bound in respect to (4.7) can be defined as:

$$\Pr \left\{ \hat{l} \in \mathbb{O} \right\} \leq \Pr \left\{ b^{(i)} < c^{(i)} \right\} + \frac{|\mathbb{O}|}{|\mathbb{O}|+1} \Pr \left\{ b^{(i)} = c^{(i)} \right\}. \quad (4.15)$$

With the same reasoning, the lower bound (4.6) can be tightened, creating a new lower bound, by considering not only all the events when $b^{(i)} < c^{(i)}$ but also some part of events when $b^{(i)} = c^{(i)}$. It can be observed that for all the events when $b^{(i)} = c^{(i)}$ the minimal value of factor $q_O/(q_O + q_W)$ is $1/(|\mathbb{W}|+1)$. Therefore the improved lower bound is:

$$\Pr \left\{ b^{(i)} < c^{(i)} \right\} + \frac{1}{|\mathbb{W}|+1} \Pr \left\{ b^{(i)} = c^{(i)} \right\} \leq \Pr \left\{ \hat{l} \in \mathbb{O} \right\}. \quad (4.16)$$

Finally, the lower and upper bounds to be used are defined by (4.16), and (4.15), respectively.

Probability of Success for Optimal Channel Sampling:

The previous section provided tight bounds for the probability of success in channel selection, knowing the samples allocation for each channel $\tilde{N}_l^{(i)}$ as defined in (4.3), and CBR β_l . Now, the values of $\tilde{N}_l^{(i)}$ can be optimized in order to obtain the maximal possible

probability of success. As the upper bound is being assessed, the usage of the upper bound derived in the previous section is justified.

Let us first find *global optimal solutions*, *i.e.*, what should be the total number of samples taken in channel l over the current and all the previous iterations $j \leq i$, *i.e.*, $\tilde{N}_l^{(i)}$, to maximize probability of success in i th iteration. Every possible solution to this problem can be represented as a L -tuple $\mathbf{n}^{(i)} = (\tilde{N}_1^{(i)}, \dots, \tilde{N}_L^{(i)})$, for which each element is a natural number. The number of all samples taken over all channels is iN , *i.e.*, $\sum_{l=1}^L \tilde{N}_l^{(i)} = iN$. In general, the total number of such combinations equals $\binom{L-1+iN}{L-1}$. However, it is reasonable to assume that in the first iteration, before any knowledge about CBR of any of channels is obtained, the *equal allocation* approach is used. As such, $\lfloor \frac{N}{L} \rfloor$ samples constitute a fixed allocation of samples per each element of each possible tuple that is possibly increased by any combination of $iN - L\lfloor \frac{N}{L} \rfloor$ samples distribution among L channels. There are $\binom{L-1+iN-L\lfloor \frac{N}{L} \rfloor}{L-1}$ possible allocations. These can be generated, *e.g.*, using *stars and bars* method, and will be denoted as a set $\mathbb{N}_{\text{global}}^{(i)}$. While the total number of possibilities rises fast with the iteration index i and the number of samples allocated per iteration N , the computational complexity of finding the global optimal solution can be significant.

The optimization problem is defined as:

$$\hat{\mathbf{n}}^{(i)} = \arg \max_{\mathbf{n}^{(i)} \in \mathbb{N}_{\text{global}}^{(i)}} \Pr \left\{ b^{(i)} < c^{(i)} \right\} + \frac{|\mathbb{O}|}{|\mathbb{O}|+1} \Pr \left\{ b^{(i)} = c^{(i)} \right\}, \quad (4.17)$$

where $\hat{\mathbf{n}}^{(i)}$ is the optimal tuple of samples taken in each channel in i -th iteration. It is solved by exhaustive search of the solution space, independently for each iteration i .

Iterative optimal solution: While the global method maximizes the probability of optimal channel selection in i th iteration, its samples allocation does not consider how the samples were allocated in previous iterations or the previous probability of success. As such this upper bound can result, *e.g.*, in the optimal cumulative number of samples allocated to decrease over consecutive iterations, *i.e.*, $\tilde{N}_l^{(j)} > \tilde{N}_l^{(i)}$ for $j < i$. This is not possible in a practical sensing system, when in each iteration a decision on samples distribution is made and cannot be changed (the historical allocation) in the next iteration. In each iteration only N samples can be allocated among L channels. Therefore, in i -th iteration ($i > 1$), the solution space is defined as a set $\mathbb{N}_{\text{iter}}^{(i)}$ of $\binom{L-1+N}{L-1}$ L -tuples $\mathbf{n}_{\text{iter}}^{(i)}$. Each element of this

tuple is defined as $\hat{N}_l^{(i-1)} + N_l^{(i)}$ where $\hat{N}_l^{(i-1)}$ is the cumulative samples allocation $\tilde{N}_l^{(i-1)}$ over all previous iterations, that has been decided as a result of optimization in the previous iterations. The chosen allocation from i -th iteration over all channels, composed of elements $\hat{N}_l^{(i)}$, is denoted as a tuple $\hat{\mathbf{n}}_{\text{iter}}^{(i)}$.

In the case of the first iteration, $\lfloor \frac{N}{L} \rfloor$ samples are initially allocated to each channel. As such only $N - L \lfloor \frac{N}{L} \rfloor$ samples have to be optimally allocated among L channels. In this case, the solution space $\mathbb{N}_{\text{iter}}^{(1)}$ is composed of $\binom{L-1+N-L \lfloor \frac{N}{L} \rfloor}{L-1}$ tuples.

The optimization problem is defined as:

$$\hat{\mathbf{n}}_{\text{iter}}^{(i)} = \arg \max_{\mathbf{n}_{\text{iter}}^{(i)} \in \mathbb{N}_{\text{iter}}^{(i)}} \Pr \left\{ b^{(i)} < c^{(i)} \right\} + \frac{|\mathbb{O}|}{|\mathbb{O}|+1} \Pr \left\{ b^{(i)} = c^{(i)} \right\}. \quad (4.18)$$

Most importantly, in this case optimization has to be carried iteratively, each time increasing i by 1, as the solution space $\mathbb{N}_{\text{iter}}^{(i)}$ depends on $\hat{\mathbf{n}}_{\text{iter}}^{(i-1)}$, being the resultant samples allocation in the previous iteration. The computational complexity of solving this problem is moderate as the solution space $\mathbb{N}_{\text{iter}}^{(i)}$ does not scale with the iteration index i , but depends only on the total number of samples assigned per one iteration N , assuming a fixed number of observed channels L .

While implementing the iterative solutions, it is important to consider that there might be many optimal solutions, *i.e.*, having the same upper bound of probability of success in i -th iteration. However, depending on the choice in i -th iteration, the achievable upper bound can vary in next iterations. The suggested solution is to evaluate in parallel all the equally optimal solutions till some of them become better than others in the next iterations.

4.1.2 Analysis of Optimal Allocation and Proposal of Unequal Samples Allocation

To observe how the optimal allocation differs from equal allocation, computer simulations have been performed. The following arbitrary parameters have been selected: four channels ($L = 4$) with $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$ and $\beta_4 = 0.8$. Samples distributions were derived using both the iterative and global methods presented in Section 4.1.1. Both upper and lower bounds for both optimal sampling methods are shown in Figure 4.1 for $N = 8$. Both UB and LB are very close to each other and visibly overlap after the third

iteration. As expected, the probability of optimal channel selection increases with a growing number of iterations. Simulation results are also presented, where in each case 100000 independent, random runs are performed to obtain sufficient statistical correctness. The simulation result is bounded by the iterative optimal UB and LB; this is as a partial confirmation of the correctness of the derived bounds. However, this optimal samples allocation approach is obtained knowing β_i values. Without this knowledge, a standard solution is *equal allocation*, meaning there are two new samples (for the considered case, when $N = 8$ and $L = 4$) obtained for each channel in each iteration. The equal allocation is visibly worse than an optimal allocation, obtaining probability of best channel selection of 0.9 nearly 6 iterations after such a success probability is achieved for the optimal samples allocation. Equivalently, the 0.9 probability of optimal channel selection will be achieved after 60% more time for the equal allocation algorithm.

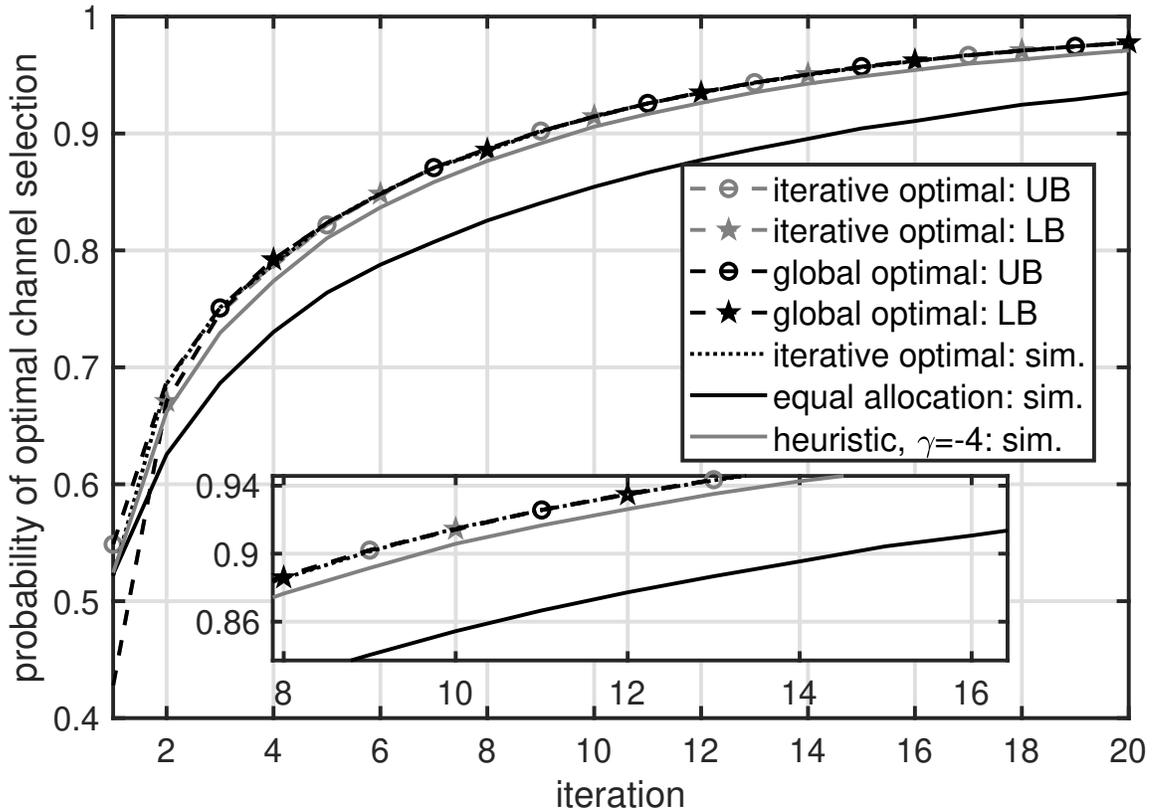


Figure 4.1: Probability of optimal channel selection versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 8$ [28].

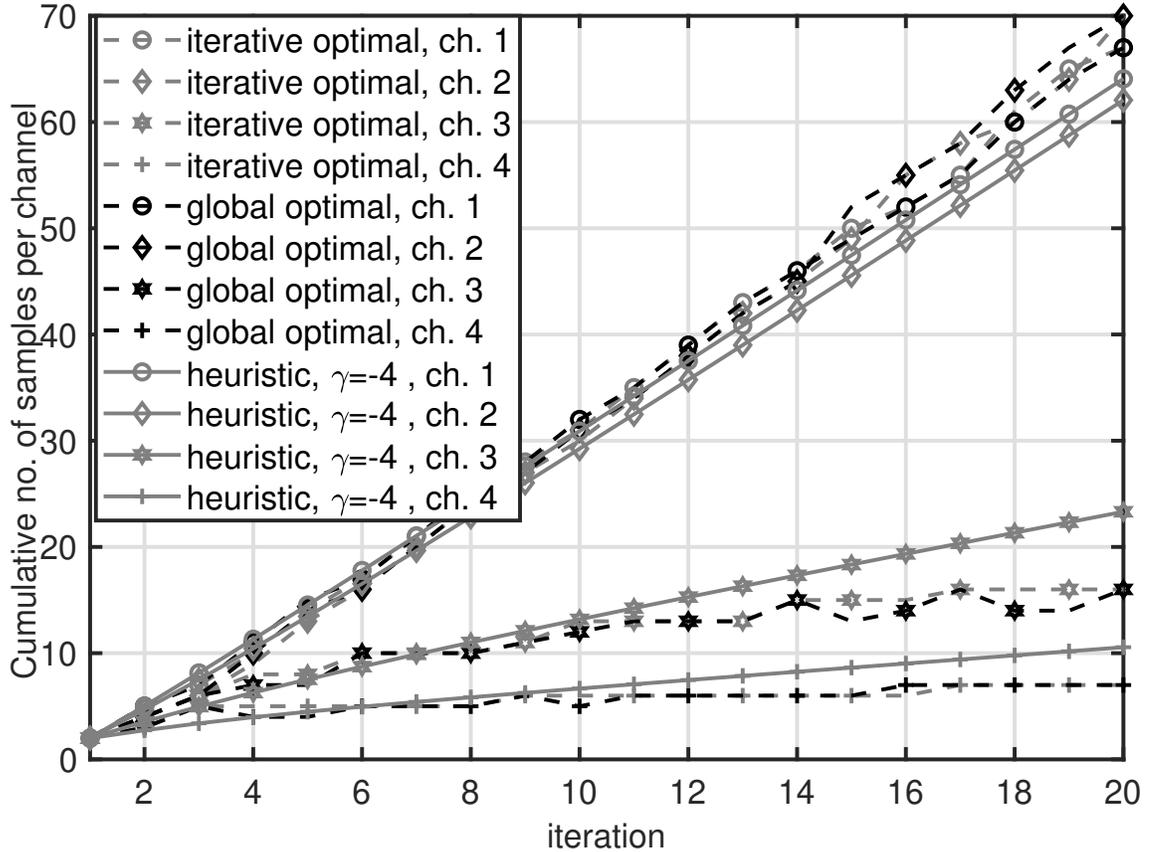


Figure 4.2: Cumulative number of sampling moments per channel (ch.) versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 8$ [28].

Let us analyze the cumulative number of samples per channel taken for the optimal allocation schemes. It is shown for the above defined set of parameters in Figure 4.2. Most interestingly, it rises nearly linearly for each channel, both for iterative and global optimal methods. However, the highest rise in number of samples is observed for the optimal channel (ch. 1) and the wrong channel of the lowest CBR (ch. 2). After 20 iterations each of these channels was sampled nearly 70 times (approximately 3.4 samples per iteration out of 8 available). At the same time, the third and fourth channels are sampled with approximately 0.8 and 0.35 samples per iteration, respectively. This result provides motivation to define an unequal samples allocation heuristic:

$$N_l^{(i)} \approx \frac{n_l^{(i)}}{\sum_{q=1}^L n_q^{(i)}} N, \quad (4.19)$$

where:

$$n_l^{(i)} = \begin{cases} \exp\left(\gamma \hat{\beta}_l^{(i-1)}\right) & \text{for } l = \hat{l} \\ \exp\left(\gamma \hat{\beta}_l^{(i-1)}\right) & \text{elsewhere,} \end{cases} \quad (4.20)$$

\hat{l} is obtained according to (4.4), \tilde{l} is the channel index of the second-best estimated channel, *i.e.*, $\tilde{l} = \arg \min_{l \in \{1, \dots, \hat{l}-1, \hat{l}+1, \dots, L\}} \hat{\beta}_l^{(i-1)}$ and γ is a design parameter. The approximation in (4.19) is required as the right-hand side values are not necessarily integers, and can require rounding to obtain $N_l^{(i)}$ summing up to N . The expected range for γ is from $-\infty$ to 0. While for $\gamma = 0$ an equal allocation is obtained, the lower the γ value, the higher portion of samples are expected to be assigned to the channels of relatively low estimated CBR.

As an initial test, this heuristic is simulated for the system configuration as mentioned above, and arbitrarily chosen $\gamma = -4$. The results are depicted in Figures 4.1 and 4.2. The probability of the best channel selection is very close to the derived upper bound, delayed approximately by a half of an iteration at the 0.9 probability level. At the same time, the cumulative number of samples per channel, averaged over all simulation runs, resembles the plots for global optimal solution.

To show that the proposed scheme works well even for sparsely sensed channels, the number of samples available per each iteration is reduced to $N = 6$. The resultant probability of the best channel selection as a function of iteration is shown in Figure 4.3. First, let us focus on the global and iterative UB and LB. The gap between the UB and LB is the highest for the first iteration and equals about 0.13, but it rapidly decreases with increasing iteration number. One interesting observation is the separation of iterative optimal bounds and global optimal bounds between sixth iteration up to the twentieth iteration. While both optimal solutions achieve probability of the best channel selection equally fast (around 0.8), the iterative solutions apparently reach a plateau at this level. This behaviour is captured by simulations, *i.e.*, “iterative optimal” series. Interestingly, around the tenth iteration the equal allocation strategy outperforms the iterative optimal solution. The greedy solution, *i.e.*, the maximization of probability of the best channel selection in each iteration, results in a temporal deadlock. Thus we conclude that there cannot exist a samples allocation strategy that always achieves the global UB in all iterations for all possible system configurations.

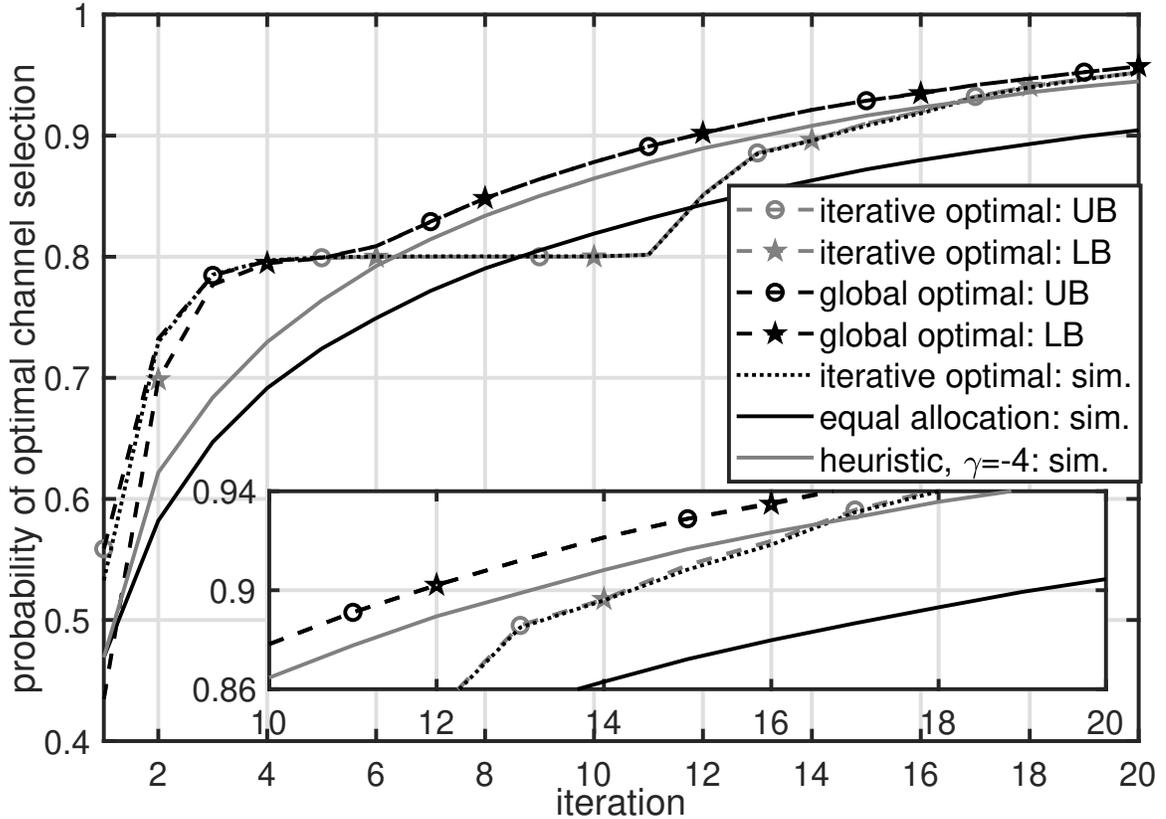


Figure 4.3: Probability of optimal channel selection versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 6$ [28].

From this perspective, it is positive that the proposed heuristic still performs well, close to the global optimal solution for higher number of iterations. While the best channel is selected with probability 0.9 after 12 iterations for the global optimal solution, the system utilizing the proposed heuristic requires one more iteration. The same reliability is achieved with equal sample allocation after 19 iterations.

A justification of the plateau in probability of the best channel selection can be found by looking at the cumulative number of sampling moments per channel, given in Figure 4.4. Until the tenth iteration in the iterative optimal solution, channel no. 1 is sampled only once. In 80% of cases this results in the estimated CBR, *i.e.*, $\hat{\beta}_1^{(i)}$, being equal to 0 as $\beta_1 = 0.2$. While the other, heavily sampled channels can achieve a zero-valued CBR with low probability, the optimal channel is successfully selected in around 80% of cases. However, to achieve the probability of optimal channel selection higher than 0.8, the first channel sensing

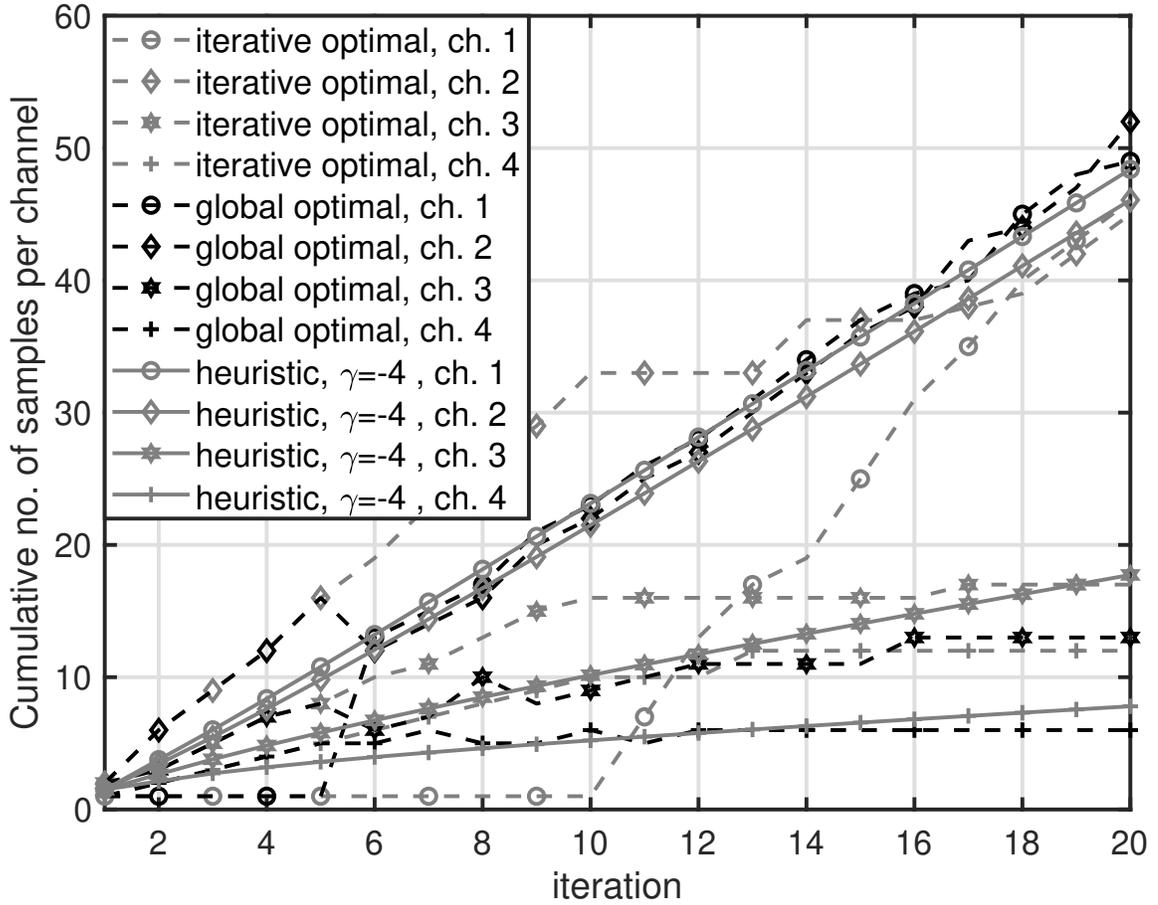


Figure 4.4: Cumulative number of sampling moments per channel versus iteration for $\beta_1 = 0.2$, $\beta_2 = 0.35$, $\beta_3 = 0.6$, $\beta_4 = 0.8$ and $N = 6$ [28].

scheme has to be changed radically. It is visible between the fifth and sixth iteration for global optimal solution when the cumulative number of samples for the first channel rises from 1 to 13. At the same time, the number of samples drawn from the second channel falls from 16 to 12. While this is globally optimal, it cannot be achieved with an iterative solution that cannot decrease the number of cumulative past samples per channel while proceeding with iterations. After the tenth iteration, the first channel starts to be sampled the most often in the case of an iterative optimal solution (visible as the highest gradient of values). Most importantly, the proposed heuristic approximates the global optimal solution well after the sixth iteration. This result suggests it can be used to improve the probability of optimal channel selection even for a limited number of samples per iteration N .

