UCLA
UCLA Electronic Theses and Dissertations

Title
Traffic-Aware Spectrum Sharing Protocols

Permalink
https://escholarship.org/uc/item/63n1n4nc

Author
Liu, Chun-Hao

Publication Date
2015

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Los Angeles

Traffic-Aware Spectrum Sharing Protocols

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Chun-Hao Liu

2015

© Copyright by
Chun-Hao Liu
2015

Abstract of the Dissertation

Traffic-Aware Spectrum Sharing Protocols

by

Chun-Hao Liu

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2015

Professor Danijela Cabric, Chair

Current radio frequency (RF) spectrum is shown to be highly underutilized in both spatial and temporal domain. One potential solution to improve the spectrum utilization is through cognitive radio (CR). Traditional CR approach adopts the sense-and-transmit strategy to exploit more transmission opportunities. After CR senses the RF channel, it will utilize this channel as long as it is not occupied by licensed users. However, there are some limitations using this approach. First, if the licensed users appear while CR is transmitting on the same channel, the collision occurs. Second, if the licensed users stop transmitting and CR does not detect this opportunity, the throughput loss occurs.

In this dissertation, we focus on studying the statistical properties of the licensed channel bands, and explore the approaches to exploit the traffic knowledge to improve the traditional CR strategy. The statistical knowledge of the busy/idle period on the licensed channel allows CR to switch to better channels to increase throughput. To this end, our goal is to develop traffic-aware spectrum sharing protocols. As a first step, we proposed a traffic classifier to classify the distribution for the idle/busy period using less number of observations and extended to the cases where there is no perfect knowledge of traffic parameters and periods. We also proposed a traffic prediction algorithm based on Markov chain derivation and estimated traffic parameters, and we analyzed the impact of estimation errors on the prediction accuracy. Based on this result, we further designed spectrum sharing protocols with multi-channel sensing strategies for CR users. The CR users are scheduled for their transmission

and cooperative sensing strategies in order to achieve maximal user throughput. We proposed to solve the scheduling optimization problem with four low-complexity strategies, i.e., sequential, parallel, sequential-parallel, and iterative-parallel sensing strategies. The optimal sensing order for sequential strategy was derived, and we also proposed several heuristics to further reduce the computation complexity for other strategies. Furthermore, this protocol is designed to be robust to both channel and traffic uncertainties.

The dissertation of Chun-Hao Liu is approved.

Songwu Lu

Lara Dolecek

Kung Yao

Danijela Cabric, Committee Chair

University of California, Los Angeles

2015

To my family.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to thank my family and my girlfriend for their strong support during my PhD life. My father, who is my role model of being perseverance and hard working; my mother, who guides me toward a better life; my brother, being excellent in academic who is my best friend and consulter; my girlfriend, showing her trust and mental support who helps me go through all these difficulties.

I would like to give my big thank to my advisor, Professor Danijela Cabric. I learned a lot from her about doing rigorous research, and she always provides me with great guidance and comments. I also thank her for giving me enough freedom to explore the wireless fields and being frank to point out my weakness. I would like to appreciate Professors Przemysław Pawełczak, Jean-Francois Frigon, and Brunilde Sanso for their great advices and contributions to our cooperated works. I am also grateful to Professors Kung Yao, Lara Dolecek, and Songwu Lu for serving on my doctoral committee and providing me with insightful feedback regarding my research.

Finally, my PhD life is abundant owing to all my colleagues Arash Azarfar, Mihir Laghate, Shailesh Chaudhari, Han Yan, Ghaith Hattab, Wesam Gabran, Eric Rebeiz, Jason Tran, Jun Wang, Paulo Urriza, Tsung-Han Yu, and Professor Liping Du. I thank them for their precious friendship and academic support. I would like to thank all my friends I met in the US for their accompany and I really treasure those good days being with them.

Vita

2003–2007	B.S. in Electronics Engineering, National Chiao Tung University, Taiwan
2007–2009	M.S. in Electronics Engineering, National Taiwan University, Taiwan
2013–2015	Ph.D. Candidate in Electrical Engineering, University of California, Los Angeles, California, USA

Publications

[J1] Arash Azarfar, Chun-Hao Liu, Jean-Francois Frigon, Brunilde Sanso, and Danijela Cabric, "Joint Transmission and Cooperative Spectrum Sensing Scheduling Optimization in Multi-Channel Dynamic Spectrum Access Networks", submitted to IEEE Transactions on Cognitive Communications and Networking, Jul. 2015.

[J2] Chun-Hao Liu, Arash Azarfar, Jean-Francois Frigon, Brunilde Sanso, and Danijela Cabric, "Robust Cooperative Spectrum Sensing Scheduling Optimization in Multi-Channel Dynamic Spectrum Access Networks", accepted to IEEE Transactions on Mobile Computing, Jul. 2015.

[J3] Chun-Hao Liu and Danijela Cabric, "Prediction of Erlang-2 Distributed Primary User Traffic for Dynamic Spectrum Access", accepted to IEEE Wireless Communications Letters, May 2015.

[J4] Chun-Hao Liu, Przemyslaw Paweczak, and Danijela Cabric, "Primary User Traffic Classification and Analysis in Dynamic Spectrum Access Networks", IEEE Journal on Selected Areas in Communications, vol. 32, no. 11, pp. 2237–2251, Nov. 2014.

[J5] Chun-Hao Liu, Jason Tran, Przemyslaw Paweczak, and Danijela Cabric, “Traffic-Aware Channel Sensing Order in Dynamic Spectrum Access Networks”, IEEE Journal on Selected Areas in Communications, vol. 31, no. 11, pp. 2312–2323, Nov. 2013.

[J6] Wesam Gabran, Chun-Hao Liu, Przemyslaw Paweczak, and Danijela Cabric, “Primary User Traffic Estimation for Dynamic Spectrum Access”, IEEE Journal on Selected Areas in Communications, vol. 31, no. 3, pp. 544–558, Mar. 2013.

[C1] Chun-Hao Liu and Danijela Cabric, “Traffic-Aware Backoff Design for IEEE 802.11(ac)”, submitted to IEEE Cognitive Computing and Networking Symposium (ICNC), Jul. 2015.

[C2] Arash Azarfar, Chun-Hao Liu, Jean-Francois Frigon, Brunilde Sanso, and Danijela Cabric, “Cooperative Spectrum Sensing Scheduling Optimization in Multi-Channel Dynamic Spectrum Access Networks”, in IEEE GLOBAL Communications Conference (GLOBECOM), Dec. 2014.

[C3] Chun-Hao Liu, Eric Rebeiz, Przemyslaw Paweczak, and Danijela Cabric, “Primary User Traffic Classification in Dynamic Spectrum Access Networks”, in IEEE GLOBAL Communications Conference (GLOBECOM), Dec. 2013.

[C4] Wesam Gabran, Przemyslaw Paweczak, Chun-Hao Liu, and Danijela Cabric, “Blind Estimation of Primary User Traffic Parameters in Dynamic Spectrum Access Networks Under Sensing Errors”, IEEE International Conference on Communications (ICC), Jun. 2013.

[C5] Chun-Hao Liu, Wesam Gabran, and Danijela Cabric, “Prediction of Exponentially Distributed Primary User Traffic for Dynamic Spectrum Access”, IEEE GLOBAL Communications Conference (GLOBECOM), Dec. 2012.

CHAPTER 1

Introduction

1.1 Introduction

Due to the proliferation of wireless devices to support the Internet of Things (IoT) in recent years, the radio frequency (RF) spectrum becomes more crowded as all devices are competing for the limited licensed spectrum resources. Furthermore, higher quality (such as data rate) for these devices is required in order to support services from voice-only to multimedia communications. This stringent requirement makes how to efficiently use the scarce spectrum resources even more important while challenging. Hence, solving the spectrum utilization and scarcity problem is definitely an urgent issue.

In order to address this problem, several spectrum measurement campaigns in [Yan05, SCZ10b, WRM09, MTM06, CYZ09, PPS11, BGR14] show that the current fixed allocation of spectrum is inefficient and causes most of channels being underutilized. One such result in [MTM06], showing the measurements completed in Chicago, is presented in Fig. 1.1. It has shown that the licensed channels are mostly used under 25% over time, leading to a large amount of spectrum vacancies, which is inefficient. In addition to [MTM06], i.e., the measurements done in USA, the authors in [PPS11] further summarized the worldwide spectrum occupancy measurement campaigns. They mentioned the average spectrum occupancy found to be 22.57% for the frequency range from 75 MHz to 3 GHz in Barcelona, Spain. Specifically the cellular GSM 900 and the UMTS uplink bands are both in a very low activity. The measurement campaign done in Singapore showed that the average spectrum occupancy is only 4.54% from 80 to 5850 MHz. The measurements done in New Zealand showed that the actual spectrum utilization from 806 MHz to 2750 MHz is only about 6.21%

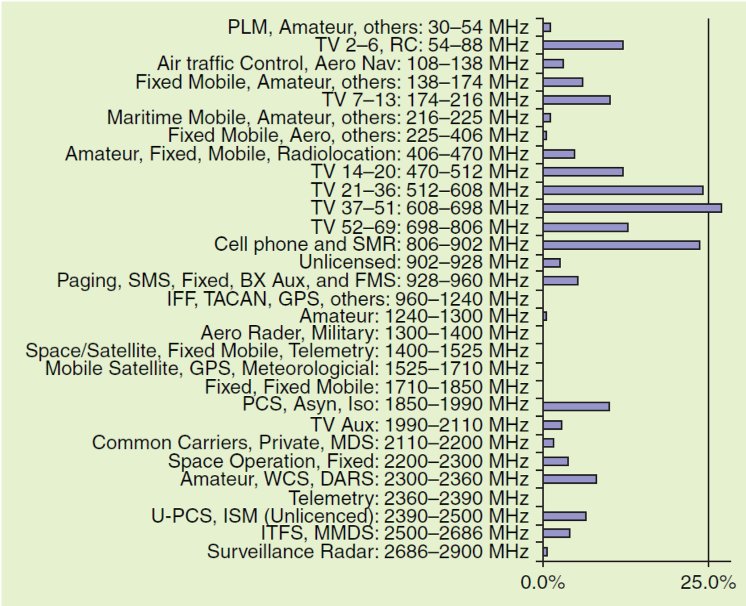


Figure 1.1: Average spectrum utilization taken over multiple locations [MTM06].

for outdoor location and 5.72% for indoor location. Another spectrum measurement results done in Waknaghat [BGR14], i.e., a rural area in India, showed that the RF spectrum from 80 MHz to 500 MHz is utilized under 15% on average. They pointed out that specifically for the frequency range from 220 MHz to 270 MHz for space operation and fixed mobile, the spectrum utilization is almost always empty. A recently published spectrum measurement work completed in Grand Forks, USA [SRK15] concluded that the spectrum occupancy of any channel over a frequency range of 824 MHz to 5.8 GHz was less than 20%. To summarize, the spectrum utilization measurement campaign results conducted over the world in the recent ten years are listed in Table 1.1. From the spectrum utilization table, we conclude that the spectrum utilization rate is very low in most radio communication bands in a broad range of countries.

1.1.1 Cognitive Radio Approach

Cognitive radio (CR) is a promising solution to solve the low spectrum utilization problem and it attracts a lot of interest in recent years. The key idea for CR is to possibly exploit all spectrum holes. The spectrum holes will be used by secondary users (SUs), temporally

Table 1.1: Summary of Spectrum Measurement

Area	Year	Utilization	Frequency	Reference
USA (Chicago)	2006	17.4%	30–2900 MHz	[MTM06]
New Zealand (Auckland)	2007	6.2%	806–2750 MHz	[CRS07]
Singapore	2008	4.54%	80–5850 MHz	[IKO08]
Spain (Barcelona)	2009	22.57%	75–3000 MHz	[LUC09]
France (Paris)	2010	10.7%	400–3000 MHz	[VMB10]
Japan (Yokosuka)	2011	11.45%	90–3000 MHz	[CVF11]
UK (Hull)	2012	9.1%	80–2700 MHz	[MRA12]
Brazil (Rio de Janeiro)	2013	19.6%	144–2690 MHz	[LS13]
Finland (Oulu)	2013	11.93%	2.4 GHz (5 MHz)	[HLK13]
Poland (Poznan)	2013	27%	223–2290 MHz	[KKP13]
China (Beijing)	2013	15.2%	440–2700 MHz	[XFC13]
India (Waknaghat)	2014	$< 15\%$	80–500 MHz	[BGR14]
USA (Grand Forks)	2015	$< 20\%$	824–5800 MHz	[SRK15]

or spatially, when their interference to primary (licensed) users (PUs) is not affecting their transmission. By using CR, the spectrum utilization would be increased tremendously and the spectrum scarcity problem caused by fix allocation can be potentially solved.

The basic function of any cognitive radio can be simplified into a cycle of exploration and exploitation. This cycle was first popularly presented in the work of Mitola and Maguire [MM99]. A simplified version of this cycle from Haykin [Hay05] is illustrated in Fig. 1.2. In the exploration stage, the CR transmitter transmits the signal through RF environment, and the CR receiver can observe the RF channel based on the received signal quality. After learning the channel information, the CR starts to exploit it. The CR receiver will send the feedback signal to the transmitter, guiding it to adjust its transmit power or channel exploring strategy. For instance, if CR can predict channel status in advance based on the PU traffic knowledge, it can switch its transmitter to search for more opportunities on other channels to avoid collisions and to increase throughput. The authors in [LCL11] presented the CR architecture for network, media access control (MAC), and physical (PHY) layers. The main functionalities were shown in [LCL11, Fig. 3]. This architecture consists with the one proposed by [Hay05], which investigated the environmental learning by several spectrum sensing techniques and transceiver optimization. In addition to the PHY layer, [LCL11] provided more compact solutions in the upper layers such as sensing scheduling, sensing access, and spectrum-aware routing algorithms.

1.1.2 Traditional Cognitive Radio Challenges

Traditional CR adopts a simple sense-and-transmit strategy [LZP08]. CR is assumed to transmit its signal slot by slot. First, CR senses the channel using spectrum sensing techniques such as energy detection, to determine if PU exists or not. Then the users can exploit the channel vacancies if we find the channel is not occupied by PUs. However, there are several limitations for this traditional approach. First, traditional CR only considers one channel at a time. This means we may miss transmission opportunities on other channels. In addition, relying on sense-and-transmit strategies can not avoid the collision/throughput

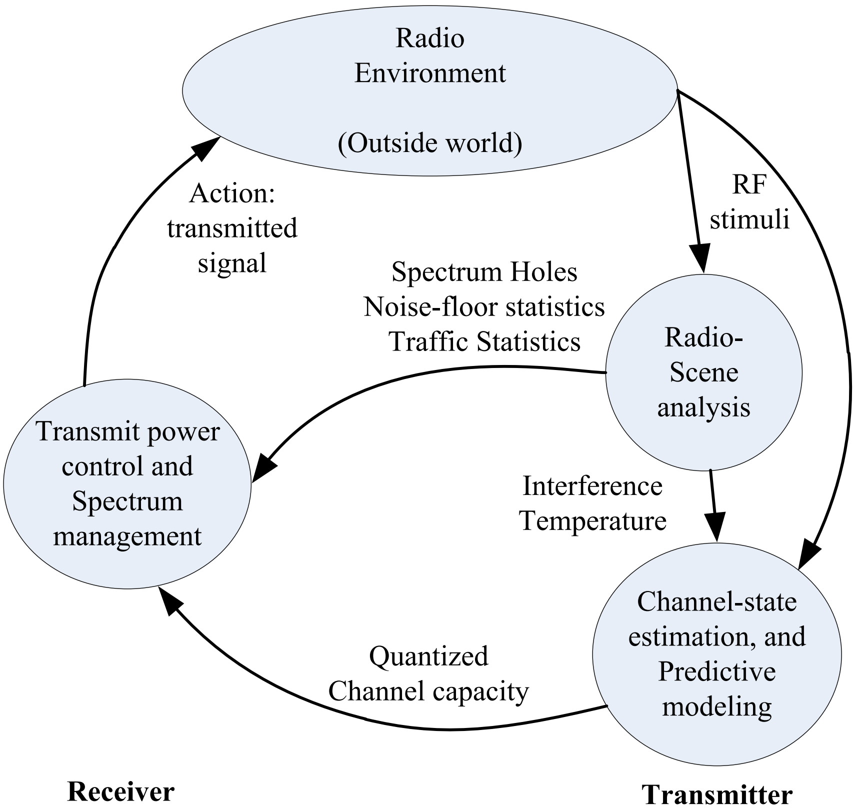


Figure 1.2: Cognitive Cycle [Hay05]

loss. For example, the PU may appear/disappear later on the same channel even if we initially sense it as idle/busy, i.e., the collision/throughput loss occurs. Second, traditional CR techniques only adopt one SU to sense the channel. In this case, the sensing time is long compared with the SU time slot. It is obvious that if CRs require longer sensing time for sensing, which means SUs have less transmission time space and it will lower the SU transmission throughput, assuming SU has a fixed time slot.

1.1.3 Motivations for Traffic-Aware Cognitive Radio Approach

In order to improve the traditional CR approach, CRs want to exploit the traffic knowledge to design a better spectrum utilization strategy. We consider multiple channels and users in a CR network. First CRs can observe traffic states and learn the traffic statistics. With the traffic statistics, the CR not only knows roughly the utilization for each channel, but it also can predict the future status for the channel. For instance, if the CR finds out the channel has large probability of being used, it should try to avoid using this channel, or to switch to other channels before the collision occurs. On the other hand, if the channel is found to be idle most of the time, the CR should spend more effort to search transmission opportunities on it. Second, CRs can cooperate multiple users to reduce the sensing time on each channel since it is a well-known result that multiple sensors can provide a better sensing accuracy within a shorter interval. Moreover, CRs can exploit the traffic to better schedule sensing process for SUs. For example, CRs may prefer to cooperate more users to sense the channel with light traffic in order to gain more transmission opportunities.

For all the advanced strategies for CR approach, the key point is that we need accurate traffic knowledge to implement them.

1.2 Overview of Related Work

There are two major categories in terms of traffic-aware cognitive radio design. The first category is the traffic analysis. Traffic analysis, including traffic classification, traffic prediction, and traffic estimation are well-developed in wireless/wired communication fields. For example, classify the Internet Protocol (IP) traffic in an IP network [NA08] or PU traffic pattern in a dynamic spectrum access (DSA) network [HPM10]. For traffic prediction, it is well-studied that if we know the PU traffic behavior in advance, we can further improve the wireless communication systems, e.g., efficiently allocating multi-carrier resource [LJX08] or minimizing transmission energy [TPC11]. Understanding traffic parameters is another famous topic, which is done through traffic estimation. Several traffic estimation techniques were proposed [TTZ12, GLP13b] to estimate the mean channel idle/busy time, and their estimation bounds were also derived in the paper. The second category is to design spectrum sharing protocols for the CR network. Most current literature focuses on the channel sensing order problem, i.e., to figure out how to schedule the SUs to sense the channels in order to achieve maximal throughput or minimal sensing delay [KS08a, CZ11].

However, none of the above mentioned works provide an efficient framework to link the traffic analysis and the CR protocol design. Specifically for the protocol design, they proposed to incorporate perfect traffic statistics in their protocol design. But in reality, the traffic statistics can not be perfectly known. We need to apply traffic estimation techniques to obtain the traffic statistics. Therefore, in this dissertation we aim at looking into deeper development of traffic analysis, while incorporating the traffic analysis into the spectrum sharing protocol design and analyze how the traffic estimation errors affect the protocol performance.

1.3 Proposed Cognitive Radio Architecture

In this dissertation we propose two tasks to design spectrum sharing protocols shown in Fig. 1.3. In the first task, we would like to understand the statistical properties of licensed channels. To achieve this, we classify the accurate PU idle/busy time distribution, and estimate some key parameters for the distribution. Based on these parameters, we further design a channel predictor which can predict future channel state given past observations. In task 2, we propose to integrate traffic statistics from task 1 into two protocols, i.e., sequential sensing and opportunistic spectrum access (OSA).

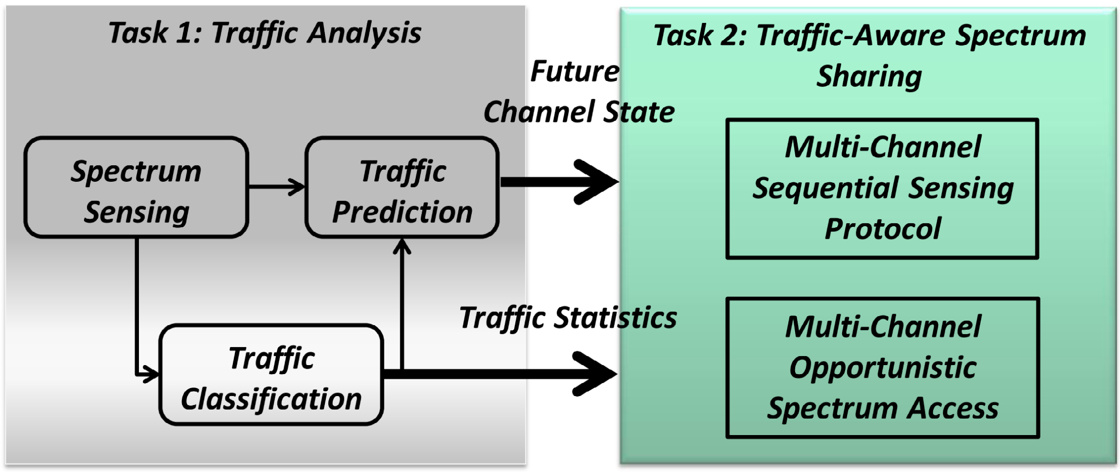


Figure 1.3: Proposed system architecture

1.3.1 Traffic Analysis

The first step in traffic analysis is traffic classification. We focuses on analytical studies of the PU traffic classification problem. Observing that the gamma distribution can represent positively skewed data and exponential distribution (popular in communication networks performance analysis literature) it is considered here as the PU traffic descriptor. We investigate two PU traffic classifiers utilizing perfectly measured PU activity (busy) and inactivity (idle) periods: (i) maximum likelihood classifier (MLC) and (ii) multi-hypothesis sequential probability ratio test classifier (MSPRTC). Then, relaxing the assumption on perfect period measurement, we consider a PU traffic observation through channel sampling. For a special case of negligible probability of PU state change in between two samplings, we propose a minimum variance PU busy/idle period length estimator. Later, relaxing the assumption of the complete knowledge of the parameters of the PU period length distribution, we propose two PU traffic classification schemes: (i) estimate-then-classify (ETC), and (ii) average likelihood function (ALF) classifiers considering time domain fluctuation of the PU traffic parameters. Numerical results show that both MLC and MSPRTC are sensitive to the periods measurement errors when the distance among distribution hypotheses is small, and to the distribution parameter estimation errors when the distance among hypotheses is large. For PU traffic parameters with a partial prior knowledge of the distribution, the ETC outperforms ALF when the distance among hypotheses is small, while the opposite holds when the distance is large.

The second step in traffic analysis is traffic prediction. In order to fully exploit the availability of PUs' unused spectrum by secondary users, traffic prediction can be used to increase system throughput. In this work we propose a PU traffic prediction algorithm based on estimated PU traffic state transition probabilities. The probabilities are obtained via constrained-time PU traffic parameters estimation assuming exponentially/Erlang-2 distributed PU ON/OFF channel utilization intervals. Moreover, we define prediction regions for the estimated parameters where the optimal traffic prediction is possible. Finally, we theoretically quantify the prediction confidence as a function of the prediction time, the total estimation time period and the number of samples used for estimation.

1.3.2 Traffic-Aware Spectrum Sharing

After traffic analysis, first we propose to design a sequential sensing protocol utilizing the traffic knowledge. We present new results on the problem of finding the best channel sensing order for multi-channel DSA networks. We start with the general assumption that all SUs cooperatively sense each PU channel at one time., i.e., to design a multi-channel sequential sensing protocol. Then, the SU sensing results are reported to a DSA base station that schedules SU transmissions in order to maximize DSA network throughput. We then assume that PU traffic parameters are not perfectly known to DSA network and change over time, and propose a novel PU channel sensing order scheme based on the quality of PU traffic estimation. We adopt a maximum likelihood estimator to estimate the traffic statistics of PU channels and derive the Cramér-Rao (CR) bounds for the PU traffic estimation performance. Based on the CR bound and its Gaussian approximation, we analyze the impact of the estimation error on the DSA network throughput by computing a new metric called sensing order confidence, i.e., the probability that the best selected sensing order is not affected by PU traffic estimation errors. Finally, we formulate a convex optimization problem to determine the minimum number of PU channel state samples required for estimating PU traffic parameters after determining a certain constraint on the sensing order confidence metric to achieve the best sensing order.

Then we extend previous results and look at a more general DSA sensing scheduling problem, i.e., to design a multi-channel opportunistic spectrum access (OSA) protocol. A key factor to design an OSA network is the spectrum sensing algorithms for multiple channels with multiple users. Multi-user cooperative channel sensing reduces the sensing time, and thus it increases transmission throughput. However, in a multi-channel system, the problem becomes more complex since the benefits of assigning users to sense channels in parallel must also be considered. A sensing schedule, indicating to each user the channel that it should sense at different sensing moments, must be thus created to optimize system performance. In this work, we formulate the general sensing scheduling optimization problem and then propose several sensing strategies to schedule the users according to network parameters with homogeneous sensors. Later on we extend the results to heterogeneous sensors and propose a robust scheduling design when we have traffic and channel uncertainty. We propose three sensing strategies, and, within each one of them, several solutions, striking a balance between throughput performance and computational complexity, are proposed. In addition, we show that a sequential channel sensing strategy is the one to be preferred when the sensing time is small, the number of channels is large, and the number of users is small. For all the other cases, a parallel channel sensing strategy is recommended in terms of throughput performance. We also show that a proposed hybrid sequential-parallel channel sensing strategy achieves the best performance in all scenarios at the cost of extra memory and computation complexity.

1.4 Thesis Organization

In this thesis, we develop two traffic-aware spectrum sharing protocols, i.e., sequential sensing protocol and opportunistic spectrum access protocol, by utilizing the traffic knowledge. In Chapter 2, we propose several classifiers to classify PU traffic distribution. We also analyze the performance for the traffic classifier if we have traffic parameter estimation errors.

A design guideline is provided for energy-efficient or delay-efficient sensing design to achieve a certain classification accuracy. In Chapter 3, we propose a traffic predictor based on the Markov chain for PUs. Then we propose a prediction region to link the traffic prediction performance and the traffic parameter estimation errors. A prediction confidence is proposed to quantify the probability that the prediction performance is not affected by the parameter estimation errors, using the prediction region. In Chapter 4, we design a sequential sensing strategy for SUs to maximize their throughput in a DSA network. We also link the throughput performance and the traffic estimation through a sensing confidence. In Chapter 5 we discuss a more general sensing scheduling problem that is solved by not only the sequential strategy, but also the parallel and sequential-parallel sensing strategies. Furthermore, the sensing strategies are designed to be robust to channel and traffic parameter estimation variations. Finally, this work is concluded in Chapter 6.

CHAPTER 2

Primary User Traffic Classification in Dynamic Spectrum Access Networks

2.1 Introduction

Understanding the PUs' channel occupancy distributions becomes important from a theoretical point of view [GLP13a], but most importantly it allows to improve seamless DSA operation [KYY12, Sec. IV-B], [RW13, Fig. 2]. For example, if SUs have sufficient knowledge about the PUs' traffic distributions, they can minimize the channel switching latency [KS08a], predict the PUs' behavior to minimize interference [LGC12] or find an optimal PU channel sensing order [LTP13]. Therefore, the SUs should accurately estimate the PUs' traffic distribution, i.e., classify the PU traffic correctly from a set of possible distributions, e.g., exponential, gamma, log-normal, and Weibull distributions as tested in [LC13]. Looking at the recent DSA/OSA applications, traffic classification can be used in Licensed Shared Access [PMP14] (LSA) systems, where traffic classification would help in identifying the behavior of individual LSA licensees [AK14] and adapting licensing rules accordingly.

2.1.1 Related Work

Traffic classification is an important research area in many telecommunication domains, e.g. in IP networks [NA08]. In parallel, analytical modeling of IP traffic has also been concerned, refer to a discussion in e.g. [Pax98, Sec. III-D]. In the DSA area, the topic has started to receive attention as well. Considering relevant works that aim at PU traffic classification, [HPM10] was the first to deal with traffic pattern classification in DSA networks. Therein, the classification of the traffic pattern was done by using the autocorrelation function of the received PU signal. Work of [HSM11] improved the classification algorithm of [HPM10] by filtering away the errors that were caused by noise and incorrect spectrum sensing. Inspired by machine learning, the authors in [DB12] proposed two behavior classifiers, namely a naive Bayesian classifier and an averaged one-dependence estimation classifier to classify the channel selection strategy for SUs. However, the authors of [HPM10, HSM11] considered the PU traffic pattern to be either stochastic or deterministic, without assigning the PU traffic to a specific distribution. Furthermore, the classifier of [DB12] did not take the distributions of PU traffic but only the mean busy/idle time into consideration. We thus conclude, to the best of our knowledge, the performance of PU traffic classification is still relatively unexplored from the theoretical point of view.

2.1.2 Our Contribution

This motivated us to perform detailed theoretical studies of PU traffic classification. Since gamma distribution (including exponential and Erlang distributions) for PU traffic modeling has been measured widely in current literature [LC13, WRM09], in this work we consider the classification of gamma-distributed PU busy/idle time collected through an error-free spectrum sensing process. The contribution of our work is fourfold:

1. We analytically derive the performance for the PU traffic classifier based on maximum likelihood using Gaussian approximation;

2. We re-evaluate a sequential algorithm based on a multi-hypothesis sequential probability ratio test [BV94], to deal with the classification problem for multiple PU traffic classes, when parameters of PU traffic classes are known in advance;

3. Considering PU channel sampling, for a special case when probability of PU period change in-between two samples (busy-to-idle-to-busy or idle-to-busy-to-idle) is negligible, we evaluate (i) a minimum variance PU state length estimator, and (ii) propose a modified maximum likelihood classifier, quantifying its performance analytically and providing design guidelines based on traffic parameters;

4. Finally, we propose (i) a PU traffic estimate-then-classify scheme which requires no complete knowledge of the PU traffic parameters, and (ii) an average likelihood function method which requires knowledge on the statistics of the PU traffic parameters when they fluctuate in time domain.

In addition, we list the important limitations of our work:

1. We assume that the set size of distributions considered for classification is finite and does not change over time;

2. The effect of spectrum sensing errors at the physical layer on the classification accuracy is not considered;

3. The calculations of classification accuracy obtained in this paper depend on the exact knowledge of a subset of traffic parameters and their stationarity.

2.2 System Model

We consider a single channel randomly accessed by a PU. To ease the analysis we disregard (i) the effect of incidental SU operation within a PU band, i.e., the injection of SU traffic into PU traffic which obfuscates the correct classification of the latter, and (ii) the effect of spectrum sensing errors. The assumption (ii) is taken consciously, as the problem of traffic classification is strictly coupled with the spectrum sensing problem and requires a separate analytical study due to its complexity. For example, in [WM10, Sec. 4.2] it has been concluded that "different energy detection thresholds (...) result in significantly different [PU traffic] distributions." Recent work of [Lop13] provides a more formal discussion on the effect of sensing errors on PU traffic analysis. Nevertheless, assumptions (i) and (ii) allow us to use the results obtained in this paper also for the non-DSA scenarios and provide a classification benchmark for interference-prone and sensing error-prone cases.

Further, we assume we can obtain traffic busy/idle periods (denoted as ON/OFF, respectively) perfectly through time-domain fine-grained spectrum sensing, as in e.g. [TTZ12, Sec. II]. This assumption, in practical terms, results in a sampling time much smaller than the shortest duration of PU traffic periods. The ON/OFF periods are denoted as a random variable X with its n independent and identically distributed realizations $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T,\ x_i \in (0, \infty)$. Those are assumed to belong to one of $\mathcal{M} = \{1, \cdots, M\}$ possible gamma distributions. The gamma distribution is chosen for its flexibility to represent: (i) exponential distribution, due to its analytical popularity [JL12b, Sec. V-B] and existence in real networks, e.g. as measured in [WMB08b, Sec. IV-A] for call arrival times in CDMA-based system; and (ii) positively skewed data, which is also confirmed through the traffic measurement, e.g. in [SCL04, Fig. 10] for call holding time in public safety systems.

Our objective is to minimize the required number of measurement periods in $\mathbf{x}$ in order to classify X to the correct distribution. We can formulate such a classification problem as a multi-hypothesis problem, i.e.,

$$X \sim f_X(x) = \begin{cases} f_1(x|\mathbf{\Theta}_1), & \mathcal{H}_1, \\ f_2(x|\mathbf{\Theta}_2), & \mathcal{H}_2, \\ \quad\vdots & \\ f_M(x|\mathbf{\Theta}_M), & \mathcal{H}_M, \end{cases} \tag{2.1}$$

where $f_X(x)$ is the hypothesized probability density function (PDF) of X, $f_j(x|\mathbf{\Theta}_j) = \frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} x^{\alpha_j - 1} e^{-\beta_j x}$ is the gamma PDF of X under hypothesis $\mathcal{H}_j$ given the shape parameter α_j and the rate parameter β_j, where $\mathbf{\Theta}_j = (\alpha_j, \beta_j)^T$ and $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is the gamma function, where again $x \in (0, \infty)$. We assume that each hypothesis $\mathcal{H}_j$ has a prior probability π_j, and we define $\mathbf{\Omega} = (\pi_1, \pi_2, \cdots, \pi_M)^T$, $\sum_{i=1}^{M} \pi_i = 1$. Without loss of generality, in this paper we assume that the elements in $\mathbf{x}$ denote either PU channel occupancy periods (ON times) or idle periods only (OFF times).

2.3 Traffic Classification with Perfect Knowledge of PU Traffic Parameters

We start with assuming a perfect knowledge of all PU traffic parameters $\mathbf{\Theta}_j = (\alpha_j, \beta_j)^T$, $\forall j \in \mathcal{M}$. Firstly, we introduce a maximum likelihood classifier (MLC) that requires a constant number of PU traffic periods, which is an optimal classifier in terms of probability of correct classification when the PDFs are known [DAB07, Sec. I] and derive its classification performance for the considered model in Section 2.2. Such an analysis, to the best of our knowledge, has not been performed before. Secondly, as a comparison to MLC, we re-introduce the multi-hypothesis sequential probability ratio test classifier (MSPRTC) using [BV94] which adopts a sequential sample test instead of using a fixed number of PU traffic periods for classification.