The above discussion shows that the proposed heuristic works well in the two considered system configurations, *i.e.*, sets of β_i , N , and L values. Consistency of this behavior should be checked for another system configurations. Additionally, a recommendation regarding choice of γ value should be provided. To do so, a set of 26 different N and L values were considered. The focus was on systems where the numbers of available samples per iteration is small, as this is the most challenging scenario for platooning using VDSA. For each of these L and N configurations, 200 random β_i sets (each of length L) were chosen with $\beta_i \in \{0, 0.1, \dots, 1\}$. The limited set of possible CBR values mimics the discrete character of CBR introduced by each vehicle transmitting in a given channel, and allows faster convergence of the simulations. Finally, for a given set of N , L and β_i parameters 100000 random runs were carried out to obtain the probability of optimal channel selection as a function of iteration index, *i.e.*, results similar to the ones presented in Figure 4.3.

For the equal sampling and unequal sampling with $\gamma \in \{-1, -2, -4, -8, -16\}$ the iteration at which the probability of optimal channel selection reaches 0.95 is stored. While the absolute number of iterations required can vary significantly between various system configurations, a normalization has been applied. That is, the number of iterations required for a given γ value is divided by the number of iterations required in equal allocation strategy. The Cumulative Density Function (CDF) of this metric is presented in Figure 4.5, where the results are estimated over all 5200 simulated system configurations. We observe that for several system configurations, x equals 0.4, *i.e.*, the required probability of the best channel selection is achieved in 40% of iterations required for the equal allocation algorithm in the case of $\gamma = -8$ or $\gamma = -16$. For 50% of cases, the best solution is $\gamma = -4$, for which the required detection quality is obtained 30% faster ($x = 0.7$) than with the equal allocation strategy. However, for all these γ values there is a small percent of system configurations resulting in a normalized number of iterations required greater than 1 ($x > 1$), *i.e.*, equal sampling is a better solution in these cases.

The highest probability of this event is obtained for $\gamma = -16$ and equals 14%. Thus, the recommended solution is to use $\gamma = -2$ since it obtains the lowest probability of being outperformed by the equally samples allocation approach, *i.e.*, occurs in 1.4% of system configurations. At the same time, this approach achieves the required probability of optimal

channel selection at least 14% faster ($x = 0.86$) across the 50% of system configurations. For this performance indicator, the recommended sampling is slightly worse than samplings with $\gamma = -4$, $\gamma = -8$ or $\gamma = -16$, but significantly better than sampling with $\gamma = -1$.

Table 4.1: Set of simulated N and L configurations

| L | N |
|-----|---------------------|
| 3 | 3,4,5,6,9 |
| 4 | 4,5,6,7,8,12 |
| 5 | 5,6,7,8,9,10,15 |
| 6 | 6,7,8,9,10,11,12,18 |

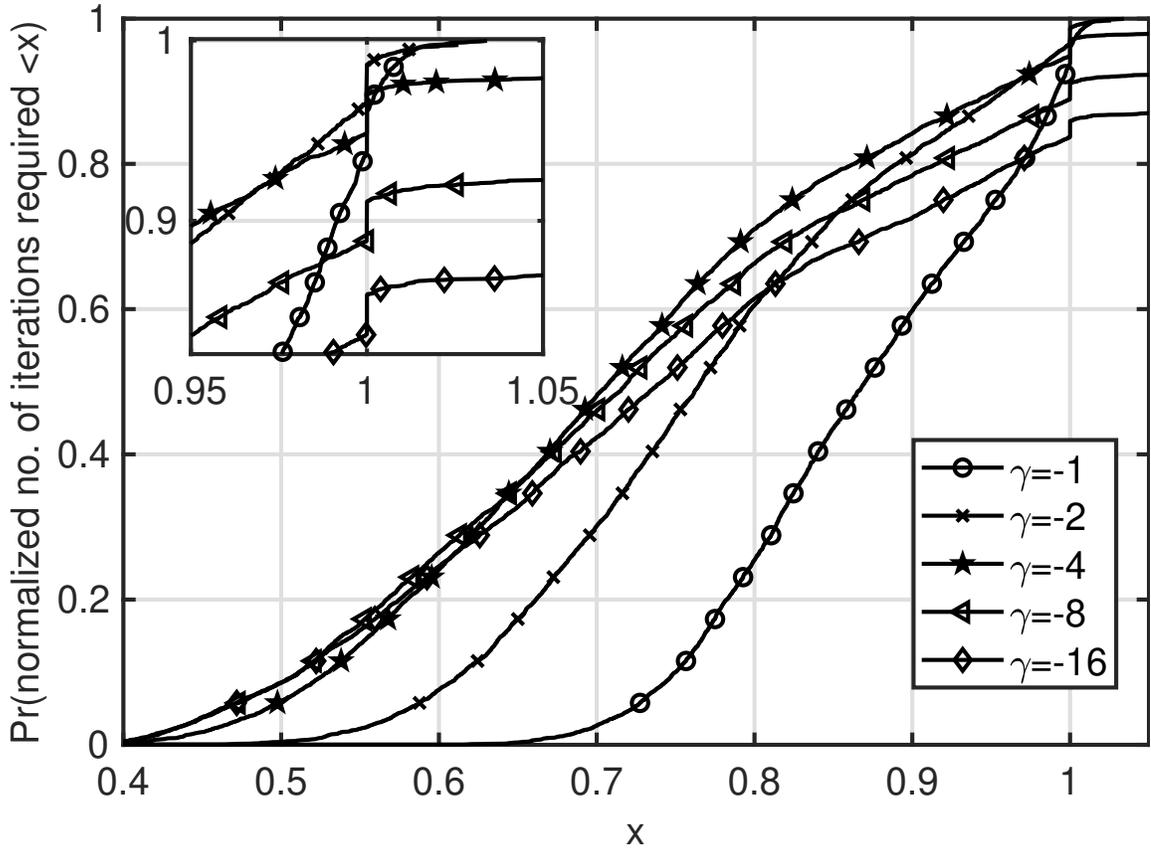


Figure 4.5: CDF of normalized number of iterations required to obtain probability of best channel selection equal 0.95 over 5200 random system configurations (varying N , L and β_i). Normalization over number of iterations required by equal allocation [28].

In the upcoming sections, if not stated differently, the unequal sampling algorithm will use $\gamma = -2$.

4.1.3 Bumblebee Behavior-based Channel Sensing and Selection in Time-varying Environment with Channel Switching Cost

In Section 4.1.1, we looked at the unequal sampling algorithm for fixed CBR and provided an analytical derivation on the optimal sampling distribution among the sensed frequency channels. In this section, we will evaluate the performance of the considered channel sampling scheme applied in the time-varying CBR scenario. To address the time-varying environment, the knowledge of the prior values of the CBR that are stored in a memory will be utilized. Two types of memories will be directly implemented in the bumblebee behavior-based algorithm proposed for the channel selection process. From the implementation perspective, the time-varying CBR is generated using the system-level V2V simulator, described in Section 4.2, and imported into the analytical simulation framework to evaluate the memory enabled bumblebee algorithm.

4.1.4 Memory-based Bumblebee Algorithm Description

We concentrated on a fully distributed scenario where each car makes its own decision regarding the selection of the best frequency channel for V2V communications. However, we would like to simultaneously benefit from access to knowledge on prior system states to improve decision making. These assumptions are based on bumblebee foraging behavior, where the insects make decisions to select the best flowers while collecting pollen. The decision to be made by each car is either to switch the channel or to stay at the currently selected one. We denote the currently (*i.e.*, at iteration i) selected channel by a given car for data transmission as $\hat{l}(i)$. Next, the set of all remaining channels (not used currently by this car) at iteration i is denoted by $\hat{\mathcal{O}}(i) = \mathcal{O} - \hat{l}(i)$. Then, the original bumblebee behaviour-based channel selection is given by comparing the CBR with the candidate channel $l^* \in \hat{\mathcal{O}}(i)$, for which the best channel reward is achieved (the lowest CBR, *i.e.*, $l^* = \arg \min_{l \in \hat{\mathcal{O}}(i)} \beta_l^{\hat{l}(i)}$).

In particular, the following decision is done:

$$\hat{l}(i+1) = \begin{cases} l^*, & \text{for } \bar{\beta}_{\hat{l}(i)}^{(i)} \geq (\bar{\beta}_{l^*}^{(i)} + \chi) \\ \hat{l}(i), & \text{for } \bar{\beta}_{\hat{l}(i)}^{(i)} < (\bar{\beta}_{l^*}^{(i)} + \chi) \end{cases}, \quad (4.21)$$

where $\bar{\beta}_{\hat{l}(i)}^{(i)}$ is the current channel cost (*i.e.*, observed CBR), $\bar{\beta}_{l^*}^{(i)}$ is the cost for channel $l^* \in \hat{\mathbb{O}}(i)$ and χ is the switching cost for the new channel. The purpose of the switching cost is to avoid frequent channel changes or even the so-called “ping-pong” effect, where the algorithm switches between two channels at each iteration. The higher the value of χ , the less dynamic the channel selection. Furthermore, as the number of stored sensing samples needs to be finite, we redefine the CBR estimate for channel l , comparing to (4.3), as follows:

$$\bar{\beta}_l^{(i)} = \begin{cases} \frac{\sum_{j=i-J}^i k_l^{(j)}}{\sum_{j=i-J}^i N_l^{(j)}} & \text{if } \sum_{j=i-J}^i N_l^{(j)} > 0 \\ \bar{\beta}_l^{(i-1)} & \text{if } \sum_{j=i-J}^i N_l^{(j)} = 0 \end{cases}, \quad (4.22)$$

where J is an arbitrarily selected number of sensing iterations.

In an original, memoryless bumblebee algorithm, the car will make channel switching decisions based on instantaneous values of the CBR for candidate channels (*i.e.*, the channels from the set \mathbb{O}). However, such potentially frequent switching can lead to the performance degradation due to the channel-switching hardware-lag, synchronization of the devices, and control signaling. As discussed previously, such cost has to be considered in the switching decision process, and was expressed by χ in (4.21). However, if the vehicle has access to the past values of CBR, the performance can be improved by using memory. Consequently, the problem of instantaneous and permanent channel switching may be mitigated, as the change of the frequency band will be done in cases when the new channel opportunity is significantly better (as it would result from the prior channel assignments). Both sliding window average (SWA) and exponentially weighted moving average (EWMA) memory models are employed via a heuristic approach (4.20). The SWA model can be represented as:

$$\bar{\beta}_l^{(i)\dagger} = \frac{\sum_{j=1}^K \bar{\beta}_l^{(i-j)}}{K}, \quad (4.23)$$

where $\bar{\beta}_l^{(i)\dagger}$ is the sliding window average for candidate channel l over last K past $\bar{\beta}_l^{(i-j)}$ ($j = 1 \dots K$) values observed in i -th time interval.

Next, the EWMA memory model is defined as:

$$\bar{\beta}_l^{(i)\ddagger} = \begin{cases} \beta_l^{(1)}, & i = 1 \\ \alpha\beta_l^{(i)} + (1 - \alpha)\beta_l^{(i-1)}, & i > 1 \end{cases}, \quad (4.24)$$

where $\bar{\beta}_l^{(i)\ddagger}$ is the exponentially weighted average for l -th channel in i -th time period, and α is the forgetting factor. The higher α implies more weight is given to the recent sample in comparison to the past weighted values. Both memory models are tested with different parameters and their performance is compared with no-memory bumblebee algorithm.

Algorithm 3 describes the proposed memory-enabled bumblebee VDSA algorithm applied to the V2V communications network, where the non-uniform spectrum sensing decisions are made, and the estimated CBRs are calculated using a heuristic approach (4.20). The proposed solution consists of two key actions, mainly: (i) sampling Interval, and (ii) transmission interval. In the former action, one vehicle performs spectrum sensing in the considered frequency band (*e.g.*, TVWS) and facilitates opportunistic channel access. Being initialized with the randomly selected channel (from the available set \mathbb{O} , in the second phase the currently optimal channel (with the lowest CBR) is selected. We also assume that during the sensing interval all the candidate channels can be evaluated by different vehicles, whereas during transmission interval only sensing of current channel is feasible. Once the channel is selected, real data transmission is realized.

4.1.5 Simulation Results

We evaluated the performance of memory-enabled non-uniform sampling-based bumblebee algorithm by means of extensive computer simulations. First, the time-varying CBR values were generated in an accurate V2V simulator (developed in C++, [92, 93]), as illustrated in Figure 4.6. Here, the total sampling instants N is set to 8, and the total number of channels is set to $L = 4$. One can observe the CBR values for the observed channels in the considered period. For three channels the CBR values oscillate slightly around specific value (*i.e.*, around 0.3 for channel 1, 0.05 for channel 2, 0.6 for channel 4, and 0.9 for channel 4). Such a setup corresponds to the situation where, for example, one channel is highly occupied by currently ongoing transmissions, whereas the other is almost empty. Moreover,

a significant change in CBR is observed in channel 1, which illustrates a situation where, for example, another car utilizing the same band appeared for awhile (*i.e.*, when overtaking the car). In further simulations, two different types of memory models are employed with the bumblebee algorithm. For SWA, the β values are stored in memory, and the bumblebee algorithm makes the switching decision based on the sliding window average given by Eq (4.23). In the EWMA model, recent β samples are given higher weight based on the forgetting factor α .

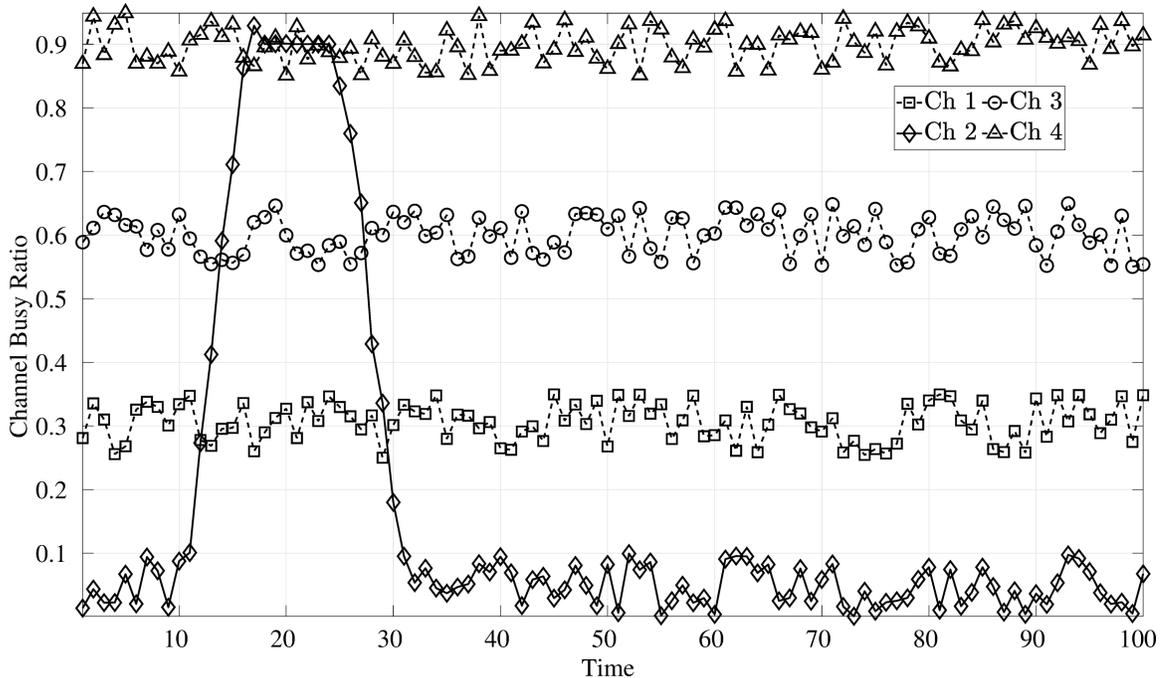


Figure 4.6: Time-varying channel busy ratio generated using the simplified birth-death process [94] with $L = 4$ [28].

First, the performance of the equal allocation approach is compared with novel heuristic-based unequal allocation scheme using the CBR values generated via the birth-death process. Figure 4.7 compares the two schemes with $\gamma = -2$, with the bar plot representing the gain difference. The gain is computed by taking the difference between the two schemes with the best channel selection probabilities as the performance metric. The gain values (in percentage points) are plotted over the right y -axis across simulation time.

Based on Figure 4.7, the heuristic approach has a consistent gain over equal alloca-

tion. This demonstrates the robust capability of the unequal heuristic approach which we proposed. Furthermore, data-driven models can be employed that consider γ as a hyperparameter that can be tuned based on the environment. This approach is outside the scope of the present dissertation, but may be studied in the future.

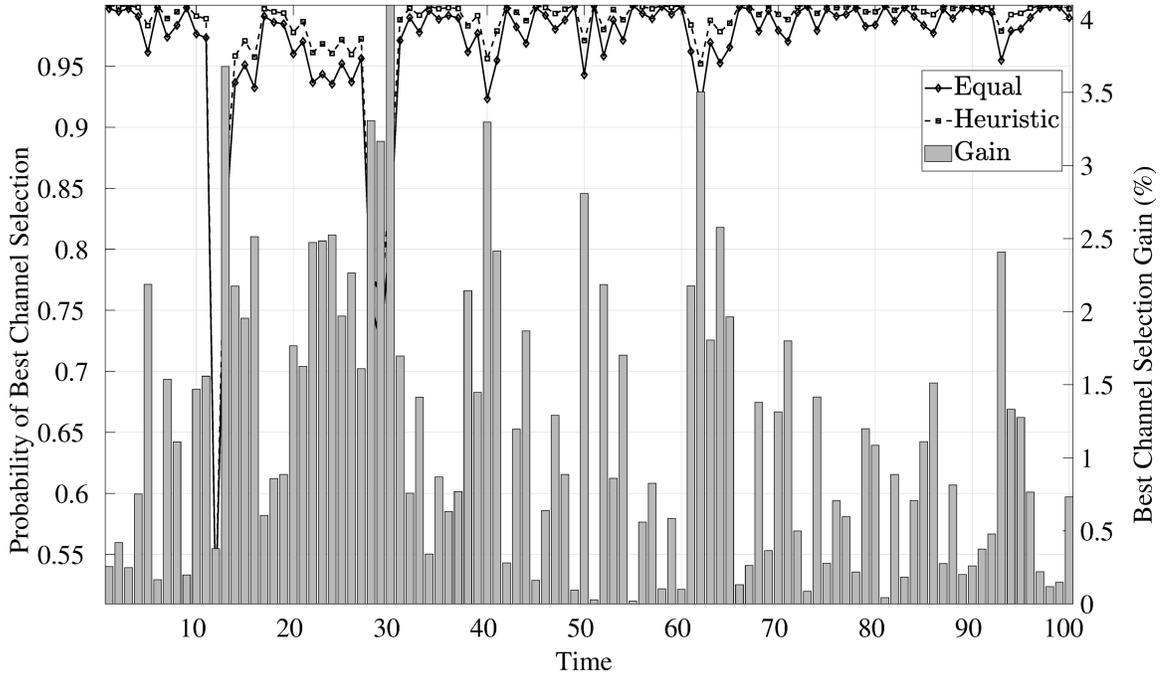


Figure 4.7: Performance comparison for equal sample allocation and the proposed heuristic approach for $\gamma = -2$. It shows that heuristic approach outperform equal allocation with no memory. The time from 10 – 30 shows a very high channel utilization where heuristic needs to adapt to the variation but due to no memory the performance is low [28].

In this work, we evaluate the best sliding window length K and the forgetting factor α values using analytical simulation. The simulation is performed with different forgetting factors and sliding window lengths, and the values which provide the best channel selection (%) are chosen to compare against the equal allocation. Figure 4.8 shows the comparison of EWMA and SWA memory schemes with different parameters, where $K = 4$ and $\alpha = 0.7$ provide the highest channel selection gain.