2.3.1 Maximum Likelihood Classifier

For the considered gamma distribution $f_j(x|\mathbf{\Theta}_j)$ the likelihood function given $\mathbf{x}$ for $\mathcal{H}_j$ can be written as

$$\begin{aligned} L_{\mathcal{H}_j}(\mathbf{x}) &= \pi_j \prod_{i=1}^{n} f_j(x_i|\mathbf{\Theta}_j) \\ &= \pi_j \prod_{i=1}^{n} \left(\frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} x_i^{\alpha_j - 1} e^{-\beta_j x_i} \right), \forall j \in \mathcal{M}. \end{aligned} \tag{2.2}$$

Then, the MLC final decision, ν, is

$$\nu = \mathcal{H}_{m \triangleq \arg\max_j L_{\mathcal{H}_j}(\mathbf{x})}. \tag{2.3}$$

To analyze the MLC classification performance for the system model considered in Section 2.2, we start with calculating the log-likelihood function $g_{\mathcal{H}_j}(\mathbf{x}) \triangleq \log L_{\mathcal{H}_j}(\mathbf{x})$ which can be represented as

$$g_{\mathcal{H}_j}(\mathbf{x}) = \log \frac{\pi_j \beta_j^{n\alpha_j}}{\Gamma(\alpha_j)^n} + \sum_{i=1}^{n} [(\alpha_j - 1) \log x_i - \beta_j x_i]. \tag{2.4}$$

Then we can calculate the probability of correct classification under $\mathcal{H}_j$ using (2.4) as

$$\begin{aligned}\Pr\{\nu = \mathcal{H}_j|\mathcal{H}_j\} &= \Pr\{g_{\mathcal{H}_j}(\mathbf{x}) > g_{\mathcal{H}_k}(\mathbf{x})\}\ \forall k \in \{\mathcal{M} - \{j\}\} \\ &= \prod_{k=1,\ k\neq j}^{M} \Pr\{g_{\mathcal{H}_j}(\mathbf{x}) - g_{\mathcal{H}_k}(\mathbf{x}) > 0\}. \end{aligned} \tag{2.5}$$

Embedding (2.4) into (2.5) we can simplify (2.5) as

$$\begin{aligned}&\Pr\{\nu = \mathcal{H}_j|\mathcal{H}_j\} \\ &= \prod_{k=1,\ k\neq j}^{M} \Pr\left\{\sum_{i=1}^{n} y_i^{(j,k)} > -\log \frac{\pi_j \beta_j^{n\alpha_j}\Gamma(\alpha_k)^n}{\pi_k \beta_k^{n\alpha_k}\Gamma(\alpha_j)^n}\right\}, \end{aligned} \tag{2.6}$$

where $y_i^{(j,k)} = \alpha_{j,k}\log x_i - \beta_{j,k}x_i$ and $\alpha_{j,k} = \alpha_j - \alpha_k$, $\beta_{j,k} = \beta_j - \beta_k$. We also define the mean and variance for the variable $y_i^{(j,k)}$ as $\mu_{j,k}$ and $\sigma_{j,k}^2$, respectively, which are derived in Appendix A.1.

We can now define $\bar{y}^{(j,k)} \triangleq \sum_{i=1}^{n} y_i^{(j,k)}$ and calculate its PDF as $f\left(\bar{y}^{(j,k)}\right) = f^{(n)}\left(y_i^{(j,k)}\right)$, where $f^{(n)}(\cdot)$ denotes the n-fold PDF convolution. Then, by calculating the cumulative distribution function (CDF) of $\bar{y}^{(j,k)}$ we can obtain an exact analytical expression for (2.6). However, due to mathematical intractability of such operations we use a simple approximation instead, which has a closed-form expression, to derive the probability of correct classification. Therefore, let us transform (2.6) as

$$\Pr\{\nu = \mathcal{H}_j|\mathcal{H}_j\} = \prod_{k=1,\ k\neq j}^{M} \Pr\{z_{j,k} > \tau_{j,k}\}, \tag{2.7}$$

where $z_{j,k} = \frac{1}{\sqrt{n\sigma_{j,k}^2}} \sum_{i=1}^{n} \left(y_i^{(j,k)} - \mu_{j,k}\right)$ and $\tau_{j,k} = -\frac{1}{\sqrt{n\sigma_{j,k}^2}} \left(\log \frac{\pi_j \beta_j^{n\alpha_j}\Gamma(\alpha_k)^n}{\pi_k \beta_k^{n\alpha_k}\Gamma(\alpha_j)^n} + n\mu_{j,k}\right)$. According to the Central Limit Theorem, as n is large enough, $z_{j,k}$ will approach a standard normal distribution, $\mathcal{N}(0,1)$. Hence we can approximate (2.7) as

$$\Pr\{\nu = \mathcal{H}_j|\mathcal{H}_j\} \approx \prod_{k=1,\ k\neq j}^{M} Q(\tau_{j,k}), \tag{2.8}$$

where $Q(\cdot)$ is the tail probability function of the standard normal distribution. Finally, the average probability of correct classification P_c for all hypotheses is derived using (2.8) as

$$\boxed{P_c = \sum_{j=1}^{M} \pi_j \Pr\{\nu = \mathcal{H}_j|\mathcal{H}_j\}.} \tag{2.9}$$

2.3.2 Multi-Hypothesis Sequential Probability Ratio Test Classifier

To compare the performance with MLC, we introduce a new classification method based on MSPRTC of [BV94]. Unlike MLC which uses a constant number of PU traffic ON (or OFF) periods, MSPRTC sequentially classifies multiple hypotheses requiring only as many PU traffic periods as needed for correct classification. We adopt MSPRTC since the authors in [BV94, Sec. III] show that it provides a good approximation to the optimal solution on the condition of a perfect a priori knowledge for all distributions, i.e. their parameters, in the sequential multi-hypothesis classification problem.

MSPRTC decision is then $\nu = \mathcal{H}_{m \triangleq \arg\max_j p^j_{N_A}}$, where the posteriori probability p^j_n is given as [BV94, Sec. II]

$$p^j_n \triangleq \pi_j \prod_{i=1}^{n} f_j(x_i|\mathbf{\Theta}_j) \left[\sum_{l=1}^{M} \pi_l \left(\prod_{i=1}^{n} f_l(x_i|\mathbf{\Theta}_l) \right) \right]^{-1}. \quad (2.10)$$

We define N_A as the first $n \geq 1$ such that $p^j_n > \frac{1}{1+A_j}$ for at least one $j \in \mathcal{M}$, where $A_j > 0$ is the design threshold.

Recalling [BV94, Sec. VII] $A_k = \frac{c}{\delta_k \gamma_k}$, where $c = \frac{\alpha}{\sum_{k=0}^{M-1} \frac{\pi_k}{\delta_k}}$, α is the total probability of incorrect decision, γ_k is the constant defined in [BV94, Sec. VI] and δ_k is the measure of probabilistic distance. In [BV94, Sec. VII] $\delta_k = \min_{k:k\neq j} D(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k))$, D is the Kullback-Leibler (KL) divergence which for two gamma distributions is defined in [Pen01, Eq. (6)] and after simplifications

$$\begin{aligned} D(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k)) &\triangleq \int_{-\infty}^{\infty} f_j(x|\mathbf{\Theta}_j) \log \frac{f_j(x|\mathbf{\Theta}_j)}{f_k(x|\mathbf{\Theta}_k)} dx \\ &= (\alpha_j - \alpha_k)\psi(\alpha_j) - \log\Gamma(\alpha_j) \\ &+ \log\Gamma(\alpha_k) + \alpha_k(\log\beta_k - \log\beta_j) + \alpha_j \left(\frac{\beta_j - \beta_k}{\beta_k} \right), \end{aligned} \quad (2.11)$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the digamma function[1]

Observation 1 *The authors of [BV94] suggest to use KL for δ_k as a descriptor of probabilistic distance for two distributions. For the squared Hellinger (SH) distance, defined*

[1] For the derivation see `http://stats.stackexchange.com/questions/11646/kullbackleibler-divergence-between-two-gamma-distributions`, retrieved December 22, 2013.

as [Van98, Ch. 14.5, pp. 211]

$$H^2(f_j(x|\mathbf{\Theta}_j),f_k(x|\mathbf{\Theta}_k)) \triangleq 1 - \int_{-\infty}^{\infty} \sqrt{f_j(x|\mathbf{\Theta}_j) f_k(x|\mathbf{\Theta}_k)} dx, \tag{2.12}$$

(note that the 0.5 constant is omitted for convenience as remarked in [Pol02, Ch. 3.3, pp. 61]), it can be shown to be the lower bound of KL divergence [Cas13, Proposition 1], i.e.,

$$D(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k)) \geq H^2(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k)). \tag{2.13}$$

We thus propose to replace δ_i used in calculating the threshold for MSPRTC, A_j, with η_j where

$$\eta_j = \min_{k:k\neq j} H^2(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k)), \tag{2.14}$$

and the SH distance between two gamma distributions (considered in the system model in Section 2.2) is derived in Appendix A.2.

Observation 2 *The procedure to calculate γ_k explained in [BV94, Sec. VII] is convolved[2]. Therefore, in numerical evaluation in Section 5.6 we will replace A_k with a single value γ for all the hypotheses. To find γ, before performing classification we sweep through $\gamma \in [0, \infty)$ to determine the desired classification probability. For example, we can set $\gamma = 0$ and obtain the first classification performance. If it does not satisfy the classification system requirement, we increase γ by a pre-defined step size $\Delta\gamma > 0$ until we reach our desired classification performance.*

2.4 Joint PU Traffic Period Estimation and Traffic Classification

So far, we have assumed the continuous observation of the PU channel state. In this section we consider a more general traffic classification problem, where the elements of $\mathbf{x}$ also need to be estimated. Therefore we relax the assumption on the continuous observation of PU

[2] *Even though we used it in [LRP13] by actually not calculating it, but sweeping through a large set of values of constant γ (Bayes classification risk minimizer) to obtain a desired classification.*

state and assume a PU channel observation at instants every T_s seconds to find the elements in $\mathbf{x}$.

First, we introduce the model for the PU period length estimation in Section 2.4.1. Then, in Section 2.4.2, we propose a minimum variance period length estimator to minimize estimation errors. Subsequently, we propose a modified MLC considering estimation error and analytically derive the approximation of its classification performance in Section 2.4.3. We then propose a modified MSPRTC considering estimation error in Section 2.4.4. Finally, in Section 2.4.5 we propose a design guideline for MLC with energy or time constraints on the spectrum sensing budget.

2.4.1 Period Estimation Noise Modeling under PU Traffic Sampling

We follow the system model shown in [Lop13, Section II, Fig. 1(a)], where a PU traffic period, i.e., ON/OFF duration $T_{\text{on}}/T_{\text{off}}$, is estimated through sampling performed at regular intervals of T_s seconds. Without loss of generality, we will focus on estimating T_{on} only, while T_{off} can be estimated using the same technique. In addition, to ease the analysis, we assume that the probability of PU state change between two samplings is negligible.

Denote $s = 1$ represents the channel being busy, while $s = 0$ represents the channel being idle. Assuming as previously that we ignore spectrum sensing errors we would like to estimate the length of T_{on} based on the set of samples obtained at T_s intervals. For the actual T_{on} we denote four time instants, i.e., t_0, t_1, t_2, and t_3: (i) t_0 is the starting point with $s = 0$, (ii) t_1 and (iii) t_2 are the transition points from $s = 0$ to $s = 1$ and $s = 1$ to $s = 0$, respectively, and (iv) t_3 is the end point with $s = 0$. After sampling the traffic, we define the nearest sampling point to t_1 as ζ_1 in region (t_0, t_1) and ζ_2 in region (t_1, t_2). Similarly, we define the nearest sampling point to t_2 as ζ_3 in region (t_1, t_2) and ζ_4 in region (t_2, t_3). In other words, ζ_i are the actual discrete channel measurement points. Then we can think of this PU channel sampling as a quantization process, i.e., there are four sources of quantization noise which are $\phi_1 = t_1 - \zeta_1$, $\phi_2 = \zeta_2 - t_1$, $\phi_3 = t_2 - \zeta_3$, and $\phi_4 = \zeta_4 - t_2$. We can now model quantization error as a uniformly distributed random variable, which implies that $\phi_i \sim \mathcal{U}(0, T_s),\ \forall i \in \{1,2,3,4\}$, where $\mathcal{U}(a,b)$ denotes the uniform distribution and a, b are the minimum and maximum value for the random variable ϕ_i, respectively.

2.4.2 T_{on} Length Estimator

We first propose a minimum variance PU period length estimator that reduces the sampling noise effect. Then we derive the average number of PU traffic samples needed for T_{on} length estimation using the proposed estimator.

2.4.2.1 Minimum Variance Estimator

First we consider T_1, i.e., the interval between two nearest $s = 0$ points, where $T_1 = \zeta_4 - \zeta_1 = T_{\text{on}} + \phi_1 + \phi_4$. Then, we consider T_2, i.e., the interval between two nearest $s = 1$ points, where $T_2 = \zeta_3 - \zeta_2 = T_{\text{on}} - \phi_2 - \phi_3$. We propose a weighted average of T_1 and T_2, i.e., $T_a = wT_1 + (1-w)T_2$ as our T_{on} estimator, where $w \in [0,1]$ is the weight that needs to be designed. We know that the mean for T_a is

$$\begin{aligned}
\mathbb{E}\{T_a\} &= w\mathbb{E}\{T_1\} + (1-w)\mathbb{E}\{T_2\} \\
&= w\mathbb{E}\{T_{\text{on}} + \phi_1 + \phi_4\} + (1-w)\mathbb{E}\{T_{\text{on}} - \phi_2 - \phi_3\} \\
&= (2w-1)T_s + \mathbb{E}\{T_{\text{on}}\}, \qquad (2.15)
\end{aligned}$$

since $\mathbb{E}\{\phi_i\} = \frac{T_s}{2},\ \forall i \in \{1,2,3,4\}$. We can observe that with $w = \frac{1}{2}$, the mean of T_a will be the same as the mean of T_{on}, resulting in T_a an unbiased estimator. Then we would like to minimize the variance of T_a to derive the optimal w. Such variance is expressed as

$$\text{Var}\{T_a\} = \frac{T_s^2}{6}(w^2 + (1-w)^2) + \text{Var}\{T_{\text{on}}\}, \qquad (2.16)$$

since $\text{Var}\{\phi_i\} = \frac{T_s^2}{12},\ \forall i \in \{1,2,3,4\}$. Taking the derivative of (2.16) with respect to w and setting it to zero, we can obtain the optimal weight as $w^* = \frac{1}{2}$. Therefore, the minimum variance estimator (MVE) is expressed as

$$T_a = \frac{1}{2}(T_1 + T_2) = T_{\text{on}} + \phi_1 - \phi_3 = T_{\text{on}} - \phi_2 + \phi_4. \qquad (2.17)$$

2.4.2.2 Average Number of PU Traffic Samples for Period Estimation using Minimum Variance Estimator

The following theorem summarizes the analytical results for the average number of PU traffic samples, N, when we adopt the proposed MVE to estimate one PU state length T_{on}.

Theorem 1 *The expected average number of traffic samples for estimating one PU period is*

$$\boxed{\mathbb{E}\{N\} = \sum_{j=1}^{M} \pi_j \mathbb{E}\{N|\mathcal{H}_j\},} \tag{2.18}$$

where

$$\mathbb{E}\{N|\mathcal{H}_j\} = \sum_{k=1}^{\infty} \frac{\Gamma(\alpha_j, k\beta_j T_s)}{\Gamma(\alpha_j)} + 1. \tag{2.19}$$

Proof: See Appendix A.3.

Corollary 1 *If hypothesis $\mathcal{H}_j$ is an exponential distribution with parameter λ, then*

$$\mathbb{E}\{N|\mathcal{H}_j\} = \frac{1}{1 - e^{\lambda T_s}}. \tag{2.20}$$

Proof: We can simplify (2.19) by assigning $\alpha_j = 1$ and $\beta_j = \lambda$, which results in

$$\begin{aligned}\mathbb{E}\{N|\mathcal{H}_j\} &= \sum_{k=1}^{\infty} \frac{\Gamma(1, k\lambda T_s)}{\Gamma(1)} + 1 = \sum_{k=1}^{\infty} \int_{k\lambda T_s}^{\infty} e^{-t} dt + 1 \\ &= \sum_{k=1}^{\infty} e^{-k\lambda T_s} + 1 = \frac{1}{1 - e^{-\lambda T_s}}.\end{aligned} \tag{2.21}$$

Corollary 2 *If hypothesis $\mathcal{H}_j$ is an Erlang distribution with parameter $\alpha_j = 2$ and $\beta_j = \lambda$ then*

$$\mathbb{E}\{N|\mathcal{H}_j\} = \frac{1 - e^{-\lambda T_s} + \lambda T_s e^{-\lambda T_s}}{(1 - e^{-\lambda T_s})^2}. \tag{2.22}$$

Proof: If α_j is an integer then $\Gamma(\alpha_j) = (\alpha_j - 1)!$ and $\Gamma(\alpha_j, k\lambda T_s) = (\alpha_j - 1)! e^{-k\lambda Ts} \sum_{l=0}^{\alpha_j - 1} \frac{(k\lambda T_s)^l}{l!}$. Plugging the above two results with $\alpha_j = 2$ into (2.19) we have

$$\begin{aligned}\mathbb{E}\{N|\mathcal{H}_j\} &= \sum_{k=1}^{\infty} e^{-k\lambda T_s} \sum_{l=0}^{1} \frac{(k\lambda T_s)^l}{l!} + 1 \\ &= \left(\sum_{k=1}^{\infty} e^{-k\lambda T_s} + 1 \right) + \sum_{k=1}^{\infty} k\lambda T_s e^{-k\lambda T_s}. \end{aligned} \tag{2.23}$$

The left hand side in (2.23) can be simply obtained from (2.21), and the right hand part in (2.23) can be calculated as $\sum_{k=1}^{\infty} k\lambda T_s e^{-k\lambda T_s} = \frac{\lambda T_s e^{-\lambda T_s}}{(1-e^{-\lambda T_s})^2}$. Combining the left hand and right hand parts completes the proof.

2.4.3 MLC under PU Period Estimation Error

To derive the MLC considering PU period estimation error, we first need to derive the modified PDF for our proposed estimator. From (2.17) we can observe that the estimated PU period length is represented by the real PU traffic periods plus two uniformly distributed variables (representing sampling noise), one for the beginning and one for the end of the PU traffic period. The PDF for the combined sampling noise, $\phi = \phi_1 - \phi_3$ or $\phi = -\phi_2 + \phi_4$, can be calculated by taking the convolution of two uniform distributions, which can be expressed as a triangular function

$$f_\Phi(\phi) = \Lambda(-T_s, T_s) = \frac{I(-\phi)}{T_s^2}\phi + \frac{1}{T_s}, \tag{2.24}$$

where $I(\phi) = 1$ if $\phi \geq 0$, else $I(\phi) = -1$. By convolving the PDF for $T_{\rm on}$ and ϕ, we can obtain the PDF for T_a, which can be derived using the following theorem.

Theorem 2 *Given a random variable $\tilde{x} = x + \phi$, where x is gamma distributed with parameters $\mathbf{\Theta} = (\alpha, \beta)$ and ϕ is triangular distributed with parameter T_s, the PDF of $\tilde{x}$ can be* *expressed as*

$$f(\tilde{x}|\mathbf{\Theta},T_s)=\begin{cases}\frac{\Gamma(\alpha+1,(\tilde{x}+T_s)\beta)-2\Gamma(\alpha+1,\tilde{x}\beta)+\Gamma(\alpha+1,(\tilde{x}-T_s)\beta)}{\Gamma(\alpha)\beta T_s^2} & \\ \quad-\frac{(\tilde{x}+T_s)\Gamma(\alpha,(\tilde{x}+T_s)\beta)}{\Gamma(\alpha)T_s^2} & \\ +\frac{2\tilde{x}\Gamma(\alpha,\tilde{x}\beta)-(\tilde{x}-T_s)\Gamma(\alpha,(\tilde{x}-T_s)\beta)}{\Gamma(\alpha)T_s^2}, & \textit{if } \tilde{x}\geq T_s, \\ \frac{\Gamma(\alpha+1,(\tilde{x}+T_s)\beta)-2\Gamma(\alpha+1,\tilde{x}\beta)+\Gamma(\alpha+1)}{\Gamma(\alpha)\beta T_s^2} & \\ \quad-\frac{(\tilde{x}+T_s)\Gamma(\alpha,(\tilde{x}+T_s)\beta)}{\Gamma(\alpha)T_s^2} & \\ \quad+\frac{2\tilde{x}\Gamma(\alpha,\tilde{x}\beta)-(\tilde{x}-T_s)\Gamma(\alpha)}{\Gamma(\alpha)T_s^2}, & \textit{if } 0\leq\tilde{x}<T_s, \\ \frac{\Gamma(\alpha+1,(\tilde{x}+T_s)\beta)-\Gamma(\alpha+1)}{\Gamma(\alpha)\beta T_s^2} & \\ \quad-\frac{(\tilde{x}+T_s)[\Gamma(\alpha,(\tilde{x}+T_s)\beta)-\Gamma(\alpha)]}{\Gamma(\alpha)T_s^2}, & \textit{if} -T_s\leq\tilde{x}<0, \\ 0, & \textit{otherwise.}\end{cases} \tag{2.25}$$

Proof: See Appendix A.4. Denote the realization for T_a as $\tilde{x}_i$. We can obtain its PDF, $f_j(\tilde{x}_i|\mathbf{\Theta},T_s)$, under hypothesis $\mathcal{H}_j$ from Theorem 2. We follow the same step in Section 2.3.1 to derive the MLC, where the likelihood function can be written as $L_{\mathcal{H}_j}(\tilde{\mathbf{x}})=\pi_j\prod_{i=1}^{n}f_j(\tilde{x}_i|\mathbf{\Theta},T_s),\forall j\in\mathcal{M}$, similarly to (2.2).

To quantify the probability of correct classification with estimation error, $\tilde{P}_c$, in a closed-form, we apply approximation in the same manner as in Section 2.3.1. First, let us assume that the sampling period T_s is not large, which means that PDF of $\tilde{x}_i$ and x_i would not significantly deviate from each other. We first replace $y_i^{(j,k)}$ with the $\tilde{y}_i^{(j,k)}=\alpha_{j,k}\log(x_i+\phi_i)-\beta_{j,k}(x_i+\phi_i)$ where ϕ_i is a realization for the quantization noise. To be able to apply (2.6) considering sampling noise we need to first find an expectation and variance of $\tilde{y}_i^{(j,k)}$, i.e., $\tilde{\mu}_{j,k}$ and $\tilde{\sigma}_{j,k}^2$, respectively. For $\tilde{\mu}_{j,k}$, since $x_i+\phi_i$ might be a negative value, the mean for $\tilde{y}_i^{(j,k)}$ might be a complex number, which can not be used in the Q function. Therefore we use $\tilde{\mu}_{j,k}=\mu_{j,k}$. On the other hand, the derivation for $\tilde{\sigma}_{j,k}^2$ is given in Appendix A.5, which is always a real number. We can now obtain the average probability of correct classification $\tilde{P}_c$ under estimation noise using (2.8) and (2.9) by replacing $\sigma_{j,k}^2$ with $\tilde{\sigma}_{j,k}^2$. It is thus imperative to emphasize that the proposed calculation method (due to above assumptions) is quite inaccurate considering all parameter combinations and needs to be taken with caution. Therefore calculation of classification performance is still considered to be an open problem. The reader is encouraged to experiment with our analytical procedure of classification based on the accompanying MATLAB code, see Section 2.6.1.

2.4.4 MSPRTC under Period Estimation Error

The proposed MSPRTC under period estimation error follows the same procedure explained in Section 2.3.2. The only adaptation is to replace the PDF $f_j(x|\mathbf{\Theta}_j)$ in (2.10) with the modified PDF $f_j(\tilde{x}|\mathbf{\Theta}_j, T_s)$ derived in (2.25).

2.4.5 A Design Guideline for Traffic Classification using MLC

There are two parameters, i.e., total observation time, T, and total number of samples, N, to be used in classification that need to be optimized. Naturally, we would like to use the smallest T or N to achieve the desired performance for MLC. To derive the performance of correct classification using period estimation $\tilde{P}_c$, we need to obtain the number of periods and the sampling period T_s. Obviously $T_s = \frac{T}{N-1}$. The average number of periods can be derived as

$$\mathbb{E}\{K\} = \sum_{j=1}^{M} \pi_j \mathbb{E}\{K|\mathcal{H}_j\}, \tag{2.26}$$

where

$$\mathbb{E}\{K|\mathcal{H}_j\} = \frac{T}{\mathbb{E}\{T_{\text{on}}|\mathcal{H}_j\}}(1 - R(T_s|\mathcal{H}_j)). \tag{2.27}$$

Here $\mathbb{E}\{K|\mathcal{H}_j\}$ is the average number of periods we can obtain under hypothesis $\mathcal{H}_j$, which is equal to the total average number of periods $\frac{T}{\mathbb{E}\{T_{\text{on}}|\mathcal{H}_j\}}$ times the successful period detection rate $1 - R(T_s|\mathcal{H}_j)$, where R is the mis-detection rate for detecting one period defined as $R(T_s|\mathcal{H}_j) = \Pr\{T_{\text{on}} < T_s|\mathcal{H}_j\} = G(T_s|\mathbf{\Theta}_j)$, where $G(\cdot|\mathbf{\Theta}_j)$ is the CDF function for a gamma distribution under hypothesis $\mathcal{H}_j$. Note that T_s and $\mathbb{E}\{K\}$ are functions of T and N, therefore we know $\tilde{P}_c$ is a function of traffic parameters $\mathbf{\Theta}_j$, $\forall j \in \mathcal{M}$, $\mathbf{\Omega}$, observation time T, and number of traffic samples N. Once the classification performance constraint ϵ is given, we can solve the optimization problem

$$\min \ T(\text{or } N) \text{ subject to } \tilde{P}_c \geq \epsilon \tag{2.28}$$

analytically.

2.5 Traffic Classification with Imperfect Knowledge of PU Traffic Parameters

We further relax the system model assumptions from Section 2.2 and consider the lack of complete information on $\mathbf{\Theta}_j$. Specifically, for the perfectly measured $\mathbf{x}$ we assume that the shape parameters α_j are known, but the rate parameters β_j, $\forall j \in \mathcal{M}$ are not.

First, we consider to treat β_j as unknown deterministic value. In this case we propose the *estimate-then-classify* (ETC) scheme to complete the traffic classification, where we estimate all PU traffic parameters before applying them to the MLC (Section 2.5.1) and MSPRTC (Section 2.5.2). Additionally, for the ETC we derive the classification performance of MLC analytically. Then, if the PU traffic parameters β_j follow a certain distribution, we propose in Section 2.5.3 the *average likelihood function* (ALF) for the classifiers.

2.5.1 Estimate-Then-Classify: Using MLC

The ML estimator of β_j for the distribution $f_j(x|\mathbf{\Theta}_j)$ can be derived by solving $\frac{\partial \prod_{i=1}^{n} f_j(x_i|\mathbf{\Theta}_j)}{\partial \beta_j} = 0$ which gives

$$\hat{\beta}_j = \alpha_j n \left(\sum_{i=1}^{n} x_i \right)^{-1}. \tag{2.29}$$

Considering MLC, the ETC scheme is based on replacing β_j with its estimate $\hat{\beta}_j$ in the PDF of x as $f_j(x|\hat{\mathbf{\Theta}}_j)$ where $\hat{\mathbf{\Theta}}_j = (\alpha_j, \hat{\beta}_j)$ and subsequently to the likelihood function defined in (2.2).

To analyze the classification performance for the proposed ETC-based MLC, we can simply use (2.9) except β_j is replaced by the corresponding mean of the estimator $\mathbb{E}\{\hat{\beta}_j\}$. To be more specific, under hypothesis $\mathcal{H}_j$, the mean is expressed as $\mathbb{E}\{\hat{\beta}_k^{-1}|\mathcal{H}_j\} = \frac{1}{n\alpha_k}\sum_{i=1}^{n}\mathbb{E}\{x_i|\mathcal{H}_j\} = \frac{\alpha_j}{\alpha_k}\beta_j^{-1}$, and the variance is expressed as $\mathrm{Var}\{\hat{\beta}_k^{-1}|\mathcal{H}_j\} = \frac{\alpha_j\beta_j^{-2}}{n\alpha_k^2}$, $\forall k \in \mathcal{M}$. Therefore as n approaches infinity, the variance for $\hat{\beta}_k^{-1}$ approaches zero, which means that $\hat{\beta}_k^{-1}$ converges to $\frac{\alpha_j}{\alpha_k}\beta_j^{-1}$ asymptotically. To conclude, we replace β_k with $\mathbb{E}\{\hat{\beta}_k\}$ and embed it into (2.9), and the probability of correct classification using ETC-based MLC can be represented as

$$\boxed{\begin{aligned}\hat{P}_c = &\sum_{j=1}^{M}\pi_j\prod_{k=1\ k\neq j}^{M}Q\left(-\frac{1}{\sqrt{n\hat{\sigma}_{j,k}^2}}\right.\\ &\left.\times\left(\log\frac{\pi_j\alpha_j^{n\alpha_k}\beta_j^{n(\alpha_j-\alpha_k)}\Gamma(\alpha_k)^n}{\pi_k\alpha_k^{n\alpha_k}\Gamma(\alpha_j)^n}+n\hat{\mu}_{j,k}\right)\right),\end{aligned}} \tag{2.30}$$

where $\hat{\mu}_{j,k}$ and $\hat{\sigma}_{j,k}^2$ are the mean and variance of $\hat{y}_i^{(j,k)} = \alpha_{j,k}\log x_i - \left(\frac{\alpha_j-\alpha_k}{\alpha_j}\right)\beta_j x_i$, respectively, which can also be derived analytically using the scheme given in Appendix A.1.

2.5.2 Estimate-then-Classify: Using MSPRTC

For the MSPRTC, we need to update the estimated posterior probabilities after collecting each new PU traffic period if all the estimated posterior probabilities defined as

$$\hat{p}_n^j \triangleq \pi_j\prod_{i=1}^{n}f_j(x_i|\hat{\mathbf{\Theta}}_j)\left[\sum_{l=1}^{M}\pi_l\left(\prod_{i=1}^{n}f_l(x_i|\hat{\mathbf{\Theta}}_l)\right)\right]^{-1}, \tag{2.31}$$

are less than or equal to the threshold. The complete algorithm for ETC-based MSPRTC is listed in Algorithm 1.

2.5.3 Average Likelihood Function: Traffic Classification with Prior Knowledge on Distribution of PU Traffic Parameters

We now consider a case when the PU traffic parameters β_j are no longer constants, but instead follow a certain distribution. When the distribution of the PU traffic parameter is known, such knowledge can be exploited by averaging the conditional likelihood function with respect to the distribution of the PU traffic parameter, which can better describe the

Algorithm 1 ETC-based MSPRTC

procedure CLASSIFIER($\mathbf{x}, f_j(x|\mathbf{\Theta}_j), M, \mathbf{\Omega}, \gamma$)

  $i \leftarrow 1$

  $\mathbf{x} \leftarrow x_1$

  Calculate $\hat{\mathbf{\Theta}}_j$ using (2.29), $\forall j \in \{1, 2, \cdots, M\}$

  Calculate estimated posteriori probability $\hat{p}_i^j$ using (2.31)

  while $\hat{p}_i^j \leq \frac{1}{1+\gamma} \forall j \in \{1, 2, \cdots, M\}$ **do**

    $i \leftarrow i + 1$

    $\mathbf{x} \leftarrow (x_1, x_2, \cdots, x_i)^T$

    Calculate $\hat{\mathbf{\Theta}}_j$ using (2.29), $\forall j \in \{1, 2, \cdots, M\}$

    Calculate estimated posteriori probability $\hat{p}_i^j$ using (2.31)

  end while

  $N_A \leftarrow i$ ▷ Stopping time

  $m \leftarrow \arg\max_j \hat{p}_{N_A}^j$

  $\nu \leftarrow \mathcal{H}_m$ ▷ Final decision

end procedure

behavior for each hypothesis. The proposed ALF under $\mathcal{H}_j$ is defined as

$$h_j(x) \triangleq \int_{-\infty}^{\infty} f_j(x|\mathbf{\Theta}_j) q_j(\beta_j) d\beta_j, \tag{2.32}$$

where $q_j(\beta_j)$ is the PDF for β_j. Hence the likelihood function in (2.2) for MLC and the posterior probability in (2.10) for MSPRTC are modified by replacing likelihood function $f_j(x|\mathbf{\Theta}_j)$ with ALF $h_j(x)$. As an example, assuming $\beta_j \sim \mathcal{U}(L_j, U_j)$ then (2.32) can be derived using (A.12) as

$$\begin{aligned} h_j(x) &= \int_{L_j}^{U_j} f_j(x|\mathbf{\Theta}_j) q_j(\beta_j) d\beta_j \\ &= \frac{x^{\alpha_j - 1}}{(U_j - L_j)\Gamma(\alpha_j)} \int_{L_j}^{U_j} \beta_j^{\alpha_j} e^{-\beta_j x} d\beta_j \\ &= \frac{\Gamma(\alpha_j + 1, L_j x) - \Gamma(\alpha_j + 1, U_j x)}{(U_j - L_j)\Gamma(\alpha_j) x^2}. \end{aligned} \tag{2.33}$$

Note that the average SH distance with ALF can be calculated by using (2.33) to replace $f_j(x|\mathbf{\Theta}_j)$ in (A.11). Also note that for the average SH distance with ALF we were unable to find a closed-form expression and it can only be computed through numerical integration.

2.6 Numerical Results

We now present MATLAB-based numerical results for the performance of the proposed PU traffic classification algorithms. We assume $M = 3$, as in [BV94, Sec. VIII] in which two distributions are considered as special cases, that is where: (i) $\alpha_1 = 1$, i.e., exponential distribution, and (ii) $\alpha_2 = 2$, i.e., Erlang distribution. Furthermore, we design two test scenarios for the classifiers, i.e., Test I and Test II, with a relatively large and small average distribution distance among hypotheses, respectively. The average distance among hypotheses is evaluated through average SH distance, H^2, which is calculated in Appendix A.2. The PU traffic parameters for Test I and Test II are summarized in Table 2.1 for the PU traffic with stable parameters and in Table 2.2 for PU traffic with fluctuating parameters, respectively. The unit for β_i is second^{-1}. We assume that each hypothesis has the same prior probability, i.e., $\pi_j = \frac{1}{M}$, i.e. a maximum entropy case. Observe that for Test II, all hypotheses have the same first moment in order to have a small average distance among hypotheses, which is different from Test I. In our simulations, PU traffic periods are generated randomly from three distributions in one realization. In case of PU sampled process we generate it by adding two uniformly distributed random variables at the beginning and the end of PU traffic process, following strictly the simplifying assumption from Section 2.4.1. Each simulation point is obtained by method of batch means (unless otherwise stated) averaging 50 classification runs, each having at least 2000 realizations for a confidence interval of 0.1.