Figure 4.9 describes the advantage of adding memory in the system and exploiting the past values. Three different memory lengths are used, *i.e.*, $K = 2, 3, 4$, with sliding

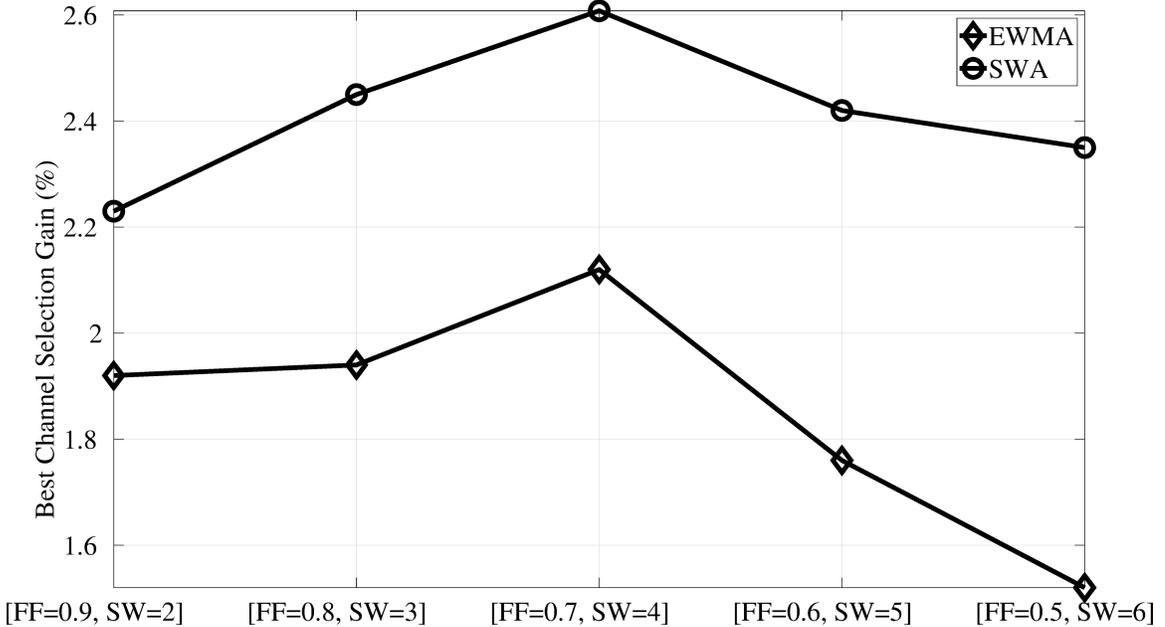


Figure 4.8: Best Channel Selection gain (%) is evaluated for different forgetting factors and sliding window memory length. The plot demonstrate the optimal values of both EWMA and SWA memory schemes computed using simulation [28].

window average compared against a memoryless scheme. It is observed that $K = 4$ provides the highest gain, with increases in the memory length (*i.e.*, $K > 4$) yielding a degradation in performance. The best channel selection gain on the right y -axis shows the gain in performance against the memoryless system.

In Figure 4.10, we evaluate the performance of an exponentially weighted moving average for $\alpha = 0.9, 0.8, 0.7$, and compare it against a memoryless scheme. It is observed that $\alpha = 0.7$ provides the highest gain with decreasing values of α below this threshold leading to diminishing performance. Similar to Figure 4.9, the best channel selection gain on the right y -axis shows the improvement versus memoryless system when operating with $\alpha = 0.7$.

4.2 Evaluation in Platooning Scenario

In this section, we present the results of an experimental evaluation of the proposed VDSA algorithm using extensive and sophisticated simulations of the autonomous car platooning scenario, where multiple vehicles follow the platoon leader using the Cooperative

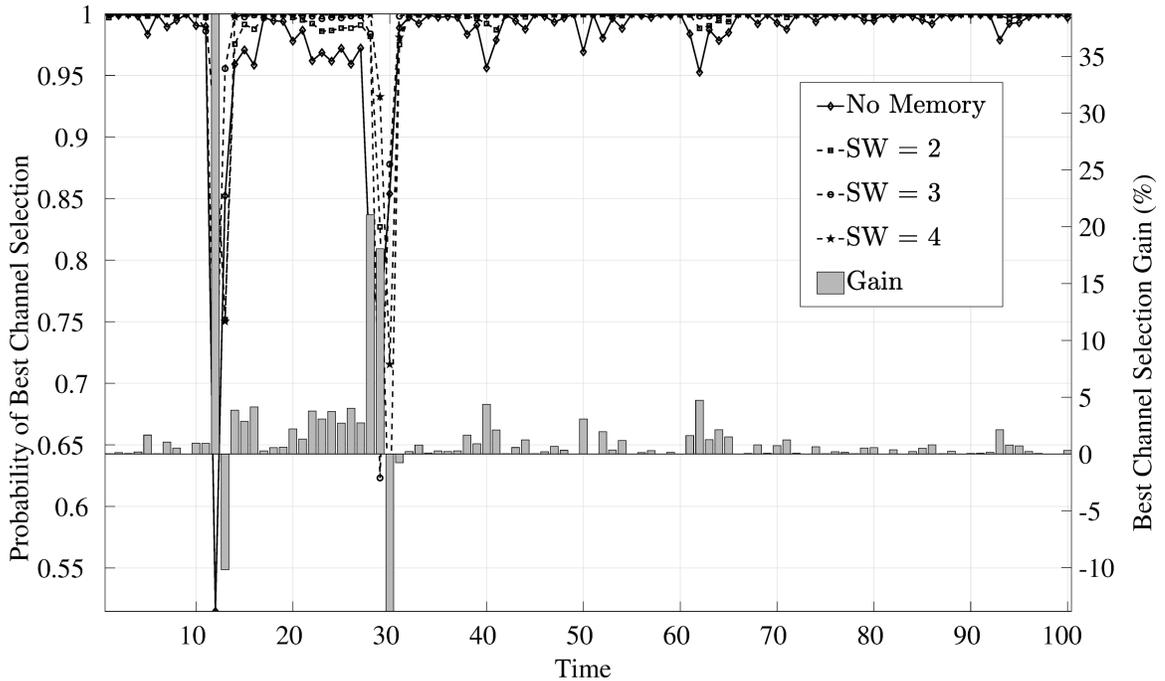


Figure 4.9: Performance comparison for heuristic approach with $\gamma = -2$ for different memory lengths and equal weight tapering. Memory length $K = 4$ was found to be optimal in simulation. The right axis shows the performance gain over memoryless case [28].

Adaptive Cruise Control (CACC) algorithm [63]. To facilitate proper CACC operation, platoon cars communicate with each other using the IEEE 802.11p protocol [91] in a dynamically selected frequency band, with the platoon leader broadcasting the mobility information to all platoon members, while the other vehicles transmit their position and movement information only to their followers. The bumblebee-based VDSA performance was evaluated using observations of the platoon leader packet success ratio (PSR), as the link between the platoon leader and its followers limits the performance of CACC operation. Furthermore, we observed the latency of selection of the best channel (*i.e.*, the delay in switching when the real CBRs facilitate the change of channel) for different channel sampling strategies.

4.2.1 Simulation Setup

To evaluate the performance of bumblebee-based VDSA for platooning, we considered a scenario with a 5 km section of a 6-lane motorway. A single platoon occupied the outer

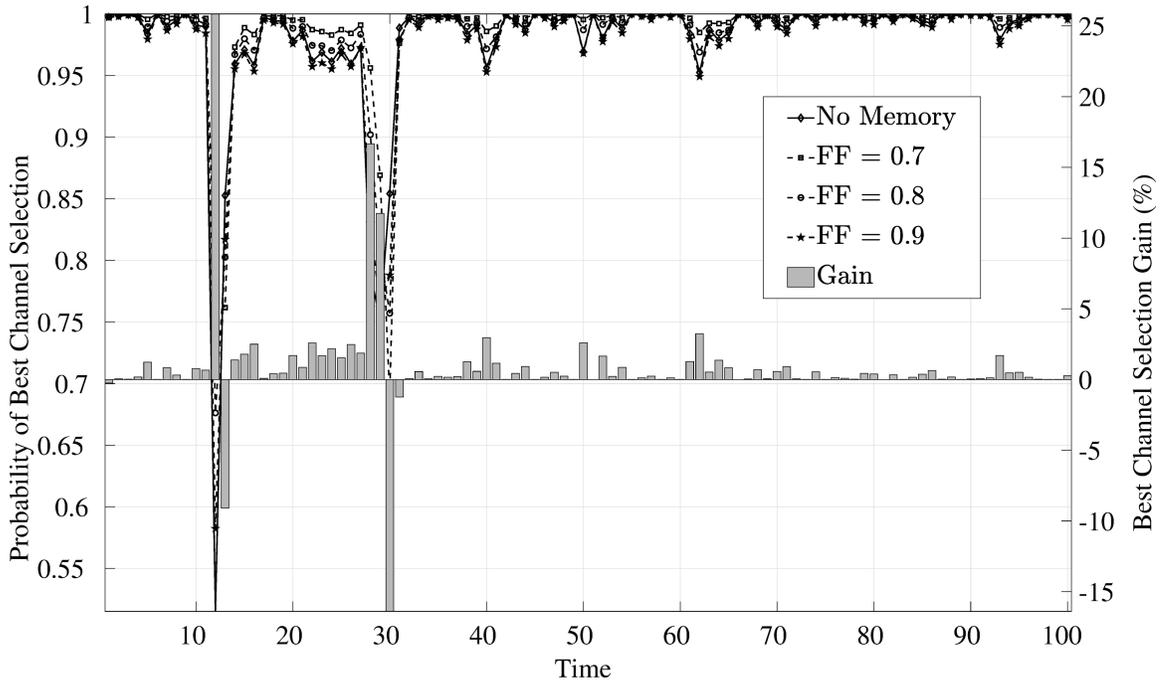


Figure 4.10: Performance comparison for the heuristic approach when $\gamma = -2$ with different exponentially weighted memory average tapering. $\alpha = 0.7$ appeared to be optimal in simulation. The axis on the right shows the performance gain over memoryless case [28].

lane, and the desired platoon inter-car spacing was 3 m. Two platooning configurations were evaluated:

- A platoon consisting of 4 cars (one leader and three followers) with an approximate platoon length of 25 m;
- A platoon comprising 10 vehicles with an approximate platoon length of 70 m.

Every platoon car transmitted its BSMV messages with a 100 ms period in the dynamically selected TVWS band, with each packet comprising of 300 bytes of data. There were four 10-MHz frequency bands available to switch between following the VDSA, with center frequencies of 490 MHz, 506 MHz, 522 MHz, and 536 MHz. The bumblebee-based VDSA procedure was applied periodically every 100 ms. Furthermore, each platoon car performed sensing of a single selected frequency band in a $32 \mu\text{s}$ interval after transmitting its own packet (with a delay between the end of transmission and the start of sensing selected randomly between 1 ms and 5 ms). The sensing results gathered in a 10 s observation window,

corresponding to $J = 100$ iterations of the VDSA algorithm, were used in the calculation of the CBR estimates for the bumblebee-based VDSA. Furthermore, the calculated CBR values were smoothed according to the EWMA rule, defined as in (4.24), with $\alpha = 0.7$. Two sensing channel selection methods were considered, with the band allocation for sensing following (4.19) presented in Section 4.1.2 for $\gamma = 0$ (uniform sampling) and $\gamma = -2$ (non-uniform sampling), respectively.

Apart from the platoon other cars occupied different lanes of the motorway. They were placed randomly following a uniform distribution with an average of 10 cars/km/lane. Each non-platoon car broadcasted the 300 B BSM messages every 100 ms in a randomly pre-selected frequency band (the band choice remained fixed for the whole simulation run duration), chosen from the set available for the platoon VDSA, with the probabilities of the available channels set as follows: $\{0.08, 0.28, 0.16, 0.48\}$. Furthermore, there were four roadside units (RSUs) placed along the motorway every 1 km, with the first and the last one transmitting 300 B messages every 2 ms in channel 1 (490 MHz), the second transmitting in channel 2 (506 MHz), and the third using channel 3 (522 MHz). These infrastructure transmitters contributed to the significant regional increase in the CBR for selected channels.

For both considered platoon configurations 10 independent simulation runs were performed, with the duration of a single simulation run set to 140 s. The main simulation parameters are summarized in Table 4.2.

4.2.2 Simulation Results for a Four Car Platoon

Simulations carried out with a 4-car platoon represent a scenario with a compact entity that performs sensing and VDSA. As the platoon vehicles follow one another closely, the whole platoon length is approximately 25 m. Thus variations in CBR measured by different platoon cars should not impact significantly the results. Figure 4.11 shows the evolution of the channel selected for transmission in the VDSA procedure for a specified simulation experiment. The correct channel is selected for both uniform and non-uniform sampling. Note

Table 4.2: Simulation parameters

| Parameters | Values |
|-------------------------------------|-----------------------------|
| Motorway length | 5 km |
| Number of cars in platoon | {4, 10} |
| Inter-car spacing in platoon | 3 m |
| Messaging periodicity | 100 ms |
| BSM message size | 300 B |
| VDSA frequency bands (channels) | {490, 506, 522, 536} MHz |
| VDSA procedure periodicity | 100 ms |
| Average number of non-platoon cars | 10 cars/km/lane |
| Non-platoon car channel probability | {0.08, 0.28, 0.16, 0.48} |
| Number and location of RSUs | 4 (@ 1, 2, 3, 4 km) |
| RSU messaging periodicity | 2 ms |
| Channels used by RSUs | (1, 2, 3, 1) |
| Number of simulations per scenario | 10 |
| Single simulation run duration | 140 s |

that the best available channel, corresponding to the choice with perfect CBR knowledge, is indicated with a solid blue line. However, when the number of samples collected per single VDSA algorithm iteration is relatively small (*e.g.*, 4 samples, which is equal to the number of vehicles in platoon), some latency in channel switching is observed. This latency is smaller in the case of non-uniform sampling, as more sensing slots are allocated to the more promising channels than for uniform sampling. Therefore, using non-uniform sampling enables the VDSA algorithm to switch faster to a better channel when CBRs change.

The ability to quickly detect the channel with the lowest CBR using non-uniform sampling impacts the performance of intra-platoon communications. Referring to Figure 4.12, we observe the estimated probability of successful reception of leader packets versus vehicle position in platoon averaged over all simulation runs. We observe a slightly higher probability that non-uniform sampling can enable cars 1 and 2 to switch faster to a better band when CBR values change. This phenomenon is further highlighted in Figure 4.13,

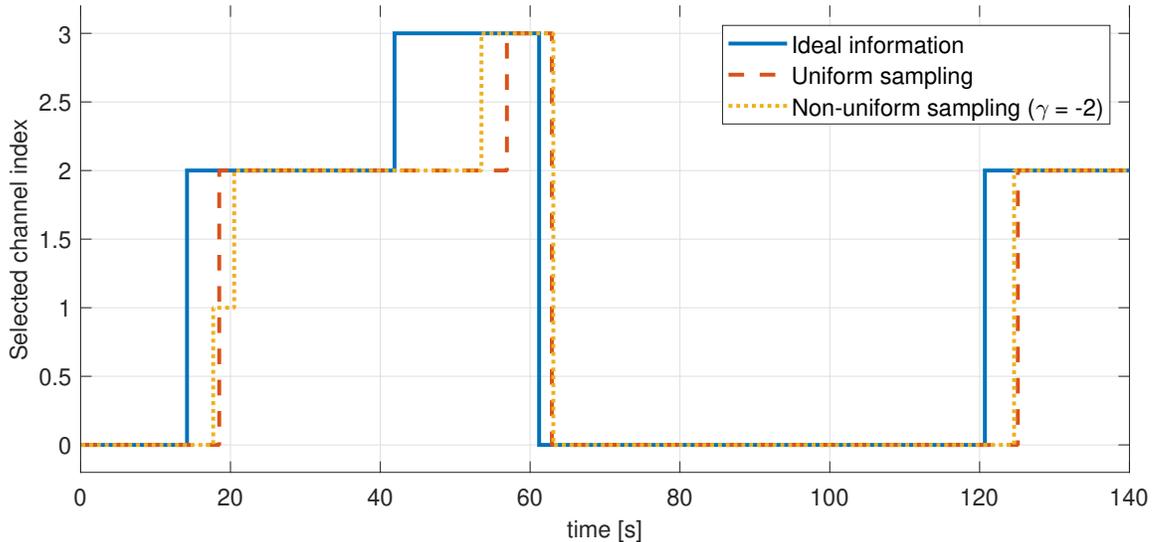


Figure 4.11: Selected channel (band) index versus time in the case of 4 vehicles platoon [28].

which presents the evolution of the successful reception of leader packets (measured in a moving 10 s observation window) versus the time for a selected simulation experiment. The reception rate drops in both cases around $t = 60$ s due to the increasing occupancy of the currently used channel. It should be noted that with non-uniform sampling the switch is performed faster, thus yielding a smaller drop in successful reception ratio.

4.2.3 Simulation Results for a Ten Car Platoon

The scenario with a 10-car platoon represents a more complicated situation, in which the platoon length is significantly larger (might exceed 70 m). In such a case, the sensing results obtained for different platoon vehicles might differ significantly, thus affecting the ability to select the lowest CBR channel in the VDSA procedure. This is clearly visible in Figure 4.15, which presents the evolution of the channel selected for transmission in the VDSA procedure in a selected simulation run. The VDSA algorithm switches frequently between bands in the uniform sampling case, and it is not able to find the best one (represented with a solid blue line) for the initial 60 s. This behavior results from the very limited number of collected samples. Only a single sample per car was collected in each iteration, thus causing the CBR values to be dependent on the location of a vehicle measuring the channel of interest. With non-uniform sampling, more sensing slots are allocated to the potentially more suitable

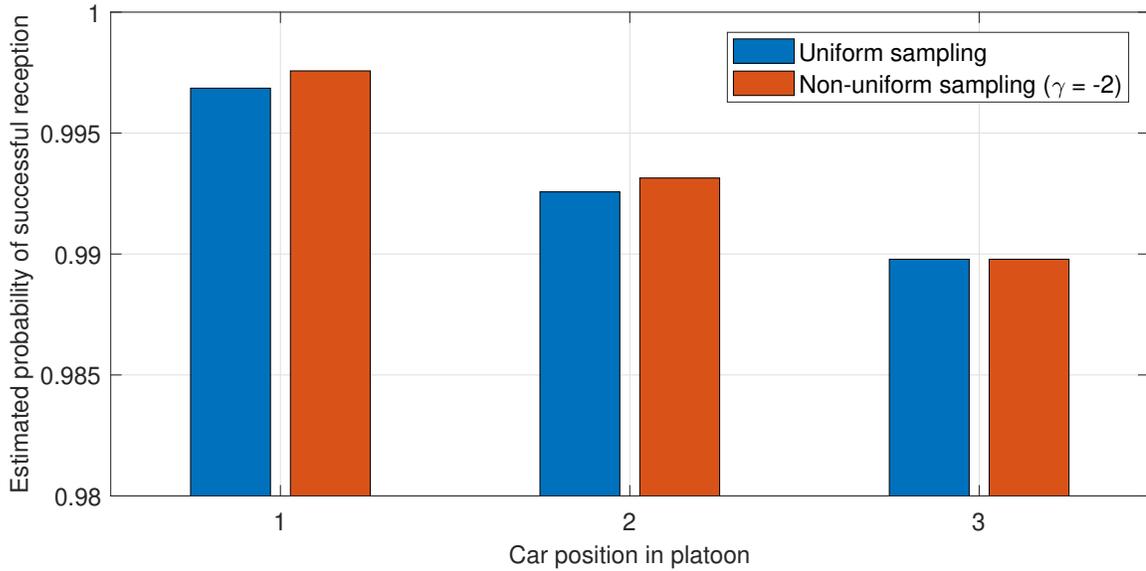


Figure 4.12: Estimates of the probability of successful reception of leader's packets versus car position in a 4-vehicles platoon [28].

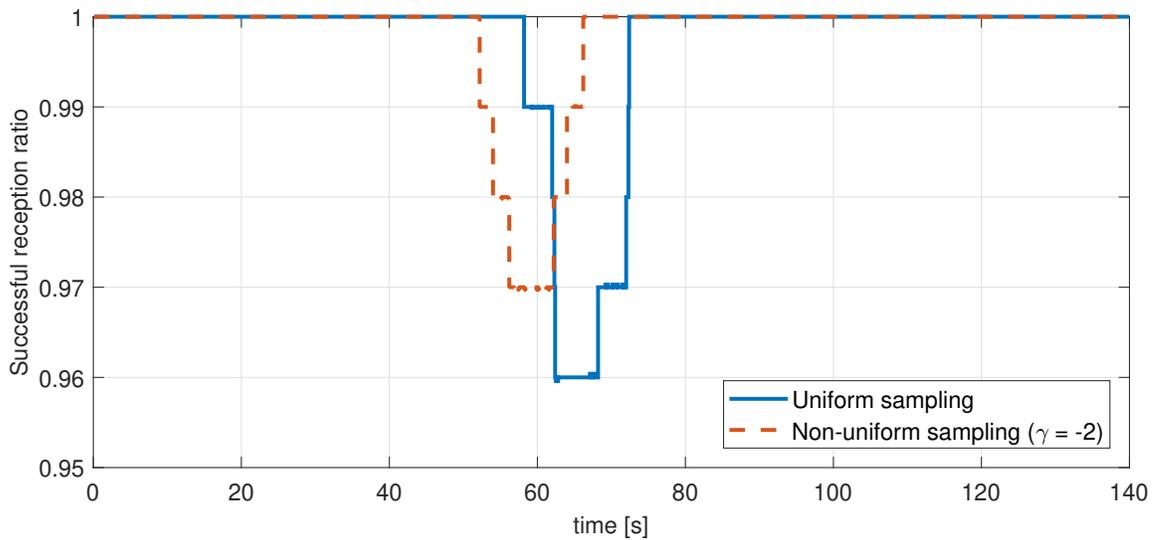


Figure 4.13: Example of time evolution of the successful reception ratio of leader's packets observed in a window of 10 s for car no. 2 in a 4-vehicles platoon [28].

channels, thus resulting in a higher level of accuracy of the CBR estimation. Thus, the selected channel with non-uniform sampling follows closely the reference (with perfect CBR knowledge) best channel, with only some latency introduced due to the averaging of results in a 10 s observation window.

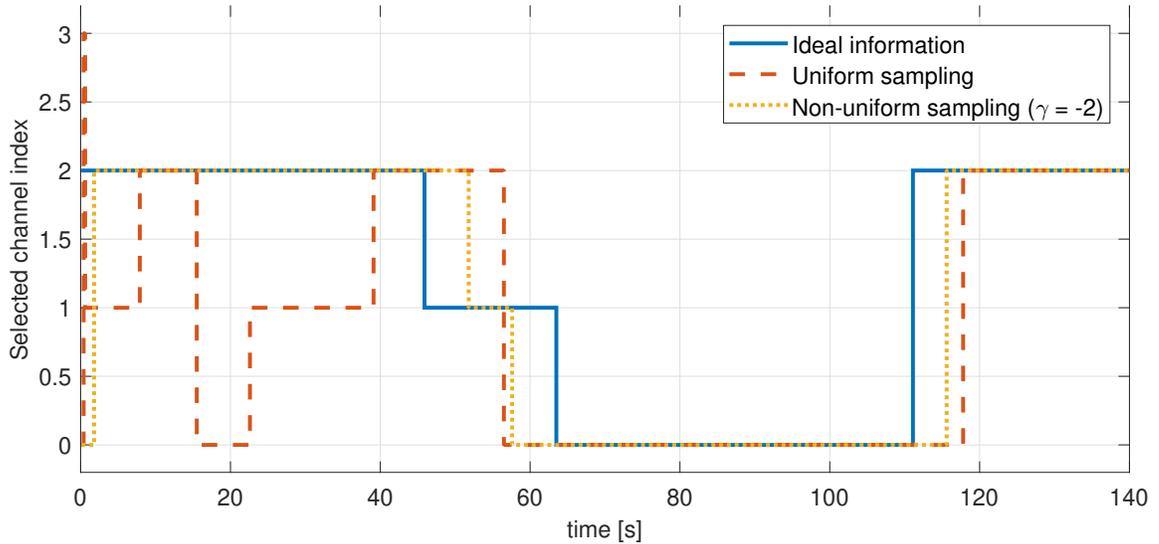


Figure 4.14: Selected channel (band) index versus time in the case of 10 vehicles platoon [28].