2.6.1 Results Reproducibility and Open Code Access

In addition, for the reproducibility of results, the source code used in generating all figures is (i) available upon request or (ii) via http://cores.ee.ucla.edu/images/d/d4/Traffic-classification-source-code-may-2014.zip. The code allows the reader to generate results for a desired set of variables and experiment with the implementation and the accuracy of the developed classifiers. Any future corrections and updates to the source code and the paper will be also available therein.

Table 2.1: Traffic Parameters (Stable)

Function	Parameters (Test I)	Parameters (Test II)
exponential	$\alpha_1 = 1,\ \beta_1 = 0.4$	$\alpha_1 = 1,\ \beta_1 = 0.4$
Erlang	$\alpha_2 = 2,\ \beta_2 = 0.3$	$\alpha_2 = 2,\ \beta_2 = 0.8$
gamma	$\alpha_3 = 0.8,\ \beta_3 = 0.5$	$\alpha_3 = 0.5,\ \beta_3 = 0.2$
Average H^2	0.1799	0.0695

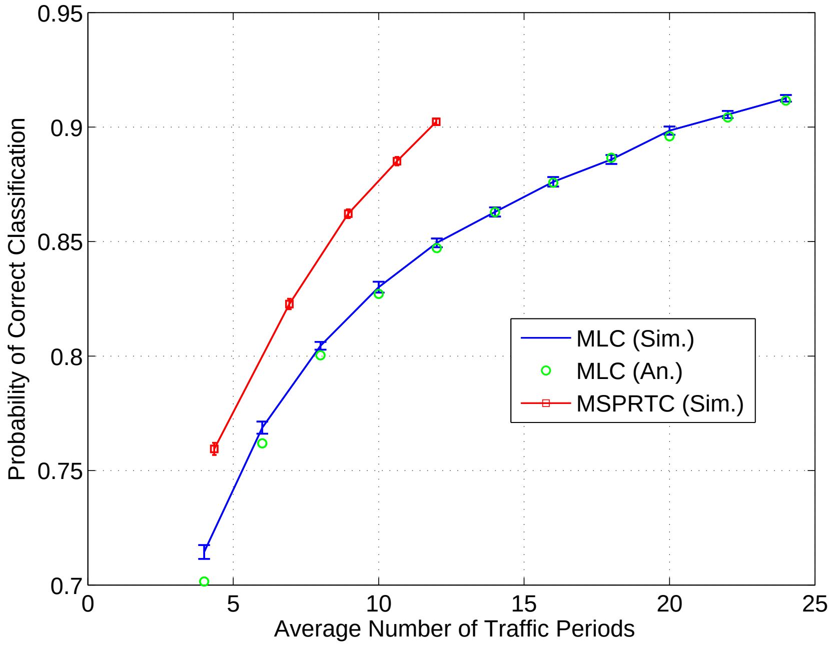
Table 2.2: Traffic Parameters (Fluctuating)

Function	Parameters (Test I)	Parameters (Test II)
exponential	$\alpha_1 = 1, \beta_1 \sim \mathcal{U}(0.4, 0.9)$	$\alpha_1 = 1, \beta_1 \sim \mathcal{U}(0.4, 0.9)$
Erlang	$\alpha_2 = 2,\ \beta_2 \sim \mathcal{U}(0.1, 0.3)$	$\alpha_2 = 2,\ \beta_2 \sim \mathcal{U}(1.2, 1.4)$
gamma	$\alpha_3 = 0.2,\ \beta_3 \sim \mathcal{U}(0.2, 0.5)$	$\alpha_3 = 3,\ \beta_3 \sim \mathcal{U}(1.1, 2.8)$
Average H^2	0.4482	0.0379

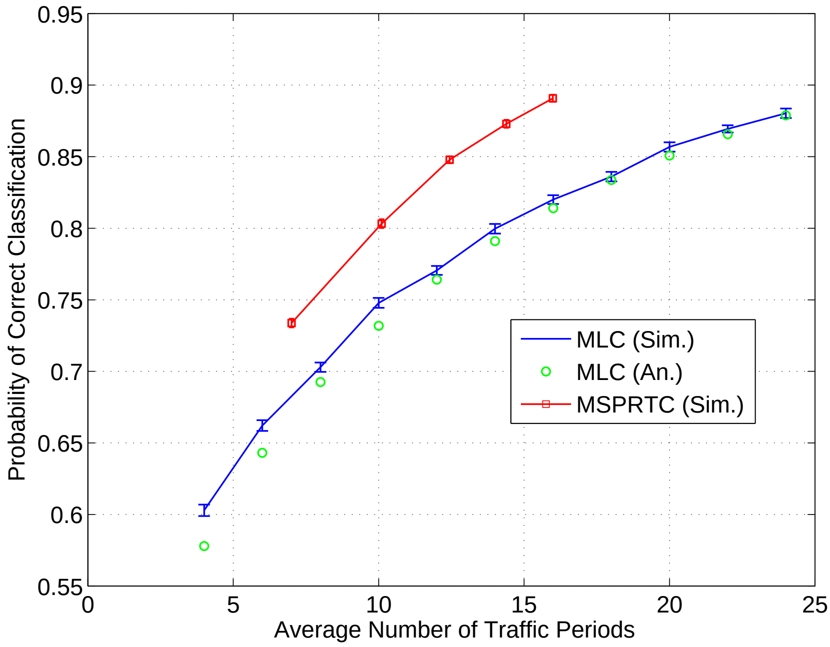
2.6.2 Traffic Classification Performance with Perfectly Sampled PU Traffic Periods and Parameters

In Fig. 2.1 we present the classification performance under perfect knowledge of PU traffic parameters and perfect sampling of traffic ON/OFF periods as a function of traffic periods n. First, we observe that under both tests the simulated MLC performance matches our derived analytical performance. Second, the MSPRTC performs better than MLC since it can achieve the same P_c using less number of PU traffic periods. Finally, our results prove the intuitive observation that for a smaller average distance among hypotheses, shown in Fig. 2.1(b), a higher number of PU traffic periods is needed to classify the correct hypotheses[3].

[3]Note that in the MATLAB implementation we are constrained by the numerical precision of 32 bit unsigned integers (due to frequent exponentiations of very small numbers) thus the analytical results are not realizable for large values of n. Also, note that while plotting the analytical results for the MLC classifier, we have used a simulation to generate statistics for mean and variance for $y_i^{(j,k)}$ and $\tilde{y}_i^{(j,k)}$, to speed up figure generation. More details are provided in the code accompanying this paper.



(a) Test I



(b) Test II

Figure 2.1: Probability of correct classification with the average number of PU traffic periods, under perfect knowledge of PU traffic periods and parameters. MLC is compared with MSPRTC. PU traffic parameters used in simulations are presented in Table 2.1. Simulation results (Sim.) are plotted to verify analytical results (An.).

2.6.3 Traffic Classification Performance with PU Traffic Period Estimation and Perfect Knowledge of Parameters

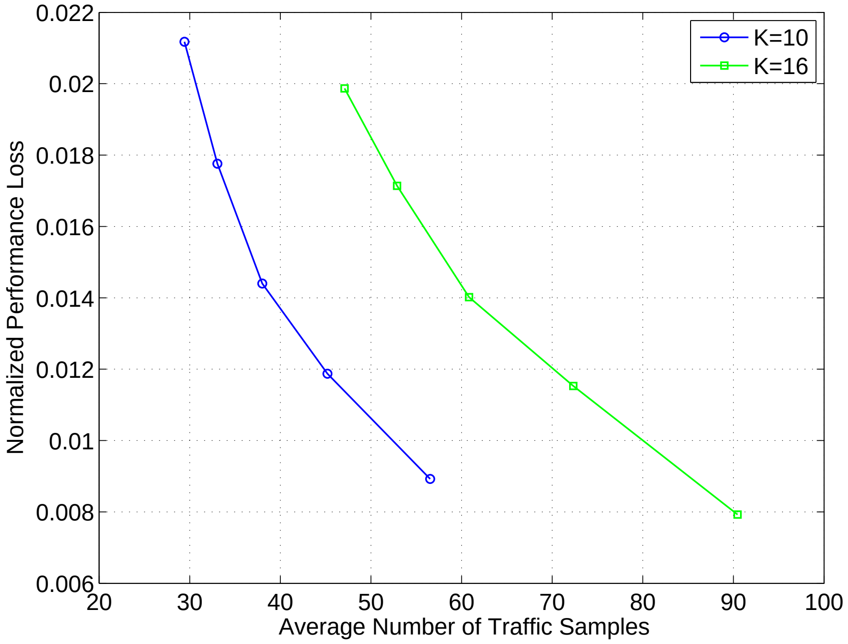
Fig. 2.2 shows the normalized performance loss $L \triangleq \frac{P_c - \tilde{P}_c}{P_c}$ for MLC with the average number of traffic samples[4] $\mathbb{E}\{N\}$, which are both functions of T_s. We consider two cases of PU traffic periods: (i) $K = 10$ and (ii) $K = 16$. First, as the average number of PU traffic samplings increases, which means we adopt a small sampling period T_s, L decreases. This is because we have higher resolution for sampling to estimate the PU traffic periods, thus resulting in a more accurate classification. Second, we observe that the performance for higher number of PU traffic periods is more sensitive to the PU traffic period estimation error. Therefore more PU traffic samples for higher number of PU traffic periods are needed to achieve the same performance as with a lower number of PU traffic periods. Finally, we show that for a small average distance among hypotheses, the performance loss is large since in this case the PU traffic classification is more sensitive to the period estimation errors.

In Fig. 2.3 we compare MLC and MSPRTC under sampling. First, as the sampling period increases, the performance of both classifiers decreases. Naturally, a longer sampling period will result in a larger estimation noise. Second, we observe that MSPRTC is as sensitive as MLC to the period estimation error. This is because both MSPRTC and MLC adopt the same likelihood function for classification, which requires accurate knowledge of the true distributions. If the noise is added into the observation, it will distort the original PDF even worse when the distance among hypotheses is small.

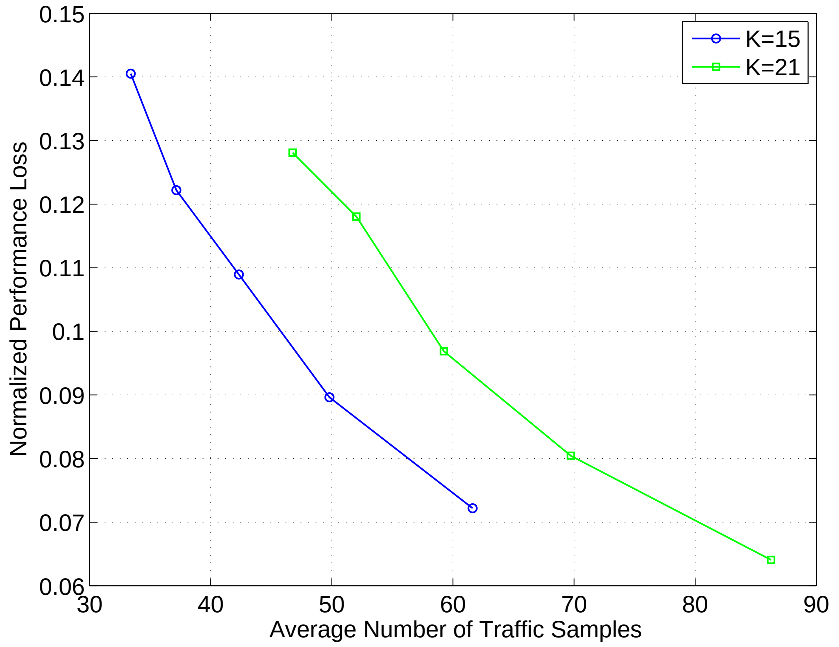
2.6.4 A Design Guideline Example for Traffic Classification using ML Classifier

We provide two examples for the design guideline shown in Section 2.4.5. First we consider the case where, given observation time T, we need to design the number of traffic samples and therefore sampling period T_s, to achieve a certain probability of correct classification. In Fig. 2.4(a) we observe that as the number of traffic samples increases the classification performance improves. This is because as the number of traffic samples increases, the period

[4]In this case we do not plot the confidence intervals as we plot the difference between the two means.

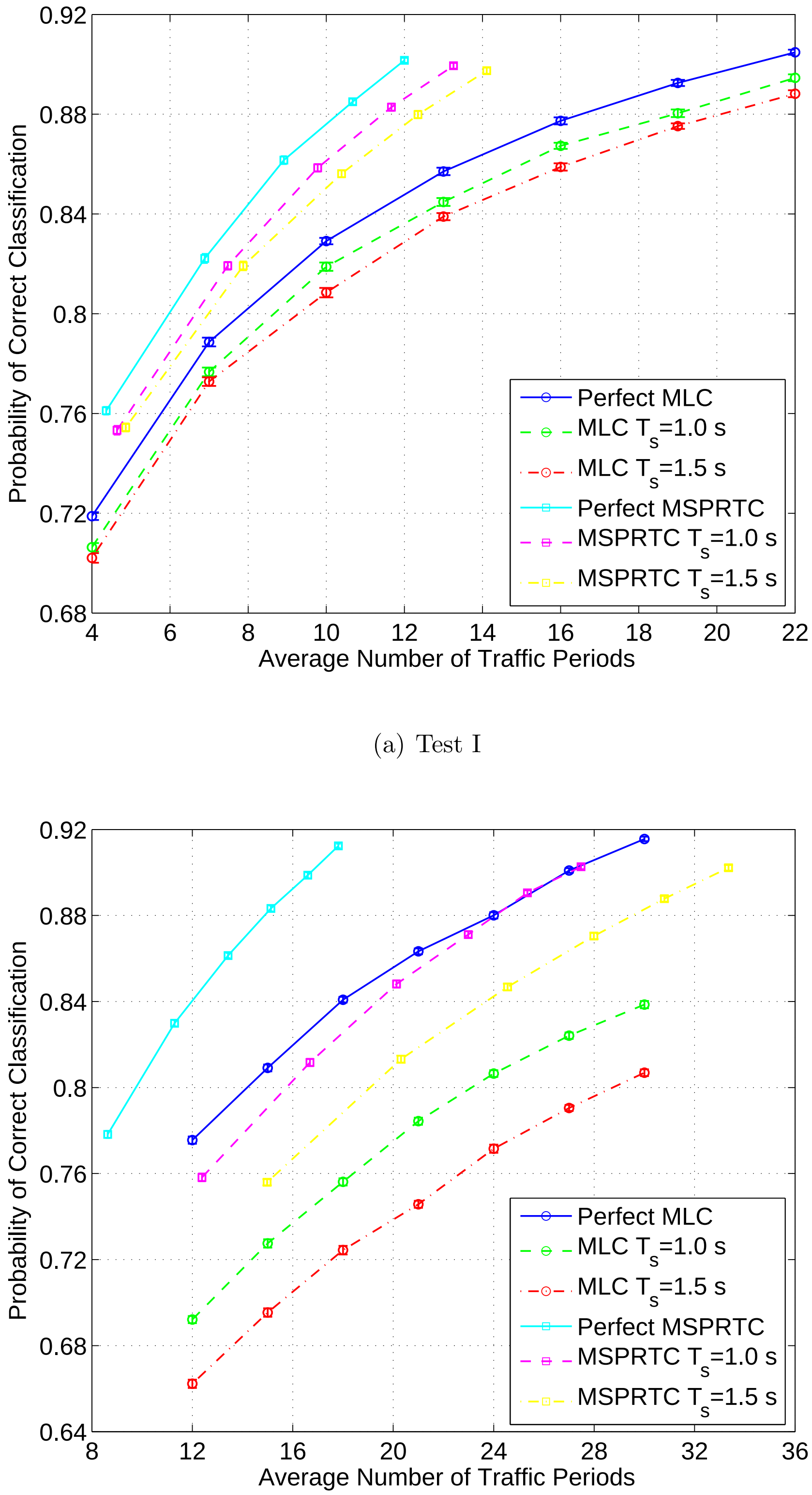


(a) Test I



(b) Test II

Figure 2.2: Normalized classification performance loss with average number of traffic samples using the minimum variance estimator for the MLC under perfect knowledge of PU traffic parameters. The PU traffic parameters used are given in Table 2.1. All results were obtained by simulations.



(a) Test I

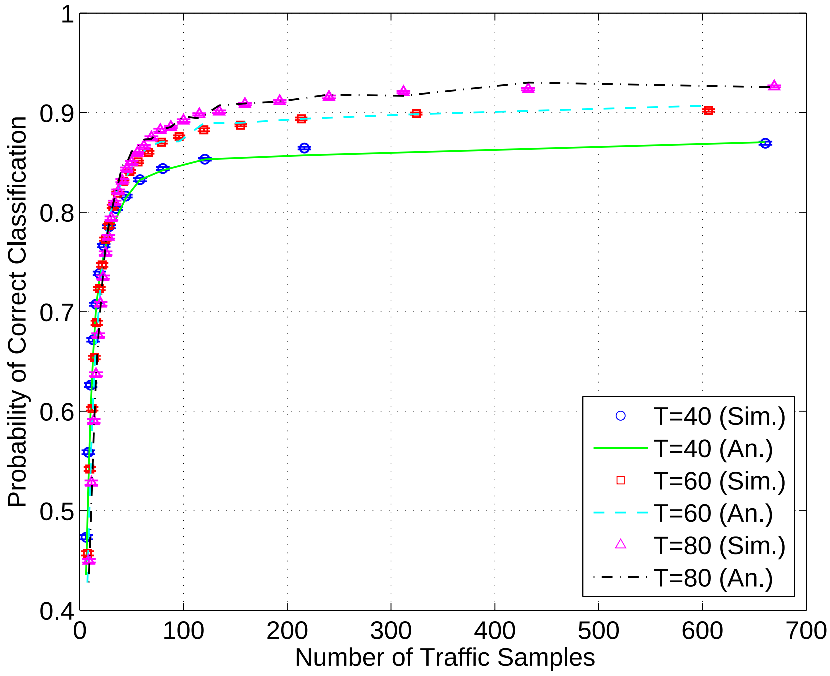
(b) Test II

Figure 2.3: Probability of correct classification with the average number of PU traffic periods using the minimum variance estimator, under perfect knowledge of PU traffic parameters. MLC is compared with MSPRTC. The PU traffic parameters used are given in Table 2.1. All results were obtained by simulations.

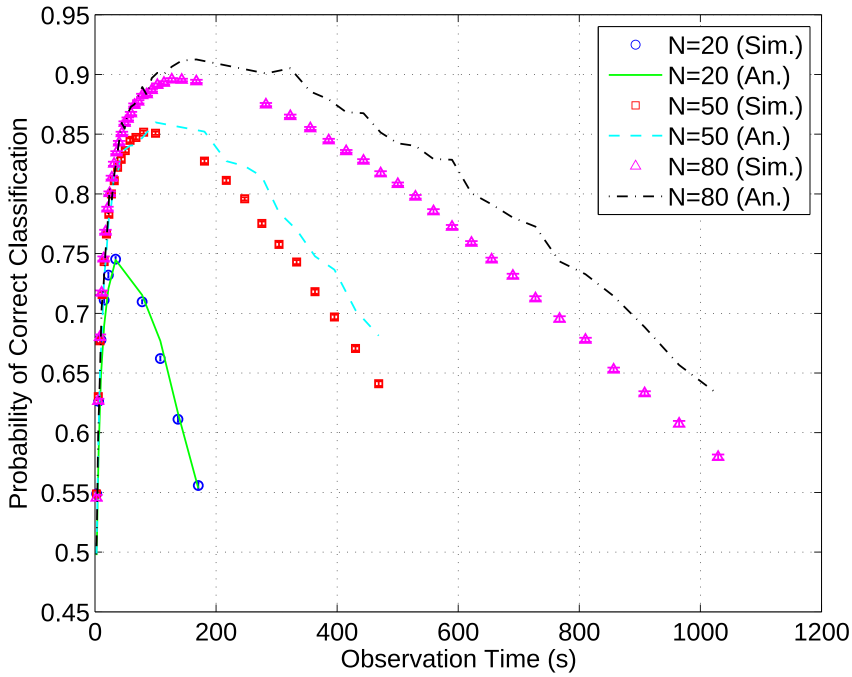
estimation errors decrease, and at the same time, we can obtain more PU traffic periods as the PU traffic period mis-detection rate decreases which is shown in (2.26). Furthermore, as the observation time increases, the classification performance also increase. Although in this case the estimation error increases, the obtained traffic periods increases. This is because the latter factor has more influence on the classification performance. In this traffic scenario, for example, given the timing constraint $T = 60$ seconds we need at least $N = 350$ traffic samples to achieve the performance $\epsilon = 0.90$. This means the constraint for the sampling rate T_s to sample this traffic should be no less than $\frac{60}{350-1} = 0.1719$ seconds to achieve the classification performance of $\epsilon = 0.90$.

Second we consider the case where, given the number of samples, we need to design the observation time to achieve a certain classification performance. From Fig. 2.4(b), the performance is a concave curve with respect to the observation time. This can be explained by the behavior of (2.26). In (2.26), $\mathbb{E}\{K\}$ versus T has a similar shape as $\tilde{P}_c$ versus T. However, to figure out the classification performance, not only $\mathbb{E}\{K\}$ but also the sampling period T_s needs to be considered to determine the classification performance. Initially, as T increases, $\mathbb{E}\{K\}$ increases, and T_s increases. Since the effect of $\mathbb{E}\{K\}$ is more significant, the classification performance increases. As T increases through the maximum point of $\mathbb{E}\{K\}$, $\mathbb{E}\{K\}$ starts to decrease. In this region T_s also increases. Therefore the performance will decrease since we obtain less traffic periods with higher estimation errors. In this traffic scenario, for example, given the energy constraint $N = 50$, we can solve for the optimal observation time $T = 100$ seconds to achieve the maximal performance $\epsilon = 0.86$. This means the optimal sampling rate T_s to sample this traffic should be set as $\frac{100}{50-1} = 2.04$ seconds to achieve $\epsilon = 0.86$. Larger and smaller T_s than the optimal T_s will both degrade the classification performance.

Finally, we see that our proposed analytical approximation matches the simulation results for small values of T_s. But as T_s increases, shown in Fig. 2.4(b), the analytical results start to deviate from the simulation results, refer again to Section 2.4.3.



(a) Sensing constraint on observation time



(b) Sensing constraint on number of traffic samples

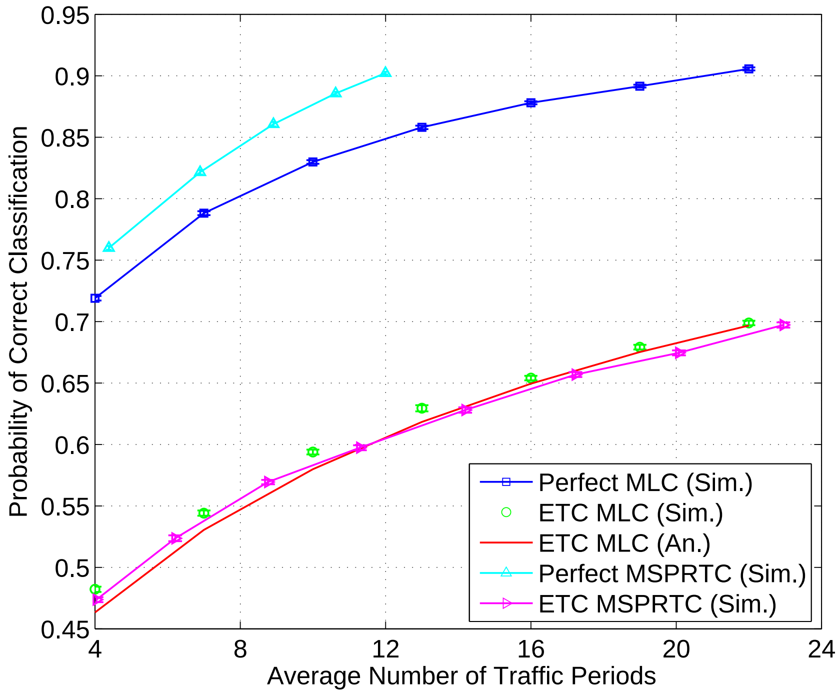
Figure 2.4: Probability of correct classification with number of traffic samples in Fig. 2.4(a) and observation time in Fig. 2.4(b) with perfect knowledge of traffic parameters using MLC. The PU traffic parameters used are shown as Test I in Table 2.1. Simulation results (Sim.) are plotted to verify analytical results (An.).

2.6.5 Traffic Classification with Perfect PU Periods and No Knowledge of Parameters

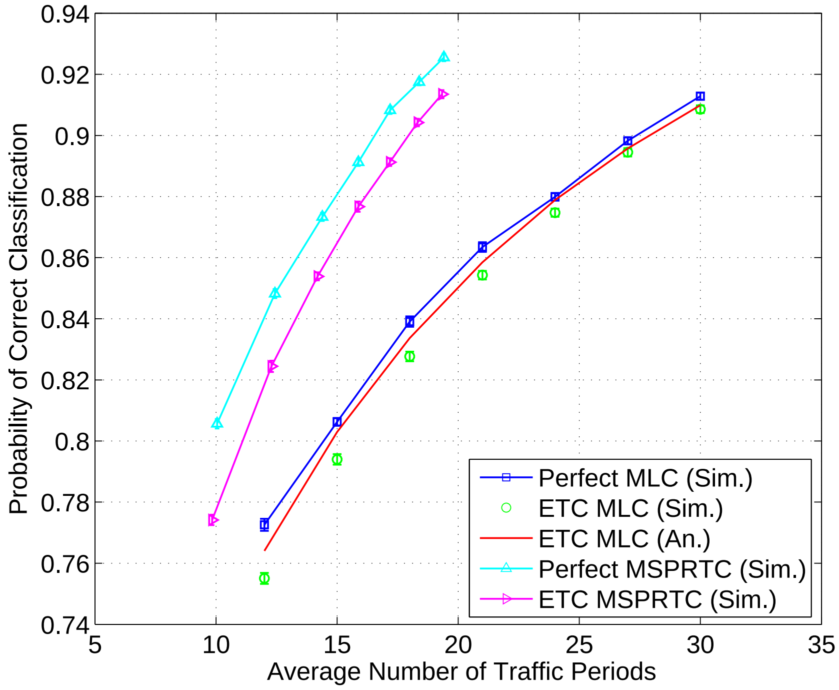
Fig. 2.5 presents the probability of correct classification with the average number of PU traffic periods assuming no knowledge of PU traffic parameters β_j. We compare MLC and MSPRTC with perfect knowledge of PU traffic parameters and the ETC method with no knowledge of traffic parameters β_j. First, we note that ETC-based method performs worse than methods using perfect parameters. Second, the ETC-based MSPRTC outperforms MLC as the distance among hypotheses is small, otherwise they perform similarly. Third, the simulation results for ETC-based MLC matches our proposed analytical results in (2.30), since the number of PU traffic periods is large enough for parameter estimation. Finally, we can observe that ETC-based method will perform worse under Test I than Test II, compared with the perfect classifiers. This is because in Test II the first moments for all hypotheses are set to be the same, hence the estimated parameters will be close to the true parameters for all hypotheses. But this is not the case for Test I since the first moments are more different for all hypotheses—which means a small parameter estimation error will cause a large classification performance degradation.

2.6.6 Traffic Classification Performance with Perfect PU Traffic Periods and Prior Knowledge of Traffic Parameters

In Fig. 2.6 we present the classification performance comparisons assuming prior knowledge about the distribution of PU traffic parameters β_j. We note that ALF-based classifiers are better than ETC-based classifiers under Test I, and the result is opposite under Test II. This is because of the fact that ALF can capture most PU traffic parameter information if the distance among hypotheses is large, i.e., the Test I case. If the distance among hypotheses is small, as in Test II, ETC-based method provides a more accurate PU traffic parameter estimation.

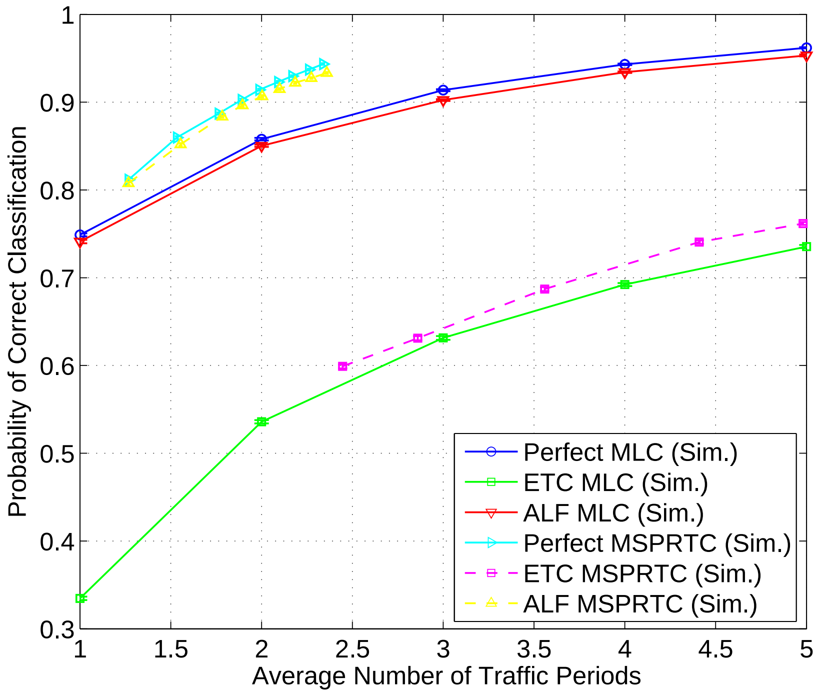


(a) Test I

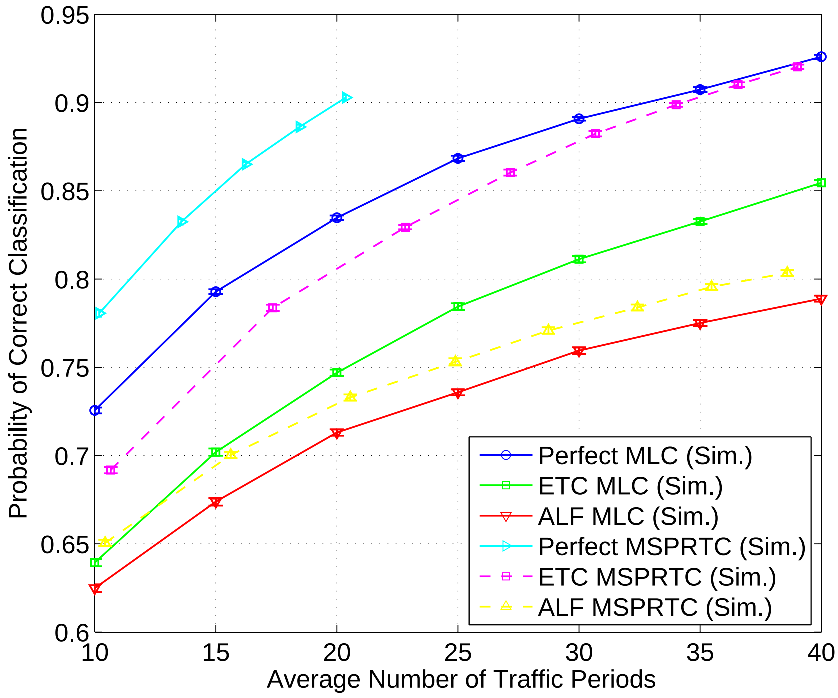


(b) Test II

Figure 2.5: Probability of correct classification with the average number of PU traffic periods, under no knowledge of PU traffic parameters β_j. MLC is compared with MSPRTC using ETC scheme. The PU traffic parameters used are shown in Table 2.1. Simulation results (Sim.) are plotted to verify analytical results (An.).



(a) Test I



(b) Test II

Figure 2.6: Probability of correct classification with the average number of PU traffic periods, with prior knowledge of PU traffic parameters β_j. MLC is compared with MSPRTC using ETC and ALF schemes. The PU traffic parameters used are shown in Table 2.2.

2.7 Summary

We proposed novel primary user (PU) traffic classification algorithms which were based on the maximum likelihood function and multi-hypothesis sequential probability ratio test classifiers, and we considered cases where the PU traffic periods and PU traffic parameters need to be estimated. In addition, we analyzed a sampling technique to estimate PU traffic periods, and a minimum variance period estimator was derived to design a traffic classifier given sensing constraints such as the number of traffic samples or observation time. Furthermore, we proposed two classifiers, estimate-then-classify (ETC) and average likelihood function (ALF) classifiers to handle the cases when there was only no/partial knowledge of PU traffic parameters.

To conclude, for PU traffic with constant and known parameters, MSPRTC, a more complicated classifier than MLC is recommended in terms of classification performance both with and without period estimation. For PU traffic with prior knowledge of parameters, the ALF-based classifier is suitable for traffic classification when the average distance among hypotheses is large. If the average distance among hypotheses is small, the ETC-based classifier is preferred to provide a good classification performance.

CHAPTER 3

Prediction of Primary User Traffic for Dynamic Spectrum Access

3.1 Introduction

In Chapter 2, we discussed how we can classify PU traffic into certain distribution. In this chapter, we assume we can accurately classify PU traffic distribution so that we can exploit the traffic distribution to predict PU states. Specifically, we propose a prediction algorithm for PUs assuming perfect sensing with exponentially and Erlang-2 distributed traffic, which are common distributions for wireless/wired traffic.

3.1.1 Related Work

The most notable work that considers PU traffic prediction can be found in [TPC11, YCZ10, LJX08, SCZ10a]. In [TPC11], an energy-minimization framework is presented based on predicting the PU traffic. The authors assume perfect prediction in order to reduce both transmission and switching energy. Achievable throughput optimization based on the correlation of different channels has been discussed in [YCZ10] with a single SU and multiple PUs in a slotted-transmission mode using spectrum usage prediction. The authors in [LJX08] adopt multi-carrier techniques to flexibly allocate channel resources over a wide bandwidth. The prediction is performed across a number of frequency bands where subcarrier allocation is achieved considering collision minimization. A Markov-based prediction method has been proposed in [SCZ10a], which is a form of the universal predictor, that allows SUs to sense the channels with most available probability.

3.1.2 Our Contribution

The contribution of this work is threefold:

1. Prediction Algorithm: In [GLP13b], we study the accuracy of PU traffic estimation in terms of the mean squared error of the traffic parameter estimates. Here, we propose an algorithm that predicts the PU traffic based on the PU traffic parameter estimates in contrary to prior works that assume perfect knowledge of traffic parameters;

2. Performance of Estimators: Our prediction algorithm does not assume any a priori knowledge of traffic parameters. Therefore, we propose maximum likelihood (ML) estimators to estimate traffic parameters. Then a general estimation performance bound is derived for any sequence obeying a Markov chain;

3. Analysis of Prediction Confidence: We further define a number of regions which are functions of the PU traffic parameters and for each region we derive conditions required to achieve minimum prediction error rate. Moreover, we quantify the prediction confidence for different regions based on the accuracy of the estimated traffic parameters. The confidence serves as a guideline for determining the required estimation accuracy needed to meet a given prediction error rate constraint.

3.2 System Model for Traffic Prediction

This section describes the models used for traffic estimation and prediction. We consider a single channel randomly accessed by a single PU, where the traffic of the PU is assumed to be stationary. Moreover, the PU ON/OFF intervals are assumed to be exponentially distributed [WMB08a, LC11, KS08a] or Erlang-2 distributed [SS14, MM14], where the probability density function of the interval, t, is given by

$$f_\lambda(t) = \begin{cases} \lambda e^{-\lambda t} & \text{for } t \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.1}$$

for the exponential traffic, and

$$f_\lambda(t) = \begin{cases} \lambda^2 t e^{-\lambda t} & \text{for } t \geq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{3.2}$$

for the Erlang-2 traffic, with $\lambda = \lambda_n$ and $\lambda = \lambda_f$ for the ON and OFF periods, respectively. As a property of these distributions, the ON and OFF parameters are λ_n and λ_f, respectively. The duty cycle u is defined by $u = \frac{\lambda_f}{\lambda_f + \lambda_n}$, indicating the fraction of time with the presence of PU. This duty cycle also represents the channel utilization, which is the key parameter for traffic estimation.

The PU traffic utilization is modeled as a semi-Markov process s_i [KS08a], where s_i represents the state of the PU at any time instant T_i. $s_i = 0$ and 1 indicate the PU is absent and present, respectively. The traffic is described by the stationary distribution and the transition probabilities. The former indicates the probabilities of the PU being absent and present at time T_i, with $\Pr\{s_i = 0\} = P_0 = 1 - u$ and $\Pr\{s_i = 1\} = P_1 = u$. The later states the probability of switching from state x to y during arbitrary time interval t. The authors in [KS08a] have derived the Laplace transform for transition probabilities of a two-state continuous-time Markov chain (CTMC) with ON/OFF length obeying arbitrary traffic distributions. The probabilities of the four possible transitions are given by

$$P_{00}(t) = 1 - u(1 - E), \tag{3.3}$$

$$P_{01}(t) = u(1 - E), \tag{3.4}$$

$$P_{10}(t) = (1 - u)(1 - E), \tag{3.5}$$

$$P_{11}(t) = u(1 - E) + E, \tag{3.6}$$

where $E = e^{-(\lambda_f + \lambda_n)t}$ for the exponential traffic [KS08a] and

$$E = \frac{e^{-\frac{(\lambda_f+\lambda_n)t}{2}}(\lambda_f + \lambda_n)^2[\cosh(\psi t) - \frac{\sinh(\psi t)}{\psi}(\frac{\lambda_f+\lambda_n}{2} - \frac{2\lambda_f\lambda_n}{\lambda_f+\lambda_n})]}{4\lambda_f\lambda_n} - \frac{e^{-(\lambda_f+\lambda_n)t}(\lambda_f - \lambda_n)^2}{4\lambda_f\lambda_n},$$

where $\psi = \sqrt{\frac{\lambda_f^2}{4} - \frac{3\lambda_f\lambda_n}{2} + \frac{\lambda_n^2}{4}}$ for the Erlang-2 traffic.

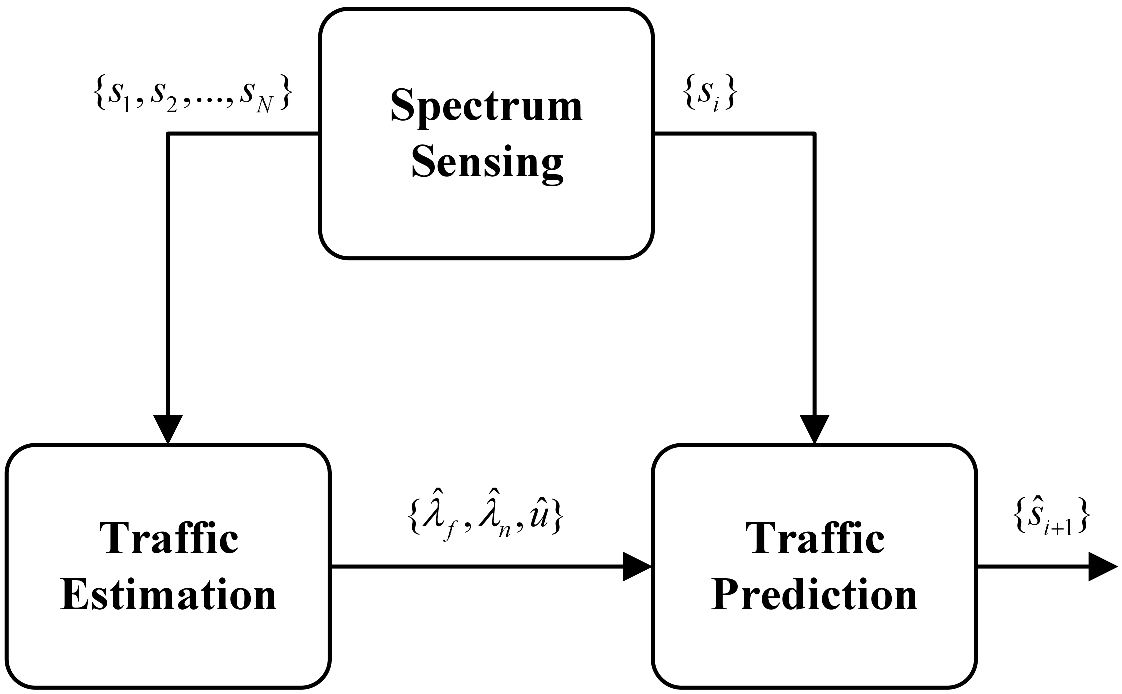


Figure 3.1: The block diagram for traffic estimation-based prediction algorithm.

3.3 Traffic Prediction

The proposed estimation and prediction algorithm is presented in Fig. 3.1 where at first, the traffic parameters are estimated using the results of spectrum sensing measured at N time instances. Spectrum sensing is assumed to be perfect. Then, the proposed algorithm predicts the future PU state $\hat{s}_{i+1}$ based on the estimated parameters $\{\hat{\lambda}_f, \hat{\lambda}_n, \hat{u}\}$ and the current PU state s_i. We analyze the impact of estimation errors on the prediction error rate, and define prediction regions which quantify the prediction confidence. The estimation and prediction procedures are elaborated in the following sections.

3.3.1 Traffic Estimation

Channel utilization as well as average PU ON/OFF interval durations need to be estimated for traffic prediction. Sampling strategies and estimation schemes determine the accuracy bounds of the traffic parameter estimates. It was shown in [GLP13b] that using uniform sampling achieves comparable estimation accuracy compared to non-uniform sampling. Besides, uniform sampling is easier to implement. Hence, in our estimation procedure, given the duty cycle u, we implement uniform sampling with a constant sampling time T_c. Note that a CTMC with uniform sampling now becomes a discrete-time Markov chain (DTMC). Then, we can write the likelihood function given N observed traffic samples as

$$L(\mathbf{s}|\Theta) = P_1^{s_1} P_0^{1-s_1} P_{00}^{n_0}(T_c) P_{01}^{n_1}(T_c) P_{10}^{n_2}(T_c) P_{11}^{n_3}(T_c), \tag{3.7}$$

where n_0, n_1, n_2,n_3 represents the number of PU state transitions of $(0 \to 0)$, $(0 \to 1)$, $(1 \to 0)$, and $(1 \to 1)$, respectively. The ML estimator of Θ is obtained by computing $\hat{\Theta} = \max_{\Theta} L(\mathbf{s}|\Theta)$. Note that Θ can be either a scalar or a vector. Assuming we have perfect knowledge of u, we can focus on estimating one parameter, for example, $\Theta = \lambda_f$. Otherwise, if we do not know all parameters, i.e., the blind case, we need to estimate $\Theta = (\lambda_f, \lambda_n)^T$ jointly. The maximum-likelihood (ML) estimates of λ_f and λ_n for an exponential traffic assuming we know u are given by

$$\hat{\lambda}_f = -(u/T_c) \log[(-\beta + \sqrt{\beta^2 - 4\alpha\gamma})/(2\alpha)], \tag{3.8}$$

$$\hat{\lambda}_n = (1-u)\hat{\lambda}_f/u, \tag{3.9}$$

where $T_c = \frac{T}{N-1}$, $\alpha = (u-u^2)(N-1)$, N is the overall number of samples used for estimation, T is the observation window time, and $\beta = -2\alpha + N - 1 - (1-u)n_0 - un_3$, $\gamma = \alpha - un_0 - (1-u)n_3$. Unfortunately, there is no closed-form solution for estimators of Erlang-2 traffic, and they can only be found numerically.

An important analysis for the ML estimator is its performance bound, i.e., the Cramér-Rao (CR) bound. It is the lower bound for the variance of the estimator and when N is large enough, the variance will approach the CR bound asymptotically. We derive CR bounds for traffic parameters applicable to the traffic states following a two-state DTMC. This result is more general than the bound derived in [GPL13], which is applicable only to the exponentially distributed traffic. First, we derive the Fisher information matrix of $\Theta = (\lambda_f, \lambda_n)^T$, and we derive their CR bounds correspondingly. The Fisher information matrix with N traffic samples is defined as $I_N(\Theta) = \mathbb{E}\{\Omega_N^2\}$, where $\mathbb{E}\{\cdot\}$ is the expectation operation, and $\Omega_N = \frac{\log \partial L(\mathbf{s}|\Theta)}{\partial \Theta}$. The following theorem states the closed form results for the Fisher information matrix omitting T_c from transition probabilities for sake of readability.

Theorem 3 *Given any traffic sequence with length N following a two-state discrete-time Markov chain, the Fisher information for traffic parameters $(\lambda_f, \lambda_n)^T$ can be represented as*

$$
\begin{aligned}
I_N(\Theta)[1,1] &= \frac{1}{P_0}(\frac{\partial P_0}{\partial \lambda_f})^2 + \frac{1}{P_1}(\frac{\partial P_1}{\partial \lambda_f})^2 + (N-1)(\frac{P_0}{P_{00}}(\frac{\partial P_{00}}{\partial \lambda_f})^2 \\
&+ \frac{P_0}{P_{01}}(\frac{\partial P_{01}}{\partial \lambda_f})^2 + \frac{P_1}{P_{10}}(\frac{\partial P_{10}}{\partial \lambda_f})^2 + \frac{P_1}{P_{11}}(\frac{\partial P_{11}}{\partial \lambda_f})^2), && (3.10)\\
I_N(\Theta)[1,2] &= I_N(\Theta)[2,1] = \frac{1}{P_0}\frac{\partial P_0}{\partial \lambda_f}\frac{\partial P_0}{\partial \lambda_n} + \frac{1}{P_1}\frac{\partial P_1}{\partial \lambda_f}\frac{\partial P_1}{\partial \lambda_n} \\
&+ (N-1)(\frac{P_0}{P_{00}}\frac{\partial P_{00}}{\partial \lambda_f}\frac{\partial P_{00}}{\partial \lambda_n} + \frac{P_0}{P_{01}}\frac{\partial P_{01}}{\partial \lambda_f}\frac{\partial P_{01}}{\partial \lambda_n} + \frac{P_1}{P_{10}}\frac{\partial P_{10}}{\partial \lambda_f}\frac{\partial P_{10}}{\partial \lambda_n} \\
&+ \frac{P_1}{P_{11}}\frac{\partial P_{11}}{\partial \lambda_f}\frac{\partial P_{11}}{\partial \lambda_n}), && (3.11)\\
I_N(\Theta)[2,2] &= \frac{1}{P_0}(\frac{\partial P_0}{\partial \lambda_n})^2 + \frac{1}{P_1}(\frac{\partial P_1}{\partial \lambda_n})^2 + (N-1)(\frac{P_0}{P_{00}}(\frac{\partial P_{00}}{\partial \lambda_n})^2 \\
&+ \frac{P_0}{P_{01}}(\frac{\partial P_{01}}{\partial \lambda_n})^2 + \frac{P_1}{P_{10}}(\frac{\partial P_{10}}{\partial \lambda_n})^2 + \frac{P_1}{P_{11}}(\frac{\partial P_{11}}{\partial \lambda_n})^2), && (3.12)
\end{aligned}
$$

where $I_N(\Theta)[i,j]$, $1 \le i,j \le 2$, is the i-th row and j-th column element in matrix $I_N(\Theta)$.

Proof: We prove it by mathematical induction.

The CR bounds for λ_f and λ_n are derived respectively as $\sigma^2_{\lambda_f} = \frac{I_N(\Theta)[2,2]}{\det(I_N(\Theta))}$ and $\sigma^2_{\lambda_n} = \frac{I_N(\Theta)[1,1]}{\det(I_N(\Theta))}$, where $\det(\cdot)$ is the determinant of a matrix.

Corollary 3 *If u is known, the CR bound for estimating λ_f and λ_n for an exponential traffic can be expressed as*

$$
\begin{aligned}
\sigma_f^2 &= \frac{u(1-E)[E+u(1-E)^2(1-u)]}{(ET_c)^2(1-u)(1+E)(N-1)}, && (3.13)\\
\sigma_n^2 &= \frac{(1-u)^2\sigma_f^2}{u^2}. && (3.14)
\end{aligned}
$$

For estimating u, we adopt a low-complexity averaging estimator [GLP13b,KS08a] which is expressed as

$$\hat{u} = \frac{1}{N}\sum_{i=1}^{N} s_i. \tag{3.15}$$

3.3.2 Traffic Prediction under Perfect and Imperfect Knowledge of Traffic Parameters

Traffic prediction is performed according to the current traffic parameter estimates with the transition probabilities. As the assumed traffic model is based on a semi-Markov process, future traffic states can be predicted using the current traffic state. Assuming perfect knowledge of traffic parameters, the predicted future traffic state is determined by comparing the transition probability pair (TPP) $\{P_{z0}(T_p), P_{z1}(T_p)\}$, where $z \in \{0, 1\}$ is the current state and T_p is the time period between the current state and the predicted state. For instance, when the current PU state is 0, the next state is predicted to be 0, with a prediction error probability of $P_{01}(T_p)$, if $P_{00}(T_p)$ is larger than $P_{01}(T_p)$, and vice versa. Combining both PU present and absent states, the theoretical prediction error probability is given by

$$P_e = P_0 \times \min\{P_{00}, P_{01}\} + P_1 \times \min\{P_{10}, P_{11}\}, \tag{3.16}$$

where T_p is omitted for convenience. If both transition probabilities of a TPP are equal, the error probability P_e is always 0.5, regardless of the predicted state. Furthermore, expression (3.16) holds for arbitrary prediction time period T_p due to the continuous-time semi-Markov process assumption. Integrating both traffic estimation and prediction procedures, the joint procedure is summarized in Algorithm 2.

3.3.3 Prediction Region

Next, we consider the influence of the traffic parameter estimation errors on the prediction error. The traffic parameter estimation error results in an error in estimating the TPPs. For example, when the current PU state is absent and the TPP is wrongly estimated, where the actual TPP follows $\{P_{00} > P_{01}\}$, but the estimated TPP follows $\{\hat{P}_{00} < \hat{P}_{01}\}$, the prediction error with traffic estimation error is P_{00}, which is larger than P_{01}. This reveals that the estimation error leads to a higher prediction error, when the magnitudes of the estimated TPP have reversed order with respect to the actual TPP. Combining the estimation error

Algorithm 2 Prediction Algorithm

procedure PREDICTOR($\hat{s}_{i+1}, s_i, s_1, s_2, \cdots, s_N, T, N, T_p$)

Calculate duty cycle $\hat{u}$ using (3.15), with observation window T and number of samples N.

Calculate $\hat{\lambda}_f$ using (3.8) for exponential traffic, or numerically by solving (3.7) for Erlang-2 traffic.

Calculate estimated TPPs (3.3)–(3.6) with $\hat{u}$, $\hat{\lambda}_f$, and prediction period T_p.

Do spectrum sensing at current time to obtain s_i.

    if $s_i == 1$ **then** ▷ Prediction starts

        if $\hat{P}_{10} \geq \hat{P}_{11}$ **then** ▷ TPP comparison

            $\hat{s}_{i+1} = 0.$ ▷ Predict the next state

        else

            $\hat{s}_{i+1} = 1.$

        end if

    else

        if $\hat{P}_{00} \geq \hat{P}_{01}$ **then**

            $\hat{s}_{i+1} = 0.$

        else

            $\hat{s}_{i+1} = 1.$

        end if

    end if

    return $\hat{s}_{i+1}$

end procedure

with the prediction error model, we modify equation (3.16) as

$$\begin{aligned} P_e = & P_0 \left[P_{01} I(\hat{P}_{00} - \hat{P}_{01}) + P_{00} I(\hat{P}_{01} - \hat{P}_{00}) \right] \\ & + P_1 \left[P_{10} I(\hat{P}_{11} - \hat{P}_{10}) + P_{11} I(\hat{P}_{10} - \hat{P}_{11}) \right], \end{aligned} \tag{3.17}$$

where $I(x)$ is the indicator function defined as $I(x) = 1$ for $x \geq 0$, and $I(x) = 0$, otherwise. The relationship of the TPPs given the current state defines the prediction regions. We will discuss separately the prediction region for exponential and Erlang-2 traffic in the following paragraphs.

For the exponential traffic, the sufficient and necessary condition for $P_{00} > P_{01}$ is expressed as

$$T_p(\lambda_f + \lambda_n) < \log(\frac{2\lambda_f}{\lambda_f - \lambda_n}), \quad \forall \lambda_f > \lambda_n, \tag{3.18}$$

and $P_{00} > P_{01}$ is always true for $\lambda_f \leq \lambda_n$. On the other hand, $P_{10} < P_{11}$ is valid if

$$T_p(\lambda_f + \lambda_n) < \log(\frac{2\lambda_n}{\lambda_n - \lambda_f}), \quad \forall \lambda_n > \lambda_f, \tag{3.19}$$

and $P_{10} < P_{11}$ always holds valid for $\lambda_n \leq \lambda_f$. It is hard to determine the prediction region as a function of (λ_f, λ_n) based on (3.18) and (3.19), accordingly, we apply the change in variables technique to facilitate defining the prediction regions. Letting $A = \lambda_f + \lambda_n$ and $B = \lambda_f - \lambda_n$, equation (3.18) becomes

$$B < \frac{A}{e^{T_p A} - 1}, \quad \forall B > 0. \tag{3.20}$$

Note that $P_{00} > P_{01}$ always hold when $B \leq 0$. Similarly, equation (3.19) becomes

$$B > \frac{A}{1 - e^{T_p A}}, \quad \forall B < 0, \tag{3.21}$$

and $P_{10} > P_{11}$ is always valid if $B \geq 0$. This preserves the uniqueness of mapping (λ_f, λ_n) into (A, B). The prediction region partitions are defined by equations (3.20) and (3.21), with transformed boundary curves C_1 and C_2, respectively. Fig. 3.2 illustrates the prediction region in terms of A and B for $T_p \in \{0.5, 1\}$ s. The valid regions are $A \geq B$ and $A + B \geq 0$, which are constrained by $\lambda_f \geq 0$ and $\lambda_n \geq 0$, respectively. Each region corresponds to a relationship of the TPPs, resulting in different prediction error rates.

For the Erlang-2 traffic, the two-dimensional region for λ_f and λ_n shown in Fig. 3.3 is categorized into four regions by C_1 and C_2, where

$$C_1 : P_{00} = P_{01} \Leftrightarrow 1 - u(1 - E) = 0.5, \tag{3.22}$$

$$C_2 : P_{10} = P_{11} \Leftrightarrow u(1 - E) + E = 0.5. \tag{3.23}$$

Note that we can not solve λ_n as a closed-form function of λ_f, so C_1 and C_2 can only be derived through numerical methods. Now we provide some important properties for this prediction region.

Property 1 *The line of symmetry for C_1 and C_2 is $\lambda_f = \lambda_n$.*

Proof: By exchanging the variables λ_f and λ_n in C_1, we get the equation of C_2. This shows that C_1 and C_2 are symmetric with respect to the line $\lambda_f = \lambda_n$.

Property 2 *The boundary points of intersection with the axes are $\lambda_f = \frac{-W(-1,-e^{-2})}{t}$ and $\lambda_n = \frac{-W(-1,-e^{-2})}{t}$, where $W(n, x)$ is the Lambert W function [CGH96] with branch n and value x.*

Proof: To figure out the point of intersection on λ_f-axis, we first need to calculate the limiting value of C_1. By L'Hospital's rule, we have $\lim_{\lambda_n \to 0} E = (1 + \frac{t\lambda_f}{2})e^{-\lambda_f t}$. Then we can calculate the boundary point by solving $(1 + \frac{t\lambda_f}{2})e^{-\lambda_f t} = \frac{1}{2}$. Applying the change of variable technique, i.e., let $z = (1 + \frac{\lambda_f t}{2})$, and plugging it into the above equation, we have $(-2z)e^{-2z} = -e^{-2}$, and $W(n, -e^{-2})$ are the solutions for this equation. Since $z \geq 1$ and we are interested in the real solution, this results in $z = -\frac{1}{2}W(-1, -e^{-2})$, which gives the boundary point λ_f as the property shows. However, since C_1 and C_2 are symmetric, the boundary point on the λ_n-axis has the same value as on the λ_f-axis.

Property 3 *The crossing points for C_1 and C_2 are the points which satisfy the equation $\lambda_f = \frac{(\pi/2+n\pi)}{t} = \lambda_n$, where $n \in \mathbb{Z}^+$.*

Proof: By solving C_1 and C_2 jointly, we have $u = 0.5$ and $E = 0$. The equation $u = 0.5$ results in $\lambda_f = \lambda_n$, which means the crossing points are located on the line of symmetry.

The equation $E = 0$ results in $\psi = j\lambda_f$, where $j = \sqrt{-1}$, and hence we have $\cosh(j\lambda_f t) = 0$. By solving the above equation, we can derive the solution shown in the property.

Property 4 *The asymptote for both C_1 and C_2 is $\lambda_f = \lambda_n$.*

Proof: We can easily show that $\lim_{\lambda_f \to \infty} E = 0$ by L'Hospital's rule; therefore, $C_1 \to 1-u = 0.5$ and $C_2 \to u = 0.5$. These two equations are both equivalent to $\lambda_f = \lambda_n$.

By observing the prediction error rate $\hat{P}_e$ and prediction region, we conclude that as long as the estimated TPPs are located in the same region as the true TPPs, the prediction error rate with estimation errors will be the same as the case with perfect estimation. In the next section, we would like to figure out how to link the traffic estimation and prediction quantitatively in a probabilistic sense.

3.3.4 Prediction Confidence

The influence of the traffic estimation error on traffic prediction is evaluated by the prediction confidence, which is the probability that the estimated parameters $(\hat{A}, \hat{B})$ or $(\hat{\lambda}_f, \hat{\lambda}_n)$ and their actual parameters (A, B) (λ_f, λ_n) are located in the same prediction region. To guarantee that they are located in the same prediction region, the sufficient condition is that the Euclidean distance between the estimated parameters and the actual parameters needs to be smaller than the minimum Euclidean distance between the actual parameters and the boundary curves C_1 and C_2.

The prediction confidence is quantified by the cumulative distribution function (CDF) of the traffic parameter estimation errors. Define two pairs of error random variables $\Delta\lambda_f = \hat{\lambda}_f - \lambda_f$ and $\Delta\lambda_n = \hat{\lambda}_n - \lambda_n$, where $\Delta A = \hat{A} - A = \Delta\lambda_f + \Delta\lambda_n$ and $\Delta B = \hat{B} - B = \Delta\lambda_f - \Delta\lambda_n$. The prediction confidence for the exponential traffic is given by

$$\begin{aligned}
&\Pr\{(\Delta A)^2 + (\Delta B)^2 < d^2\} \\
&= \Pr\{(\Delta\lambda_f + \Delta\lambda_n)^2 + (\Delta\lambda_f - \Delta\lambda_n)^2 < d^2\} \\
&= \Pr\{(\Delta\lambda_f)^2 + (\Delta\lambda_n)^2 < \frac{d^2}{2}\},
\end{aligned} \tag{3.24}$$

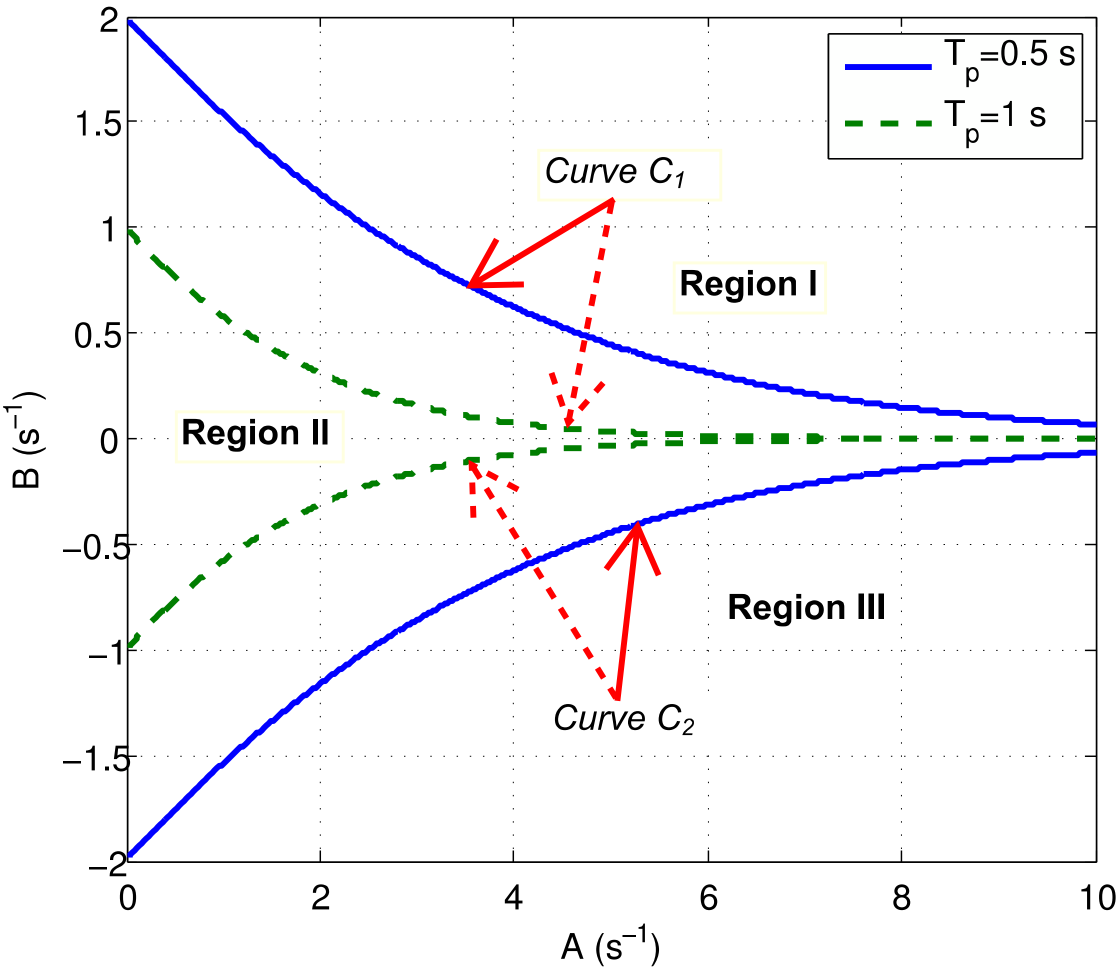


Figure 3.2: Prediction regions with prediction period $T_p = 0.5$ s and $T_p = 1$ s.

where d is the minimum Euclidean distance between (A, B) and the boundary curves, which is solved numerically by

$$\text{Minimize} \quad d = \sqrt{(\tilde{A} - A)^2 + (\tilde{B} - B)^2}$$

$$\text{subject to} \quad \begin{cases} \tilde{B} = \frac{\tilde{A}}{e^{T_p \tilde{A}} - 1}, & \forall B \geq 0, \\ \tilde{B} = \frac{\tilde{A}}{1 - e^{T_p \tilde{A}}}, & \forall B < 0. \end{cases}$$

For the Erlang-2 traffic, the prediction confidence can be derived as

$$\Pr\{(\Delta\lambda_f)^2 + (\Delta\lambda_n)^2 < d^2\}, \tag{3.25}$$

where $\Delta\lambda_f = \hat{\lambda}_f - \lambda_f$,$\Delta\lambda_n = \hat{\lambda}_n - \lambda_n$, and d is the minimum distance from (λ_f, λ_n) to the nearest boundary curves C_1 or C_2. d can be solved numerically by minimizing $d =$

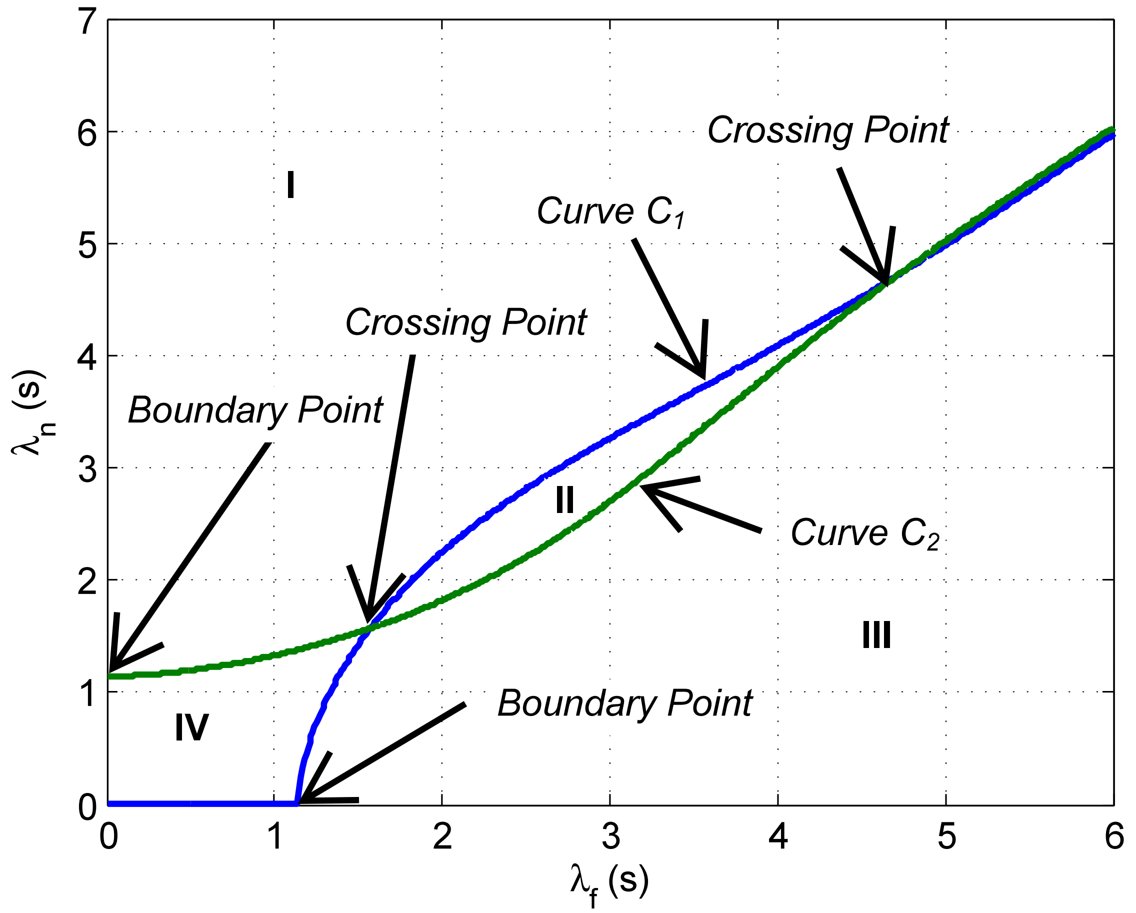


Figure 3.3: Prediction regions with prediction period $T_p = 1$s.

$\sqrt{(\tilde{\lambda}_f - \lambda_f)^2 + (\tilde{\lambda}_n - \lambda_n)^2}$, subject to $(\tilde{\lambda}_f, \tilde{\lambda}_n) \in \{C_1 \cup C_2\}$.

Note that $\Delta\lambda_f$ and $\Delta\lambda_n$ are the estimation errors which can be interpreted as the noise added to λ_f and λ_n, respectively. Therefore, we model this noise as Gaussian random variables, which is a general assumption for ML estimation [Mil11]. Their means are $\mathbb{E}\{\Delta\lambda_f\} = m_f$ and $\mathbb{E}\{\Delta\lambda_n\} = m_n$[1], where $\mathbb{E}\{\cdot\}$ denotes expectation. Their variances are $\mathrm{Var}\{\Delta\lambda_f\} = \sigma_f^2$ and $\mathrm{Var}\{\Delta\lambda_n\} = \sigma_n^2$.

Define a random variable $Z = (\Delta\lambda_f)^2 + (\Delta\lambda_n)^2$ as the sum of the square errors, which is a Chi-Square random variable due to the Gaussian assumption for the estimation errors.

[1] We assume the ML estimators for both $\hat{\lambda}_f$ and $\hat{\lambda}_n$ are asymptotically unbiased, i.e., $m_f \approx 0$ and $m_n \approx 0$.

The correlation coefficient of $\hat{\lambda}_f$ and $\hat{\lambda}_n$ is ρ^2 where

$$\rho = \frac{\mathbb{E}\{\Delta\lambda_f \Delta\lambda_n\} - m_f m_n}{\sigma_f \sigma_n}. \tag{3.26}$$

The CDF of Z is represented as [Sim02, Eq. (5.31)]

$$\begin{aligned} F_Z(z) &= 1 + \exp(-\frac{1}{4}(\theta - \phi)z) I_0(\frac{1}{4}\phi z) \\ &- 2Q_1(\frac{\sqrt{(\sigma^2 - 2\zeta\sqrt{1-\rho^2})z}}{2\zeta\sqrt{1-\rho^2}}, \frac{\sqrt{(\sigma^2 + 2\zeta\sqrt{1-\rho^2})z}}{2\zeta\sqrt{1-\rho^2}}), \end{aligned} \tag{3.27}$$

where $\sigma^2 = \sigma_f^2 + \sigma_n^2$, $\zeta = \sigma_f \sigma_n$, I_0 is the modified Bessel function of the first kind with order zero, Q_1 is the first-order Marcum Q-function, and

$$\phi = \frac{\sqrt{\sigma^4 - 4\zeta^2(1-\rho^2)}}{\zeta^2(1-\rho^2)}, \quad \theta = \phi + \frac{\sigma^2}{\zeta^2(1-\rho^2)}.$$

Combining equations (3.24), (3.25), and (3.27), given a traffic estimation error, the prediction confidence κ are $F_Z(d^2/2)$ for the exponential traffic and $F_Z(d^2)$ for the Erlang-2 traffic.

3.4 Numerical Results

This section presents numerical results for the prediction error rate yielded by different prediction algorithms. Moreover, we discuss the joint traffic estimation and prediction design issues. In our simulations, the traffic states are generated randomly with different traffic parameters following the semi-Markov process described in Section 3.2. Each simulation result is calculated by averaging over 10000 realizations.