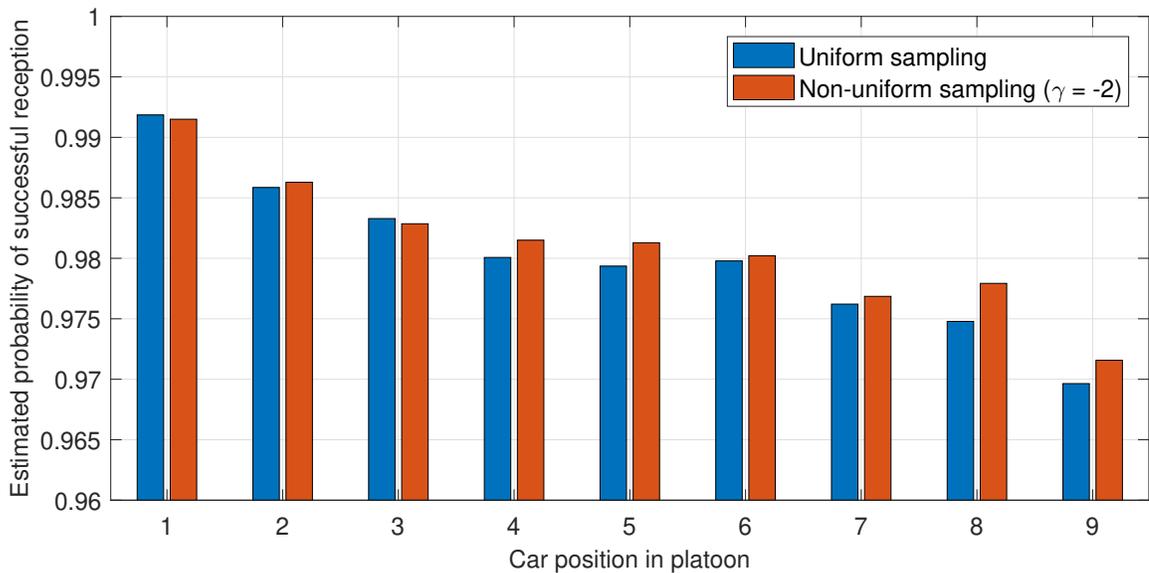


Figure 4.15: Estimates of the probability of successful reception of leader's packets versus car position in a 10-vehicles platoon [28].

The improved ability to find the lowest CBR channel with the non-uniform sampling approach is also reflected probability estimates for successful reception of the leader packets, as presented in Figure 4.15 (results averaged over 10 simulation runs). We observe that with non-uniform sampling the VDSA algorithm is capable of reducing losses of transmitted packets. Loss reduction is particularly apparent for cars at the tail of the platoon, where

the reception rate is usually lower due to higher channel attenuation.

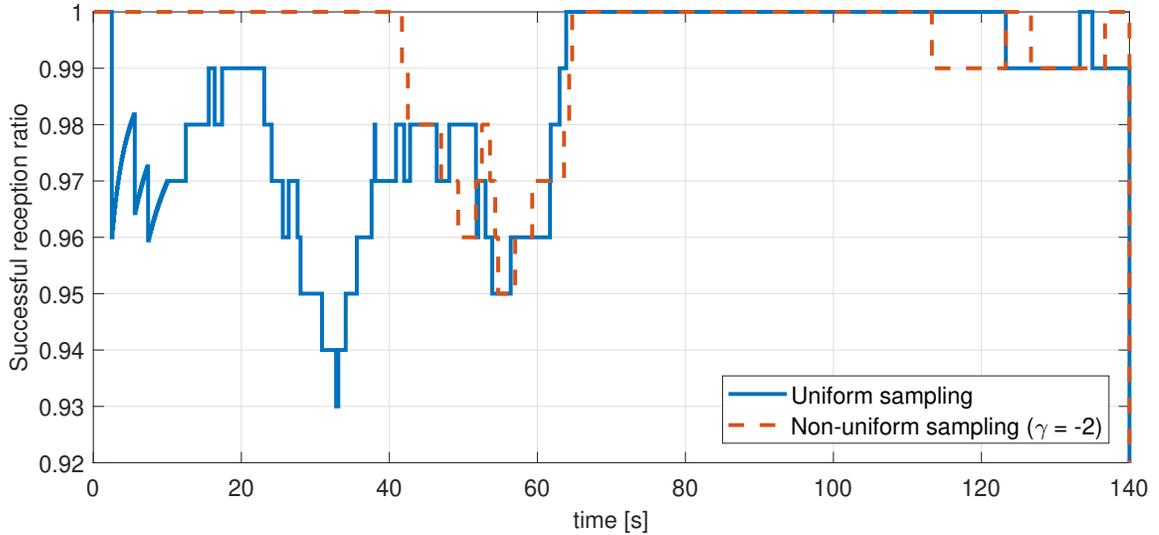


Figure 4.16: Example of time evolution of the successful reception ratio of leader’s packets observed in a window of 10 s for car no. 9 in a 10-vehicles platoon [28].

Figure 4.16 shows the evolution of the successful reception of leader packets (measured within a moving 10 s observation window) versus time for a selected simulation run. With uniform sampling, the VDSA algorithm is unable to find a channel guaranteeing sufficient quality of transmission for the initial 60 s, thus resulting in a temporary packet error rate over 5%, which is unacceptable for autonomous platooning. Overall, for a platooning scenario employing the bumblebee-based VDSA algorithm, the intra-platoon communication channels can be selected in a dynamic manner. However, the implementation needs to be provided with a fairly accurate set of CBR values. These values can be reliably obtained by providing a sufficient number of sensing samples for the prospective frequency bands. With non-uniform sampling the values are secured faster for the channels of interest (*i.e.*, the channels with low CBR) than with uniform sensing slots allocation.

4.3 Hardware Validation

Connected vehicles are capable of exchanging traffic environmental information via Basic Safety Messages (BSM) within a transmission range of 500 meters [7]. BSMs can contain

information such as current position, speed of the vehicle and direction, and can provide critical time-sensitive support for vehicular applications such as road safety systems and self-driving car ecosystems. Connecting vehicles by leveraging both wireless communication and networking solutions have been exhaustively studied, especially with respect to Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). The IEEE 802.11p Dedicated Short Range Communication (DSRC) standard [2] was the first framework designed to meet demands of the Vehicular Network (VANET) architecture. However, despite several of its initial advantages, IEEE 802.11p also possess a number of shortcomings such as low reliability, hidden node problem, unbounded delay and sporadic V2I connectivity [3]. Alternatively, the Third Generation Partnership Project (3GPP) specified an approach to vehicular connectivity called Cellular Vehicle-to-Everything (C-V2X) in Release 14, where the direct communication is supported between vehicles to accommodate the latency requirements for time-sensitive vehicular applications. C-V2X offers several features designed to support ITS applications with respect to coverage, mobility support, reliability, and scalability. In particular, 3GPP added two new modes, Mode 3 and Mode 4 in order to supplement the Proximity Services (ProSe) for V2V communications [5]. LTE enables wide coverage and data transmission reliability with low latency. For instance, LTE Mode 3 enables V2V communication via a sidelink channel with the support of an eNodeB *i.e.*, resources are scheduled with the help of base-station or centralized entity. In LTE Mode 4, vehicles can autonomously schedule resources using sensing-based semi-persistent scheduling (SPS) in an out-of-coverage scenario. Vehicles reserve the spectral resources that are sensed as vacant or with minimal interference in order to accommodate BSM messages. However, the dedicated LTE channels are unable to support cellular V2X communications in densely populated urban areas. To overcome the shortage of spectral resources, Vehicular Dynamic Spectrum Access (VDSA) has been proposed as a viable solution that leverages the under-utilized spectrum in licensed bands [95–97]. Without loss of generality, we will use digital television (DTV) spectrum from 470 MHz to 520 MHz for our experiment with VDSA-based C-V2X since the primary (licensed) users of this band are relatively stable when compared to other wireless frequency bands. The primary users of the DTV band possess a more predictable utilization of frequency bands. It is assumed that vehicles within the vicinity are designed

to individually detect the available channels for unlicensed usage [7, 23].

4.3.1 Bumblebee Model for C-V2X VDSA

Stochastic time-varying channel environments make it difficult to employ VDSA efficiently. Hence, a robust VDSA framework is required. The bumblebee-based VDSA algorithm is well-suited for independent decision-making, and hence can perform very well in a vehicular V2V communication environment. We also integrated our proposed bumblebee-based VDSA algorithm with a realistic C-V2X testbed and evaluated its feasibility for C-V2X.

Algorithm 3 Bumblebee-based VDSA Algorithm

- 1: **procedure** BUMBLEBEEVDSAALGORITHM(SC, \mathbf{T}, C, S)
 - 2: **Initialize:**
 Initialize random sub-channels at 2685 MHz
 - 3: **Sampling Interval :**
 - 4: **Radio:1** Perform spectrum sensing in DTV band for 1000 sub-frames
 - 5: Select the best sub-channels with energy $\leq \mathbf{T}$
 - 6: Configure **Radio:0** with selected sub-channels
 - 7: **Transmission Interval [Radio:0, Radio:1]:**
 - 8: Start the packet transmission
 - 9: Monitor the current SC set and keep storing energy values $E_i \in SC_i$
 - 10: **if** $\bar{S}_c \leq (\bar{S}_i - C)$ **then**
 - 11: Keep using same sub-channels
 - 12: **else**
 - 13: Switch to new sub-channels based on the memory
 - 14: **end if**
 - 15: **end procedure**
-

Algorithm 3 describes the bumblebee VDSA algorithm for cellular V2X in detail. In this work, we employed bumblebee-based VDSA algorithm in the Digital Television (DTV) band since the primary users of this band are not rapidly time-varying when compared to

other wireless frequency bands. Additionally, the primary users of the DTV band possess a more steady occupancy of the frequency band [98]. We implement the algorithm in two steps: (i) *Sampling Interval*, and (ii) *Transmission Interval*. For this algorithm, we make use of full duplex communications where two independently tunable radios are used. **Radio:0** performs the C-V2X communications using the LTE-Sidelink protocol stack with Frequency Division Duplexing (FDD), whereas **Radio:1** performs the spectrum monitoring simultaneously and independently of the LTE Sidelink. We initialize our VDSA model by assigning random SC resources to **Radio:0** C-V2X allocated at 2685 MHz LTE band 7 and start monitoring spectrum across the DTV band from 470 MHz to 520 MHz simultaneously for 1000 ms. Since wired connections are employed for V2V communications, LTE band 7 was used instead of 5.9 GHz ITS band. Once we have the energy values E for set SC based on the threshold value “**T**”, we sort them and select the best sub-channels based on the bandwidth requirements. We employed 10 MHz and 5 MHz bandwidths, and did not consider higher bandwidth values in order to reduce the computation load on our system.

Once **Radio:1** has the desired SC_i values, channel switching can be triggered, *i.e.*, **Radio:0** can switch from the 2685 MHz band 7 to the DTV band with the selected sub-channels. During the transmission interval, **Radio:1** keeps monitoring the entire DTV spectrum. It stores the time-domain energy detection values E for sub-channels in the memory and compares their mean values with the current sub-channels at periodic interval α . This periodic interval can be tuned based on the environment, *i.e.*, for rural highway where the environment does not change that often the value can be kept large whereas for urban environment the low value is needed to compensate for high channel variation. If the mean energy values for the remaining sub-channels is greater than the current sub-channels, **Radio:0** keep on using the same sub-channels or else it switches to the new sub-channels as selected by **Radio:1**. Equation (4.25) describes the switching decision, where \bar{S}_c is the mean signal strength for the current sub-channels, \bar{S}_i is the mean energy values of sub-channels which were selected from the channel set.

$$\begin{array}{l} \text{Switching} \\ \text{Decision} \end{array} = \begin{cases} \bar{S}_c \leq (\bar{S}_i - C), & \text{“Stay”} \\ \text{Otherwise,} & \text{“Switch”} \end{cases} \quad (4.25)$$

Switch cost C helps in reducing the unnecessary sub carrier switching as these can adversely affect the performance due to resource selection and allocation delays.

4.3.2 OpenAirInterface-based Experimentation Testbed

The experimental testbed for evaluating the VDSA performance for bumblebee algorithm was implemented using the OpenAirinterface (OAI) [99–101] LTE platform. OAI is an open-source LTE experimentation and prototyping platform created by the Mobile Communications Department at EURECOM for realistic 4G/LTE research. OAI is a fully open-source software-defined radio (SDR) based solution that implements complete LTE protocol stack using software. All elements of a LTE architecture: eNodeB (eNB), User Equipment (UE), and core-network (CN) are implemented for a standard Linux-based PC interfaced with a USRP SDR platform. For this testbed, we are using the LTE-Sidelink [102] branch of the OAI project. LTE-Sidelink provides support for two Device-to-Device (D2D) scenarios.

- *Off-Network Scenario*, where none of the UEs are connected to any network infrastructure and need to perform the scheduling assignment autonomously. In this work, the scheduler allocates the resource blocks randomly for a given bandwidth.
- *Relay-Network Scenario*, where at least one UE is connected to the network and acts as a relay for the data traffic from/to the out-of-coverage UE(s). The scheduling assignment for the in-coverage scenario is performed by the base station.

We will be focusing only on the scenario 1 for this dissertation, where both the UEs are outside the coverage area and they use sidelink communications to facilitate V2V. Figure 4.17 shows the complete protocol stack for the LTE V2V as implemented on the OAI platform. We integrated our bumblebee-based VDSA in the PHY and MAC layers of the testbed to make it a full-duplex UE that can simultaneously sense DTV channels (*Radio:1*)

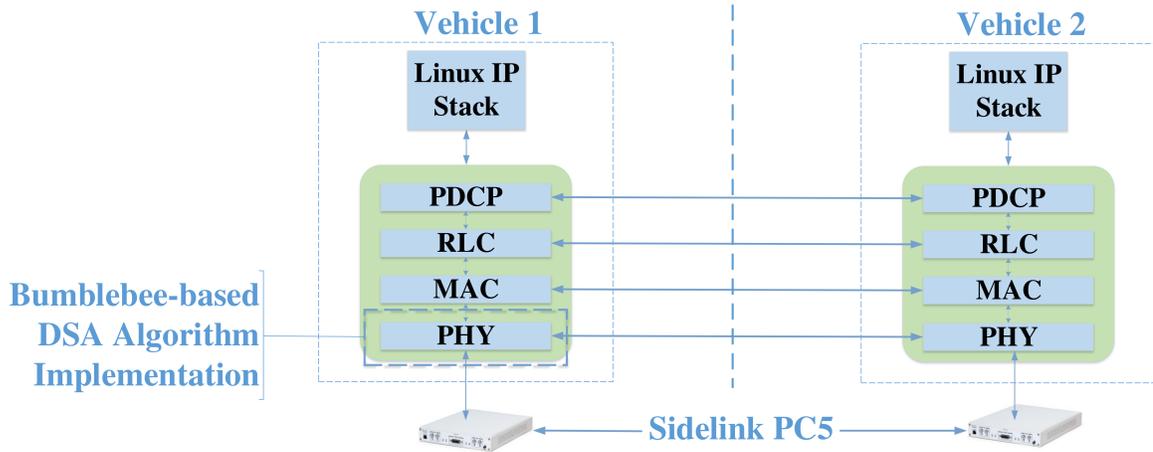


Figure 4.17: C-V2X Protocol Stack implemented on OAI testbed. The PHY layer is modified to integrate the bumblebee-based VDSA algorithm while reusing the higher layer setup from OpenAirinterface. For the C-V2X Mode 4, random selection is employed on the PSSCH (Physical Sidelink Shared) at a given channel pool [28].

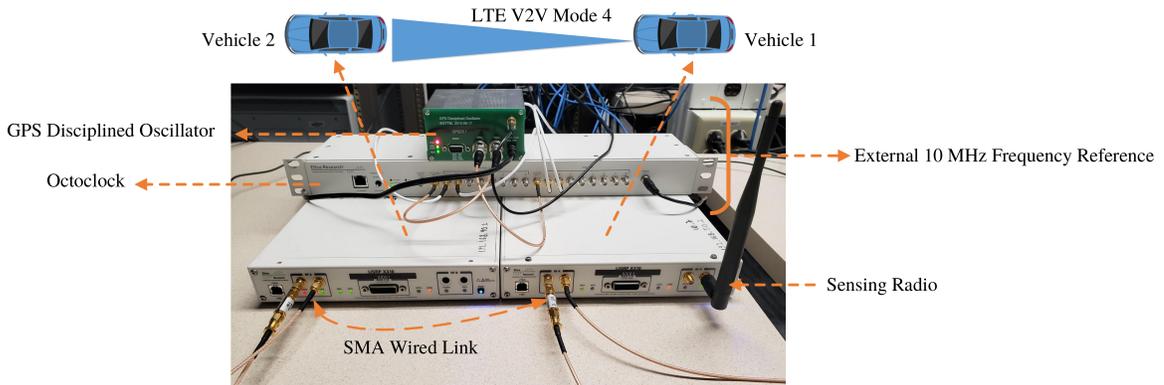


Figure 4.18: C-V2X radio testbed consisting of two USRP X310. One X310 is equipped with one SBX daughterboard for LTE transceiver link, whereas another X310 has two SBX daughterboard for simultaneous C-V2X and spectrum sensing functionality. For the C-V2X link, a wired connection is used whereas the VDSA band for the sub-channels is monitored using the VERT2450 Antenna with 3 dBi gain. The Octoclock and GPSDO are used to provide the external 10 MHz frequency reference in order to achieve sidelink synchronization between the radios [28].

and facilitate V2V communication (*Radio:0*). We employed two NI USRP X310 SDR radios using SBX-120 daughterboards for our bumblebee-based OAI testbed. The *Vehicle:1* X310

Table 4.3: Configuration parameters for bumblebee-based VDSA testbed.

| Testbed Parameters | Values |
|------------------------------|-------------------|
| C-V2X Bandwidth[Radio:0] | 10 MHz and 5 MHz |
| VDSA [Radio:1] Sampling Rate | 15.36 MSps |
| FFT Bin Resolution [Radio:1] | 1024 |
| VERT2450 Antenna Gain | 3 dBi |
| C-V2X Dedicated Frequency | 2685 MHz |
| VDSA Frequency Band | 470 MHz – 520 MHz |

supports independent operation of both LTE and sensing of the DTV spectrum from 470 MHz – 520 MHz by using two different daughterboards on the SDR whereas the *Vehicle:2* X310 only has one daughterboard for C-V2X communications. For spectrum sensing, time-domain energy detection is implemented with *Vehicle:1* sweeping the 50 MHz band in five steps. We use a dwell time of 20 ms in each 10 MHz bin and then switch to the new bin. An FFT size of 1024 is used to provide the necessary resolution for accurate detection of LTE sub-carriers.

We are not performing dynamic spectrum access at subcarrier level, *i.e.*, individual subcarriers are not selected from the entire 50 MHz bandwidth. Rather, entire 10 MHz contiguous block is selected based on bumblebee-based VDSA and then random subcarriers are selected from the block. Implementing subcarrier level spectrum allocation on top of OAI will be the focus of the future research. Figure 5.12 shows the testbed diagram, where we are employing two X310 radios that are connected using an Octoclock and a GPS disciplined oscillator (GPSDO). Octoclock and GPSDO are employed for the providing the 10 MHz reference signal that is required to provide the sidelink synchronization between the radios. The SBX daughterboards for C-V2X connectivity on both radios are attached using SMA cables whereas a separate SBX daughterboard on *Vehicle:1* is used for performing spectrum sensing and dynamic spectrum access.

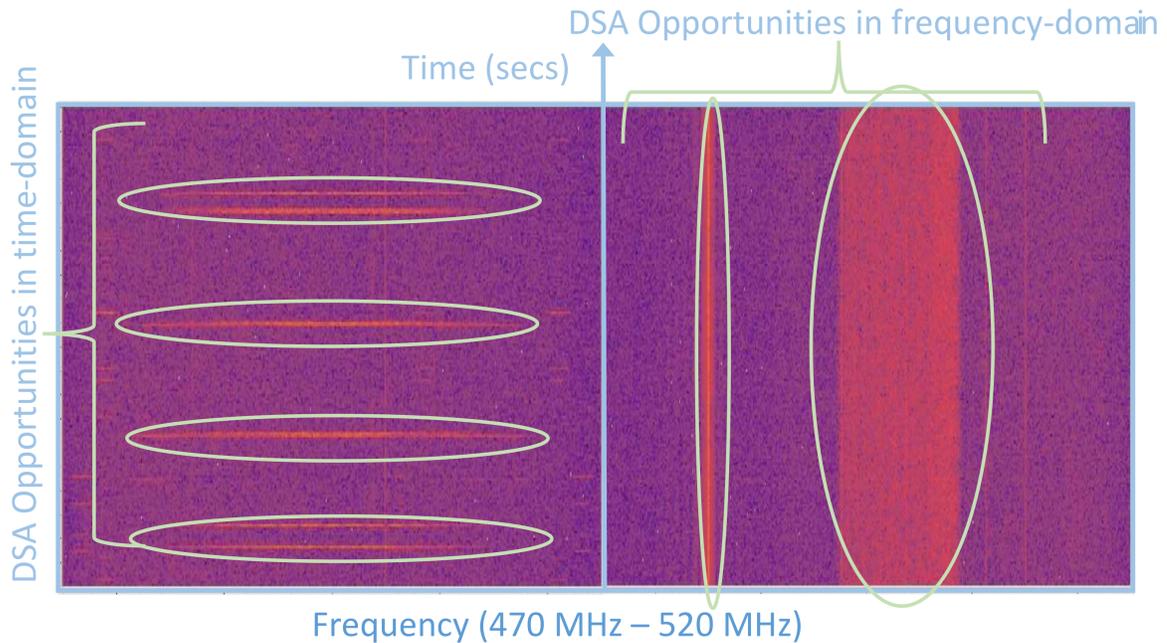


Figure 4.19: The power spectral density (PSD) for DTV band from 470 MHz to 520 MHz across 5 minutes time domain. There are spectrum access opportunities across both time and frequency and due to the stationary nature of the band these opportunities are static [28].

4.3.3 Measurement and Results

The DTV band possess stationary characteristics due to its primary users, *i.e.*, TV broadcasters that have a relatively stable spectral usage. This makes it very attractive for dynamic spectrum access. We use the DTV UHF band from 470 MHz to 520 MHz. Figure 4.19 shows the power spectral density (PSD) color-map across spectral and temporal domains for five minutes. The color variation is from -110 dBm to -50 dBm, increasing in intensity as we move to high PSD values. There are various time and frequency spectrum access opportunities available in the 50 MHz wide spectrum. Due to the primary user's predictable nature, the spectral opportunities and channel characteristics will not vary drastically. Figure shows two transmissions around 500.25 MHz and 513.25 MHz, which are being continuously accessed, whereas the spectrum from 473 MHz to 489 MHz is being utilized in an episodic manner.