3.4.1 Prediction Error Rate

Fig. 3.4 shows the prediction error rate with respect to the prediction period with exponential traffic. First, our proposed algorithm achieves the prediction error rate of a predictor with

[2]The correlation coefficient approaches 1 if the mean of the ML estimators $\hat{\lambda}_f$ and $\hat{\lambda}_n$ approach λ_f and λ_n respectively (see Appendix A.6 for detailed derivation for the exponential traffic). Here we set $\rho = 1 - \epsilon$ with $\epsilon = 10^{-7}$ in our simulations.

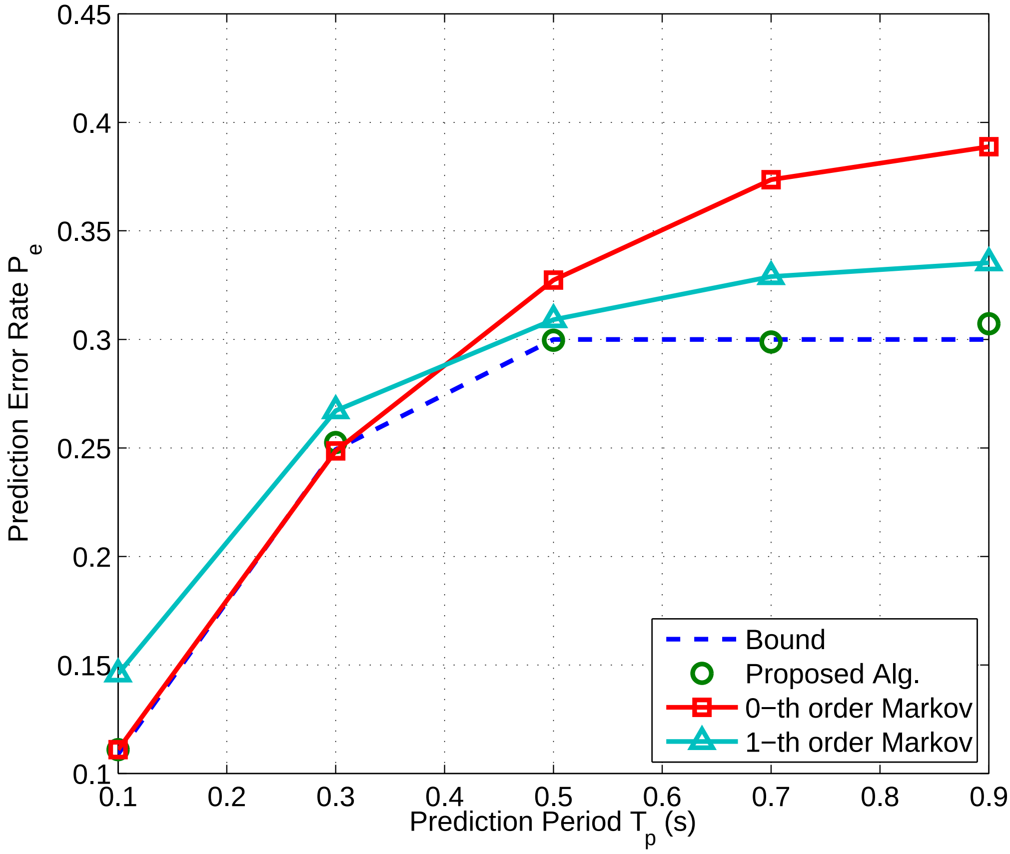


Figure 3.4: Variation of the prediction error with the prediction period T_p with exponential traffic. The error is plotted for the proposed algorithm and the algorithm presented in [SCZ10a]. The used parameters are $T = 50$ s, $N = 300$, $u = 0.3$, and $\lambda_f = 0.9$ s^{-1}.

perfect knowledge of traffic parameters. Second, the prediction error rate increases as the prediction period becomes longer. This is because the traffic becomes more uncorrelated, resulting in more uncertainty in traffic prediction. Third, we compare our algorithm with the universal predictor presented in [SCZ10a] which does not rely on the traffic distribution, but only depends on past traffic states. It is called the k-th order Markov predictor since it relies on k past states for prediction. The 0-th and 1-th order Markov predictors are shown to yield the highest prediction accuracy and are hence used in this work as a reference. Since our prediction algorithm exploits the characteristics of the traffic distribution, it outperforms the two Markov predictors under all circumstances.

Fig. 3.5 shows the prediction error rate with respect to the prediction period. First, we compare the prediction error rate for the same prediction algorithm but different traffic models, i.e., Erlang-2 and exponential distributions. In order to have a fair comparison, we set λ_f in the Erlang-2 distribution as equal to twice the value in the exponential distribution. Under this setting, they have the same first moment. The simulation figure shows that the prediction error rate for Erlang-2 traffic is higher than that of exponential traffic. This is because Erlang-2 traffic has a lower traffic sample correlation compared with exponential traffic. The result is not surprising since both Erlang-2 and exponential distributions are special cases of a gamma distribution but the shape parameter for Erlang-2 is of higher order. This increases the difficulties to have accurate prediction. Finally, we observe that the analytical results for prediction error rate match the simulation results and we can easily show that the prediction error rate will saturate to $\min\{u, 1-u\}$ as the prediction period goes to infinity.

The variation of the prediction error rate with the channel utilization u is presented in Fig. 3.6 with exponential traffic. The results are plotted for our proposed algorithm, and the algorithm proposed in [SCZ10a] using 0-th and 1-th order Markov predictors. Our proposed algorithm matches the analytical bound given by equation (3.16). Moreover, the prediction error rates decrease when the channel utilization approaches 0 or 1. This is caused by the decrease in the traffic uncertainty. Furthermore, our proposed algorithm achieves the best prediction error rate among the three prediction algorithms under all channel utilizations.

3.4.2 Estimation Performance

Fig. 3.7 shows the standard deviation for the joint ML estimators of λ_f and λ_n with respect to the number of traffic samples with Erlang-2 traffic. First, we observe that as the number of samples increases, the standard deviation decreases. This is because more traffic samples provide more information for the estimator and therefore the estimation accuracy will increase. However, the standard deviation for $\hat{\lambda}_n$ is much higher than $\hat{\lambda}_f$ in this figure. This can be explained by noting that the PU average OFF duration ($2\lambda_f^{-1} = 2.2$s) is larger than

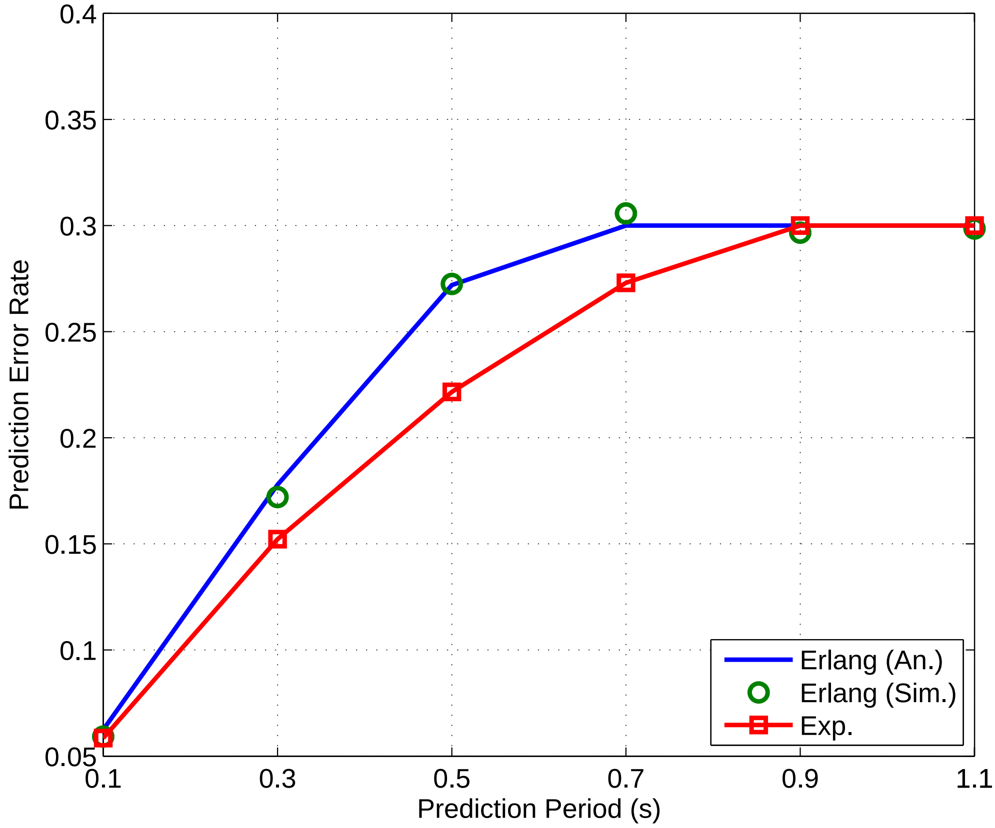


Figure 3.5: Prediction error rate comparisons for exponential (Exp.) and Erlang-2 traffic with the prediction period. Analytical (An.) and simulated (Sim.) results for Erlang-2 predictors are compared. The used parameters are $u = 0.3$, $\lambda_f = 0.9\text{s}^{-1}$ for Erlang-2 distribution and $\lambda_f = 0.45\text{s}^{-1}$ for exponential distribution.

the average ON duration ($2\lambda_n^{-1} = 0.95\text{s}$). Given an observation time T, we are likely to get more samples from the OFF period than the ON period, and thus we can estimate the OFF period more accurately. Second, we verify that the CR bounds serve as the lower bounds for the estimators. We will later use these bounds to calculate the prediction confidence analytically.

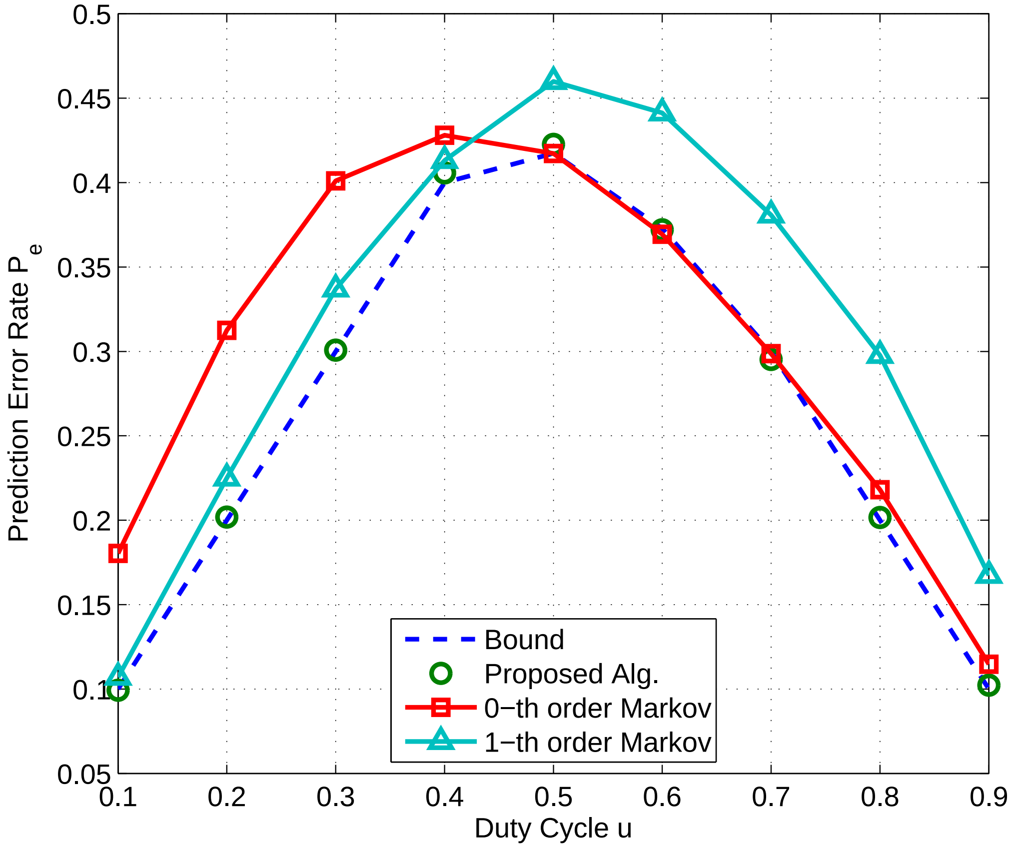


Figure 3.6: Variation of the prediction error with the channel utilization u with exponential traffic. The error is plotted for the proposed algorithm and the algorithm presented in [SCZ10a]. The used parameters are $T = 50$ s, $N = 300$, $T_p = 1$ s, and $\lambda_f = 0.9$ s^{-1}.

3.4.3 Prediction Confidence

The proposed analytical expression for the CDF, equation (3.27), is plotted with simulation results in Fig. 3.8 for the exponential traffic. The simulation and analytical results provide a good match indicating that the assumption of the Gaussian distribution of the estimation errors as well as the unbiased assumption for the estimators are valid. Moreover, as N increases, the CDF curve moves upwards. This can be explained by the fact that the variances σ_f^2 and σ_n^2 decrease as N increases, hence, causing an increase in the CDF values. Therefore, a higher prediction confidence is expected as the number of samples is increased.

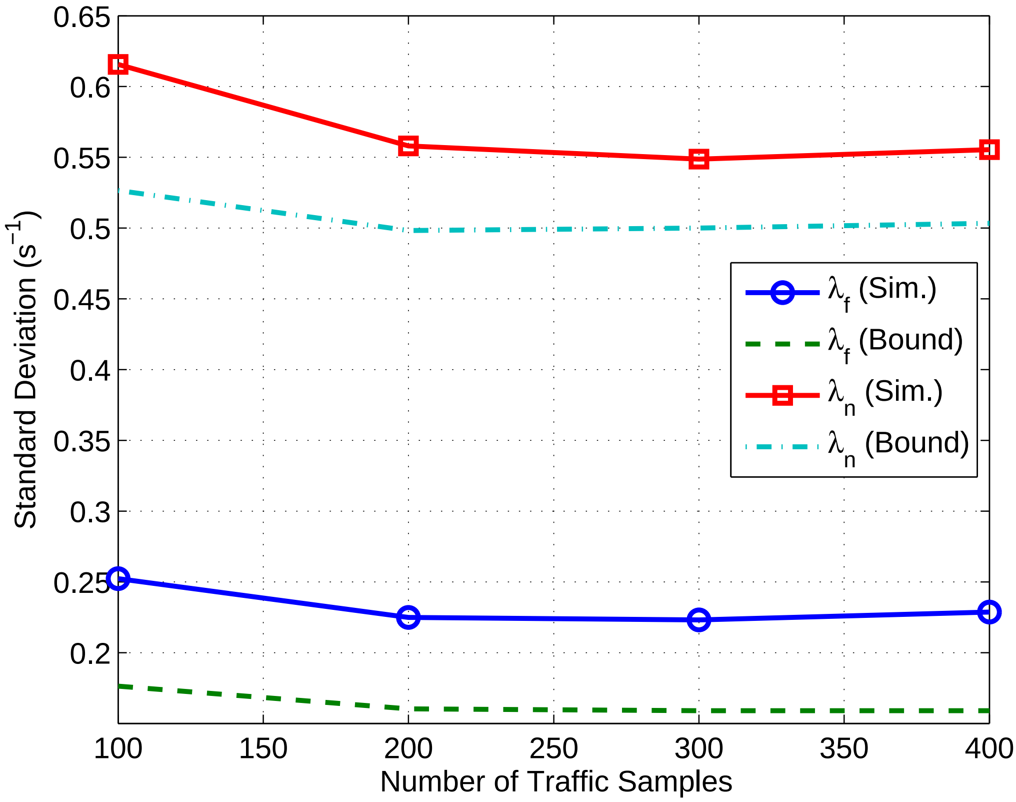


Figure 3.7: Simulated (Sim.) and analytical bound of standard deviation for Erlang-2 traffic estimators with number of traffic samples. The used parameters are $u = 0.3$, $\lambda_f = 0.9\text{s}^{-1}$, $\lambda_n = 2.1\text{s}^{-1}$, and $T = 50\text{s}$.

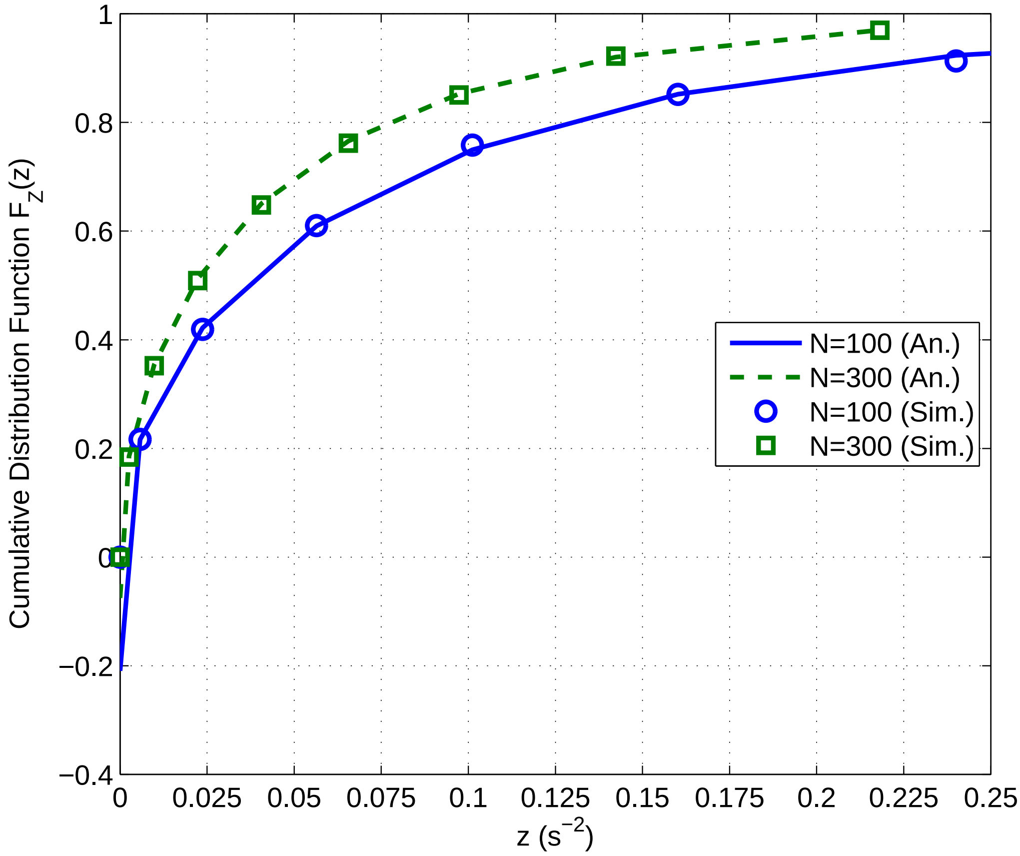


Figure 3.8: CDF for the sum of squares of the estimation errors of $\hat{\lambda}_f$ and $\hat{\lambda}_n$ with exponential traffic. Analytical results (An.) and simulation results (Sim.) are presented. The used parameters are $T = 50$ s, $u = 0.3$, and $\lambda_f = 0.4$ s^{-1}.

The variation of the prediction confidence with the prediction period T_p is illustrated in Fig. 3.9 for different number of samples N with exponential traffic. The simulation results match the analytical results more accurately for larger N. Moreover, larger N decreases the estimation errors of λ_f and λ_n, resulting in a higher prediction confidence. It is worth noting that longer T_p leads to higher prediction confidence when the actual parameters (A, B) are located in Region I or III, and vice versa. Furthermore, as T_p increases, the boundary curves move inward, thus, the minimum Euclidean distance d increases, and the probability of $(\hat{A}, \hat{B})$ and (A, B) being located in the same region increases.

The prediction confidence for different channel utilization u under different typical val-

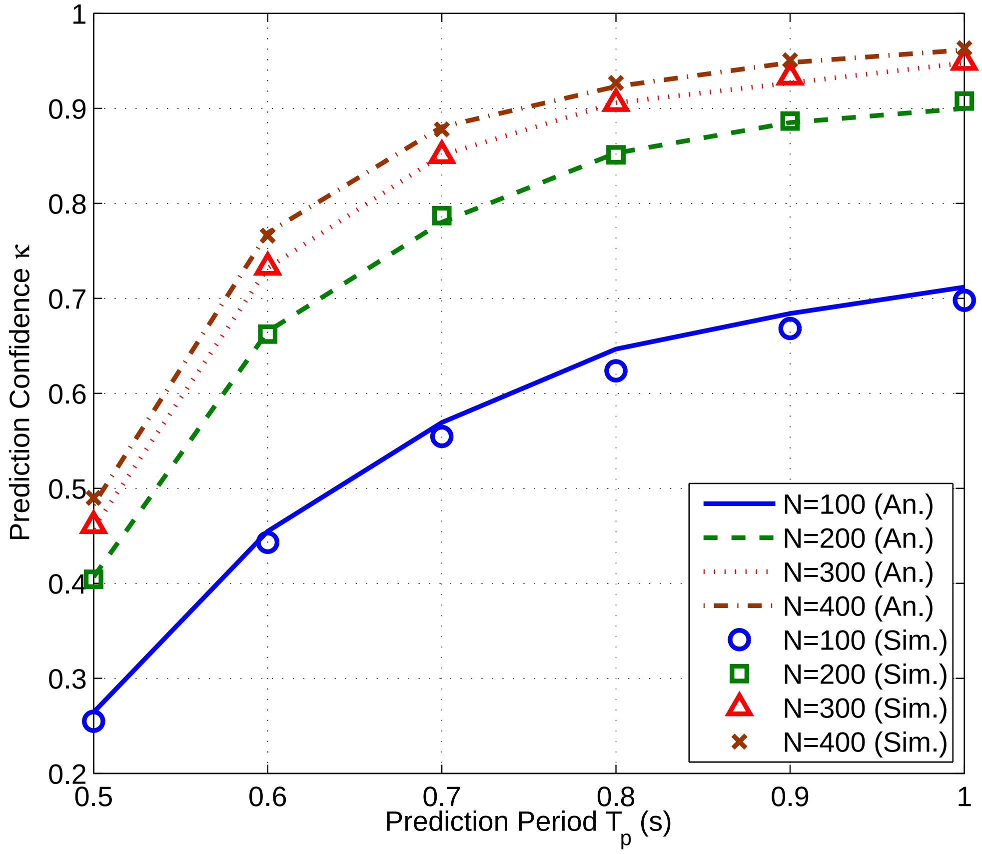


Figure 3.9: Variation of the prediction confidence with the prediction period T_p with exponential traffic. The prediction confidence is plotted for different numbers of samples N. Analytical results (An.) and simulation results (Sim.) are presented. The used parameters are $T = 50$ s, $u = 0.3$, and $\lambda_f = 0.9$ s^{-1}.

ues of λ_f [GLP13b] and prediction periods T_p is presented in Fig. 3.10. The trend of the confidence level is determined by the position of the parameters (A, B) and the boundary curves C_1 and C_2. The traffic parameters determine the position of (A, B), and the larger prediction period makes the boundary curves C_1 and C_2 closer. As u increases, (A, B) move from the bottom right to the top left with a slope of -1. Therefore, different combinations of λ_f and T_p result in different trends of the confidence level. Given $\lambda_f = 0.9$ s^{-1} and $T_p = 1$ s, (A, B) move from $(9, -7.2)$ to $(1, 0.8)$ as u is swept from 0.1 to 0.9. The parameters (A, B) initially move toward C_2, then away from C_2, then toward C_1, and finally away from C_1.

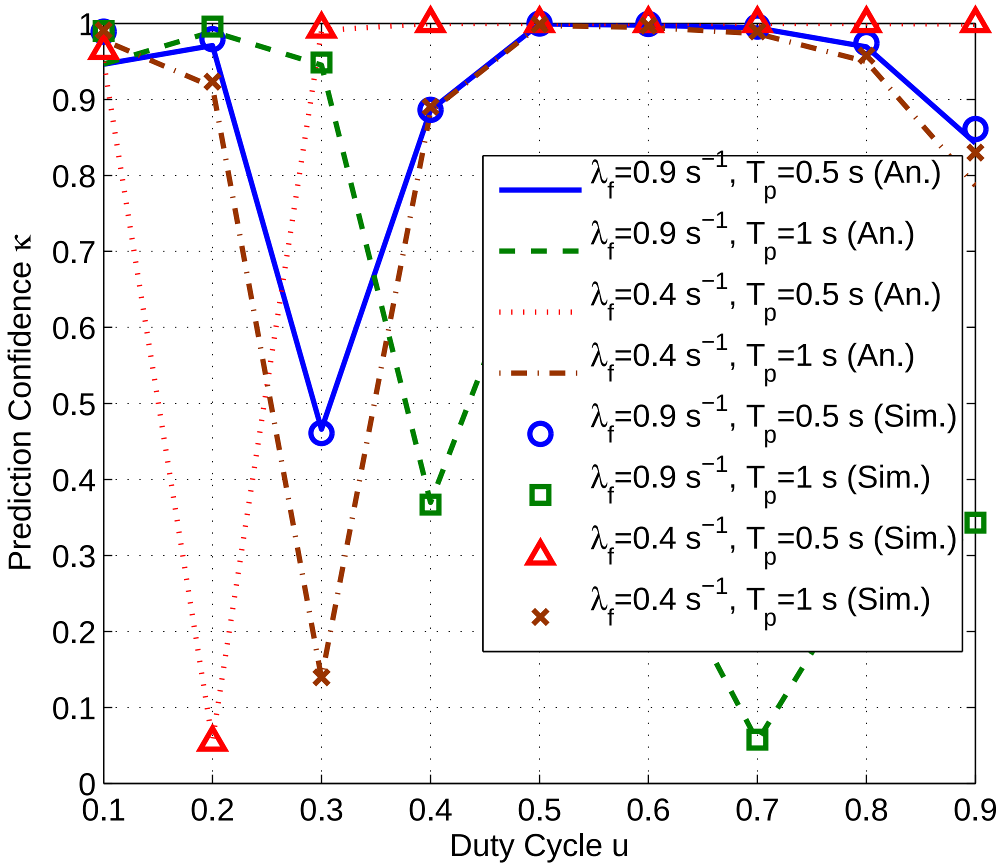


Figure 3.10: Variation of the prediction confidence with the duty cycle u with exponential traffic. The prediction confidence is plotted for different combinations of λ_f and T_p. Analytical results (An.) and simulation results (Sim.) are presented. The used parameters are $T = 50$ s, $N = 300$.

This explains the dips in the prediction confidence for different λ_f and T_p.

Fig. 3.11 shows the prediction confidence with the prediction period for Erlang-2 traffic. First we observe that as the prediction period increases, the prediction confidence decreases initially and then it increases. This can be explained by looking at the prediction region. The traffic parameters $(\lambda_f, \lambda_n) = (0.9, 2.1)$ are located in region IV which is close to curve C_2 for $t = 0.5$ in this example. At this point, we can expect to have a low prediction confidence since the traffic estimation error can easily cause the estimated traffic parameters to be located outside region IV. As t increases to 0.6, C_2 starts to move downward and (λ_f, λ_n) is moved

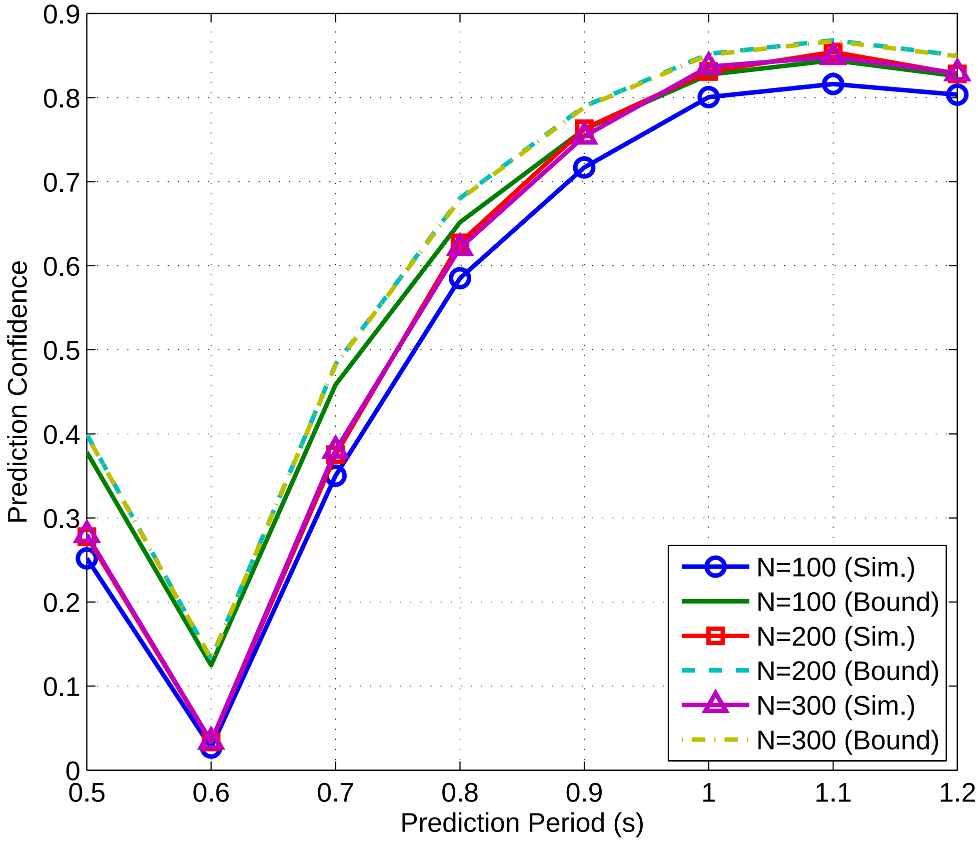


Figure 3.11: Simulated (Sim.) and analytical bound of prediction confidence with prediction period for Erlang-2 traffic. The used parameters are $u = 0.3$, $\lambda_f = 0.9\text{s}^{-1}$, and $T = 50$s.

to region I. At this point, C_2 gets closer to (λ_f, λ_n); therefore, the prediction confidence decreases. Moreover, as t increases beyond 0.6, C_2 moves downward and farther away from (λ_f, λ_n). Hence in this case, the prediction confidence increases. Note that as the prediction period approaches infinity, d becomes a constant and hence κ will saturate. Second, as the number of traffic samples increases, the prediction confidence increases. This is because as the sample increases, the estimation accuracy for the traffic parameters increases. Therefore, the estimator would have a higher chance to stay in the same region as the true parameters. Finally, we verify that (3.25) serves as an upper bound for the simulation results since we adopt the CR lower bounds for the variance of both traffic estimators.

3.4.4 Joint Traffic Estimation and Prediction Design Issues

The prediction confidence serves as a guideline for jointly designing traffic estimation and prediction algorithms. Achieving a low prediction error rate and a high confidence level for different prediction periods are the design goals. Traffic parameters need to be estimated accurately enough by properly choosing T and N in order to achieve a low prediction error rate, while maintaining sufficient prediction confidence κ by lowering the variance of estimation errors.

3.5 Summary

In this work, we proposed a primary user traffic prediction algorithm. The algorithm was based on the traffic estimation of the PU traffic parameters assuming exponentially and Erlang-2 distributed ON/OFF channel utilization intervals. The prediction accuracy was analyzed in terms of the estimation accuracy of the PU state transition probabilities, that were calculated using the estimated traffic parameters. Furthermore, prediction regions were defined for the actual and estimated traffic parameters, which linked the prediction error with the traffic parameters estimation error. Finally, we quantified the prediction confidence in terms of the cumulative distribution function of the Chi-Squared-distributed variance of the estimated parameters. The prediction confidence analysis provided guidelines for selecting the number of samples used for estimation as well as the total estimation time period needed to achieve a certain prediction error.

CHAPTER 4

Traffic-Aware Channel Sensing Order in Dynamic Spectrum Access Networks

4.1 Introduction

In Chapters 2 and 3 we did the traffic analysis to obtain PU traffic knowledge. In the following chapters, we would like to design/improve OSA protocols by exploiting the traffic knowledge. In the event of multiple SUs and PU channels, it is important to have a well-designed PU channel selection protocol and channel sensing order scheme, i.e. the process of ordering PU channels to sense to optimize a predefined objective (throughput, delay, etc.) of a DSA network.

For instance, in a centralized DSA network, PU channel selection and sensing order scheduling can be done by the base station (or a coordinator). The DSA base station would then decide on the best sensing policy based on the PU channel statistics and would schedule SU transmissions accordingly when a channel is sensed to be vacant. In order to acquire the PU traffic statistics of each channel (which vary over time, making exact a priori knowledge of PU traffic impossible), a highly accurate PU traffic estimation algorithm is needed.

4.1.1 Related Work

Techniques of estimating the PU traffic parameters were studied in, e.g., [SCZ10a, GLP13a] (and the references therein), in isolation from the channel sensing order problem. Considering channel sensing order, [KS08a] was the first work to investigate this aspect (to the best of the authors' knowledge) and introduced the concept of semi-Markov chain to model the availability of PUs (assuming perfect knowledge of the PU traffic statistics). Therein, it was proposed to sense the channels in descending order of the PU idle probability to find the spectrum opportunities with a small delay. In [KS08b], the authors improved the sensing order results of [KS08a]. Their objective was to determine the optimal sensing sequence that minimizes the average delay in finding idle channels with the constraint of the total capacity of the system being above a certain threshold. The approach was not only based on the idle probability of the PU channel (again, assumed to be known perfectly), but also on sensing time and channel capacity. In [CZ11], the authors proposed a simple channel sensing order for SUs by sensing the channels in descending order of their achievable rates. Therein they proved that the SUs should stop sensing at the first sensed-free channel for maximum throughput without requiring a priori knowledge of PU traffic. This work considered maximization of the throughput for one SU to transmit in one channel while neglecting the transmission opportunities in other channels. The authors of [JLF09] investigated the optimal sensing order with opportunistic transmissions by considering both adaptive and non-adaptive modulations for SUs to exploit changes in PU channel physical layer parameters. The problem of PU traffic parameter estimations was not considered in this work. Finally, [ZGT08] considered finding an optimal channel sensing order (through a constrained Markov decision process) taking collision probability between PU and DSA network into consideration. As in all above referred works, the authors also assumed that the arrival and departure rates for each PU channel were known perfectly in advance.