We employ full-duplex radios for simultaneous monitoring of the DTV band and C-

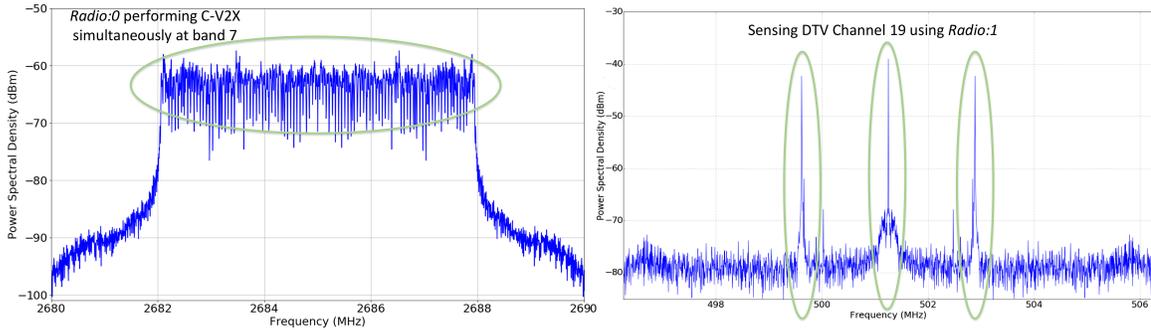


Figure 4.20: Monitoring the DTV channel 19 at 500 MHz using *Radio:1* while simultaneously performing C-V2X communication using *Radio:0*. In the experiment, sensing radio employing bumblebee algorithm monitors DTV channels whereas C-V2X radio keeps the communication active. Sensing radio discovers DTV band with better channel quality and perform the channel switching [28].

V2X communications. Figure 4.20 shows the full duplex capability of our bumblebee-based VDSA system. *Vehicle:1* performs C-V2X communication with 25 PRBs using the LTE protocol stack while simultaneously monitoring the DTV spectrum for spectrum access opportunities. Only 10 and 5 MHz of bandwidths are being utilized for C-V2X in order to reduce the computation since performing spectrum sensing with larger bandwidth leads to late packet replies and link breakage. The *Radio:1* is able to sense three narrowband transmissions across 496 MHz to 506 MHz. For sensing the entire 50 MHz DTV spectrum, we start the channel sensing at 470 MHz, sense for 1 ms and store the channel energy value in the memory. We collect the energy values for the entire 50 MHz spectrum and after some processing make the switching decision for single block memory length. We have employed different memory lengths to evaluate the bumblebee algorithm. Using more energy values for making the switching decision leads to better accuracy but also adds to the latency and memory usage of the system.

In Figure 4.21, we have computed the latency values for the different memory block sizes and channel bandwidths for 5 MHz and 10 MHz. LTE supports different bandwidths based on the availability, so the latency has been evaluated for different bandwidth values. Figures shows that as we increase the memory size, the latency increases as we need to process more data to make switching decisions. However, increasing the memory size leads

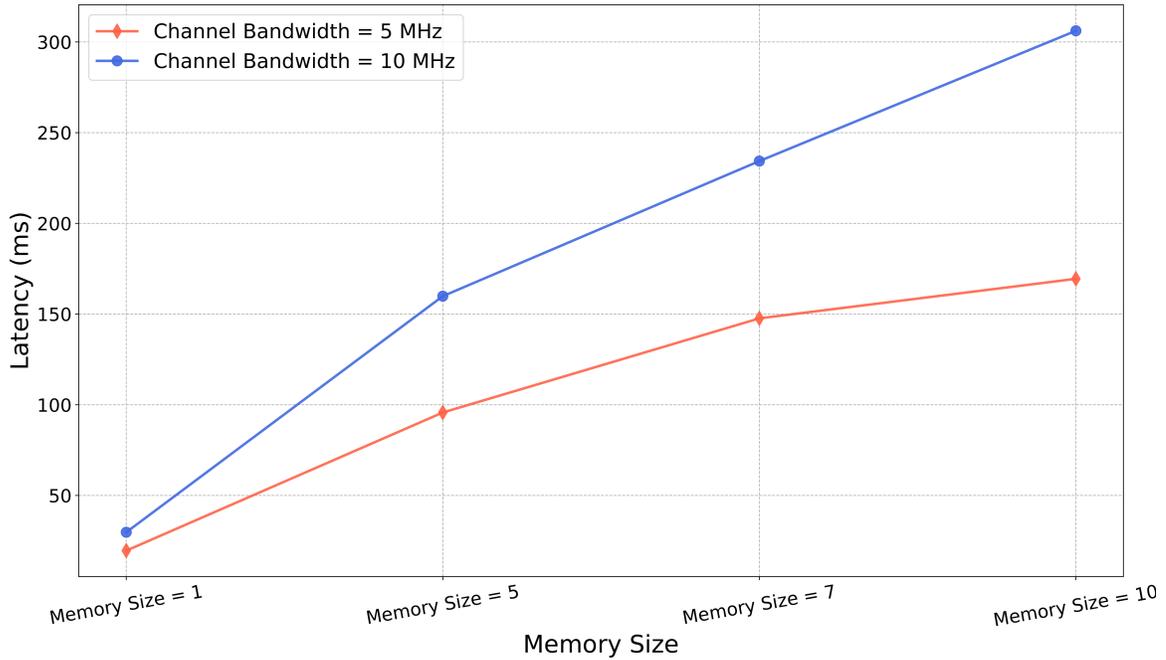


Figure 4.21: Latency comparison for bumblebee-based VDSA algorithm for different memory block size. Two channel bandwidth 5 MHz and 10 MHz are employed for resource allocation [28].

to better accuracy as we have better estimate of the channel quality. Consequently, there is a trade-off between accuracy and latency that depends on the channel environment. For an urban channel environment, larger memory size is required to accurately estimate the channel quality, even if the switching decision leads to higher latency. Since selecting the wrong subchannels can result in low packet delivery ratio (PDR). In [68], we evaluated the PDR for the bumblebee-based algorithm with a random channel switching. From this experimentation we observe large gap in performance due to erroneous selection by the random channel access scheme. In a highway environment, low memory size can be employed as the channel characteristics do not change frequently.

4.4 Chapter Summary

In this chapter, we presented a framework for a memory-enabled bumblebee foraging algorithm for vehicular platoon communications. The optimized unequal sampling allocation

heuristic is proposed to estimate the Channel Busy Ratio β with sufficiently high accuracy. Based on the results obtained, the unequal sampling allocation approach outperforms the equal sampling allocation scheme with the proposed sub-optimal allocation heuristic. We have also implemented two memory models that are integrated with the bumblebee foraging algorithm to leverage available memory, which boosts the probability of the best channel selection. Sliding window average and exponentially weighted moving average memory schemes are employed and their performance is compared against the memoryless model. Using the SWA memory scheme, different memory lengths are utilized, and, similarly, different forgetting factors are used for the EWMA scheme. The simulation results show the bumblebee algorithm with unequal sampling allocation heuristic provides higher accuracy compared to the equal allocation scheme, especially in a scenario where sensing resources are scarce.

Hardware validation was also performed to evaluate the feasibility of memory-enabled vehicular dynamic spectrum access framework. We implemented memory-based VDSA framework on top of an open-source software-based platform of LTE system. We performed switching at 2.685 GHz LTE band 7, to DTV spectrum around 500 MHz, and demonstrated the feasibility for C-V2X system.

Chapter 5

Passive Opportunistic RF

Localization of Connected Vehicles

In this chapter, we present a novel opportunistic approach for passive RF localization designed for detecting connected vehicles, especially those referred to as “phantom cars” ,*i.e.*, vehicles intentionally faking their position/velocity information to surrounding vehicles and the communication network. Current state-of-the-art approaches for vehicle localization mostly rely on either: (i) self-reported position/velocity updates obtained via navigation technologies such as GPS [103], or (ii) cooperative communication approaches involving multiple vehicles exchanging situational awareness information with each other [104]. In both cases, these approaches assume all vehicles truthfully share alocation information, which might not be the case if malicious users such as phantom cars are present. Referring to Figure 5.1, the proposed approach extracts location information of a target vehicle without the needing to cooperate with it. Multiple sensors surrounding the target vehicle opportunistically and passively measure RF emission characteristics (*e.g.*, received signal strength, time difference of arrival) of on-board widely available wireless signals (*e.g.*, 5G, Bluetooth, WiFi). These characteristics are processed via data fusion and bounded via vehicle dynamics behavioral models before being compared against reported positions. There are several components used to perform this type of vehicle localization, such as traffic flow modeling, data fusion, and RF emission propagation characterization. The focus of this chapter is

the last component and how wireless localization techniques are used to establish positions and velocities of these vehicles. To evaluate the feasibility of the proposed approach, a custom-built computer simulation platform was developed via a multi-institution research effort. The platform accurately models the vehicular environment and its associated RF emissions characteristics. This dissertation presents a simple hardware field experiment evaluating the performance of proposed approach incorporating RF localization, while data fusion, and vehicle behavioral dynamics modules are included to help illustrate the viability of identifying phantom car attacks within a non-cooperative framework.

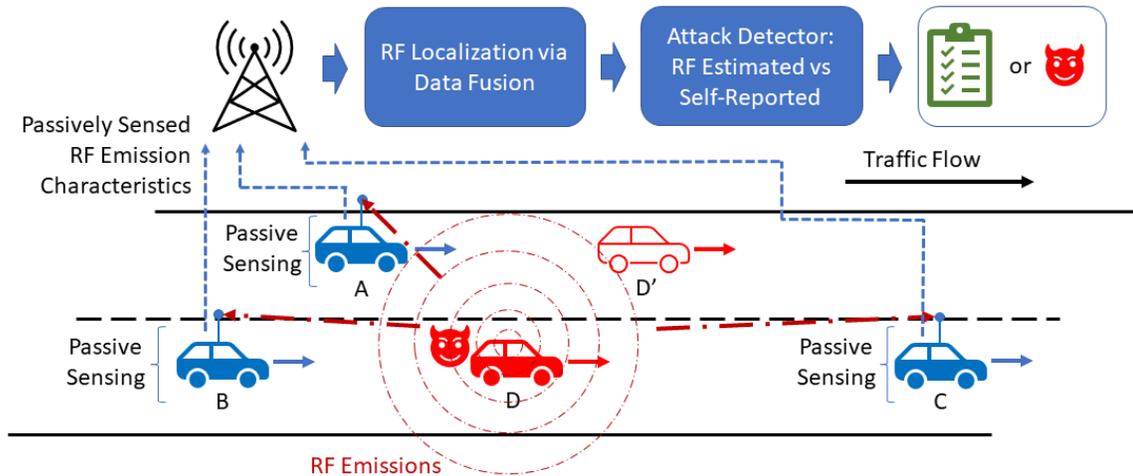


Figure 5.1: Illustration of the proposed phantom car attack detection framework, where Vehicles A, B, and C are passively sensing RF emissions coming from Vehicle D. These RF emissions can be any form of wireless signal, including 5G cellular communications, WiFi, Bluetooth, and tire pressure measurement sensor (TPMS) transmissions. It is important the surrounding vehicles are capable of detecting those signals although it is not necessary to decode transmissions. From these intercepted RF emissions, each vehicle extracts RF characteristics such as received signal strength (RSS) and arrival time values. These RF measurements are communicated to the network along with the precise position of Vehicles A, B, and C. The network performs RF localization via data fusion using these measurements, and then compares with self-reported position information to determine whether the network is under attack [24].

5.1 Overall Framework

The approach taken to achieve the goal of the overall multi-institutional research effort is shown in Figure 5.1. It consists of three fundamental building blocks: the **RF sensor network**, the **data fusion center**, and the **joint RF emission/traffic flow model**. The RF sensors are installed alongside a length of roadway and is designed to measure RF emissions and post-process them before forwarding them to the data fusion center. Each RF sensor possesses a sufficient level of intelligence to adapt and dynamically calibrate its detection thresholds to increase detection and reduce false positives. The data fusion center is responsible for taking all the RF emission measurements from the RF sensor network and estimating location and trajectory information of every detected vehicle on the target roadway. We also use a joint RF emissions/traffic flow model to characterize the RF propagation in vehicle environment so that this information can be used for several initialization and training tasks in the data fusion center. Furthermore, the data fusion center is capable of providing feedback to the RF sensor network and the joint RF emission/traffic flow model such that they can fine tune their operations to improve effectiveness and accuracy of the measurements. Consequently, the data fusion center is at the center of the proposed approach.

Two types of prevalent road environments will be considered, namely, a multi-lane highway segment and a multi-lane arterial road segment (about 2.5 miles). In the road environment, each traveling vehicle can be associated with different types of RF emissions (*e.g.*, WiFi, Bluetooth, TPMS) either emitted by the vehicle themselves or by device they carry. RF emissions to be studied were all less than 6 GHz in frequency. Moreover, all RF emissions resulted from COTS wireless devices commonly founded in typical road vehicles. RF emission detection techniques included in this work were received signal strength indicator (RSSI) and time-of-arrival (TOA). Other techniques such as angle-of-arrival (AOA) and time-difference-of-arrival (TDOA) will not be studied although they could be used in any future work resulting from this project. Localization and tracking framework will assume to possess a network RF sensor connected to a centralized cloud center (CCC). RF sensors will measure RF emissions and perform some adaptive signal processing to further enhance

detection process, while the CCC is responsible for the data fusion of these RF emission measurements and the calculation of each detected vehicle's location and trajectory. Performed exclusively in the CCC based on RF emission measurements collected by the RF sensors. Data fusion converts this measurement information into identification, localization, and target tracking information of multiple vehicles traveling on the road. This operation produced location and the continuous trajectories of vehicles in the time-space domain. We used models from both micro- and macroscopic perspectives in order to facilitate vehicle localization and tracking. Microscopic models describe the movements of individual vehicles on the road that interact with other vehicles and the operating environment (such as road geometry), while the macroscopic models capture the characteristics of a traffic stream at an aggregate level. We describe a joint analytical model that captures the characteristics of RF emission propagation and traffic flow. Different from statistical models, the model will base on the physical properties of the RF propagation and the behavior of vehicles, which will result in realistic and deterministic characterization of RF propagation. Building on the traffic-RF analytical model, it consists of the two integrated layers (vehicular traffic and RF emissions) and provides a high-fidelity simulation environment across varying conditions. The simulation platform will be used to train the data fusion framework in order to initialize it, and to predict the traffic-RF emission dynamics based on previous RF sensor inputs. (Communication protocol layers (*e.g.*, link layer, network layer) will not be included in this platform since it only focuses on RF emission information and not the information contained within the transmission.

This work included implementing the EM emissions and sensing module which, will be discussed in the following section. The remaining modules were implemented by other collaborators on the project, and are out of the scope of this dissertation. Although, there was involvement by other collaborators on the development and implementation, the creation of these modules were primarily the work of this dissertation's author.

5.2 EM Emissions and Sensing Module

The purpose of this module is to simulate the electromagnetic emissions from each vehicle to each base station (or sensor). Using the road geometry and base station indices, the power received can be determined by one of several electromagnetic models. The output of the system is a vector of the received signal strength values at each base station and vehicle ID. There are many different models that can be used to simulate the propagation of an electromagnetic wave from transmitter to a receiver, from a simple path loss model to complex ray tracing models that include multiple forms of wave-environment interactions. We chose to first implement a Two-Ray Model [105]. This model is slightly more complex than a simple path loss model and includes the effects of the electromagnetic wave reflecting off of the ground. The received power is determined from the electromagnetic field, which is composed of line-of-sight (LOS) and ground reflection (GR) components. Referring to Figure 5.1, a propagation path exists between vehicle i and base station/sensor j , for all vehicles and all base stations/sensors across every time instant.

To calculate the total electric field, we can sum the magnitudes of the components: $|E_{TOT}| = |E_{LOS} + E_g|$, where E_{LOS} and E_g correspond to the electromagnetic fields of the LOS and ground reflection, respectively. The total electromagnetic field as a function of distance is:

$$|E_{TOT}d| = 2 \frac{E_0 d_0}{d} \sin\left(\frac{\theta_\delta}{2}\right), \quad (5.1)$$

where E_0 is the free-space electromagnetic field (in V/m) at some reference distance d_0 (in m), d is the separation distance between the transmitter and the receiver (in m), and θ_δ is phase different between the LOS and GR E-field components. The phase difference is calculated as:

$$\theta_\Delta = \frac{2\pi\Delta}{\lambda}, \quad (5.2)$$

where λ is the wavelength (in m) Δ is the difference between the LOS path and the ground reflection path, which can be approximated as, $\Delta \approx \frac{2h_t h_r}{d}$. The values of h_t and h_r are the heights (in m) of the transmitting and receiving antennas.

Finally, the received power can be calculated (in W):

$$P_r(d) = \frac{|E_{TOT}|G_r\lambda^2}{480\pi^2} \quad (5.3)$$

where G_r is the gain of the receiver in linear units. There are further simplifications for (5.1), depending the TX and RX separation and the heights of the antennas. However, these approximations limit the simulation condition and minimally reduce the computation. The received powers are calculated for vehicle i and base station/sensor j and then the information is output. For this module, we assume that the transmitter frequency bands are those employed by Verizon's cellular network which are band 2, 4, 5, 13, 46, 48, 66 in North America. This work can be easily extended to other cellular bands and networks without loss of generality. Vehicle antenna height will be assumed to be 1 m.

The concept of using *Signals of Opportunity* (SOP) to passively intercept EM emissions from vehicles in order to extract RSS information that can subsequently be used by the data fusion center as illustrated in Figure 5.2. Cellular base stations intercept EM emissions originating from a road or highway where vehicles are located. The emissions are labeled by their ID_i information contained within the transmission header. Using the emission signal strength, the RSS of each vehicle EM emission is calculated by every base station and/or sensor at every sampling time instant. All of this information is collected, calculated, and forwarded to the data fusion center for vehicle localization. The concept of received signal strength-based localization is simple, where signal power is inversely proportional to the square of the distance between transmitter and receiver [106]. However, in multipath environments the problem becomes complex and can lead to large distance measurement errors. Channel models are employed in the literature to estimate the RF signal strength due to large-scale (propagation loss) and small-scale (multipath) which are later mapped to range estimate values [107].

In this dissertation, a simple two-ray channel model is implemented as a baseline within the overall simulator, which is later enhanced by 3GPP extended vehicular A channel model (EVA) [61]. Since there is potentially multipath propagation influencing each EM emission originating from every vehicle and intercepted by every base station and/or sensor, a two-ray channel model is applied to each emission to capture the distortion experienced by these

signals. Consequently, base station k collects RSS values $P_{i,k}$ and $P_{j,k}$ for vehicles ID_i and ID_j , respectively, while base station m collects RSS values $P_{i,m}$ and $P_{j,m}$ for vehicles ID_i and ID_j , respectively.

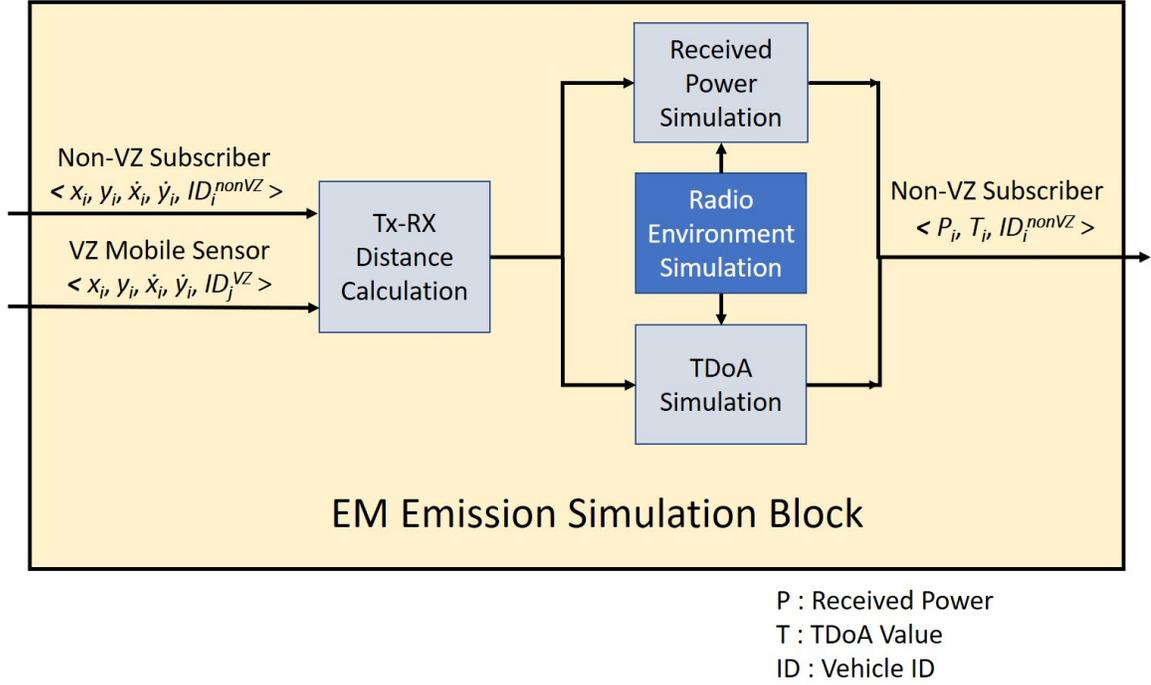


Figure 5.2: Input/output definition of vehicle electromagnetic (EM) emissions generation module [24].

5.3 Simulation Workflow

For the overall project, we created a comprehensive computer simulation environment shown in Figure 5.2. In this simulation environment, we employed SUMO to create realistic traffic traces which are then fed to our C-V2X channel module to create realistic wireless conditions. The free space path loss model (FSPL) states that for a certain frequency in free space, the power of radio signal attenuates proportionally to d^2 , where d is the line-of-sight (LOS) transmitter and receiver separation distance. However, in real-world radio environments and more specifically for C-V2X, LOS communications are not always possible and signal propagation can be affected by various physical characteristics like,

reflection, refraction, diffraction, scattering, and their combination. The VZ subscriber cars are assumed to be connected to the fusion center (FC) where the localization estimates are extracted by employing hybrid RSS-TDoA localization.