4.1.2 Our Contribution

The contribution of this work is threefold:

1. PU Channel Sensing Order Algorithm and Its Derivation: A sensing order algorithm maximizing the throughput for SUs is proposed, assuming first a perfect knowledge of PU traffic statistics and PU channel physical layer parameters;

2. PU Traffic Estimation Modeling and Analysis: For discrete PU traffic models, parameters of PU traffic are estimated using Maximum Likelihood (ML) estimation. More-

over, the Cramér-Rao (CR) bound is derived for the adopted estimators by finding their Fisher information matrix;

3. Analysis of Throughput for the Proposed Channel Sensing Order Algorithm Considering the Traffic Estimation Process: The proposed sensing order algorithm is analyzed as a function of PU traffic estimation errors. We derive the throughput loss in terms of a new sensing metric called *sensing order confidence*, which is derived from the CR bound and Gaussian approximation for the proposed PU traffic estimators' errors. We form a convex optimization problem to determine the minimum number of PU traffic samples needed to obtain a certain level of sensing order confidence on the chosen sensing order. In addition, traffic estimation and our sensing order algorithm are discussed when the traffic is non-stationary.

4.2 System Model

Following the model and assumptions of [KS08a, KS08b, CZ11, JLF09], in this paper, we consider a DSA network consisting of M PU channels of known capacity C_i and K SUs, $\forall i \in \mathbf{M}$, where $\mathbf{M} = \{1, 2, \cdots, M\}$ and $M \ll K$. Each SU can be reconfigured to be a transceiver or a sensor, i.e., it can either transmit on or sense one PU channel at a time [KS08a, KS08b, CZ11]. The transmission structure for each PU is assumed to follow a synchronous, slotted frame within slot time T, implying the activity for the PU is constant during one slot time [PPS09, JL12a]. To enhance the detectability of the PU channels and to minimize the time spent on sensing, the SUs jointly sense one channel at a time via a cooperative sensing scheme [BZ09, LZP08]. In turn, the sensing results are forwarded to a DSA base station which makes a collective decision on whether or not there is a PU present on a channel. To simplify analysis, this paper considers the case where the DSA controller sensing decision is free of spectrum sensing errors, unless explicitly specified otherwise. Note that spectrum sensing free decision is a well justified assumption as under the cooperative sensing with large K sensing errors are amply eradicated. If a PU vacancy in channel i is detected within sensing time $T_I^{(i)} < T$, the DSA base station schedules one SU to switch to a transceiver

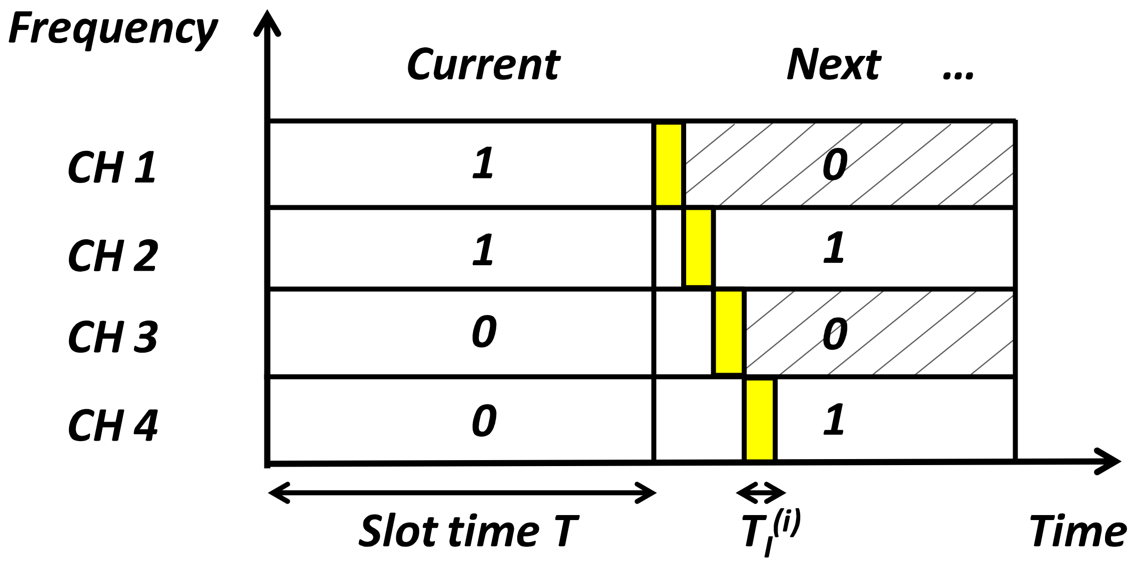


Figure 4.1: An example of time and frequency domain DSA transmission structure for $M = 4$ PU channels. Current state $\mathbf{s} = (1, 1, 0, 0)$, and next state $\hat{\mathbf{s}} = (0, 1, 0, 1)$. Sensing order $A = (1, 2, 3, 4)$.

and transmit on that channel. The remaining SUs continue to sense the rest of the channels in an order governed by a predefined algorithm. Our task is to design such algorithm, i.e., a sequential sensing order A for SUs to sense all PU channels, which maximizes the overall DSA network throughput on a slot basis.

PU traffic is modeled as a Markov process [ZTS07, SZ08], where s represents the current state of the PU. States $s = 0$ and $s = 1$ indicate the PU absence and presence, respectively. This traffic is characterized by the steady-state distribution and the transition probabilities. Probability of the PU absence is described by $P_0 \triangleq \Pr\{s = 0\} = 1 - u$ and the probability of PU presence is $P_1 \triangleq \Pr\{s = 1\} = u$. The transition probabilities describe the probability of switching from the current state s to next state $\hat{s}$ after a constant slot time T. The probability of transition from state x to state y is denoted as P_{xy} with four possible probabilities given by tuple $\{P_{00}, P_{01}, P_{10}, P_{11}\}$ with $u = \frac{P_{01}}{P_{01}+P_{10}}$, $P_{00} + P_{01} = 1$ and $P_{10} + P_{11} = 1$. Define $\mathbf{s} \triangleq (s_1, s_2, \cdots, s_M)$ and $\hat{\mathbf{s}} \triangleq (\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_M)$, where s_i is the current PU traffic state and $\hat{s}_i$ is the next PU traffic state for the i-th channel. Note each PU follows its own Markov process, i.e., s_i is defined by its own traffic parameters $\left\{u^{(i)}, P_{00}^{(i)}, P_{10}^{(i)}\right\}$.

4.3 A Sensing Order Algorithm to Maximize Throughput in Multi-Channel DSA Networks

We begin by explaining in detail our proposed sensing order algorithm and its relationship to the throughput in our system model. The DSA throughput gains of this sensing order algorithm are dependent on the accuracy of the traffic statistic estimates of the PUs on each channel. Therefore, we show the need of a sensing order confidence metric which will govern the level of impact of estimation errors on the throughput loss. This sensing order confidence metric is derived from the CR bounds of the maximum likelihood traffic estimator adopted in this work. Deployed SU networks will manually specify or tune this metric according to network conditions (number of channels, MAC structure, etc.) and desired throughput. Once the sensing order confidence is specified, SU networks will know how accurately they will need to estimate the traffic before proceeding to sense channels and transmit. We show how a convex optimization problem can be formulated to minimize the number of traffic samples to take in a certain time period to estimate PU traffic statistics to a desirable accuracy, which is determined by the sensing order confidence. Lastly, we will discuss how to use the combination of traffic estimation and the proposed sensing order algorithm when the traffic is non-stationary.

4.3.1 SU Network Throughput and Sensing Order Algorithm

Define a sensing order $A = (a_1, a_2, \cdots, a_M)$ for SUs in M channels, which is a permutation of the set $\mathbf{U}$, i.e., $\mathbf{U}$ is a set that contains all the permutations of the sequence $(1, 2, \cdots, M)$, and $|\mathbf{U}| = M!$. For example, Fig. 4.1 shows a sensing order $(a_1, a_2, a_3, a_4) = (1, 2, 3, 4)$ for 4 channels. The SUs sense the PU channels in this order using the required sensing time $T_I^{(i)}$ for the i-th channel. Assume SUs have the current state information of PU channels, s_i for the i-th channel. After sensing the next time slot, the SUs will obtain the information of the next state $\hat{s}_i$. If it is sensed as available, the rest of the slot time, $T - \sum_{j=1}^{i} T_I^{(a_j)}$, can be used to transmit data. Otherwise, the SUs will fail to find an opportunity to transmit in their respective queues. We define the effective rate for data transmission for all channels in one slot given the next state information after sensing as the system throughput $R(A)$ [CZ11, NLW10, PLG09]

$$R(A) = \sum_{i=1}^{M} \left(\frac{T - \sum_{j=1}^{i} T_I^{(a_j)}}{T} \right) C_{a_i} I(\hat{s}_{a_i}), \tag{4.1}$$

whereas $I(x)$ is the indicator function defined as $I(x) = 1$ for $x = 0$, and $I(x) = 0$, otherwise. The expected throughput $\mathbb{E}\{R(A)\}$ is defined as the average throughput, where $\mathbb{E}\{\cdot\}$ denotes the expectation.

The problem is to maximize $\mathbb{E}\{R(A)\}$ by deciding on the proper sensing order A. We introduce the following theorem.

Theorem 4 *The sensing order to maximize $\mathbb{E}\{R(A)\}$ can be achieved by sensing the channels in descending order of the sensing factor $F_i = \frac{C_i \Pr\{\hat{s}_i = 0\}}{T_I^{(i)}}$, $\forall 1 \leq i \leq M$.*

Proof: Assuming we sense channels in a sequence $A = (a_1, a_2, \cdots, a_l, a_{l+1}, \cdots, a_M)$ in descending order of F_i. Define $A' = (a_1, a_2, \cdots, a_{l+1}, a_l, \cdots, a_M)$ which changes any two consecutive elements a_l and a_{l+1} in A, $\forall 1 \leq l \leq M-1$. Subtracting $\mathbb{E}\{R(A')\}$ from $\mathbb{E}\{R(A)\}$ we have

$$\begin{aligned} &\mathbb{E}\{R(A)\} - \mathbb{E}\{R(A')\} = \\ &\mathbb{E}\left\{ \frac{T_I^{(a_{l+1})}}{T} C_{a_l} I(\hat{s}_{a_l}) \right\} - \mathbb{E}\left\{ \frac{T_I^{(a_l)}}{T} C_{a_l+1} I(\hat{s}_{a_{l+1}}) \right\} = \\ &\frac{T_I^{(a_{l+1})}}{T} C_{a_l} \Pr\{\hat{s}_{a_l} = 0\} - \frac{T_I^{(a_l)}}{T} C_{a_{l+1}} \Pr\{\hat{s}_{a_{l+1}} = 0\}. \end{aligned} \tag{4.2}$$

Since we sense the channels in descending order of F_i, we have $F_{a_l} \geq F_{a_{l+1}}$, which implies that $\mathbb{E}\{R(A)\} - \mathbb{E}\{R(A')\} \geq 0$, and therefore $\mathbb{E}\{R(A')\} \leq \mathbb{E}\{R(A)\}$. Hence, interchanging any two consecutive elements (a_k, a_l) with $F_{a_k} \geq F_{a_l}$ will decrease the expected throughput. However, any sequence $B \in \mathbf{U}$ can be obtained by a finite interchanging algorithm starting from sequence A (see Appendix A.7), with each interchanging step of two consectuive elements (a_k, a_l), $\forall k < l$. Therefore, since $F_{a_k} \geq F_{a_l}$, $\forall k < l$, we have $\mathbb{E}\{R(A)\} \geq \mathbb{E}\{R(B)\}$, which proves the theorem.

For example, given $M = 5$, assume sequence A is sorted by the proposed sensing factor. If we would like to interchange A to a sequence $B = (a_2, a_3, a_5, a_1, a_4)$, we can follow the process

$$A = (a_1, a_2, a_3, a_4, a_5) \to (a_2, a_1, a_3, a_4, a_5) \to (a_2, a_3, a_1, a_4, a_5) \to (a_2, a_3, a_1, a_5, a_4)$$
$$\to (a_2, a_3, a_5, a_1, a_4) = B.$$

Each step interchanges only two consecutive elements (a_k, a_l), $\forall k < l$, so it decreases the expected throughput, which implies $\mathbb{E}\{R(A)\} \geq \mathbb{E}\{R(B)\}$.

The sensing factor F_i is composed of three parts: the capacity, sensing time, and idle probability of the PU channel, as the throughput is related to all of these values. Since the capacity and sensing time are known before deciding a sensing order, the accuracy of the estimation of PU idle probabilities affects the sensing order the most. The idle probability $\Pr\{\hat{s}_i = 0\}$ can be obtained by parameter $1 - u^{(i)}$. With the knowledge of the present state s_i, we can estimate the idle probability as $\Pr\{\hat{s}_i = 0\} = P^{(i)}_{s_i 0}$, where $P^{(i)}_{s_i 0}$ is the transition probability from the present state to state zero for the i-th channel. In addition, if we consider sensing errors (given by false alarm and mis-detection probabilities, denoted by $P^{(i)}_f$ and $P^{(i)}_m$ for the i-th channel, respectively[1]) in our model, the idle probability given the current estimated state $\tilde{s}_i$ can be derived as

$$\Pr\{\hat{s}_i = 0\} = \begin{cases} P^{(i)}_m P^{(i)}_{00} + (1 - P^{(i)}_m) P^{(i)}_{10}, & \text{if } \tilde{s}_i = 1, \\ (1 - P^{(i)}_f) P^{(i)}_{00} + P^{(i)}_f P^{(i)}_{10}, & \text{if } \tilde{s}_i = 0, \end{cases} \tag{4.3}$$

(see detailed derivation in Appendix A.10).

The proposed sensing order algorithm is shown in Algorithm 3. To analyze if the errors of idle probability estimations negatively impact the correctness of our sensing order algorithm, we next define and analyze the sensing order confidence metric.

4.3.2 Sensing Order Confidence Metric and Its Derivation

We start with the derivation of our proposed sensing order confidence metric without sensing errors, i.e., $P^{(i)}_f = P^{(i)}_m = 0 \; \forall i \in \mathbf{M}$. First, we motivate the introduction of this metric. Then,

[1]The sensing error is assumed to be independent for each traffic sample. The estimated PU traffic sample is denoted by $\tilde{s}$. It follows that $\tilde{s} = 1$ if (i) $s = 1$ and no mis-detection error occurred, or (ii) $s = 0$ and a false alarm error occurred. Similarly, $\tilde{s} = 0$ if (i) $s = 1$ and a mis-detection error occurred, or (ii) $s = 0$ and no false alarm error occurred.

Algorithm 3 Proposed Sensing Order Algorithm

```
procedure SENSING(A, C_i, T_I^(i), P_m^(i), P_f^(i))
    i ← 1
    while i ≤ M do
        spectrum sensing at current time slot to obtain s̃_i
        i ← i + 1
    end while
    i ← 1
    while i ≤ M do
        if s̃_i == 1 then
            F_i = C_i (P_m^(i) P_00^(i) + (1 − P_m^(i)) P_10^(i)) / T_I^(i)
        else
            F_i = C_i ((1 − P_f^(i)) P_00^(i) + P_f^(i) P_10^(i)) / T_I^(i)
        end if
        i ← i + 1
    end while
Sort sequence {F_1, ..., F_M} in descending order as the sequence A
    return A
end procedure
```

where line 10 is $F_i = \frac{C_i\left(P_m^{(i)}P_{00}^{(i)}+(1-P_m^{(i)})P_{10}^{(i)}\right)}{T_I^{(i)}}$ and line 12 is $F_i = \frac{C_i\left((1-P_f^{(i)})P_{00}^{(i)}+P_f^{(i)}P_{10}^{(i)}\right)}{T_I^{(i)}}$

we formally introduce our adoption of an ML PU state transition probability estimator and derive its CR bound[2]. Lastly, we show how the estimator and its bounds can be used to derive the sensing order confidence metric.

4.3.2.1 Throughput Loss Derivation

As discussed earlier, when PU traffic statistics are not perfectly known, the sensing order algorithm will not always produce the best sequence, and, as a result, incur DSA throughput loss. We define the normalized throughput loss L as the normalized difference between the expected throughput with estimated transition probabilities and expected maximal through-

[2] For further discussions and derivations of traffic estimation in terms of sensing errors, please refer to [GLP13a].

put when transition probabilities are perfectly known. This is expressed as

$$L = 1 - \frac{\mathbb{E}\{R(\hat{A})\}}{\mathbb{E}\{R(A^o)\}}, \tag{4.4}$$

where $\hat{A}$ is the sensing order sequence obtained by the proposed sensing factor with estimated transition probabilities, and A^o is the sensing order sequence obtained by the proposed sensing factor with perfect transition probabilities. The expected maximal throughput is derived as

$$\mathbb{E}\{R(A^o)\} = \sum_{\mathbf{s}} \sum_{\hat{\mathbf{s}}} \Pr\{\mathbf{s}, \hat{\mathbf{s}}\} R(A^o), \tag{4.5}$$

where $\Pr\{\mathbf{s}, \hat{\mathbf{s}}\}$ is the joint probability for current state $\mathbf{s}$ and next state $\hat{\mathbf{s}}$. The expected estimated throughput is derived in a similar way as

$$\begin{aligned}\mathbb{E}\{R(\hat{A})\} &= \mathbb{E}_{\mathbf{s}} \left\{ \sum_{i=1}^{M!} \kappa_{\mathbf{s}}^i \mathbb{E}_{\hat{\mathbf{s}}}\{R_{\hat{\mathbf{s}}}(A^i)\} \right\} \\ &= \sum_{\mathbf{s}} \sum_{i=1}^{M!} \sum_{\hat{\mathbf{s}}} \kappa_{\mathbf{s}}^i \Pr\{\mathbf{s}, \hat{\mathbf{s}}\} R_{\hat{\mathbf{s}}}(A^i), \end{aligned} \tag{4.6}$$

where $\kappa_{\mathbf{s}}^i = \Pr\{\hat{A} = A^i | \mathbf{s}\}$ is the conditional probability of the estimated order equals to the i-th order $A^i \in \mathbf{U}$, given current state $\mathbf{s}$ (later we will denote $\kappa_{\mathbf{s}}^i$ as the sensing order confidence metric), and $R_{\hat{\mathbf{s}}}(A^i)$ is the throughput with sensing order A^i given the next state $\hat{\mathbf{s}}$, defined as (4.1) replacing A with A^i. Before defining and deriving $\kappa_{\mathbf{s}}^i$ for order A^i, we first discuss the estimators for estimating transition probabilities and their analysis in the next section.

4.3.2.2 ML Transition Probability Estimators

Assume we acquire N PU traffic samples[3] that populate a state vector $\mathbf{z}_N = (z_1, z_2, \cdots, z_N)$ under perfect sensing from a PU channel, where $z_i \in \{0, 1\}$, $\forall 1 \le i \le N$. We formulate the

[3] Recall that in this paper a PU sample is the spectrum sensing result in a time slot. Therefore, the total number of samples is equivalent to the time duration with the unit of a time slot.

likelihood function $L(\mathbf{z}_N)$ using the Markov chain property to be

$$\begin{aligned} L(\mathbf{z}_N) = \Pr\{\mathbf{z}_N\} &= \Pr\{z_1\} \prod_{i=1}^{N-1} \Pr\{z_{i+1}|z_i\} \\ &= u^{z_1}(1-u)^{1-z_1} \prod_{i=1}^{N-1} P_{z_i z_{i+1}} \\ &= u^{z_1}(1-u)^{1-z_1} P_{00}^{n_0} P_{01}^{n_1} P_{10}^{n_2} P_{11}^{n_3} \\ &= u^{z_1}(1-u)^{1-z_1} P_{00}^{n_0}(1-P_{00})^{n_1} P_{10}^{n_2}(1-P_{10})^{n_3}, \end{aligned} \tag{4.7}$$

where n_0, n_1, n_2, and n_3 represent the number of state transitions of $(0 \to 0)$, $(0 \to 1)$, $(1 \to 0)$, and $(1 \to 1)$ respectively. It is known in ML estimation that the unknown variables P_{00} and P_{10} are obtained by solving

$$\frac{\partial \log L(\mathbf{z}_N)}{\partial P_{00}} = 0, \tag{4.8}$$

$$\frac{\partial \log L(\mathbf{z}_N)}{\partial P_{10}} = 0. \tag{4.9}$$

Then the estimators are derived as

$$\hat{P}_{00} = \frac{n_0}{n_0 + n_1}, \tag{4.10}$$

$$\hat{P}_{10} = \frac{n_2}{n_2 + n_3}, \tag{4.11}$$

where $\hat{P}_{01} = 1 - \hat{P}_{00}$ and $\hat{P}_{11} = 1 - \hat{P}_{10}$.

4.3.2.3 ML Estimator CR Bound

We adopt the CR bound, the analytical lower bound for the variance of any unbiased ML estimator, to quantify the performance of our adopted estimators. ML estimators achieve this bound as the number of traffic samples approaches infinity when certain conditions are satisfied [Mil11, Ch. 12]. The CR bound is obtained through the following Lemma.

Lemma 1 *The CR bounds for estimators $\hat{P}_{00}$ and $\hat{P}_{01}$ are V_0 and V_1, respectively, which are expressed as*

$$V_0 = \frac{P_{00}(1 - P_{00})}{P_0(N-1)}, \tag{4.12}$$

$$V_1 = \frac{P_{10}(1-P_{10})}{P_1(N-1)}. \tag{4.13}$$

Proof: See Appendix A.8. Note that as N goes to infinity, the bound goes to zero. This demonstrates that the accuracy of the estimator increases as the number of samples used for estimation increases.

4.3.2.4 The Sensing Order Confidence Metric $\kappa_{\mathbf{s}}^i$

In this section, we first derive analytically the sensing order confidence metric $\kappa_{\mathbf{s}}^i$ using the CR bound of the estimators derived in Section 4.3.2.3. In addition, we define the average sensing order confidence metric κ of obtaining the best order A^o.

We start first with the two channel case, i.e., $M = 2$. We consider sensing two channels in any order (a_i, a_{i+1}) with perfect traffic estimation and $(\hat{a}_i, \hat{a}_{i+1})$ with traffic estimation errors. Given the current state $\mathbf{s} = (s_{a_i}, s_{a_{i+1}})$, the sensing order confidence for two channels only for the sensing order not being affected by traffic estimation errors, is given as $\kappa_{\mathbf{s},2}^i$. It is defined as

$$\begin{aligned}
\kappa_{\mathbf{s},2}^i &\triangleq \Pr\{(\hat{a}_i, \hat{a}_{i+1}) = (a_i, a_{i+1})\} = \Pr\left\{\hat{F}_{a_i} \geq \hat{F}_{a_{i+1}}\right\} \\
&= \Pr\left\{\frac{C_{a_i}\hat{P}_{b_i0}^{(a_i)}}{T_I^{(a_i)}} \geq \frac{C_{a_{i+1}}\hat{P}_{b_{i+1}0}^{(a_{i+1})}}{T_I^{(a_{i+1})}}\right\} \\
&= \Pr\left\{\frac{\hat{P}_{b_i0}^{(a_i)}}{\hat{P}_{b_{i+1}0}^{(a_{i+1})}} \geq \epsilon_i\right\} = \Pr\left\{\hat{P}_{b_i0}^{(a_i)} - \epsilon_i\hat{P}_{b_{i+1}0}^{(a_{i+1})} \geq 0\right\} \\
&= \Pr\left\{\hat{P}_{b_i0}^{(a_i)} - P_{b_i0}^{(a_i)} - \epsilon_i\left(\hat{P}_{b_{i+1}0}^{(a_{i+1})} - P_{b_{i+1}0}^{(a_{i+1})}\right)\right. \\
&\qquad \left. + P_{b_i0}^{(a_i)} - \epsilon_i P_{b_{i+1}0}^{(a_{i+1})} \geq 0\right\} \\
&= \Pr\left\{e^{(a_i)} - \epsilon_i e^{(a_{i+1})} \geq \epsilon_i P_{b_{i+1}0}^{(a_{i+1})} - P_{b_i0}^{(a_i)}\right\},
\end{aligned} \tag{4.14}$$

where $b_i = s_{a_i}$, $\epsilon_i = \frac{C_{a_{i+1}}T_I^{(a_i)}}{C_{a_i}T_I^{(a_{i+1})}}$, $\hat{F}_{a_i} = \frac{C_{a_i}\hat{P}_{b_i0}^{(a_i)}}{T_I^{(a_i)}}$ is the estimated sensing factor, $\hat{P}_{b_i0}^{(a_i)}$ is the estimated transition probability, and $e^{(a_i)} = \hat{P}_{b_i0}^{(a_i)} - P_{b_i0}^{(a_i)}$ is the traffic estimation error for the a_i-th channel. We model the traffic estimation errors $e^{(a_i)}$ and $e^{(a_{i+1})}$ as Gaussian random variables, which is a general assumption for ML estimation if the number of samples used in the estimator is large enough [Mil11, Ch. 12]. Notice that $\mathbb{E}\left\{e^{(a_i)}\right\} = \mathbb{E}\left\{\hat{P}_{b_i0}^{(a_i)}\right\} - P_{b_i0}^{(a_i)} = 0$ and $\mathbb{E}\left\{e^{(a_{i+1})}\right\} = \mathbb{E}\left\{\hat{P}_{b_{i+1}0}^{(a_{i+1})}\right\} - P_{b_{i+1}0}^{(a_{i+1})} = 0$. We define $\mathrm{Var}\left\{e^{(a_i)}\right\} = \mathrm{Var}\left\{\hat{P}_{b_i0}^{(a_i)}\right\} \triangleq \sigma_{a_i}^2$ and $\mathrm{Var}\left\{e^{(a_{i+1})}\right\} = \mathrm{Var}\left\{\hat{P}_{b_{i+1}0}^{(a_{i+1})}\right\} \triangleq \sigma_{a_{i+1}}^2$. The distribution of the sum of two Gaussian independent variables is also normally distributed as $J = e^{(a_i)} - \epsilon_i e^{(a_{i+1})} \sim \mathcal{N}(0, \sigma_{a_i}^2 + \epsilon_i^2 \sigma_{a_{i+1}}^2)$. Hence, (4.14) is expressed as

$$\kappa_{\mathbf{s},2}^i = Q\left(\frac{\epsilon_i P_{b_{i+1}0}^{(a_{i+1})} - P_{b_i0}^{(a_i)}}{\sqrt{\sigma_{a_i}^2 + \epsilon_i^2 \sigma_{a_{i+1}}^2}}\right), \tag{4.15}$$

where $Q(\cdot)$ is the tail probability function of the standard normal distribution. In addition, we derive the approximate closed-form expression for $\kappa_{\mathbf{s},2}^i$ in (4.15) by using the CR bound (4.12) and (4.13) for $\sigma_{a_i}^2$ and $\sigma_{a_{i+1}}^2$, respectively, which is expressed as

$$\kappa_{\mathbf{s},2}^i \approx Q\left(\frac{\epsilon_i P_{b_{i+1}0}^{(a_{i+1})} - P_{b_i0}^{(a_i)}}{\sqrt{\frac{P_{b_i0}^{(a_i)}\left(1 - P_{b_i0}^{(a_i)}\right)}{P_{b_i}^{(a_i)}\left(N_{a_i} - 1\right)} + \epsilon_i^2 \frac{P_{b_{i+1}0}^{(a_{i+1})}\left(1 - P_{b_{i+1}0}^{(a_{i+1})}\right)}{P_{b_{i+1}}^{(a_{i+1})}\left(N_{a_{i+1}} - 1\right)}}}\right), \tag{4.16}$$

where N_{a_i} is the number of samples for the a_i-th channel.

Next, we consider the sensing order confidence metric for M channels. Assume the i-th sensing order is $A^i = (a_1, a_2, \cdots, a_M)$ with corresponding state vector $\mathbf{s}$. The sensing order confidence metric $\kappa_{\mathbf{s}}^i$ of the estimated best sensing order $\hat{A} = (\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_M)$ is achieved as long as the order for any two consecutive sequence is not affected by the estimation errors, which is represented as

$$\begin{aligned}
\kappa_{\mathbf{s}}^i &= \Pr\{\hat{A} = A^i\} \\
&= \Pr\{[(\hat{a}_1, \hat{a}_2) = (a_1, a_2)] \cap [(\hat{a}_2, \hat{a}_3) = (a_2, a_3)] \cap \cdots \\
&\qquad \cap [(\hat{a}_{M-1}, \hat{a}_M) = (a_{M-1}, a_M)]\}.
\end{aligned} \tag{4.17}$$

Note that the order for any two consecutive sequence is not affected by other consecutive sequences, therefore the events in (4.17) are independent, i.e.,

$$\kappa_{\mathbf{s}}^{i} = \prod_{i=1}^{M-1} \Pr\{(\hat{a}_i, \hat{a}_{i+1}) = (a_i, a_{i+1})\} = \prod_{i=1}^{M-1} \kappa_{\mathbf{s},2}^{i}$$
$$= \prod_{i=1}^{M-1} Q\left(\frac{\epsilon_i P_{b_{i+1}0}^{(a_{i+1})} - P_{b_i 0}^{(a_i)}}{\sqrt{\sigma_{a_i}^2 + \epsilon_i^2 \sigma_{a_{i+1}}^2}} \right), \tag{4.18}$$

by using $\kappa_{\mathbf{s},2}^{i}$ from (4.16).

Finally, the sensing order confidence metric of the best sensing order, denoted as $\kappa_{\mathbf{s}}^{[o]}$, is derived by applying the best sensing order A^o to (4.18). In addition, the average sensing order confidence metric for the best sensing order is expressed as

$$\kappa = \mathbb{E}_{\mathbf{s}}\{\kappa_{\mathbf{s}}^{[o]}\}. \tag{4.19}$$

4.3.3 Minimizing the Required Number of Samples for Traffic Estimation: A Convex Optimization Problem

From the estimation point of view, using more PU traffic samples, i.e., a longer traffic estimation time, will increase the accuracy of obtaining the transition probabilities. Although higher sensing order confidence can be achieved with more traffic samples or longer estimation time, it is realistic to consider energy-efficient, delay-aware traffic estimation. To do so, we strive to find the minimum number of samples for each channel to satisfy the desired sensing order confidence constraint, since each channel has its own traffic statistics. The optimization problem becomes

$$\min \sum_{i=1}^{M} N_i \text{ subject to } \kappa_{\mathbf{s}}^{[o]} \geq \eta, \tag{4.20}$$

where N_i $\forall i \in \mathbf{M}$, is the number of samples used for estimating the i-th channel, and η is the minimum tolerated sensing order confidence metric. Expression $\kappa_{\mathbf{s}}^{[o]}$ is a function of N_i since it contains the variances for the estimated transition probabilities, which are shown through the CR bounds in (4.12) and (4.13). Note that the problem of (4.20) is purely theoretical, as in a practical network implementation of the optimal sensing order A^o is not known.

The objective function is a linear combination of M variables. Therefore, it is a convex function. The constraint can be shown to be a convex constraint[4] by the following Lemma.

Lemma 2 $\kappa_{\mathbf{s}}^{[o]}$ *is a concave function.*

Proof: See Appendix A.9. Since both the objective function and its constraint are convex, we show that the optimization problem is a standard convex optimization problem which can be solved by numerous methods (e.g. interior point method [BV04]) using computer optimization tools (e.g. CVX [GB11]) efficiently.

4.3.4 Discussion: Traffic Estimation and The Sensing Order Algorithm in Non-Stationary Traffic

We assume stationary PU traffic in our system model, which means the traffic of all channels is estimated at once before the proposed sensing order algorithm starts. However, in the real wireless scenario, the traffic statistics may change slowly according to the utilization of PUs. This implies we need an algorithm which can track the traffic statistics together with the proposed sensing order algorithm simultaneously. A sliding window method is a good candidate to solve this problem. For example, in the initial phase, we adopt a long time period to estimate each channel and obtain their statistics with high accuracy. Then, we start the tracking phase by sensing the channels using the proposed sensing order. After sensing each channel, we collect the samples and update the statistics within a window length of N_w samples. The samples located outside the window are abandoned. The statistics are updated gradually as time goes assuming the statistics do not vary rapidly with time, which is a reasonable assumption.

4.4 Numerical Results

In this section, we verify our proposed algorithm and analysis with computer simulations. The simulation model here follows directly the one introduced in Section 4.2. First, we

[4] A standard convex constraint is $f \leq \zeta$, where f is a convex function. It is equivalent to $-f \geq -\zeta$, where $-f$ is a concave function.

compare the performance of our proposed sensing order algorithm with the state of the art sensing order algorithms in [KS08a,KS08b,CZ11]. Then, we verify the CR bound of PU traffic estimators and compare it with the simulated variance of our proposed ML estimators of the PU traffic parameters. Furthermore, we show the analytical and simulated results of the average sensing order confidence metric with respect to the number of samples used in traffic estimation and the DSA network throughput loss with respect to the average sensing order confidence metric. Lastly, we demonstrate two design examples of using traffic estimation and the proposed sensing order algorithm. The examples will show the minimum required number of samples for PU traffic estimation given a sensing order confidence constraint.

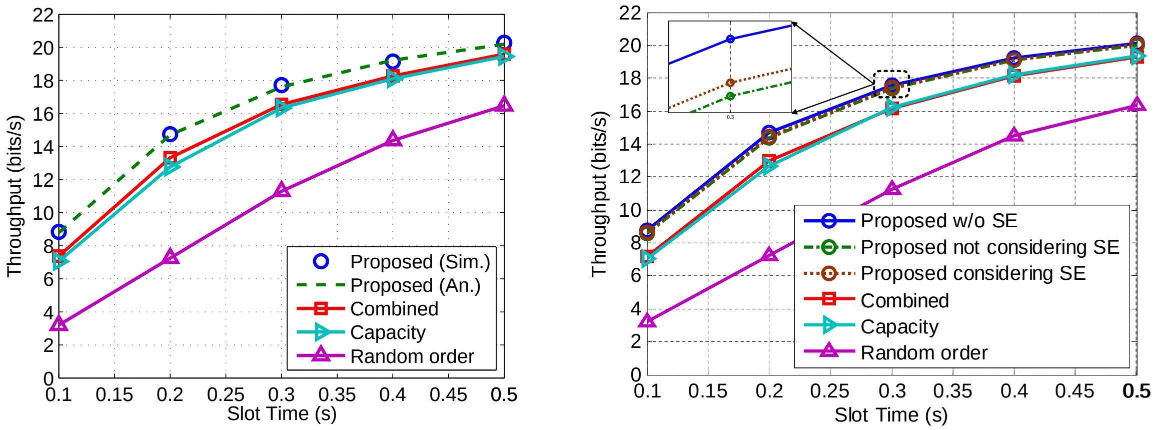
4.4.1 DSA Throughput with Perfect PU Traffic Estimation

Fig. 4.2 shows the throughput comparisons for different sensing order algorithms assuming perfect PU traffic estimation. We choose $M = 10$ PU channels in our scenario. For the system parameters, we choose the constant traffic parameters $\mathbf{P_{00}}$ and $\mathbf{P_{10}}$, which are vectors containing the elements $P_{00}^{(i)}$ and $P_{10}^{(i)}$, respectively, for the i-th channel $\forall i \in \mathbf{M}$. For the physical layer parameters, we choose constant capacity vector $\mathbf{C}$ and sensing time vector $\mathbf{T_I}$, which contains the elements C_i and $T_I^{(i)}$, respectively, for the i-th channel $\forall i \in \mathbf{M}$. $T_I^{(i)}$ is chosen to be less than or equal to 50 ms and C_i is less than or equal to 10 bits/s, which are reasonable values following [KS08b]. Each simulation result is calculated by averaging over 10^4 realizations.

Under perfect sensing for the current PU states, Fig. 4.2(a) demonstrates that as the PU slot time T increases, the DSA throughput increases as well. This is because as long as T increases, there is more time for SU transmissions, i.e., enlarging $T - T_I^{(i)}$. Furthermore, we can observe that the analytical result of our proposed sensing order algorithm, derived in (4.5), matches the simulation result. It is clear from Fig. 4.2(a) that our proposed sensing order algorithm outperforms other well-known sensing order algorithms, such as the combined algorithm which merges the two algorithms proposed in [KS08a, KS08b], the capacity-based algorithm in [CZ11], and the random order algorithm. The combined algorithm adopts the sorting factor $\frac{\Pr\{\hat{s}_i=0\}}{T_I^{(i)}}$ to sense the channels. The capacity-based algorithm considers simply sensing the channels in descending order of C_i. The random order algorithm senses the channels randomly. By considering the traffic and physical layer sensing parameters together, our proposed algorithm performs the best over the other sensing order algorithms.

Fig. 4.2(b) takes imperfect sensing for current PU states into account. First, we can observe the throughput degradation when we have sensing errors with $P_f^{(i)} = P_m^{(i)} = 0.1, \forall i \in \mathbf{M}$. If we do not include sensing errors into our sensing factor (when sensing errors are actually present), i.e., we set $P_f^{(i)} = 0$ and $P_m^{(i)} = 0$ in Algorithm 3 (refer to lines 10 and 12), the performance degradation from the perfect sensing case is even worse. Second, since the combined method also takes transition probabilities into consideration, its throughput will degrade as well. However, the capacity and random order method does not rely on transition probabilities. Therefore, they show the same performance as the case without sensing errors.

To validate the stochastic performance of the sensing order algorithms, we plot the experimental cumulative distribution function (CDF) of the throughput R in Fig. 4.3 under perfect sensing, i.e. $P_f^{(i)} = P_m^{(i)} = 0, \forall i \in \mathbf{M}$. The CDF curve is obtained based on 100 randomly generated parameter sets $\mathcal{S} = \{\mathbf{C}, \mathbf{T_I}, \mathbf{P_{00}}, \mathbf{P_{10}}\}$ with 10^4 random realizations for parameters $\{\mathbf{s}, \hat{\mathbf{s}}\}$ in each set. To obtain values for each element in $\mathcal{S}$: (i) parameters $P_{00}^{(i)}$ in $\mathbf{P_{00}}$ and $P_{10}^{(i)}$ in $\mathbf{P_{10}}$ are chosen uniformly and randomly from $\mathcal{U}(0,1)$, where $\mathcal{U}(a,b)$ denotes the uniform distribution within a range of minimum and maximum value of a and b, respectively, for the i-th channel $\forall i \in \mathbf{M}$; (ii) the capacity C_i in $\mathbf{C}$ and sensing time $T_I^{(i)}$ in $\mathbf{T_I}$ are also chosen to be uniformly distributed in the range $\mathcal{U}(0, 2C_{\text{avg}})$ and $\mathcal{U}(0, 2T_I^{\text{avg}})$, where C_{avg} and T_I^{avg} are the average capacity and average sensing time, respectively. Because the curve representing our proposed algorithm lays on the right-hand side of the other curves, we conclude that our algorithm also outperforms state of the art approaches in a stochastic manner.



(a) Throughput within the PU slot time T without sensing errors. The DSA throughput is plotted for the proposed algorithm, combined algorithm [KS08a, KS08b], capacity algorithm [CZ11], and random order algorithm.

(b) Throughput within the PU slot time T with sensing errors (SE) $P_f^{(i)} = P_m^{(i)} = 0.1\ \forall i \in \mathbf{M}$. The DSA throughput is plotted for the proposed algorithm without sensing errors, the proposed algorithm not considering sensing errors (assuming, conversely, $P_f^{(i)} = 0$ and $P_m^{(i)} = 0$ in F_i in Algorithm 3, the proposed algorithm considering sensing errors (set $P_f^{(i)} = 0.1$ and $P_m^{(i)} = 0.1$ in F_i in Algorithm 3, the combined algorithm [KS08a, KS08b], the capacity algorithm [CZ11], and the random order algorithm.

Figure 4.2: DSA throughput performance evaluation considering both with and without sensing errors. The chosen parameters are $M = 10$, $\mathbf{P_{00}}$=(0.1, 0.2, 0.5, 0.3, 0.6, 0.3, 0.2, 0.7, 0.8, 0.9), $\mathbf{P_{10}}$=(0.7, 0.2, 0.3, 0.5, 0.4, 0.8, 0.9, 0.1, 0.6, 0.2), $\mathbf{T_I}$=(20, 10, 30, 40, 10, 20, 40, 40, 30, 50) ms, and $\mathbf{C}$=(2, 8, 5, 1, 6, 3, 9, 7, 10, 1) bits/s.

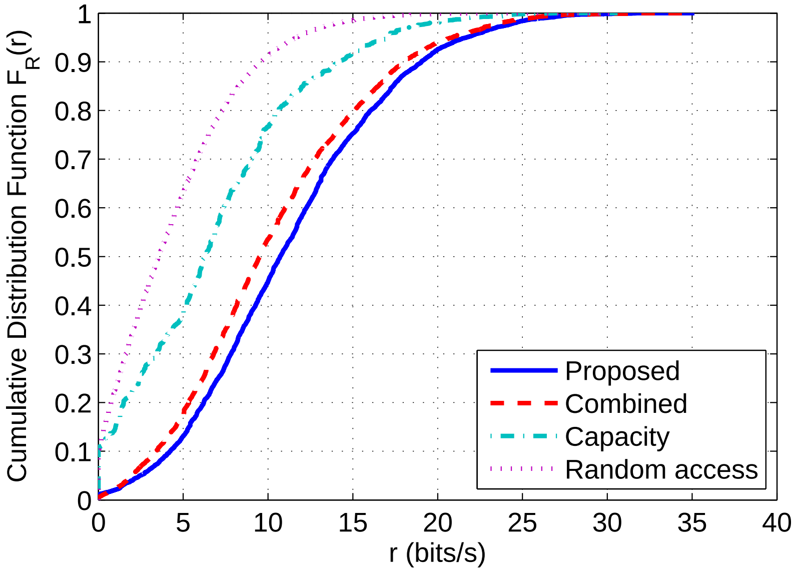
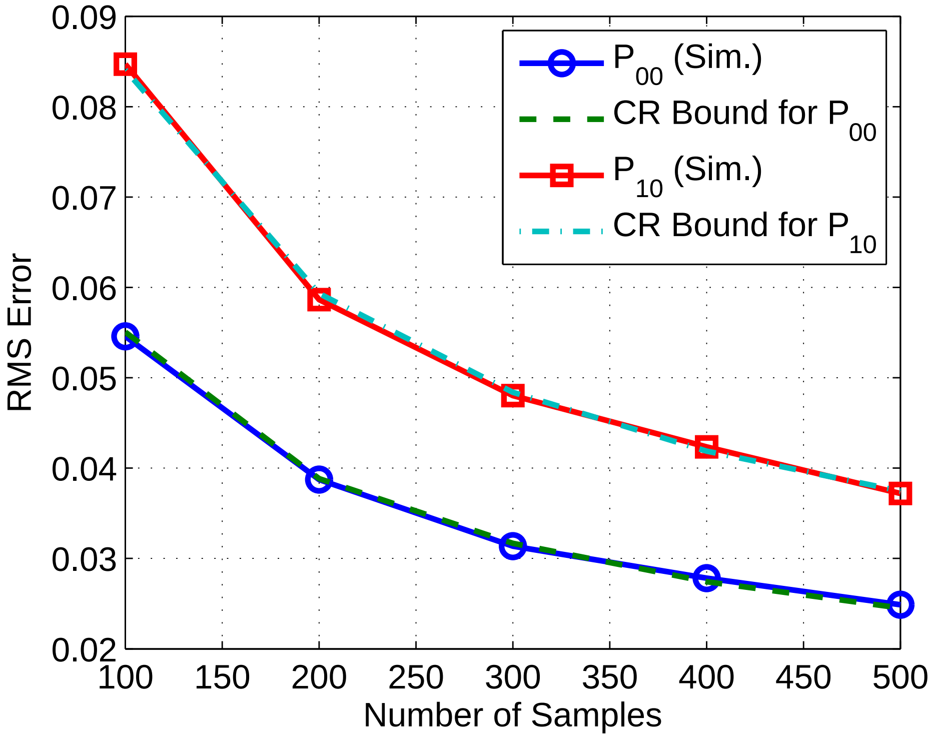


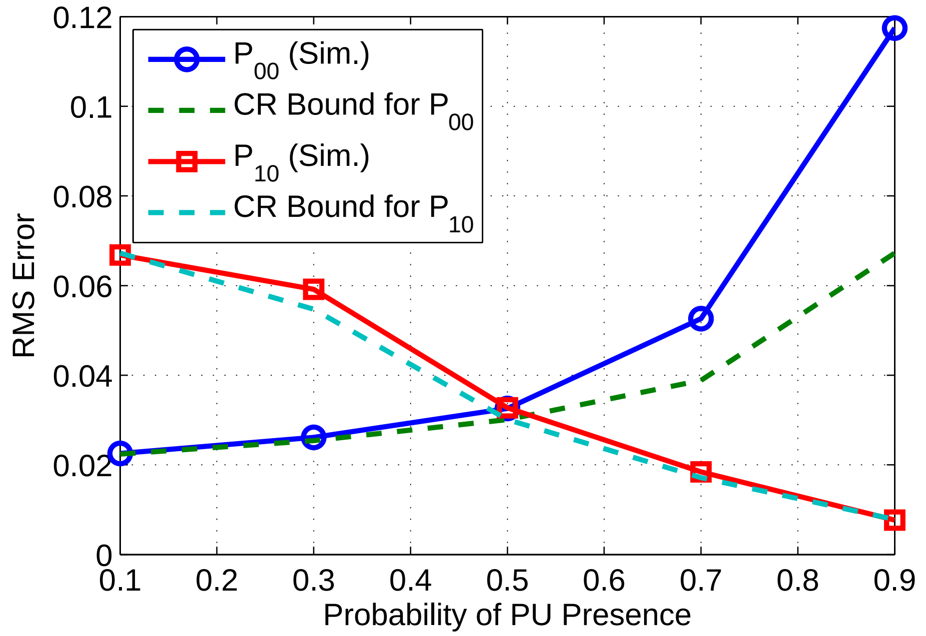
Figure 4.3: Experimental CDF comparisons of the DSA throughput obtained from each realization under perfect sensing. The CDF is plotted for our proposed algorithm, combined algorithm [KS08a, KS08b], capacity algorithm [CZ11], and random order algorithm. The used parameters are $M = 10$, $T_I^{\text{avg}} = 25$ ms, $T = 0.1\,\text{s}$, $C_{\text{avg}} = 5\,\text{bits/s}$, $P_{00}^{(i)} \sim \mathcal{U}(0,1)$, and $P_{10}^{(i)} \sim \mathcal{U}(0,1)$.

4.4.2 PU Traffic Estimation Performance Under Perfect Sensing

To measure the traffic estimation performance, we consider only one channel as multiple channels can follow the same estimation process. Fig. 4.4 depicts the PU traffic estimation performance. The traffic parameters are chosen as $P_{00} = P_{10} = 0.7$ in Fig. 4.4(a), and $N = 200$, $P_{01} = 0.1$ in Fig. 4.4(b). The metric we use to evaluate the performance is the root-mean-square (RMS) error of the estimated transition probabilities, i.e., taking the square root of the mean-squared-error (MSE) of the estimated parameters. Fig. 4.4(a) demonstrates that as the number of samples increases, the RMS error decreases. Improved accuracy for the estimation is obtained by taking more traffic samples into consideration. Furthermore, the estimator $\hat{P}_{00}$ is shown to have better performance than $\hat{P}_{10}$. This can be explained by focusing at $u = 0.3$ in this example. Since $P_0 = 1 - u = 0.7$, we know that there is higher chance for zero to be present in the sequence. This increases the number of the estimation information n_0 and n_1 for $\hat{P}_{00}$, which results in more accurate estimation. In addition, the RMS error achieves the CR bound as long as the number of samples is large enough, which verifies the ML estimation property stated in [Mil11]. Fig. 4.4(b) plots the RMS error with respect to u. Note that CR bound remains a lower bound for the PU estimators. But for the estimator $\hat{P}_{00}$, the RMS error deviates away from the CR bound when u is high. This is because, as u grows larger, the channel has a higher probability of being occupied. Most of the traffic samples are sensed busy, and therefore the measured transition numbers n_0 and n_1 are close to zero. The collection of such a small amount of transition numbers is not sufficient for the accurate PU traffic estimation, especially when the number of samples is not large enough (e.g. $N = 200$ in our simulations). In addition, $\hat{P}_{10}$ becomes more accurate as u increases, since it has more information on n_2 and n_3 given a constant number of samples. However, as we explained in Fig. 4.4(a), the performance of $\hat{P}_{00}$ and $\hat{P}_{10}$ depends on u. This phenomenon is clearly shown here, i.e., $\hat{P}_{10}$ is better than $\hat{P}_{00}$ when $u > 0.5$, and $\hat{P}_{00}$ is better than $\hat{P}_{10}$ when $u < 0.5$.



(a) RMS error of estimated transition probabilities with the number of samples. The parameters used are $P_{00} = 0.7$, $P_{10} = 0.7$.



(b) RMS error of estimated transition probabilities with the probability of PU presence u. The parameters used are $N = 200$, $P_{01} = 0.1$.

Figure 4.4: Traffic estimation performance evaluation: CR bound and simulation results (Sim.).

4.4.3 Sensing Order Confidence Metric and DSA Throughput Loss Under Perfect Sensing

Fig. 4.5 plots the average sensing order confidence, i.e., (4.19) versus the number of samples used for traffic estimation. In this simulation, we apply the same number of samples used for PU traffic estimation in each channel, which is more convenient for demonstrating the results. The simulation results are shown under two different traffic parameter sets named 'Test 1' and 'Test 2'. We define the probability of PU presence vector as $\mathbf{u} = \{u^{(1)}, u^{(2)}, \cdots, u^{(M)}\}$. From the simulation results, we can first observe that the average sensing order confidence increases when the number of samples increases. This is because, once we estimate the traffic parameters more accurately, the resulting sensing order will be with high probability closer to the best one. Secondly, we can observe that the simulation results match the analytical results. The results demonstrate that our approximation for the ML estimation errors being Gaussian distributed and application of the CR bound for the variance are accurate enough for deriving the average sensing order confidence.

Fig. 4.6 plots the normalized throughput loss L with respect to the average sensing order confidence κ. The figure shows that, as the average sensing order confidence increases, the throughput loss decreases. This is because increasing the average sensing order confidence translates to a higher probability of obtaining the best sensing order (i.e., we have a larger chance to achieve the maximal throughput). Note the analytical result for L matches the simulation result when the sensing order confidence metric $\kappa_{\mathbf{s}}^{i}$ is known. As $\kappa_{\mathbf{s}}^{i}$ is derived by (4.18), L deviates from the analytical results to the simulation result when the average sensing order confidence is small. This can be explained by the Gaussian assumption we make for the ML estimators' errors. When the number of samples is not large enough, the ML estimators' errors will not approach a Gaussian distribution. Therefore $\kappa_{\mathbf{s}}^{i}$ is not accurate in this region. The figure provides a guideline for choosing the sensing order confidence metric in a system perspective, since we can get the information on the throughput loss in our system given a corresponding average sensing order confidence. As in the example shown in Fig. 4.6, if we would like to design a system with a normalized throughput loss of less

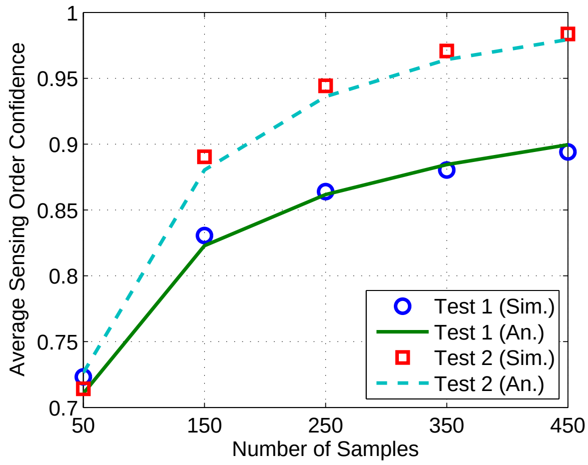


Figure 4.5: Average sensing order confidence with the number of samples. Analytical (An.) results and simulation (Sim.) results are presented. The parameters used are $M = 6$, $\mathbf{u} = (0.2, 0.1, 0.6, 0.5, 0.4, 0.3)$, $\mathbf{T}_I = (20, 10, 30, 40, 10, 20)$ ms, $\mathbf{C} = (2, 8, 5, 1, 6, 3)$ bits/s. Test 1 adopts $\mathbf{P_{00}} = (0.87, 0.90, 0.76, 0.72, 0.73, 0.83)$. Test 2 adopts $\mathbf{P_{00}} = (0.80, 0.90, 0.40, 0.50, 0.60, 0.70)$.

than 1% (note we have done the simulations up until this point), we need to design a traffic estimation algorithm which keeps the average sensing order confidence higher than 57%.

4.4.4 Design Examples

In this section, we show two design examples for solving the proposed convex optimization problem under perfect sensing. For a given state vector $\mathbf{s}$, we need to determine the minimum number of samples for PU traffic estimation to achieve certain confidence. Note that confidence maps to equivalent throughput loss. We consider two designs:

Design 1

$$\min \ N \ \textit{subject to} \ \kappa_{\mathbf{s}}^{[o]} \geq \eta; \tag{4.21}$$

Design 2 (See Section 4.3.3)

$$\min \ \sum_{i=1}^{M} N_i \ \textit{subject to} \ \kappa_{\mathbf{s}}^{[o]} \geq \eta. \tag{4.22}$$

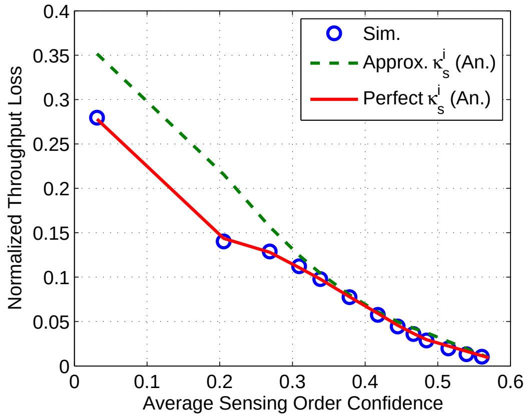


Figure 4.6: Normalized throughput loss with the average sensing order confidence. Simulation (Sim.) results are compared to the analytical (An.) results. Both approximated (Approx.) and perfect sensing order confidence are plugged into the analytical expression for normalized throughput loss. The parameters used are $T = 0.15\,\text{s}$, $M = 6$, $\mathbf{u}$=(0.2,0.1,0.6,0.5,0.4,0.3), $\mathbf{P_{00}}$=(0.95, 0.94, 0.92, 0.89, 0.88, 0.93), $\mathbf{T}_I$=(20,10,30,40,10,20) ms, $\mathbf{C}$=(2,8,5,1,6,3) bits/s.

The first design optimizes a single variable N used for all channels, while the second design optimizes M variables used for each channel. Obviously *Design 1* solves a sub-optimal solution while *Design 2* solves the optimal solution since each channel should be optimized according to its own traffic parameters, i.e., *Design 1* is a special case of *Design 2*. It has been proved in Section 4.3.3 that both designs are solving a convex optimization problem, so we can solve it numerically by using optimization tools, e.g., those available in MATLAB.

Denote the optimal solutions N^o and $N_i^o\ \forall i \in \mathbf{M}$ for *Design 1* and *Design 2*, respectively. Since the number of samples can only be taken in integers, the optimal solution should be rounded to the nearest integer which is greater than itself, i.e., $\lceil N^o \rceil$ and $\lceil N_i^o \rceil$. Note that the rounding process preserves the convex constraint, since the sensing order confidence function is an increasing function with respect to number of samples. We compare these two solutions in terms of total number of samples needed for estimation, i.e., $M\lceil N^o \rceil$ and $\sum_{i=1}^{M} \lceil N_i^o \rceil$, for *Design 1* and *Design 2* respectively.

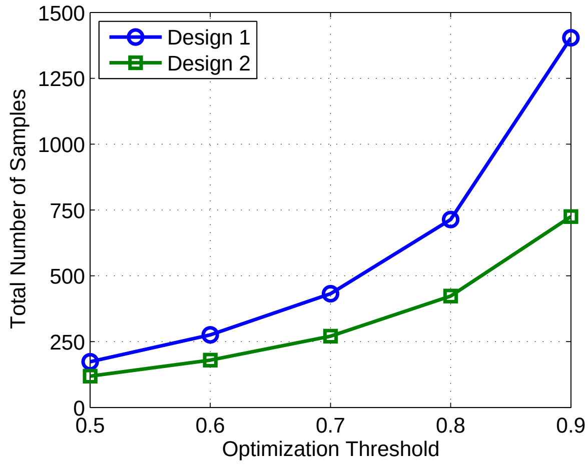


Figure 4.7: Total number of samples with optimization threshold. Comparison of optimizing single number of sample used for every channel with optimizing all number of samples used for each channel. The used parameters are $M = 6$, $\mathbf{u} = (0.2, 0.1, 0.6, 0.5, 0.4, 0.3)$, $\mathbf{P_{00}} = (0.87, 0.90, 0.76, 0.72, 0.73, 0.83)$, $\mathbf{T}_I = (20, 10, 30, 40, 10, 20)\,\mathrm{ms}$, $\mathbf{C} = (2, 8, 5, 1, 6, 3)\,\mathrm{bits/s}$, current state vector $\mathbf{s} = (1, 1, 1, 1, 1, 1)$.

Fig. 4.7 presents the total number of samples with optimization threshold η. First, as the sensing order confidence constraint threshold increases, the total number of samples increases. Since we have a tighter constraint, we need more PU traffic samples to raise the estimation accuracy. Second, we see that the optimal solution requires much less traffic samples than the sub-optimal solution. The optimal solution saves at most 48.36% samples from the sub-optimal solution for $\eta = 0.9$. But the optimal solution takes longer time to be solved, so there is a tradeoff between the duration of the traffic estimation process and the computational complexity.

4.5 Summary

In this work, we proposed a sensing order algorithm to maximize the throughput of secondary users transmitting on multiple primary user channels. The algorithm was based on deriving the sensing factor which was dependent on the traffic statistics of the primary users and sensing parameters of the secondary user, assuming a discrete-time Markov chain channel utilization model and a slotted transmission structure for primary users. Furthermore, we proposed the use of a maximum likelihood estimator to estimate the traffic statistics and quantify the estimation performance using the Cramér-Rao bound. Finally, we analyzed the performance of the proposed sensing order algorithm together with the traffic estimation. We derived the sensing order confidence metric of obtaining the best sensing order analytically by approximating the estimator errors with the Gaussian distribution and applying the Cramér-Rao bound. The system throughput loss was derived according to the sensing order confidence, which linked to the traffic estimation duration. We also formed a convex optimization problem in order to select the number of samples needed to sense each channel for a certain level of sensing order confidence.

CHAPTER 5

Robust Cooperative Spectrum Sensing Scheduling Optimization in Multi-Channel Dynamic Spectrum Access Networks

5.1 Introduction

In addition to the sequential sensing order optimization shown in Chapter 4 which is analyzed considering all physical layer details, this chapter is also the first work which formulates the general sensing strategy problem and addresses the compromises that exists between parallel and sequential sensing strategies, i.e., assigning less users to each channel in order to sense a large number of channels in parallel versus the benefits of assigning multiple users to each channel to cooperatively sense the same channel. We propose several structured sensing strategies to maximize system throughput, and we investigate the tradeoff among these strategies under various circumstances. Finally, we discuss the robust design when the proposed sensing strategies encounter uncertainty in PU channel occupancy and detection signal-to-noise ratio (SNR).

5.1.1 Related Work

Sensing strategies have so far been mostly investigated in what relates to sensing order optimization and acquiring the stopping time in a sequential manner where channels are sensed one after the other. A multi-user network is investigated in [AFS13] where channels are being sensed in parallel, but the only parameter used for the decision making is channel occupancy without considering the impact of cooperative sensing. In [DRW09], the authors proposed a scheduling scheme for spectrum sensing based on the idea that when a channel is free, the channel can be sensed with a lower time resolution set based on a backoff scheme. [RG08] proposed a robust routing schedule to maximize the social network utility subject to the variance constraint. [LWC14] proposed an online decision scheduling algorithm to determine the sensing period together with a sequential detection for spectrum sensing, which is robust to short-term channel change and possible data outliers.

5.1.2 Our Contribution

The contribution of this work is thus threefold:

1. We introduce and formulate the general problem of sensing strategy for optimal sensing allocation of SUs to maximize system throughput. However, due to implementation and analysis complexity, the general problem is not solved. This is one of the limitation of this work;

2. Three classes of structured sensing strategies, i.e., sequential[1], parallel, and sequential-parallel multi-channel sensing strategies are proposed, resolved with optimal and heuristic algorithms, and compared in presence of homogeneous and heterogeneous sensors;

3. A robust optimization for the proposed strategies is provided to investigate how the sensing strategy decision is affected when there is uncertainty for the detected PU SNR and channel occupancy.

5.2 System Model and Problem Formulation

We consider a DSA network with N SUs transceiver pairs and M channels as shown in Fig. 5.1. Similar to most of the works in the literature, channels and PUs' activity in the channels are assumed to be fully independent [KS08a]. PUs are assumed to transmit

[1]The optimal sequential strategy has been proposed in the previous chapter. But in the previous chapter, we assume arbitrary sensing time. Here we consider the practical physical layer sensing method to obtain the sensing time considering user cooperation.

synchronously on the channels in a time-slotted fashion with a slot duration equal to T. Note that the time slot length in our work is the period during which the channel and the traffic statistics can be considered almost invariant. At the beginning of each time slot, the SU central network controller determines the sensing strategy for SUs to maximize the total expected spectrum opportunities for transmission. A spectrum sensing strategy includes the time schedule (e.g., sensing order) and job schedule (which users sense which channels). After the users finish sensing a channel, sensing results are sent to the central controller where they will be merged to make the final scheduling decision. Since our work mostly focuses on the sensing scheduling aspects of the problem, the transmission delay of sensing results to the controller is not considered[2].

The channel gain between the i-th SU transceiver pair operating on the m-th channel is denoted as $h_{m,i}$, and the channel gain from PU transmitter to the i-th SU receiver operating on the m-th channel is denoted as $g_{m,i}$. We thus define the i-th SU transmission capacity on the m-th channel as $C_{m,i} = B_m \log_2(1 + \Gamma_{m,i})$, where B_m is the bandwidth of the m-th channel, $1 \leq m \leq M$, $\Gamma_{m,i} = \frac{P_i h_{m,i}^2}{\sigma_m^2}$ is the received SNR for the i-th SU on the m-th channel, P_i is the transmission power for the i-th SU, and σ_m^2 is the noise power on the m-th channel. For simplicity, we assume $\Gamma_{m,i}$ are the same for all SUs, which will be reduced to Γ_m and hence C_m. For heterogeneous sensors, we define the corresponding detection SNR as $\gamma_{m,i} = \frac{\Psi_m g_{m,i}^2}{\sigma_m^2}$, where Ψ_m is the transmission power of the primary user at channel m and $g_{m,i}$ is the channel gain from PU to the i-th SU receiver on the m-th channel. Note that for simplicity we assumed channel sensing is performed at the SU receiver node. For the homogeneous case, the detection SNR is denoted by γ. The probability for the m-th channel of being occupied by primary users is assumed to be known at the central controller as u_m, where u_m can be estimated or measured efficiently [LC13] in the training phase, as will be discussed in Section 5.5.

[2]In the literature, some works addressed this issue. For instance, constraints on the number of reporting sensors is discussed in [HSN12].

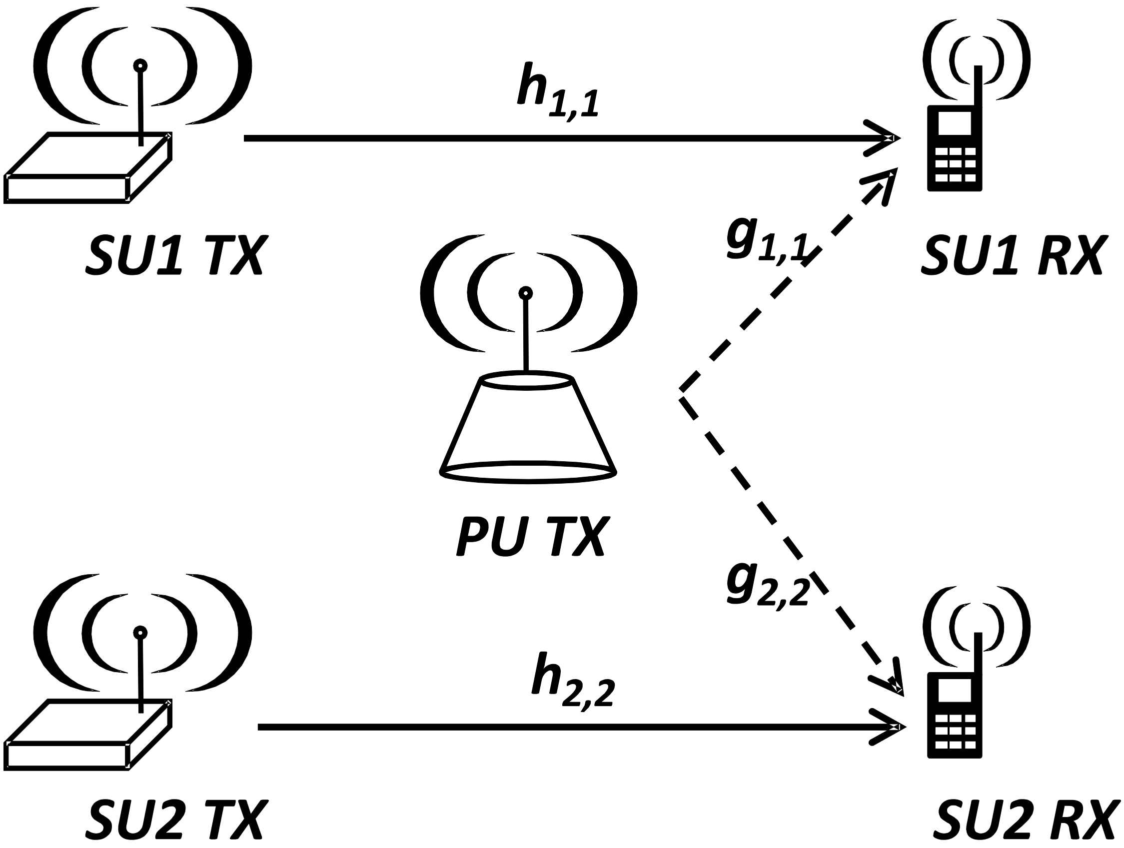


Figure 5.1: The network model with a primary user transmitter (TX), two SU TXs, and two SU receivers (RXs).

5.2.1 Cooperative Spectrum Sensing

Cooperative sensing is a well-known solution to enhance sensing performance [GS05]. The reason is that as the collective decision is made with several individual sensing results, the requirement of sensing accuracy for each individual user can be lowered, hence the sensing time can be reduced. In a time slotted DSA network, since the status of the channel does not change during one time slot, minimizing the sensing time for the channel implies increasing the expected transmission throughput [LZP08] for SUs.

5.2.1.1 Primary User Detection

Consider a secondary receiver that needs to detect primary users sending pilot signals on a particular channel [CTB06]. Let τ be the sensing time and assume that the receiver's sampling frequency is f_s such that $N_s = \tau \times f_s$ samples are gathered to make the decision of whether a channel is occupied by a primary user. The minimum sensing time required to satisfy the given detection quality under additive white Gaussian noise (AWGN) channel by the optimal detector, i.e., the matched filter is equal to:

$$\tau = \frac{\left[Q^{-1}(P_f) - Q^{-1}(P_d)\right]^2}{\gamma f_s}, \tag{5.1}$$

where γ is the detected SNR and P_d is the probability of detection, defined as the probability of detecting the primary user when it is present. P_f is the probability of false-alarm defined as the probability of wrongly finding the channel occupied when it is actually vacant. Note that even though we consider only AWGN channels, the discussion can be extended to any detection model as long as P_d and P_f are represented as a function of SNR. For example, P_d and P_f as a function of fading parameters can be found in [LZP08, Section V]. In addition, we choose the sensing sampling frequency as the Nyquist frequency which equals to two times the corresponding channel bandwidth in our simulations.

5.2.1.2 Fusion Rules

Sensing results reported by different users may be combined in different manners, known as fusion rules [GS05]. In what follows, we discuss *OR* and *AND* hard fusion rules because they are commonly used in the literature and also they provide bounds for the more general rule *k-out-of-N*. Assume all N users are homogeneous, i.e., they have the same P_d and P_f. Thus, the cumulative probability of detection and false-alarm are given as $Q_d = 1-(1-P_d)^N$, and $Q_f = 1-(1-P_f)^N$ for the *OR* fusion rule respectively, and as $Q_d = P_d^N$, and $Q_f = P_f^N$ for the *AND* fusion rule respectively.

From equation (5.1), the minimum cooperative sensing time by N homogeneous users to satisfy the Q_d and Q_f is expressed as

$$\tau = \begin{cases} \frac{\left[Q^{-1}(1-\sqrt[N]{1-Q_f})-Q^{-1}(1-\sqrt[N]{1-Q_d})\right]^2}{\gamma f_s} & \text{for the } OR \text{ Rule,} \\ \frac{\left[Q^{-1}(\sqrt[N]{Q_f})-Q^{-1}(\sqrt[N]{Q_d})\right]^2}{\gamma f_s} & \text{for the } AND \text{ Rule.} \end{cases} \tag{5.2}$$

Throughout the chapter, we also define $\tau_{m,n}$ as the cooperative sensing time of channel m by n sensors. Equation (5.2) provides two important insights. First, for any channel m (we thus drop the channel index), τ_n is a decreasing function of n. The other insight is related to the number of cooperative sensors. As illustrated in Fig. 5.2, the sensing time gain $\tau_n - \tau_{n+1}$, i.e., adding another user to the process of cooperative sensing, decreases when n increases. The most improvement in cooperative sensing time is therefore obtained when two users cooperate instead of sensing a channel by one user. This behavior promotes the idea of distributing the users more evenly among channels. We will use this result in Section 5.3.3. Note that in the discussions above, it is assumed that the detection SNR γ is given and the same for all users. The case with heterogeneous sensors (different detection SNRs thus P_d and P_f for different users) and the case with a random SNR will be investigated in Section 5.4 and 5.5, respectively.