Table 5.1: Simulation Parameters for C-V2X Channel Model

| Scenario | Environment | Frequency Range (GHz) | PLE | σ (dB) |
|----------|-------------|-----------------------|-----|---------------|
| UMi SC | LOS | 2 - 73.5 | 2.0 | 2.9 |
| | NLOS | 2 - 73.5 | 3.1 | 8.1 |
| UMi OS | LOS | 2 - 60.0 | 1.9 | 4.7 |
| | NLOS | 2 - 60.0 | 2.8 | 8.3 |
| UMa | LOS | 2 - 73.5 | 2.0 | 4.6 |
| | NLOS | 2 - 73.5 | 2.7 | 10.0 |

In this dissertation, we have employed Close-in (CI) channel model with three different types of scenario *i.e.*, urban micro-cellular street canyon (UMi SC), urban micro-cellular open square (UMi OS) and urban macro-cellular [108]. The simulation parameters used for CI channel model are described in detail in Table 5.1. For this project, we have limited the frequency to 5.9 GHz as that is current channel allocated for V2V and V2I applications [109]. The path-loss exponent (PLE) α and shadow fading (σ) can be tuned based on different channel conditions. The equation for the CI channel model is given as [108]:

$$PL^{CI}(f, d)[dB] = FSPL(f, 1, m)[dB] + 10n \log_{10}(d) + \chi_{\sigma}^{CI}, \quad (5.4)$$

where n denotes the PLE (describing path loss in dB in terms of decades of distances beginning at 1 m) and d is the Tx-Rx separation distance. The quantity $FSPL(f, 1, m)[dB]$ denotes the free space path loss in dB at a Tx-Rx separation distance of 1 m at the carrier frequency f and is given as:

$$FSPL(f, 1, m)[dB] = 20 \log_{10} \left(\frac{4\pi f}{c} \right). \quad (5.5)$$

Referring to Eq. (5.4), shadow fading (SF) is expressed as [108]:

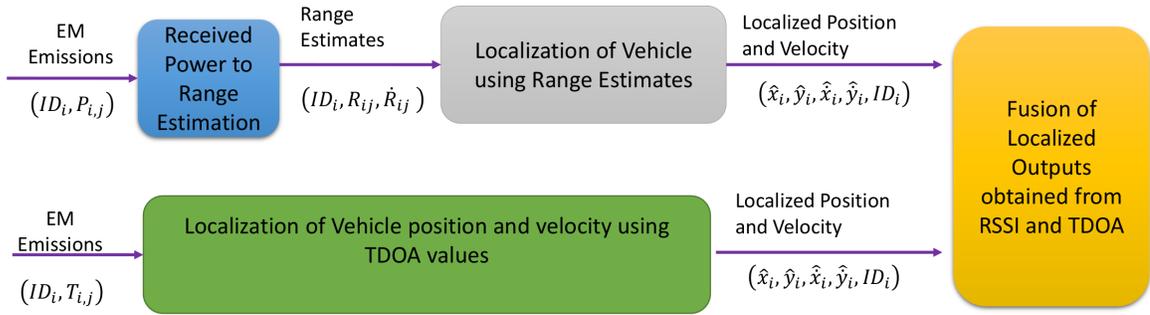
$$\mathcal{O}_{\sigma}^{CI} = A - nD, \quad (5.6)$$

where, A represents $PL^{CI}(f,d)[\text{dB}] - \text{FSPL}(f,1\text{m})[\text{dB}]$, and D denotes $10 \log_{10}(d)$. Shadow fading's standard deviation is given as [108]:

$$\sigma^{\text{CI}} = \sqrt{\sum \frac{(A - nD)}{N}}, \quad (5.7)$$

where, N is the number of path loss data points. Now, path-loss exponent (PLE) n can be obtained by minimizing $\sum (A - nD)$, thus yielding:

$$n = \frac{\sum DA}{\sum D^2}. \quad (5.8)$$



$T_{i,j}$ Time from TDOA

$P_{i,j}$ Power from RSSI

OLS Ordinary Least Square

Figure 5.3: Multi-modality data-fusion employing RSS and TDoA for non-VZ subscriber localization [24].

Figure 5.3 describes the data-fusion module, which takes the received signal strength (RSS) $P_{i,j}$ (Modality = Power) and time-difference of arrival (TDoA) $T_{i,j}$ (Modality = Time) estimates of the subscriber vehicle with identification ID_i and outputs the localization estimate of non-subscriber vehicles. TDOA and RSS are extracted from a single vehicle signal ‘RF emission’ employing 5G C-V2X standard (specifically, LTE Mode 4). The major assumption in the simulation framework for hybrid RSS-TDoA fusion is that measurement uncertainty should not decrease as a result of fusion. The module computes the x and y location estimate independently using RSS and TDoA-based localization and then fuse the data based on weights and pass it to the fusion center for decision-making.

Ordinary Least Square (OLS) algorithm is used for RSS-based localization whereas for TDoA maximum likelihood estimate (MLE) is employed in order to avoid convergence issues. Assume L_a, L_t, L_r are the localization measurement values of each method and where L_a, L_t, L_r is equal to $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n$, the weighting factors of each sensor are defined as W_1, W_2, \dots, W_n . The Covariance Intersection method takes convex combination of mean and co-variance estimates to fuse different random variables [110] and is given by:

$$P_{cc}^{-1} = \omega_1 P_{a_1 a_1}^{-1} + \dots + \omega_n P_{a_n a_n}^{-1} \quad (5.9)$$

$$P_{cc}^{-1} c = \omega_1 P_{a_1 a_1}^{-1} a_1 + \dots + \omega_n P_{a_n a_n}^{-1} a_n \quad (5.10)$$

where a_1, \dots, a_n are defined as the means and $P_{aa} \dots P_{nn}$ as the co-variances of L_a, L_t, L_r . If n pieces of information, labeled as a_1, \dots, a_n , are to be fused together to yield an output, C and $\sum_{i=1}^n W_i = 1$. For the simulation, two use-cases are considered and the proposed localization algorithm is evaluated against baseline RSS and TDoA.

1. **Stationary Sensors and Moving Emitter:** In this scenario, we assume that sensors are deployed close to the base-station and are fixed. Using this setup, high accuracy can be achieved as the location estimate of sensors are approximately equal to ground truth. As C-V2X is rolled out for V2V applications, this use-case will be easier to implement and integrate with the location server of network providers.
2. **Moving Sensors and Moving Emitters:** An out-of-coverage scenario where there is no base-station vehicles will employ LTE Mode 4 to self-allocate spectrum resources using SPS and will be able to localize other vehicles based on this use-case in GPS-denied environments.

5.3.1 Use Case 1: Stationary Sensors and Moving Emitter

For proper benchmarking of our proposed hybrid RSS-TDoA algorithm, the localization estimates are first calculated using individual RSS and TDoA modules, and are then compared with the hybrid algorithm. Figure 5.4 shows the localization estimates computed using received signal power and ordinary least square algorithm. The emitter vehicle follows a straight trajectory along the highway given by ground truth. The simulated highway is

850 meters long and is 4.7 meters wide per lane. The traffic is assumed to be bidirectional but in this work, the emitter is following one lane and has straight trajectory. We have utilized three base-stations that are deployed randomly to cover the entire simulation highway. The distance measurement error is high when the vehicle is outside the trilateration zone due to poor wireless link. The estimated position is smoothed out by applying the Kalman Filter on the localization output from the different algorithms.

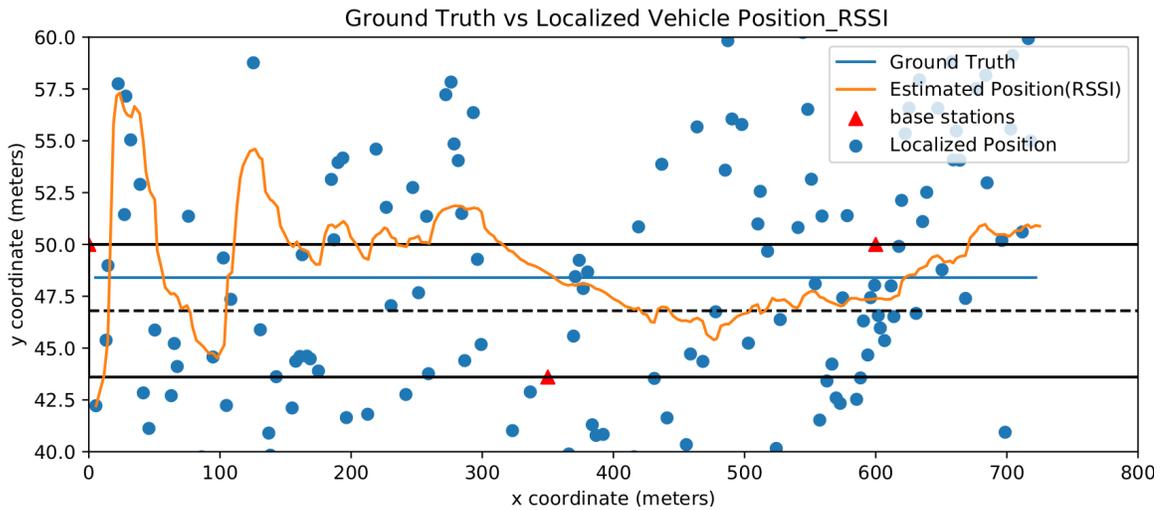


Figure 5.4: RSSI-based localization using static sensors [24].

In Figure 5.5, the localization estimates are computed using maximum likelihood estimation with input as time-difference of arrival. We do not assume the perfect synchronization between the vehicles, and the timing drift is simulated using Gaussian noise with mean of 0 and standard deviation of 1 ns. Due to the timing drift, we see large positional errors where the performance is worse than simple RSSI-based localization. Finally, in Figure 5.6 we combine the RSSI and TDoA estimates and use our fusion algorithm to get the final trajectory. The accuracy is improved drastically for the estimates which are inside the trilateration area, but outside we see higher errors compared to RSSI due to poor TDoA results.

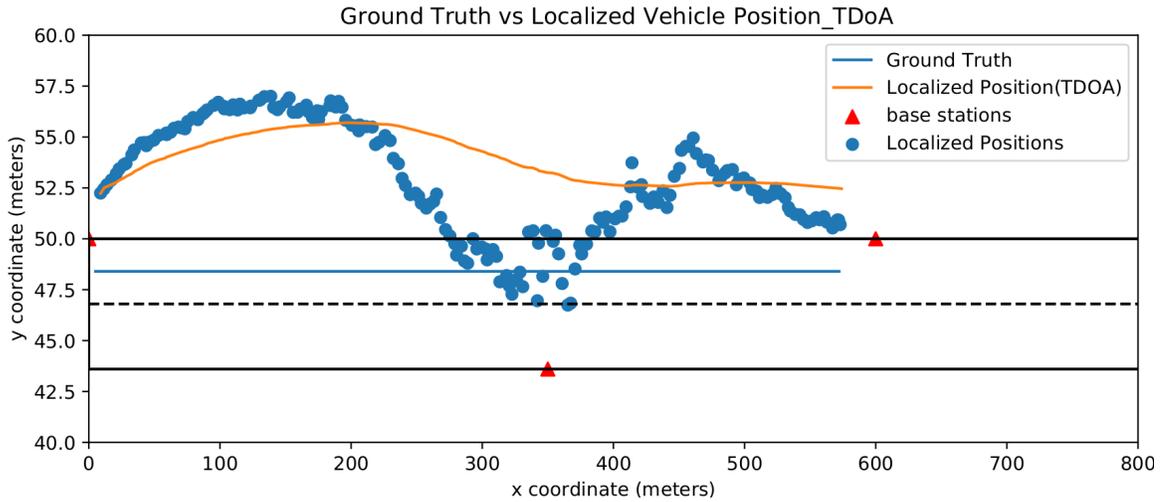


Figure 5.5: TDoA-based localization using static sensors [24].

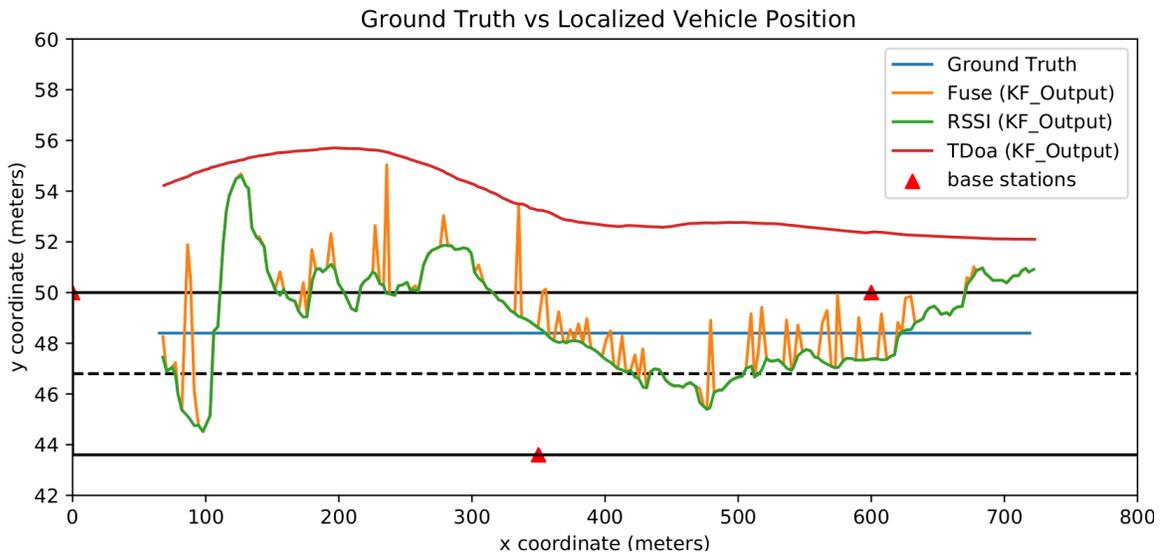


Figure 5.6: Hybrid RSS-TDoA fusion-based localization using static sensors [24].

5.3.2 Use Case 2: Moving Sensors and Moving Emitter

The moving sensors scenario is useful when we are in out-of-coverage scenario. Vehicles can schedule resources autonomously using SPS and can start V2V transmission. Localizing non-VZ subscriber vehicles in this use-case is difficult as both sensors and emitters move continuously. Figure 5.7 shows the RSS-based localization of moving emitter and sensor location is also continuously updated. It is very critical for the VZ subscriber vehicles to exchange traffic data in real-time to get good accuracy. Due to mobility of sensors, the

RSS-based localization accuracy is particularly bad in low-connectivity zones. Figure 5.8 describes the TDoA-based localization using moving vehicles. Maintaining synchronization with mobile sensors is not a trivial task but in this project, we assume the same timing error as the static case. In Figure 5.9, we show our proposed hybrid RSS-TDOA fusion scheme using moving sensors. Due to high positional errors using RSS localization, the output of the hybrid fusion algorithm is closely aligned with the TDoA for the entire simulation run. The simulation time was not large enough for the hybrid algorithm to give more weight to the RSS, leading to the current output.

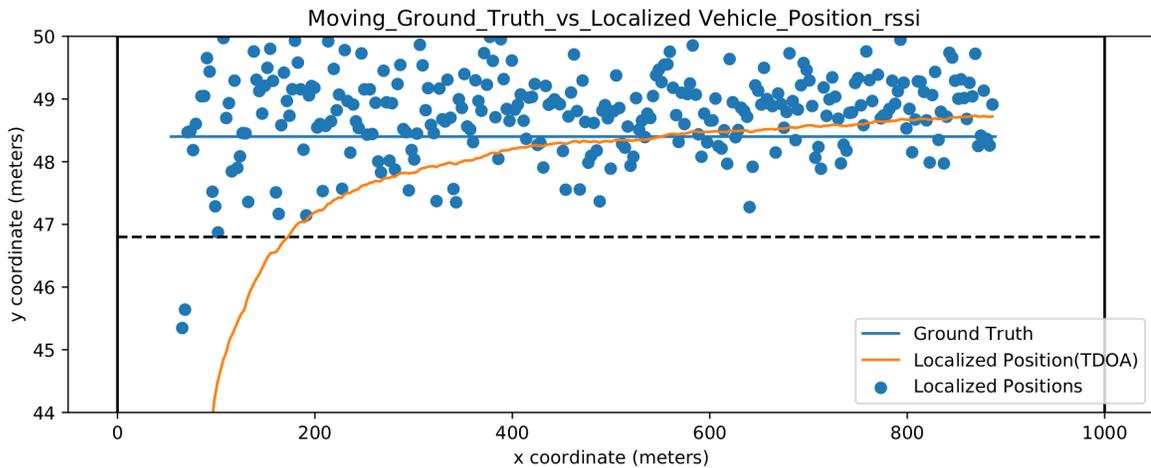


Figure 5.7: RSS-based localization using moving sensors [24].

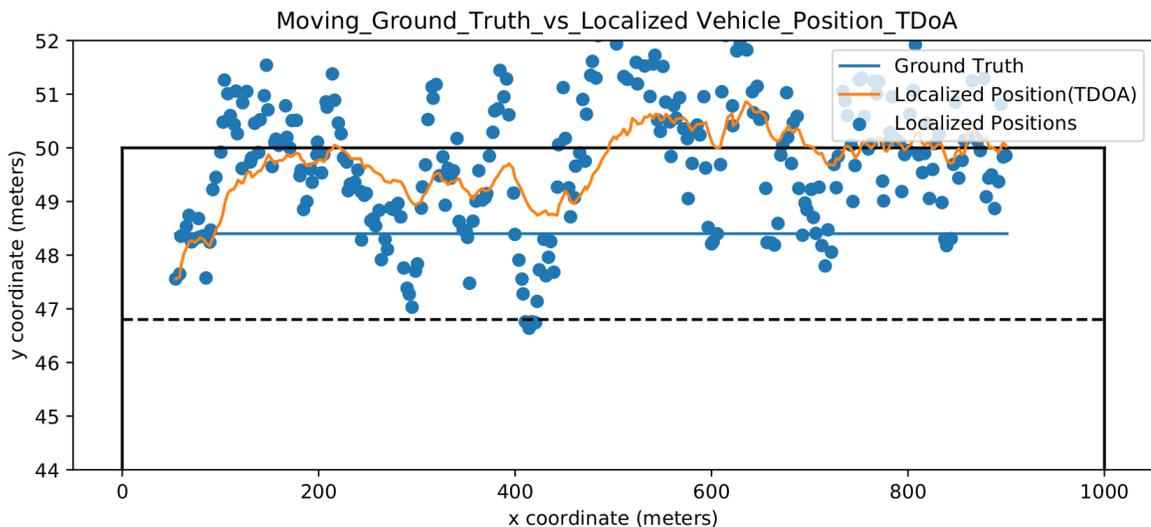


Figure 5.8: TDoA-based localization using moving sensors [24].

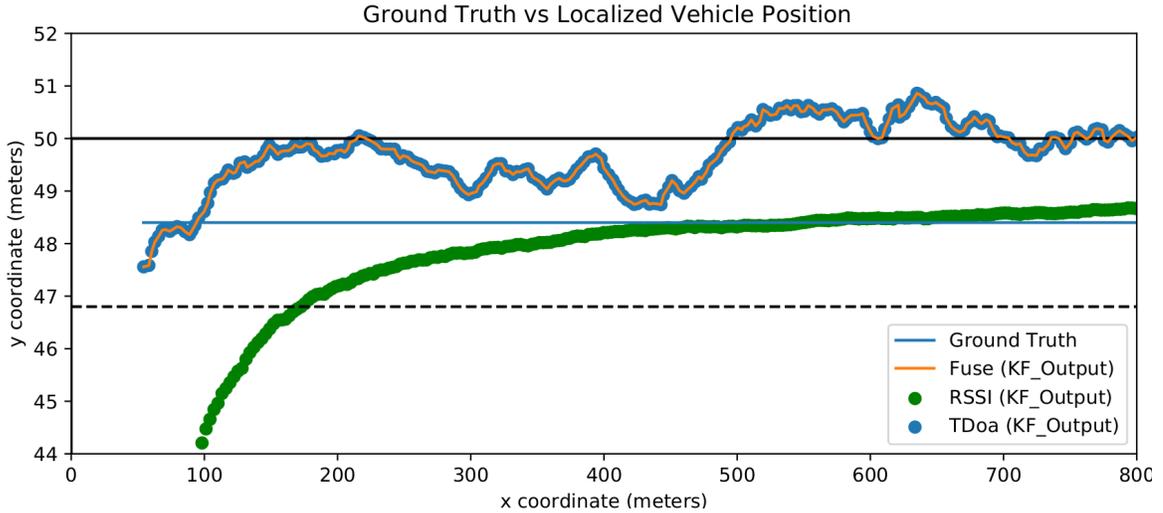


Figure 5.9: Hybrid RSS-TDoA fusion-based localization using moving sensors [24].

5.4 Small-Scale Field Experimentation

We conducted a small-scale field experiment in order to evaluate the performance of our simulation framework and to demonstrate feasibility of proposed RF localization concept using low-cost radio hardware communicating over-the-air with both emitter and sensor vehicles in motion. Four different scenarios labeled *A*, *B*, *C*, and *D* were employed. Scenarios *A* and *B* are “sanity checks” scenarios to ensure the sensor vehicle can detect the emitter vehicle. In scenario *A*, we kept the emitter stationary, whereas in scenario *B* the emitter was moving along straight line with respect to the static sensor. Scenarios *C* and *D* explore effects of SOP-based localization when only emitter vehicle is moving and when all vehicles (emitter, sensors) are in motion. In scenario *C*, all the sensor nodes were kept stationary and emitter was moving in a straight path. Finally, in scenario *D* all vehicles and emitter were moving along a straight line.

5.4.1 Experimental Setup

The experimental setup consisted of three RTL-SDR dongles, which were used as sensor nodes, and one ADALM-PLUTO [30] acting as an emitter node. Figure 5.12 describes the hardware testbed equipment as well as the software modules employed for the small-scale field experiment. Four smartphones were also employed alongside software-defined radios

Table 5.2: Configuration parameters for emitter and sensor.

| Testbed Parameters | Values |
|-----------------------------------|-----------|
| Emitter Sampling Rate [Pluto SDR] | 15 Ksps |
| Transmit Power | 8 dBm |
| Sensor Sampling Rate [RTL-SDR] | 2.4 MSps |
| LTE Antenna Gain | 8 dBi |
| Localization Channel Band | 915.1 MHz |



Figure 5.10: Experiments performed in straight North/South direction for its capacity to drive three bikes and one emitter in a straight lane. Four different scenarios were evaluated using the proposed localization technique.

to capture the GPS coordinate values of the emitter and sensor nodes. GPS Logger [111] android utility was employed to capture GPS coordinates with periodic intervals of 10 Hz. The I/Q sample measurements were performed using RTL-SDR software-defined radios and the post processing was conducted on laptops running Linux. In the realistic prototype I/Q samples captured by different radio-ends will be sent to the fusion center (FC). The fusion center can process the data and generate real-time estimate of the location information. The information can then be broadcasted back to the subscriber vehicles to mitigate GPS errors or prevent attacks from malicious users. The laptops had i5 Intel processor with eight cores and 3.41 GHz clock cycle running Ubuntu 20.04. The sensor node software was

implemented using the librtlsdr library [112] where the radio locks to the emitter frequency channel and logs the I/Q samples every 100 ms. The emitter node was implemented using GNURadio library [113]. It generates a narrowband pulse and transmits it continuously over the 915.1 MHz ISM band. The measurement samples collected by three sensor radios are later combined in fusion center (single laptop) to generate output data. The measurements are analyzed using Python NumPy package [114] and measurement plots are generated. The timing drift caused by different sensor nodes is subtracted during post processing by aligning the time-stamped I/Q samples with GPS coordinate logs. The RSSI values are upsampled by the factor of ten to align the RSSI and GPS values for localization.

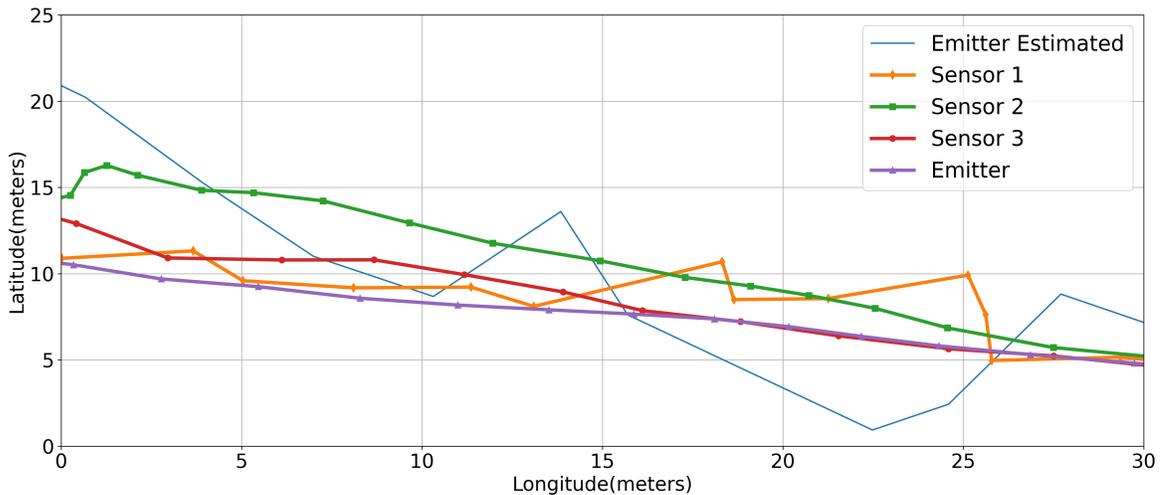


Figure 5.11: Raw estimated emitter trajectory, ground truth emitter trajectory, and trajectory of the three sensors.

Table 5.2 describes the configuration parameters employed for the hardware testbed. The narrowband sine pulse generated by emitter was centered around the 915.1 MHz ISM band with a 15 KHz bandwidth and transmit power of eight dBm. The emitter flow-graph implementation was done using GNURadio digital signal processing (DSP) framework [113]. The particular ISM band was chosen based on the spectrum measurement which showed low interference levels in that particular geographical area.

The experiment was conducted in the parking lot of Worcester Polytechnic Institute, Gateway campus in straight North/South direction. Figures 5.10 describes the venue of experiment, where the total distance was 100 *m* marked by yellow line. We initially at-

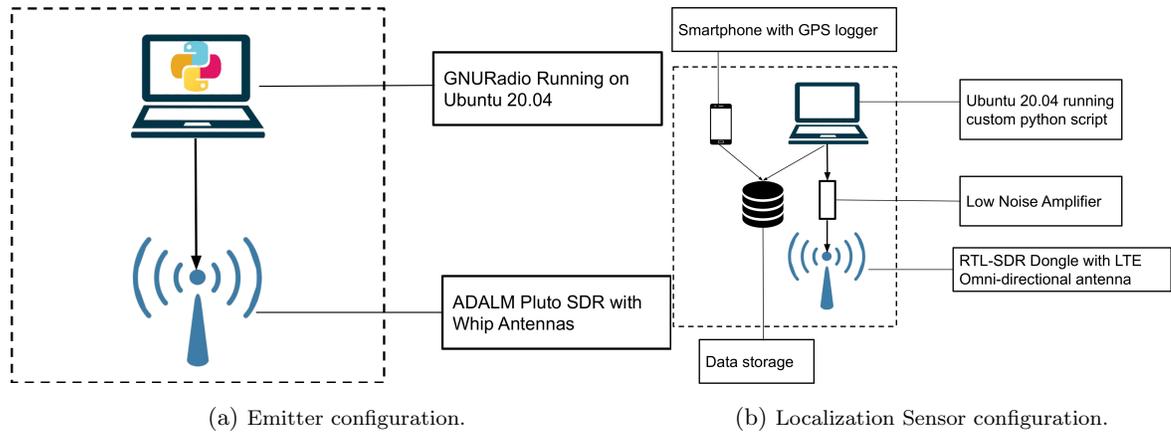


Figure 5.12: Localization testbed consisting of ADALM Pluto and RTL-SDR software-defined radios along with smartphone which is equipped with GPS logger.

tempted to use rooted 4G LTE cellular phones as mobile emitter, but later cancelled this plan due to adequate cellular access at the location. All three RTL-SDR software defined radios had a sampling rate of 2.4 MSps around the emitter center frequency which is an ISM band to intercept tone. The I/Q samples collected during the experiment were logged as *.csv* file for post-processing. The LTE antennas were reinforced with low noise amplifier (LNA) to boost the receiver sensitivity, especially at the edge of coverage. GPS locations were continuously logged for each sensor and emitter with time-stamps to correlate with the I/Q samples.

5.4.2 Localization Results

During processing, RSS measurements from RTL-SDR and location values from GPS data loggers required resampling to ensure proper time alignment. The RSS measurements were logged with the frequency of 1 Khz to reduce the effect of interference and multipath in the measurements. During the experiment, we were observing the power spectral density (PSD) of the emitter tone at 915.1 MHz and saw some unwanted signals at 914.3 and 915.8 Mhz which were filtered out during post-processing. The Ordinary Least Square (OLS)-based RSS localization algorithm was used in order to compute the position estimates using I/Q samples from the sensor nodes. The TDoA localization was not performed due to large

timing errors incurred by the RTL-SDR dongle internal clock. The hardware experiment was conducted in line-of-sight (LOS) conditions with considerable signal to noise ratio (SNR). The SNR was greater than 20 dB for the entire experiment, as sensors were moving close to the emitter.

Figure 5.11 describes the trajectory for the emitter and sensors, where latitude is on x-axis and longitude is in y-axis. The estimated emitter values are overlaid over the figure and we see the distance measurement error correlates to our simulation framework.

5.5 Chapter Summary

In this chapter, we implemented a comprehensive Python-based simulator framework in order to evaluate and test custom localization methods and communication protocols. We also proposed a hybrid RSS-TDoA localization approach which was implemented using the simulation framework and was compared against baseline RSS and TDoA localization techniques. We also conducted a small-scale field experiment using RTL-SDR and Pluto software-defined radios for hardware validation. Our proposed approach enhances localization in GPS-denied environments and detects phantom attack.

Chapter 6

Research Achievements and Future Work

The research achievements of this dissertation include a novel Bumblebee-based dynamic spectrum access algorithm which was integrated into both 802.11p and C-V2X MAC layer, and whose feasibility was demonstrated using software and hardware simulation frameworks. The algorithm was enhanced with a memory, and two suitable models were evaluated based on the vehicular network topology and the channel environment. The algorithm was also implemented using a software-defined radio to obtain a realistic performance baseline.

6.1 Research Outcomes

In this dissertation, considerable achievements have been made in the area of dynamic spectrum access applied to V2V platooning and C-V2X within a vehicular networking framework. The achievements are summarized as follows.

- **Novel Bumblebee-based DSA:** The potential utility of a bumblebee- inspired memory-based decision mechanism within a VDSA framework is demonstrated. Channel reward levels stored in memory are weighed against switch costs to decide whether to stay on the current channel or move to a different channel. Channel reward information is frequently updated in memory through periodic sampling, which provides

vehicles with a more accurate estimate of the degree to which channels differ in their quality for a given vehicular environment.

- **Memory Enabled Bumblebee Algorithm:** In this research vector, we presented a framework for a memory-enabled bumblebee foraging algorithm for vehicular platoon communications. The optimized unequal sampling allocation heuristic is proposed to estimate the Channel Busy Ratio β with sufficiently high accuracy. The unequal sampling instant allocation approach outperforms the equal sampling allocation scheme with the proposed sub-optimal allocation heuristic. We have also implemented two memory models that are integrated with the bumblebee foraging algorithm to leverage available memory, which boosts the probability of the best channel selection. Sliding window average and exponentially weighted moving average memory schemes are employed and their performance is compared against the memoryless model. Different memory lengths and forgetting factors were used for the SWA and EWMA schemes, respectively. The simulation results show the bumblebee algorithm with unequal sampling allocation heuristic provides higher accuracy compared to the equal allocation scheme, especially in a scenario where the sensing resources are scarce. We also evaluated our memory-enabled VDSA algorithm with an open-source 4G/LTE testbed platform with full protocol stack for both E-UTRA Node B (eNodeB) and user-equipment (UE). Specifically, we utilized the OpenAir-Interface (OAI) LTE-Sidelink platform, which was designed for C-V2X applications. Using a full-duplex radio architecture, *i.e.*, simultaneously employing one radio chain for C-V2X and another radio for sensing we demonstrated the novelty of our proposed approach for vehicular C-V2X.
- **Phantom Car Attack Detection Via Passive Opportunistic RF Localization:** In this work, we implemented a comprehensive Python-based simulator framework to evaluate and test custom localization methods and communication protocols. We also proposed a hybrid RSS-TDoA localization approach which outperform baseline RSS and TDoA by significant delta. The performance evaluation was conducted using our the simulation framework and was compared against baseline RSS and

TDOA localization techniques. We also conducted a small-scale field experiment using RTL-SDR and Pluto software-defined radios for hardware validation. The experiment demonstrated the feasibility of our proposed hybrid localization approach using signals of opportunity in a realistic multipath environment. Our proposed approach enhances localization accuracy in GPS-denied environments and can detect phantom attacks.

6.2 Future Work

Based on the research contributions presented in Chapters 4 and 5, potential future work in this field is as follows.

- First, we will explore a holistic memory model that can adapt to different time-varying conditions. The model parameters can be tuned by employing machine learning. A new heuristic can also be investigated where γ can be dynamically adjusted based on the sampling instants and CBR variations of the channels. Finally, we would like to perform over-the-air tests of the proposed algorithm using software-defined radio (SDR) testbed technology.
- Second, we will also conduct an on-road study with full-duplex radios mounted on cars. We will compute the optimal switching cost and switching cost evaluation interval for urban and highway scenarios. The on-road study will be conducted using C-V2X, which is a widespread networking technology for V2V. The MAC layer of *OpenAirInterface-5G* will be modified to perform scheduling assignment on sub-carrier level. Both radios will be equipped with full-duplex capabilities and will perform off-network V2V communication.

Appendix A

Appendix: Channel Busy Ratio Modeling

A.1 binomial_based_channel_occupancy.py

Listing A.1: Binomial Distribution based Channel Busy Ratio

```
import time
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import bernoulli, binom
totaliter = 20
t = np.arange(totaliter)#
K = 2 # total number of channels
Nc = 10
plist = [0.3,0.1,0.8]
runs=10000
channelidx = np.argmin(plist)
zlist = [0,1,2]
test = []
for z in zlist:
    probchanest_z_1 = np.zeros((totaliter,runs))
```

```

for r in range(runs):
    sumnk = np.zeros((K,), dtype=int)
    channeleestimate = np.zeros((totaliter, K))
    for iteration in range(len(t)):
        nk = np.zeros((K,), dtype=int)
        if iteration== 0:
            nk = (Nc//K)*np.ones((K,), dtype=int)
            sumnk = sumnk+nk
            for i in range(K):
                CBR = bernoulli.rvs(plist[i], size=sumnk[i])
                channeleestimate[iteration, i] = CBR.mean()
            probchanest_z_1[iteration, r] =
                (np.argmin(channeleestimate[iteration, :], axis=0) == channelidx)
        else:
            np.nan_to_num(channeleestimate[iteration-1], copy=False)
            temp = channeleestimate[iteration-1, :]#previous beta estimate
            for tap in range(len(temp)):
                nk[tap] =
                    np.floor(((1/(temp[tap]+0.001))**z)/
                        (np.sum(np.divide(1, (temp+0.001))**z))*Nc)
            while np.sum(nk)>Nc:
                all_positive = (nk>0).nonzero()
                ind=np.random.randint((len(all_positive),1))
                nk[ind]=nk[ind]-1
            nk[nk<0] = 1
            while np.sum(nk)<Nc:
                ind=np.random.randint((len(nk),1))
                nk[ind]=nk[ind]+1
            sumnk = sumnk+nk
            for i in range(K):
                CBR = bernoulli.rvs(plist[i], size=sumnk[i])
                channeleestimate[iteration, i] = CBR.mean()
            probchanest_z_1[iteration, r] =
                (np.argmin(channeleestimate[iteration, :], axis=0) == channelidx)

test.append(np.mean(probchanest_z_1, axis=1))

```

```
plt.figure(figsize=(10,8))
plt.title('Beta Values = [0.1,0.3,0.8], N=6, l=2')
plt.plot(test[0], '--k', linewidth=2.0, label='z = 0')
plt.plot(test[1], '--gs', linewidth=2.0, label='z = -1')
plt.plot(test[2], '--bd', linewidth=2.0, label='z = -2')
plt.plot(test[3], '--r<', linewidth=2.0, label='WIA')
plt.plot(test[4], '--y^', linewidth=2.0, label='MWIA')
plt.xlabel('Number of Iterations', fontsize=16)
plt.ylabel('Probability of Selecting Best Channel', fontsize=16)
plt.grid()
plt.legend(loc='lower right')
plt.axis('tight')
plt.savefig('./weighted_moving_average_1.jpg')
```

A.2 cbr_analytical_evaluation.m

```
clear;
clc;
% close all
tic;
load('/home/ksgill/Desktop/Publications/IEEE-Access-2/Simulation/Matlab/data/
    cbr_all_car_den_3.mat')
runs=10000; % parallel in time
iterations=10;% in time
chosen_channel= cell(3,1);
N=10;% no of sensing moments
channel_width=4;
real_beta_all = cbr_all_car_density_3(:,101:200)';
%% Bumblebee variables
bumblebee_estimated_channel_cbr = zeros(length(real_beta_all), 3, channel_width)
;
bumblebee_est_beta = zeros(10000,4);
for rb=1:length(real_beta_all)
real_beta = real_beta_all(rb,:);
```

```

[~,chosen_idx] = min(real_beta);
zlist=[-2];%[0,-0.5, -1, -2, -5, -8, -10];% for 0- equal allocation, positive
    values should work
% [chosen_channel{1},chosen_channel{2},chosen_channel{3},chosen_channel{4},...
%     chosen_channel{5},chosen_channel{6},chosen_channel{7},...
%     chosen_channel{8},chosen_channel{9},chosen_channel{10}] = deal(zeros(runs,
    iterations));
[chosen_channel{1},chosen_channel{2},chosen_channel{3}] = deal(zeros(runs,
    iterations));
for z=1:length(zlist)
    beta_est=zeros(runs,iterations,length(real_beta));
    sum_k=zeros(runs,iterations,length(real_beta));% cummlates k over
        iterations
    sum_Nl=zeros(runs,iterations,length(real_beta));% cummlates N_l over
        iterations
    for r=1:runs
        for i=1:iterations
            if i == 1
                Nl=round(N/length(real_beta))*ones(1,length(real_beta));
                % possibly sum does not meet N - randomly add or remove one
                % sample till N is met
                while sum(Nl)>N
                    all_positive=find(Nl>0);
                    ind=randi(length(all_positive),1);
                    Nl(ind)=Nl(ind)-1;
                end
                while sum(Nl)<N
                    ind=randi(length(Nl),1);
                    Nl(ind)=Nl(ind)+1;
                end
                for l=1:length(real_beta)
                    if Nl(l)>0
                        instant_k=sum(rand([1 Nl(l)])<real_beta(l));
                    else
                        instant_k=0;
                    end
                    sum_k(r,i,l)=instant_k;
                end
            end
        end
    end
end

```

```

        sum_Nl(r,i,1)=Nl(1);
    end
else
    beta_est=squeeze(sum_k(r,i-1,:)./sum_Nl(r,i-1,:));
    [~,ind2] = sort(beta_est, 'ascend');
    beta_est(ind2(1)) = beta_est(ind2(2));
    if (z == 1)
        Nl=round(N*(beta_est+0.001).^zlist(z)/sum((beta_est+0.001).^zlist(z)));
    elseif (z ==4)
        betax = 1-beta_est;
        if sum(betax) == 0 || isempty(betax(betax<0))
            Nl= round(N/length(real_beta))*ones(1,length(real_beta));
        else
            Nl= floor((betax*N)/sum(betax));
        end
    end

    if (z == 1 || z == 2)
        Nl=round(N*exp(beta_est*zlist(z))/sum(exp(beta_est*zlist(z))));
    elseif (z == 3)
        temp = squeeze(sum_Nl(r,i-1,:));
        Nl=round(N*exp(beta_est*zlist(z).*temp)/sum(exp(beta_est*zlist(z).*temp)));
    else
        disp('Error\n\n');
        return;
    end

    %Without- problem if beta_est equals 0-> Inf or NaN
    % two loops below: randomly improve allocation so that number
    % of samples is N.
    while sum(Nl)>N
        all_positive=find(Nl>0);
        ind=randi(length(all_positive),1);
        Nl(ind)=Nl(ind)-1;
    end
end

```

```

        while sum(Nl)<N
            ind=randi(length(Nl),1);
            Nl(ind)=Nl(ind)+1;
        end

        for l=1:length(real_beta)
            if Nl(l)>0
                instant_k=sum(rand([1 Nl(l)])<real_beta(l));
            else
                instant_k=0;
            end
            sum_k(r,i,l)=sum_k(r,i-1,l)+instant_k;
            sum_Nl(r,i,l)=sum_Nl(r,i-1,l)+Nl(l);
        end
    end
end
instant_est_beta=squeeze(sum_k(r, :, :)/sum_Nl(r, :, :));
[~,ind]=min(instant_est_beta,[],2);
for iter=1:iterations
    all_min_ind=find(instant_est_beta(iter, :)==instant_est_beta(iter, ind(
        iter)));
    if length(all_min_ind)>1
        ind(iter)=all_min_ind(randi(length(all_min_ind),1));
    end
end
chosen_channel{z}(r,:)=ind;
bumblebee_est_beta(r,:) = instant_est_beta(10,:);
end
bumblebee_estimated_channel_cbr(rb,z,:) = mean(instant_est_beta,1);
end
disp(rb);
end
c1 = [0, 0.4470, 0.7410];
c2 = [0.8500, 0.3250, 0.0980];
c3 = [0.4660, 0.6740, 0.1880];
c4 = [0.3010, 0.7450, 0.9330];
%% Bumblebee Plots

```

```

bumblebee_estimated_channel_cbr_h1 = squeeze(bumblebee_estimated_channel_cbr
    (:,1,:));
bumblebee_estimated_channel_cbr_h2 = squeeze(bumblebee_estimated_channel_cbr
    (:,2,:));
bumblebee_estimated_channel_cbr_h3 = squeeze(bumblebee_estimated_channel_cbr
    (:,3,:));
return;
%%
linewidthc = 2.5;
figure(1);
hold on;
grid on;
plot(cumsum(abs(squeeze(bumblebee_estimated_channel_cbr(:,1,1))-real.beta_all
    (:,1))), 'color', c1, 'LineWidth', linewidthc);
plot(cumsum(abs(squeeze(bumblebee_estimated_channel_cbr(:,2,1))-real.beta_all
    (:,1))), 'color', c3, 'LineWidth', linewidthc);
plot(cumsum(abs(squeeze(bumblebee_estimated_channel_cbr(:,3,1))-real.beta_all
    (:,1))), 'color', c4, 'LineWidth', linewidthc);
h = legend('Equal Allocation', '$H_{3}~z = -4$', '$H_{4}~z = -4$');
h.NumColumns = 1;
set(h, 'Interpreter', 'latex', 'FontSize', 14);
title("N = 10, l = 4")
xlabel('Time');
ylabel('Cumulative Absolute Error |\beta_{est}-\beta_{real}|');
set(gca, "box", "on", "FontSize", 20, "LineWidth", 1.5);
%%
toc;
hold on;
grid on;

hold on;
grid on;

plot([1:iterations], mean(chosen_channel{1}==chosen_idx, 1), '-s', 'Color', c1, '
    LineWidth', linewidthc, 'MarkerSize', 8);
plot([1:iterations], mean(chosen_channel{2}==chosen_idx, 1), '-o', 'Color', c1, '
    LineWidth', linewidthc, 'MarkerSize', 8);

```

```

plot([1:iterations],mean(chosen_channel{1}==chosen_indx,1),'--k' , 'LineWidth', 2
    .5);
plot([1:iterations],mean(chosen_channel{2}==chosen_indx,1),'-d' , 'Color',c3, '
    LineWidth', linewidthc, 'MarkerSize',8);
plot([1:iterations],mean(chosen_channel{3}==chosen_indx,1),'-^' , 'Color',c2, '
    LineWidth', linewidthc, 'MarkerSize',8);
plot([1:iterations],mean(chosen_channel{6}==chosen_indx,1),'-<' , 'Color',c2, '
    LineWidth', linewidthc, 'MarkerSize',8);
plot([1:iterations],mean(chosen_channel{7}==chosen_indx,1),'->' , 'Color',c2, '
    LineWidth', linewidthc, 'MarkerSize',8);
plot([1:iterations],mean(chosen_channel{8}==chosen_indx,1),'-*' , 'Color',c4, '
    LineWidth', linewidthc, 'MarkerSize',8);
plot([1:iterations],mean(chosen_channel{9}==chosen_indx,1),'-p' , 'Color',c4, '
    LineWidth', linewidthc, 'MarkerSize',8);
plot([1:iterations],mean(chosen_channel{10}==chosen_indx,1),'-h' , 'Color',c4, '
    LineWidth', linewidthc, 'MarkerSize',8);
axis tight
xlabel('Iteration');
ylabel('Probability of Best Channel Selection');
set(gca,"box","on","FontSize",20, "LineWidth", 1.5);
title(strcat('$l\sim\sim',num2str(channel_width),'~N\sim\sim',num2str(N), '~\beta = $ [' ,
    num2str(real_beta),']'),'FontSize', 20, 'Interpreter','latex');
subtitle('$H_1-Z_{\{Heuristic\}},H_2-W_{\{eighted\}}~I_{\{nverse\}}~A_{\{llocation\}},H_3-z_{\{
    exponent\}},H_4-z_{\{exponent~with~variance\}}$','Color',[0.25,0.25,0.25], '
    Interpreter','latex');

legendstr = strcat('$Equal Allocation','~H_{3}~z = -4~',' ~H_{4}~z = -4$~');
legendstr = strcat('z=',string(zlist));
for i=1:length(legendstr)
    if i == 1 || i==2 || i==3
        if i ==3
            legendstr(i) = 'Equal Allocation';
        else
            legendstr(i) = strcat('$H_1$: ',' z=(' ,num2str(zlist(i)),')');
        end
    elseif i == 4
        legendstr(i) = strcat('$H_2$:',' $WIA$');
    end
end

```

```

elseif i == 5 || i==6 || i==7
    legendstr(i) = strcat('$H_3$', ' z=(', num2str(zlist(i)), ')');
else
    legendstr(i) = strcat('$H_4$', ' z=(', num2str(zlist(i)), ')');
end
end
h = legend(legendstr);
set(h, 'Interpreter', 'latex', 'FontSize', 14);
t = [1:iterations];
indexOfInterest = (t <= 10) & (t >= 4); % range of t near perturbation
for i=1:3
    signal{i} = mean(chosen_channel{i}==chosen_idx, 1);
    signal{i} = signal{i}(indexOfInterest);
end
% bestoneshort = max(best_one, [], 1);
axes('position', [.15 .15 .5 .5])
box on % put box around new pair of axes
hold on;
grid on;
plot(t(indexOfInterest), signal{2}, '-d', 'Color', c3, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{3}, '-s', 'Color', c2, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{1}, '--k', 'LineWidth', 2.5);
plot(t(indexOfInterest), signal{4}, '-d', 'Color', c3, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{5}, '-^', 'Color', c2, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{6}, '-<', 'Color', c2, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{7}, '->', 'Color', c2, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{8}, '-*', 'Color', c4, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), signal{9}, '-p', 'Color', c4, 'LineWidth', linewidthc, '
    MarkerSize', 8);

```

```
plot(t(indexOfInterest), signal{10}, '-h', 'Color', c4, 'LineWidth', linewidthc, '
    MarkerSize', 8);
plot(t(indexOfInterest), bestonestshort(indexOfInterest), 'hr', 'LineWidth', 2.5, '
    MarkerFaceColor', 'r');
set(gca, "box", "on", "FontSize", 20, "LineWidth", 1.5);
axis tight
```

Bibliography

- [1] L. Wang, R. F. Iida, and A. M. Wyglinski, "Performance Analysis of EDCA for IEEE 802.11p/DSRC Based V2V Communication in Discrete Event System," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5.
- [2] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, July 2011.
- [3] A. Papathanassiou and A. Khoryaev, "Cellular V2X as the essential enabler of superior global connected transportation services," *IEEE 5G Tech Focus*, vol. 1, no. 2, p. 2017, 2017.
- [4] Qualcomm C-V2X. [Online]. Available: <https://www.qualcomm.com/news/onq/2017/02/24/accelerating-c-v2x-toward-5g-autonomous-driving>
- [5] A. Nabil, K. Kaur, C. Dietrich, and V. Marojevic, "Performance analysis of sensing-based semi-persistent scheduling in C-V2X networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–5.
- [6] J. Kane and W. Rinehart, "Use of the 5.850-5.925 GHz band," *The Center for Growth and Opportunity*, 2022.
- [7] S. Chen, A. M. Wyglinski, S. Pagadarai, R. Vuyyuru, and O. Altintas, "Feasibility analysis of vehicular dynamic spectrum access via queueing theory model," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.
- [8] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, 1967.

-
- [9] X. Zhang, R. Chai, and F. Gao, “Matched filter based spectrum sensing and power level detection for cognitive radio network,” in *2014 IEEE global conference on signal and information processing (GlobalSIP)*. IEEE, 2014, pp. 1267–1270.
- [10] V. Turunen, M. Kosunen, A. Huttunen, S. Kallioinen, P. Ikonen, A. Parssinen, and J. Ryyanen, “Implementation of cyclostationary feature detector for cognitive radios,” in *2009 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*. IEEE, 2009, pp. 1–4.
- [11] W. Lee, M. Kim, and D.-H. Cho, “Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3005–3009, 2019.
- [12] H. Hartenstein and K. L. Laberteaux, *VANET: vehicular applications and inter-networking technologies*. Wiley: 1st Ed, 2010.
- [13] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, “Vehicular Channel Characterization and Its Implications for Wireless System Design and Performance,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.
- [14] B. Heinrich, “Do bumblebees forage optimally, and does it matter?” *American Zoologist*, vol. 23, no. 2, pp. 273–281, 1983.
- [15] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford University Press, Inc., 1999.
- [16] D. Karaboga, “An Idea Based On Honey Bee Swarm For Numerical Optimization,” Erciyes University, Kayseri, Turkey, Tech. Rep., 2005.
- [17] M. R. Jabbarpour, A. Jalooli, E. Shaghaghi, R. M. Noor, L. Rothkrantz, R. H. Khokhar, and N. B. Anuar, “Ant-based Vehicle Congestion Avoidance System Using Vehicular Networks,” *Engineering Applications of Artificial Intelligence*, vol. 36, no. 11, pp. 303–319, 2014.

- [18] A. R. Deshmukh and S. S. Dorle, “Bio-inspired optimization algorithms for improvement of vehicle routing problems,” in *Emerging Trends in Engineering and Technology (ICETET), 2015 7th International Conference on*. IEEE, 2015, pp. 14–18.
- [19] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilén, R. C. Reinhart, and D. J. Mortensen, “Multi-objective reinforcement learning-based deep neural networks for cognitive space communications,” in *Cognitive Communications for Aerospace Applications Workshop (CCAA), 2017*. IEEE, 2017, pp. 1–8.
- [20] P. V. R. Ferreira, R. Paffenroth, and A. M. Wyglinski, “Interactive multiple model filter for land-mobile satellite communications at ka-band,” *IEEE Access*, 2017.
- [21] F. Riaz, S. Ahmed, I. Shafi, M. Imran, N. I. Ratyal, and M. Sajid, “White Space Optimization Using Memory Enabled Genetic Algorithm in Vehicular Cognitive Radio,” in *IEEE 11th International Conference on Cybernetic Intelligent Systems (CIS)*, Aug 2012, pp. 133–140.
- [22] R. Poli, J. Kennedy, and T. Blackwell, “Particle Swarm Optimization,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [23] K. S. Gill, B. Aygun, K. N. Heath, R. J. Gegeer, E. F. Ryder, and A. M. Wyglinski, “Memory matters: Bumblebee behavioral models for vehicle-to-vehicle communications,” *IEEE Access*, vol. 6, pp. 25 437–25 447, 2018.
- [24] A. M. Wyglinski, T. Wickramaratne, D. Chen, N. Kirsch, K. S. Gill, T. Jain, V. Garg, T. Li, S. Paul, and X. Zhang, “Phantom car attack detection via passive opportunistic RF localization, in progress,” *IEEE Access*, 2022.
- [25] K. Gill, K. N. Heath, R. J. Gegeer, E. F. Ryder, and A. M. Wyglinski, “On the capacity bounds for bumblebee-inspired connected vehicle networks via queuing theory,” in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–6.
- [26] K. S. Gill, K. McClintick, N. Kanthasamy, J. Tolbert, D. Nguyen, S. Nguyen, G. Wernsing, V. Moore, I. Gelman, A. O’Neil, N. Schubert, C. Coogan, K. Murdy, B. Mahan, S. Halama, K. N. Heath, E. F. Ryder, R. J. Gegeer, and A. M. Wyglinski,

- “Experimental test-bed for bumblebee-inspired channel selection in an ad-hoc network,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.
- [27] K. S. Gill, K. N. Heath, S. Chuks, A. Haider, R. J. Gegear, E. F. Ryder, and A. M. Wyglinski, “Bumblebee-inspired C-V2X dynamic spectrum access testbed using openairinterface,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [28] K. S. Gill, P. Kryszkiewicz, P. Sroka, A. Kliks, and A. M. Wyglinski, “Memory enabled bumblebee-based dynamic spectrum access for platooning environments,” *submitted, IEEE Transactions on Vehicular Technology*, 2022.
- [29] P. Sroka, P. Kryszkiewicz, M. Sybis, A. Kliks, K. S. Gill, and A. Wyglinski, “Distributed vehicular dynamic spectrum access for platooning environments,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [30] ADALM-PLUTO software defined radio. [Online]. Available: <http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview/>
- [31] R. Gegear and T. Lavery, *The Effect of Variation Among Floral Traits on the Flower Constancy of Pollinators (Chapter 1)*. In *Cognitive Ecology of Pollination: Animal Behavior and Floral Evolution*. Cambridge, UK: Cambridge University Press, 2001.
- [32] D. Goulson, “Foraging Strategies of Insects for Gathering Nectar and Pollen, and Implications for Plant Ecology and Evolution,” *Perspectives in Plant Ecology, Evolution and Systematics*, vol. 2, no. 2, pp. 185–209, Dec. 1999.
- [33] B. Chittka, “Sensorimotor Learning in Bumblebees: Long-Term Retention and Reversal Training,” *The Journal of Experimental Biology*, vol. 201, no. 4, pp. 515–24, 1998.
- [34] J. M. Biernaskie, S. C. Walker, and R. J. Gegear, “Bumblebees Learn to Forage like Bayesians,” *American Naturalist*, vol. 174, no. 3, pp. 413–423, 2009.

- [35] D. Goulson, S. A. Hawson, and J. C. Stout, "Foraging bumblebees avoid flowers already visited by conspecifics or by other bumblebee species," *Animal Behaviour*, vol. 55, no. 1, pp. 199–206, Jan. 1998.
- [36] A. Ghasemi, M. A. Masnadi-Shirazi, M. Biguesh, and F. Qassem, "Channel assignment based on bee algorithms in multi-hop cognitive radio networks," *IET Communications*, vol. 8, no. 13, pp. 2356–2365, 2014.
- [37] S. I. Suliman, I. Musirin, R. Mohamad, M. Kassim, and M. F. M. Idros, "An efficient constructive heuristic for optimizing frequency allocation in cellular network," in *Industrial Engineering and Applications (ICIEA), 2017 4th International Conference on*. IEEE, 2017, pp. 310–316.
- [38] R. Baskaran, M. S. Basha, J. Amudhavel, K. P. Kumar, D. A. Kumar, and V. Vijayakumar, "A bio-inspired artificial bee colony approach for dynamic independent connectivity patterns in VANET," in *2015 International conference on circuits, power and computing technologies [ICCPCT-2015]*. IEEE, 2015, pp. 1–6.
- [39] F. Riaz, S. Ahmed, I. Shafi, M. Imran, N. I. Ratyal, and M. Sajid, "White space optimization using memory enabled genetic algorithm in vehicular cognitive radio," in *2012 IEEE 11th International Conference on Cybernetic Intelligent Systems (CIS)*. IEEE, 2012, pp. 133–140.
- [40] A. R. Deshmukh and S. S. Dorle, "Bio-inspired optimization algorithms for improvement of vehicle routing problems," in *2015 7th International Conference on Emerging Trends in Engineering & Technology (ICETET)*. IEEE, 2015, pp. 14–18.
- [41] R. J. Gegear and T. M. Laverty, "Effect of a colour dimorphism on the flower constancy of honey bees and bumble bees," *Canadian journal of zoology*, vol. 82, no. 4, pp. 587–593, 2004.
- [42] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer , Tech. Rep., 2005.

- [43] M. R. Jabbarpour, A. Jalooli, E. Shaghaghi, R. M. Noor, L. Rothkrantz, R. H. Khokhar, and N. B. Anuar, “Ant-based vehicle congestion avoidance system using vehicular networks,” *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 303–319, 2014.
- [44] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Bilen, R. C. Reinhart, and D. J. Mortensen, “Multi-objective reinforcement learning-based deep neural networks for cognitive space communications,” in *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*. IEEE, 2017, pp. 1–8.
- [45] P. V. R. Ferreira, R. Paffenroth, and A. M. Wyglinski, “Interactive multiple model filter for land-mobile satellite communications at ka-band,” *IEEE Access*, vol. 5, pp. 15 414–15 427, 2017.
- [46] F. Riaz, S. Ahmed, I. Shafi, M. Imran, N. I. Ratyal, and M. Sajid, “White space optimization using memory enabled genetic algorithm in vehicular cognitive radio,” in *2012 IEEE 11th International Conference on Cybernetic Intelligent Systems (CIS)*. IEEE, 2012, pp. 133–140.
- [47] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization,” *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [48] H. Hakoyamai, “The Ideal Free Distribution When the Resource Is Variable,” *Behavioral Ecology*, vol. 14, no. 1, pp. 109–115, 2003.
- [49] K. S. Gill and A. M. Wyglinski, “Heterogeneous cooperative spectrum sensing test-bed using software-defined radios,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, pp. 1–5.
- [50] D. Stephens and J. Krebs, *Foraging Theory*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1987.
- [51] D. B. Fogel, “An introduction to simulated evolutionary optimization,” *IEEE transactions on neural networks*, vol. 5, no. 1, pp. 3–14, 1994.

- [52] T. A. Monheim, “Personal communications services: The wireless future of telecommunications,” *Fed. Comm. LJ*, vol. 44, p. 335, 1991.
- [53] M. Rahnema, “Overview of the GSM system and protocol architecture,” *IEEE Communications magazine*, vol. 31, no. 4, pp. 92–100, 1993.
- [54] D. Astély, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, “LTE: the evolution of mobile broadband,” *IEEE Communications magazine*, vol. 47, no. 4, pp. 44–51, 2009.
- [55] USRP n210 software defined radio. [Online]. Available: <https://www.ettus.com/product/category/USRP-Embedded-Series/>
- [56] M. Pätzold, *Mobile radio channels*. John Wiley & Sons, 2011.
- [57] S. Eichler, “Performance evaluation of the IEEE 802.11p WAVE communication standard,” in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. IEEE, 2007, pp. 2199–2203.
- [58] Y. J. Li, “An overview of the DSRC/WAVE technology,” in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer, 2010, pp. 544–558.
- [59] Y. Hu, J. Feng, and W. Chen, “A LTE-cellular-based V2X solution to future vehicular network,” in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, May 2018, pp. 2658–2662.
- [60] K. Bilstrup, E. Uhlemann, E. G. Ström, and U. Bilstrup, “Does the 802.11p MAC method provide predictable support for low delay communications?” in *1st ETSI TC ITS Workshop, Sophia Antipolis, France*, 2009.
- [61] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, “Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 124–130, JUNE 2018.

- [62] J. Lioris, R. Pedarsani, F. Y. Tascikaraoglu, and P. Varaiya, “Doubling Throughput in Urban Roads by Platooning,” in *Proc. IFAC Symposium on Control in Transportation Systems*, Istanbul Turkey, 2016.
- [63] R. Rajamani, Han-Shue Tan, Boon Kait Law, and Wei-Bin Zhang, “Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, July 2000.
- [64] R. G. Rajeswar and R. Ramanathan, “An Empirical study on MAC layer in IEEE 802.11p/WAVE based Vehicular Ad hoc Networks,” *Procedia Computer Science*, vol. 143, pp. 720 – 727, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050918321410>
- [65] K. Harrison, S. M. Mishra, and A. Sahai, “How much white-space capacity is there?” in *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, April 2010, pp. 1–10.
- [66] J. van de Beek, J. Riihijarvi, A. Achtzehn, and P. Mahonen, “Tv white space in europe,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 178–188, Feb 2012.
- [67] S. Chen, R. Vuyyuru, O. Altintas, and A. M. Wyglinski, “Learning-based channel selection of vdsa networks in shared tv whitespace,” in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, Sep. 2012, pp. 1–5.
- [68] K. S. Gill, K. McClintick, N. Kanthasamy, J. Tolbert, D. Nguyen, S. Nguyen, G. Wernsing, V. Moore, I. Gelman, A. O’Neil, N. Schubert, C. Coogan, K. Murdy, B. Mahan, S. Halama, K. N. Heath, E. F. Ryder, R. J. Gegear, and A. M. Wyglinski, “Experimental test-bed for bumblebee-inspired channel selection in an ad-hoc network,” in *2018 IEEE 88th Vehicular Technology Conf. (VTC-Fall)*, Aug 2018, pp. 1–5.

- [69] P. Sroka, P. Kryszkiewicz, and A. Kliks, "Radio environment maps for dynamic frequency selection in V2X communications," in *2020 IEEE 91st Vehicular Technology Conference: VTC2020-Spring*, 2020, accepted for publication.
- [70] S. Mirjalili, "Genetic algorithm," in *Evolutionary algorithms and neural networks*. Springer, 2019, pp. 43–55.
- [71] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
- [72] Geometry-based efficient propagation model vehicles for v2v communications (GEMV2). [Online]. Available: <http://vehicle2x.net/>
- [73] M. Boban, J. Barros, and O. K. Tonguz, "Geometry-based vehicle-to-vehicle channel modeling for large-scale simulation," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4146–4164, 2014.
- [74] B. Aygun, M. Boban, J. P. Vilela, and A. M. Wyglinski, "Geometry-based propagation modeling and simulation of vehicle-to-infrastructure links," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–5.
- [75] Simulation of urban mobility (SUMO). [Online]. Available: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/
- [76] G. Stern, "Queueing systems, volume 2: computer applications," 1978.
- [77] H.-P. Shiang and M. Van Der Schaar, "Queueing-based dynamic channel selection for heterogeneous multimedia applications over cognitive radio networks," *IEEE Transactions on multimedia*, vol. 10, no. 5, pp. 896–909, 2008.
- [78] A. Azarfar, J.-F. Frigo, and B. Sanso, "Analysis of cognitive radio networks based on a queueing model with server interruptions," in *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012, pp. 1703–1708.

- [79] H. Li and Z. Han, “Socially optimal queuing control in cognitive radio networks subject to service interruptions: To queue or not to queue?” *IEEE Transactions on Wireless Communications*, vol. 10, no. 5, pp. 1656–1666, 2011.
- [80] S. Chen, A. M. Wyglinski, S. Pagadarai, R. Vuyyuru, and O. Altintas, “Feasibility analysis of vehicular dynamic spectrum access via queueing theory model,” *IEEE Communications Magazine*, vol. 49, no. 11, pp. 156–163, 2011.
- [81] V. D. Khairnar and K. Kotecha, “Performance of vehicle-to-vehicle communication using IEEE 802.11 p in vehicular ad-hoc network environment,” *arXiv preprint arXiv:1304.3357*, 2013.
- [82] D. Bertsekas and R. Gallager, *Data networks*. Athena Scientific, 1992.
- [83] S. Fowler, C. H. Häll, D. Yuan, G. Baravdish, and A. Mellouk, “Analysis of vehicular wireless channel communication via queueing theory model,” in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1736–1741.
- [84] M. Boban, J. Barros, and O. K. Tonguz, “Geometry-based vehicle-to-vehicle channel modeling for large-scale simulation,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4146–4164, 2014.
- [85] Dynamic Spectrum Alliance, “Model rules and regulations for the use of television white spaces v2.0,” 2017. [Online]. Available: <http://dynamicspectrumalliance.org/wp-content/uploads/2018/01/Model-Rules-and-Regulations-for-the-use-of-TVWS.pdf>
- [86] R. J. Gegear and T. M. Laverty, “Flower constancy in bumblebees: a test of the trait variability hypothesis,” *Animal Behaviour*, vol. 69, no. 4, pp. 939–949, 2005.
- [87] K. S. Gill, K. N. Heath, R. J. Gegear, E. F. Ryder, and A. M. Wyglinski, “On the capacity bounds for bumblebee-inspired connected vehicle networks via queueing theory,” in *2018 IEEE 87th Vehicular Technology Conference (VTC-Spring)*, in press.
- [88] MATLAB. [Online]. Available: <https://www.mathworks.com/products/matlab.html>

- [89] A. Bazzi, “Congestion control mechanisms in IEEE 802.11p and sidelink C-V2X,” in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 1125–1130.
- [90] J. J. Lehtomki, M. Lopez-Bentez, K. Umebayashi, and M. Juntti, “Improved channel occupancy rate estimation,” *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 643–654, 2015.
- [91] “IEEE Standard for Information technology–Telecommunications and information exchange between systems local and metropolitan area networks–Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” 2016.
- [92] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister, and I. Thibault, “3GPP C-V2X and IEEE 802.11p for vehicle-to-vehicle communications in highway platooning scenarios,” *Ad Hoc Networks*, vol. 74, pp. 17–29, 2018.
- [93] M. Sybis, V. Vukadinovic, M. Rodziejewicz, P. Sroka, A. Langowski, K. Lenarska, and K. Wesolowski, “Communication aspects of a modified cooperative adaptive cruise control algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4513–4523, 2019.
- [94] M. L. Jakobsen, T. Pedersen, and B. H. Fleury, “Simulation of birth-death dynamics in time-variant stochastic radio channels,” in *23th International Zurich Seminar on Communications (IZS 2014)*. ETH-Zürich, 2014.
- [95] P. Wang, B. Di, H. Zhang, K. Bian, and L. Song, “Cellular V2X communications in unlicensed spectrum: Harmonious coexistence with VANET in 5G systems,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5212–5224, Aug 2018.
- [96] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Computer networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

- [97] S. Chen, R. Vuyyuru, O. Altintas, and A. M. Wyglinski, “On optimizing vehicular dynamic spectrum access networks: Automation and learning in mobile wireless environments,” in *2011 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2011, pp. 39–46.
- [98] S. Pagadarai, A. M. Wyglinski, and R. Vuyyuru, “Characterization of vacant UHF TV channels for vehicular dynamic spectrum access,” in *2009 IEEE Vehicular Networking Conference (VNC)*, Oct 2009, pp. 1–8.
- [99] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, “Openairinterface: A flexible platform for 5G research,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.
- [100] C. Y. Yeoh, M. H. Mokhtar, A. A. A. Rahman, and A. K. Samingan, “Performance study of LTE experimental testbed using Openairinterface,” in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, Jan 2016, pp. 617–622.
- [101] X. Jiang and F. Kaltenberger, “Demo: an LTE compatible massive MIMO testbed based on openairinterface,” in *WSA 2017; 21th International ITG Workshop on Smart Antennas*, March 2017, pp. 1–2.
- [102] Openairinterface-based LTE-sidelink test-bed. [Online]. Available: <https://gitlab.eurecom.fr/oai/openairinterface5g/tree/LTE-sidelink>
- [103] C. Sanders and Y. Wang, “Localizing spoofing attacks on vehicular GPS using vehicle-to-vehicle communications,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 656–15 667, 2020.
- [104] S. Fujii, A. Fujita, T. Umedu, S. Kaneda, H. Yamaguchi, T. Higashino, and M. Takai, “Cooperative vehicle positioning via V2V communications and onboard sensors,” in *2011 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, 2011, pp. 1–5.
- [105] D. Rajaveerappa and F. H. Belgasseem, “A two ray propagation model for finding mo-

- bile location over plain microcellular environment,” in *2010 International Conference on Computer and Communication Technology (ICCCCT)*. IEEE, 2010, pp. 408–414.
- [106] R. Sari and H. Zayyani, “RSS localization using unknown statistical path loss exponent model,” *IEEE Communications Letters*, vol. 22, no. 9, pp. 1830–1833, 2018.
- [107] K. S. Gill, S. Nguyen, M. M. Thein, and A. M. Wyglinski, “Three-way deep neural network for radio frequency map generation and source localization,” *arXiv preprint arXiv:2111.12175*, 2021.
- [108] K. Haneda, J. Zhang, L. Tan, G. Liu, Y. Zheng, H. Asplund, J. Li, Y. Wang, D. Steer, C. Li, T. Balercia, S. Lee, Y. Kim, A. Ghosh, T. Thomas, T. Nakamura, Y. Kakishima, T. Imai, H. Papadopoulos, T. S. Rappaport, G. R. MacCartney, M. K. Samimi, S. Sun, O. Koymen, S. Hur, J. Park, C. Zhang, E. Mellios, A. F. Molisch, S. S. Ghassamzadeh, and A. Ghosh, “5g 3gpp-like channel models for outdoor urban microcellular and macrocellular environments,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–7.
- [109] L. Shi and K. W. Sung, “Spectrum requirement for vehicle-to-vehicle communication for traffic safety,” in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, 2014, pp. 1–5.
- [110] B. Chen, G. Hu, D. W. Ho, and L. Yu, “Distributed covariance intersection fusion estimation for cyber-physical systems with communication constraints,” *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4020–4026, 2016.
- [111] GPS logger. [Online]. Available: <https://www.basicairdata.eu/projects/android/android-gps-logger/getting-started-guide-for-gps-logger/>
- [112] RTL-SDR module. [Online]. Available: <https://github.com/librtlsdr/librtlsdr>
- [113] GNURadio DSP framework. [Online]. Available: <https://www.gnuradio.org/>
- [114] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern,

M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2>