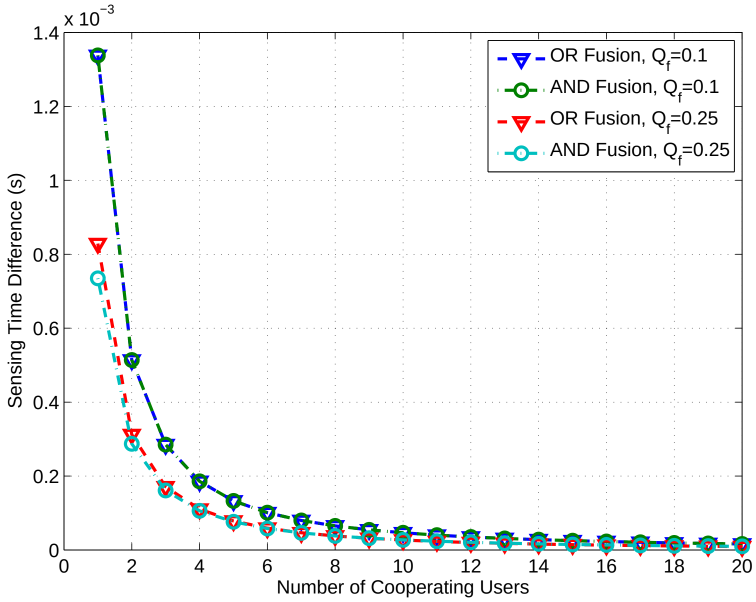


Figure 5.2: The gain of incorporating more users into the cooperative sensing process decreases when the number of users increases. The used parameters are $f_s = 5$ kHz, $Q_d = 0.9$, and SNR is -5 dB.

5.2.2 Problem Formulation

We define the beginning of a time slot as the reference point $t = 0$, and the elapsed time when the sensing process for channel i is finished as $T_I^{(i)}$. Note that $T_I^{(i)}$ depends on the sensing order and user allocation schemes, and the sensing time for channel i depends on the number of users allocated to it. As illustrated in Fig. 5.3(a), if channel i is found available, it is a potential spectrum opportunity with duration $T - T_I^{(i)}$. The expected throughput obtainable from the spectrum opportunity of channel i is thus equal to $C_i(1 - u_i)(T - T_I^{(i)})$. The elapsed time for a channel which is not sensed can be assumed to be T (no throughput gain). Our objective is to maximize the total expected normalized throughput R from all channel spectrum opportunities[3] by deciding the optimal sensing strategy, i.e.,

$$\max_{\mathcal{A}} \mathbb{E}\{R(\mathcal{A})\} = \sum_{i=1}^{M} \frac{(T - T_I^{(i)}(\mathcal{A}))C_i(1-u_i)}{T}, \tag{5.3}$$

where $\mathbb{E}\{\cdot\}$ is the expectation operation and $\mathcal{A}$ is a sensing strategy. Note for any channel i, $T_I^{(i)}(.)$ is a function of $\mathcal{A}$.

5.3 Spectrum Sensing Strategies with Homogeneous Sensors

A sensing strategy determines the order in which the channels are sensed and the number of users which sense a channel. In addition, the sensing strategy should also provide the timing schedule for each user to sense different channels. The optimal sensing strategy, which is called *general strategy* in this work, includes any possible strategy to sense a set of channels. For instance, consider the scenario in Fig. 5.3(b) with 4 channels and 3 users. Channel 1, 2, and 4 are sensed respectively by users 1, 2, and 3 starting from the beginning of the slot. To sense channel 3, there are 3 possibilities: i) User 1 solely senses channel 3 when it finishes its job sooner than user 2 and 3. Hence sensing channel 3 is finished at $T_I^{(3)} = \tau_{1,1} + \tau_{3,1}$; ii) User 1 waits for user 3 to finish its job and then they cooperatively sense channel 3 and $T_I^{(3)} = \tau_{4,1} + \tau_{3,2}$; iii) Both users 1 and 3 wait also for user 2 and then sense channel 3 cooperatively and $T_I^{(3)} = \tau_{2,1} + \tau_{3,3}$. As shown in the figure, it is assumed that option (ii) is the optimal solution. However, due to the large number of possible solutions, solving for the general strategy is highly cumbersome and can not be done efficiently in a timely manner, and it is also difficult to be implemented in practice. Therefore, in this section, to simplify the general sensing strategy, we propose three classes of multi-channel sensing strategies with particular structures. Each strategy can be considered as a sub-optimal scheme for the general strategy. In other words, we assume specific sensing strategies and,

[3]In this work, our focus is on the spectrum opportunity detection part and our objective is thus to maximize the potential throughput for other transmitting users which do not participate in the sensing process. The transmission scheduling problem where users participate both in sensing and transmission is out of the scope of this work and remains as our future work. The potential throughput in (5.3) thus represents an upper bound on the actual network throughput when joint sensing and transmission assignment of users is taken into account.

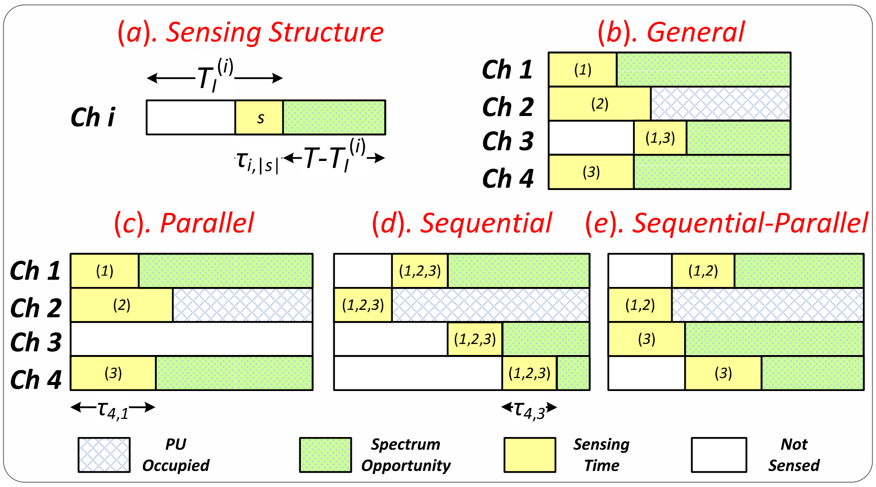


Figure 5.3: (a) Sensing structure when channel i is sensed by a subset s with $|s|$ users. (b) A general sensing strategy where user 1 waits for user 3 to finish its job and then they sense channel 3 cooperatively. (c) An example of parallel strategy where channel 3 is not sensed. (d) An example of sequential strategy with the channel (Ch) sensing order (Ch 2,Ch 1,Ch 3,Ch 4) by all users. (e) Sequential-parallel strategy where (Ch 2, Ch 1) are sensed sequentially by users 1 and 2 and in parallel for (Ch 3, Ch 4) by user 3.

given this strategy, we provide the optimal answer. We assume in this section that we have homogeneous sensors with the same detected SNR γ.

The three proposed strategies are: i) a *sequential strategy* where all channels, which can be sensed in T, are sensed cooperatively by all N users in a sequential manner, ii) a *parallel strategy* where channels are cooperatively sensed in parallel with a subset of users, and iii) a mixture of sequential and parallel strategies called *sequential-parallel strategy* where different sets of channels are sensed in parallel, but channels in each set are sensed in a sequential manner. An example for each strategy is provided in Fig. 5.3.

5.3.1 Sequential Sensing Strategy

The sequential strategy with cooperative sensing was first discussed in Chapter 4. We briefly review this strategy and show an example in Fig. 5.3(d). Given a list of users and channels, it is assumed that all users cooperatively sense each channel, and channels are thus sensed one by one. Therefore, the cooperative sensing time of any channel m by N sensors is given by $\tau_{m,N}$. The sensing order is defined as $\mathcal{A} = (a_1, a_2, \ldots, a_M)$ which is a permutation of $(1, 2, \ldots, M)$, e.g., $a_1 = 3$ implies that the first channel being sensed cooperatively by all users is channel 3. The expected spectrum opportunity throughput can then be re-written from equation (5.3) as

$$\max_{\mathcal{A}} \mathbb{E}\{R(\mathcal{A})\} = \sum_{i=1}^{M} \frac{(T - T_I^{(a_i)})C_{a_i}(1 - u_{a_i})}{T}, \tag{5.4}$$

where $T_I^{(a_i)} = \min\left\{\sum_{j=1}^{i} \tau_{a_j,N}, T\right\}$. In the previous chapter, it is proved that the optimal sensing order is found by sorting the channels in decreasing order of $\frac{C_j(1-u_j)}{\tau_{j,N}}, j = 1, \ldots, M$.

5.3.2 Parallel Sensing Strategy

In this strategy, channels are sensed in parallel and the central controller makes the decision on the number of users who should sense each channel. Intuitively, when no user is assigned to a channel, the channel is not sensed and no spectrum opportunity throughput is available for this channel. Each user thus senses only one channel. An example of this strategy is illustrated in Fig. 5.3(c). The optimization problem can be represented as

$$\max_{\mathcal{A}} \mathbb{E}\{R(\mathcal{A})\} = \sum_{i=1}^{M} \frac{(T - \tau_{i,k_i})C_i(1 - u_i)}{T}, \text{ s.t. } \sum_{i=1}^{M} k_i = N, \tag{5.5}$$

where k_i is the number of users assigned to channel i and $\mathcal{A} = \mathbf{k} = (k_1, k_2, \ldots, k_M)$. This is a classical integer programming problem. In the following, we first discuss a dynamic programming (DP) solution, and then a heuristic solution. At the end, the condition to have an integer assignment is relaxed and a relaxed optimization problem is discussed.

5.3.2.1 Dynamic Programming

As a resource allocation problem, we propose the following dynamic programming (DP) solution to find the optimal assignment [Ber05]. The *stage* of the DP is the channel number. Thus, starting from channel 1, we must decide at each stage, how many of the remaining users should be assigned to the particular channel considered. The decision variable is the number of users, the instantaneous payoff is the throughput which may be obtained from this channel, and the value function $v_k(n)$ is the total expected throughput which can be obtained from the optimal assignment from now on when k channels and n users remain. Transition possibilities naturally depend on the remaining number of users. Then, the Bellman equation can be written as

$$v_k(n) = \max_{0 \le j \le n} \left\{ \frac{(T - \tau_{k,j}) C_k (1 - u_k)}{T} + v_{k-1}(n - j) \right\}. \tag{5.6}$$

The terminal condition is when no users remain to be assigned, i.e., $v_k(0) = 0$, $\forall k$. We thus have $v_1(n) = \frac{(T - \tau_{M,n}) C_M (1 - u_M)}{T}$, meaning that in the last stage, any remaining users should be assigned to the last channel (channel M). The DP is finite, so it is solved by backward induction, and the maximal throughput is equal to $v_M(N)$. Note that since the channels are sensed in parallel, sensing order and the order of channels in the DP are irrelevant.

In the proposed DP solution, we choose the users as resource to be assigned since the DP has a lower runtime complexity compared to the case where channels are assigned. Consider an optimal assignment $\mathcal{A}^o = (k_1, \ldots, k_M)$ with a given number of users. Assume one new user is assigned to channel i to achieve an optimal allocation; for any other channel j, we should thus have the condition $\mathbb{E}\{R(k_1, \ldots, k_i + 1, \ldots, k_M)\} \ge \mathbb{E}\{R(k_1, \ldots, k_j + 1, \ldots, k_M)\}$, $\forall j \ne i$, and it can be simplified as $(\tau_{i,k_i} - \tau_{i,k_i+1}) C_i (1 - u_i) \ge (\tau_{i,k_j} - \tau_{i,k_j+1}) C_j (1 - u_j)$. Therefore, each new user is added to a channel i with currently k_i assigned users which has the highest $(\tau_{i,k_i} - \tau_{i,k_i+1}) C_i (1 - u_i)$ value. The DP algorithm for the parallel sensing strategy is presented in Algorithm 4.

5.3.2.2 Greedy Heuristic

The high execution complexity of the DP solution prompts the need to have a low-complexity heuristic. A simple, yet efficient solution is a greedy heuristic that puts more users on a channel with a higher product of the channel capacity C_i and the probability of availability $1 - u_i$. We thus propose

$$k_i = \left[N \frac{C_i (1 - u_i)}{\sum_{j=1}^{M} C_j (1 - u_j)} \right], \tag{5.7}$$

Algorithm 4 Pseudo Algorithm for Parallel DP Solution

procedure PARALLEL DP

    for $m = 1 : M$

    for $n = 1 : N$

    $R(m, n) = (T - \tau_{m,n})C_m(1 - u_m)$

    end for

    end for

    $R(:, 0) = 0$

    $\mathcal{A} = 0$

    while $N > 0$ **do**

        $m^* = \arg\max_m \Delta R = R(m, k_m + 1) - R(m, k_m)$, $1 \leq m \leq M$

        $k_{m^*} = k_{m^*} + 1$

        $N = N - 1$

    end while

end procedure

where $[\cdot]$ is the rounding operation. Since the sum of k_i values derived from equation (5.7) is not necessarily N, if $N - \sum_{j=1}^{M} k_j > 0$ the remaining $N - \sum_{j=1}^{M} k_j$ users are assigned to the channel with the maximum $C_j(1 - u_j)$, otherwise $\sum_{j=1}^{M} k_j - N$ additional users are eliminated from the channels starting with maximum $C_j(1 - u_j)$. We refer to this heuristic in the figures as "Par-GH (An.)" and "Par-GH (Sim.)".

5.3.2.3 Constraint Relaxation

In this section, we propose to relax the constraint of the optimization problem in equation (5.5) where k_i is not necessary an integer. This helps us to derive a bound for the parallel strategy. It can be easily shown that the objective function $\mathbb{E}\{R(\mathcal{A})\}$ is not a simple concave function, yielding to a non-convex optimization programming solution that, given the reduced size, it can still be optimally solved by brute-force search. The detailed derivation for relaxed optimal k_i are provided in Appendix A.11.

5.3.3 Analytical Comparisons for Sequential and Parallel Strategies with Homogeneous Channels

In this section, we compare the analytical throughput performance for the sequential and parallel strategies assuming that all channels have the same capacity C and channel occupancy rate u, i.e., the channels are homogeneous. It is complex to analytically derive the throughput performance for heterogeneous channels since it depends on multiple channel capacities and occupancy rates. We therefore only focus our effort on obtaining analytical results for the homogeneous case to gain a better insight on the conditions, e.g., number of channels, number of users, capacity, and occupancy rate, which make one scheme better than the other, as fewer variables are involved. Let us start with the parallel strategy.

In the parallel scheme, it can be observed that in practical scenarios, it is always better to sense more channels than to cooperatively sense fewer channels. For the case of similar channels, assume there are two channels and two users ($M = N = 2$). The throughput when each user senses a channel (no sensing cooperation) can be given by $2C(1-u)(T-\tau_1)$ (channel index was dropped). The throughput of cooperatively sensing only one channel is given by $C(1-u)(T-\tau_2)$. Cooperatively sensing the same channel in the parallel strategy is thus optimal when $C(1-u)(T-\tau_2) > 2C(1-u)(T-\tau_1) \Rightarrow 2\tau_1 - \tau_2 > T$. Note this condition is rarely met, so it can be claimed that when channels are similar, it is better, for the parallel scheme, to distribute the users as much as possible to sense and exploit more channels. Given this insight and based on what we observed in Fig. 5.2, the total throughput of the parallel scheme is represented as

$$\mathbb{E}\{R_{\text{Hom}}^{\text{Par}}\} = \frac{(M-r)[C(1-u)(T-\tau_L)] + r[C(1-u)(T-\tau_{L+1})]}{T}, \tag{5.8}$$

where $L = \lfloor \frac{N}{M} \rfloor$, $\lfloor \cdot \rfloor$ is the floor operator, and r is the reminder of the division, i.e., $r = \text{mod}(N, M)$. The intuition is as follows. We should first assign L users to each channel and the remaining r users are distributed among r channels, so r channels will be sensed by $L+1$ users and the others by L users. We can also derive the analytical throughput performance of the parallel sensing greedy heuristic algorithm for homogeneous channels. Since channels are assumed similar, users are evenly assigned to channels and the remaining users are assigned to one of the channels. Defining $Q = [\frac{N}{M}]$, the throughput obtained by this heuristic can thus be given by:

$$\mathbb{E}\{R_{Hom}^{Par-GH}\} = \begin{cases} \frac{N(T-\tau_1)C(1-u)}{T} & \text{if } N < M, \\ \frac{M(T-\tau_Q)C(1-u)}{T} & \text{if } MQ = N \text{ and } N \geq M, \\ \frac{(M-1)(T-\tau_Q)C(1-u)}{T} & \\ +\frac{(T-\tau_{N-(M-1)Q})C(1-u)}{T} & \text{if } MQ < N \text{ and } N \geq M, \\ \frac{(M-t)(T-\tau_Q)C(1-u)}{T} & \\ +\frac{(T-\tau_{Q-MQ+N-(t-1)Q})C(1-u)}{T} & \text{if } MQ > N, N+(t-1)Q < MQ, \\ & MQ \leq N + tQ, \text{ where } t \in \mathbb{Z}^+, \\ & \text{and } N \geq M. \end{cases} \tag{5.9}$$

For the sequential model, the sensing time of each channel is τ_N, so at most $\lfloor \frac{T}{\tau_N} \rfloor$ channels can be sensed. Let us define $K = \min\{\lfloor \frac{T}{\tau_N} \rfloor, M\}$. The total throughput is thus given by

$$\mathbb{E}\{R_{\text{Hom}}^{\text{Seq}}\} = \frac{C(1-u)\sum_{i=1}^{K}(T - i\tau_N)}{T} = \frac{KC(1-u)\left(T - \frac{K+1}{2}\tau_N\right)}{T} \tag{5.10}$$

Using equations (5.8) and (5.10), we are able to find the operating regions where one of the strategies outperforms the other, as will be illustrated in Section 5.6.

5.3.4 Sequential-Parallel Strategy

We propose in this section a hybrid strategy named sequential-parallel. As can be seen in the example provided in Fig. 5.3(e), channels are divided into several subsets, where within each channel subset, a subset of users are adopting sequential cooperative sensing. In other words, within each channel subset, a sequential strategy is followed while different channel subsets are sensed in parallel. The decision to be made is thus to find the channel subsets, the assignment of users to each subset and the sequential sensing order within each subset.

We define a function $R_s(\mathcal{S}_m, n)$ which is the maximum expected throughput obtainable from sequentially and cooperatively sensing by n users the channel subset $\mathcal{S}_m$. From Section 5.3.1, we already have the optimal sequential strategy within one channel subset. With this type of structure, the throughput maximization problem is indeed a Knapsack problem [MT90] where we are looking for the best 2-tuples $(\mathcal{S}_m, n)$ to put in the knapsack. In the following, a dynamic programming model and a greedy heuristic are proposed to solve this problem.

5.3.4.1 Dynamic Programming

Given the function R_s, the state variable in the DP equation is represented by $(\mathcal{S}, n)$, where $\mathcal{S}$ is a subset of channels, not sensed yet, and n is the number of remaining users, not assigned to any channel set. The decision is one of the subsets of $\mathcal{S}$ and the number of users assigned to it. Therefore, the total number of possible actions is equal to $2^{|\mathcal{S}|}(n+1)$. The Bellman equation can be given by

$$v(\mathcal{S}, n) = \max_{0 \leq j \leq n, \mathcal{X} \subseteq \mathcal{S}} \left\{ R_s(\mathcal{X}, j) + v(\mathcal{S} - \mathcal{X}, n - j) \right\}, \tag{5.11}$$

where $\mathcal{X}$ is the decision variable which is a subset of $\mathcal{S}$. The DP model is of infinite-horizon, so it can be solved by value iteration [Ber05]. As soon as we reach any state with $v(\emptyset, n)$ or $v(\mathcal{S}, 0)$, the ongoing payoff is zero and the solution is terminated.

5.3.4.2 Greedy Heuristic

Similar to the classical Knapsack problem, the greedy approach starts with the 2-tuple whose ratio of throughput versus the number of users is maximum. When a channel subset and the number of users assigned to this subset are decided, the algorithm is continued for the remaining users and channels. The greedy algorithm can be found in Algorithm 5. As discussed in [MT90], the greedy heuristic is guaranteed to have a performance higher than half of the optimal result.

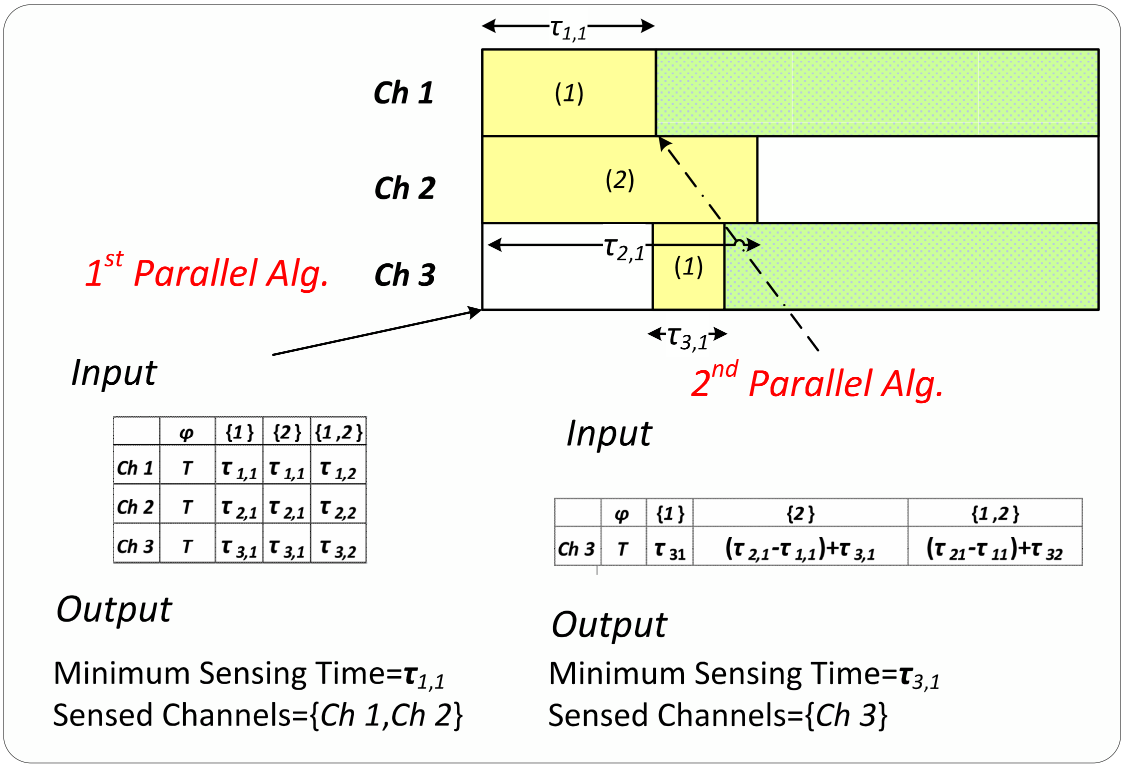
Algorithm 5 Pseudo Algorithm for Sequential-Parallel Greedy Heuristic Solution

```
procedure SEQUENTIAL-PARALLEL GREEDY HEURISTIC
    while N > 0 do
        Select 2-tuple (S*, n*) with maximum R_s(S_m, n)/n
        Remove all entries (S_m, n) if S_m ∩ S* ≠ ∅
        Remove all entries (S_m, n) if n > N − n*
        N = N − n*
    end while
end procedure
```

5.3.5 Iterative-Parallel Solution for the General Spectrum Sensing Strategy

The solutions provided for the parallel sensing strategy raise the idea of using an iterative solution for the general model. Let us call the solution algorithm proposed for the parallel strategy *ParallelStrategy*($\cdot$), which could be any of the proposed solutions. In the first iteration of the iterative approach for the general model, a decision based on the parallel strategy is made to sense some channels in parallel. Those channels are removed from the list and for the remaining channels, a new decision is made based on the parallel strategy. Iterations are continued until all channels are sensed (or until the end of time slot). However, we observe that in the parallel strategy users are all synchronized, while in the iterative solution, sensing time of the channels may be different, and hence users finish their first assigned job in different time instants.

To be able to employ the parallel strategy iteratively, we use the following approach, which is illustrated with an example in Fig. 5.4. The parallel strategy has been called in the beginning of the time slot and the optimal decision is to sense channel 1 by user one and channel 2 by user two. Now, consider the point that sensing the first channel (shortest sensing time) is finished and user one becomes idle. We call this point the new reference point where a new decision is made. The remaining sensing time of channel 2 is known, so if a new job (sensing channel 3) is assigned to a designated user who is currently busy, we have to wait first for this user to finish its current job which takes $(\tau_{21} - \tau_{11})$ time unit, and



	φ	{1}	{2}	{1,2}
Ch 1	T	$\tau_{1,1}$	$\tau_{1,1}$	$\tau_{1,2}$
Ch 2	T	$\tau_{2,1}$	$\tau_{2,1}$	$\tau_{2,2}$
Ch 3	T	$\tau_{3,1}$	$\tau_{3,1}$	$\tau_{3,2}$

	φ	{1}	{2}	{1,2}
Ch 3	T	τ_{31}	$(\tau_{2,1}-\tau_{1,1})+\tau_{3,1}$	$(\tau_{21}-\tau_{11})+\tau_{32}$

Figure 5.4: An example of the proposed Iterative-Parallel solution for the general sensing model.

then the user starts the new job. It is equivalent to assume that the sensing time of any remaining channel, here channel 3, by this designated user, starting at new reference point, is the sum of remaining sensing time of the channel being sensed by this designated user and the original sensing time of the remaining channel, which is equal to $(\tau_{21} - \tau_{11}) + \tau_{31}$. Similarly, the cooperative sensing time of channel 3 is updated to $(\tau_{21} - \tau_{11}) + \tau_{32}$ because user one should wait for user two to finish and then join for a cooperative sensing. For cooperative sensing, as discussed in Section 5.2.1, all collaborators should start at the same time, so the updated sensing time for a channel is defined based on the longest remaining job. As described in Algorithm 6, in each iteration, the list of remaining channels is updated and based on the remaining job of the users, the table of all sensing times by different subsets of users is recalculated. By modifying the length of the time slot in each iteration (as throughput function is linear versus the time slot), the reference point is redefined. Note that $\tau_s^{\square}$ is a matrix with $M \times 2^N$ entries which keeps the sensing time of each channel sensed cooperatively by a subset of users, as discussed in Section 5.2.1.2. After each iteration, some channels remain which are still not sensed. Therefore, a new decision is made only considering the remaining channels. It is worth noting that this algorithm is run offline in the beginning of the time slot (similar to a DP) to find the optimal strategy, then the strategy is followed and applied to the time slot. It is clear that since we are running the parallel algorithm, we are maximizing the instantaneous payoff, so the proposed solution is a myopic solution and not necessarily optimal.

5.3.6 Memory Usage and Computational Complexity Discussion

It is not possible to solve the throughput optimization problem in the general sensing strategy in polynomial time, since all permutations of M channels along with all ways to divide N users among M channels need to be considered. In terms of the memory space complexity, the maximum memory space required for it is $\mathcal{O}(2^M 2^N)$ to keep the sensing time of any subset of channels by any subset of users, where $\mathcal{O}(\cdot)$ is the big O notation. For the other proposed strategies, the memory space and computational complexity are summarized in Table 5.1. Note that for the sequential-parallel strategy, even though their big O complexity

Algorithm 6 Pseudo Algorithm for Iterative-Parallel Solution

```
procedure ITERATIVE-PARALLEL SOLUTION
    Function IterativeParallel(N, M, τ_s^[], T).
    CH = 1 : M
    while IsNotEmpty(CH) & T > 0 do
        SensingSchedule=ParallelStrategy(N,CH,τ_s^[],T).
        CH = CH-SensedChans (SensingSchedule).
        T = T-MinSensingTime (SensingSchedule).
        τ^M=UPDATE(T,τ_s^[],SensingSchedule).
        Solution=[Solution SensingSchedule].
    end while
end procedure
```

is the same, DP has more lower order computation terms than the heuristic (e.g., $3n^3+2n^2+n$ versus n^3 while both are $\mathcal{O}(n^3)$). In addition, in calculating the computational complexity, the execution time to fill the required data structures, already taken into account in the memory usage, is not considered to avoid repetition.

Table 5.1: Memory space and computational complexity of homogeneous sensing strategies.

Strategy	Memory	Computation
Sequential	$\mathcal{O}(M)$	$\mathcal{O}(M^2)$
Parallel-DP	$\mathcal{O}(MN)$	$\mathcal{O}(MN^2)/\mathcal{O}(M^2N)$
Parallel-Heuristic	$\mathcal{O}(1)$	$\mathcal{O}(M)$
Sequential-Parallel-DP	$\mathcal{O}(2^MN)$	$\mathcal{O}(2^{2M}N^2)$
Sequential-Parallel-Heuristic	$\mathcal{O}(2^MN)$	$\mathcal{O}(2^{2M}N^2)$
Iterative-Parallel-DP	$\mathcal{O}(MN)$	$\mathcal{O}(M^2N^2)$

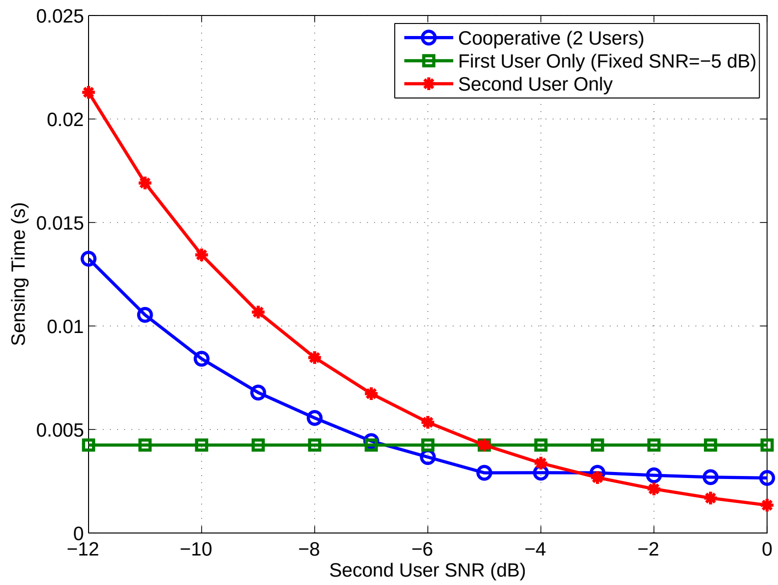


Figure 5.5: Cooperative sensing time by *AND* fusion rule versus single-user sensing to satisfy aggregated sensing accuracy of $Q_d^* = 0.9$ and $Q_f^* = 0.15$. Other parameters are $f_s = 4$ kHz, noise power $\sigma^2 = 10^{-5}$ W and the detection SNR for the first user $\gamma_1 = -5$ dB.

5.4 Spectrum Sensing Strategies with Heterogeneous Sensors

In the previous section, it was assumed that the users are homogeneous in sensing with the same detection SNR γ for all channels and all users. Due to different distances between the SUs and PU transmitter, as well as channel variations, sensors in practice may have different detection SNRs and consequently different sensing times to fulfill a given sensing accuracy. In this section, the optimal sensing strategy is therefore investigated assuming different detection SNR values $\gamma_{m,i}$ for SU i in channel m. In this work we assume a more realistic scenario where the cooperative sensing time is the same for all participating users and sensing reports are sent to the controller at the same time. We do not address the scenarios where the sensing duration is different for different users and the fusion center has to wait to receive the last report to make its decision. Considering this assumption, a key observation is that, depending on the fusion rule, the cooperative sensing time to meet target Q_f^* and Q_d^* values may even increase with additional users if a user with a significantly lower detection SNR cooperates. Fig. 5.5 illustrates this fact. Only between the crossover points at -7 dB and -3 dB should the users cooperatively sense the channel to achieve the lowest sensing time. Below a -7 dB SNR for the second user, the channel should be sensed only by the first user, which has a significantly better SNR than the second user. Similarly, when the second user SNR is above -3 dB, only the second user should sense the channel for the best performance.

For each channel m and within a set of sensors $\mathcal{S}$, with known SNRs, we can thus find the optimal subset of users $\mathcal{S}_m^* \subseteq \mathcal{S}$ to cooperatively sense this channel. The cooperative sensing time for a channel m will be the minimum sensing time such that the target sensing accuracy is satisfied. Let us call it τ_m^*. In the following, we discuss how τ_m^* is found for a channel m. It should be noted that finding τ_m^* is just an initial step in the scheduling optimization procedure and we then come back in Section 5.4.1 to the original optimization problem, which is to maximize the potential throughput by providing solutions to our proposed scheduling strategies.

Consider a subset $\mathcal{S}$ of all N users with an SNR vector $(\gamma_{m,1}\gamma_{m,2}\dots\gamma_{m,N})$. If we assume that all users in $\mathcal{S}$ cooperatively sense channel m, the minimum cooperative sensing time $\tau_{m,\mathcal{S}}$ can be found by solving the following optimization model

$$\min_{\tau,\Upsilon} \tau, \text{ s.t. } Q_d \geq Q_d^*, \ Q_f \leq Q_f^*, \tag{5.12}$$

where τ is the cooperative sensing time and $\Upsilon = (\epsilon_1 \dots \epsilon_N)$ is the detection threshold of all users. For the matched filter detection, we have [CTB06] $P_{f,i} = Q(\frac{\epsilon_i}{\sqrt{\tau f_s \theta_i \sigma_w^2}})$ and $P_{d,i} = Q(\frac{\epsilon_i - \tau f_s \theta_i}{\sqrt{\tau f_s \theta_i \sigma_w^2}})$, where ϵ_i is the detection threshold, τ is the sensing time, θ_i is the PU signal power so that $\gamma_i = \frac{\theta_i}{\sigma_w^2}$, σ_w^2 is the noise power and f_s is the sampling frequency. Channel index has been dropped in all parameters above.

For *AND* fusion rule, we also have that $Q_f = \prod_{i\in\mathcal{S}}(P_{f,i})$ and $Q_d = \prod_{i\in\mathcal{S}}(P_{d,i})$, where $P_{f,i}$ and $P_{d,i}$, $1 \leq i \leq N$ are the probability of false alarm and detection for the i-th user in $\mathcal{S}$, respectively. Note they are both functions of τ and detection thresholds Υ. We thus have $|S|$ equations and $|S|+1$ variables (τ and ϵ_i), and we are looking for the minimum τ and a feasible Υ. The optimization problem above can be solved by a solver to find the minimum cooperative sensing time $\tau_{m,\mathcal{S}}$.

The sensing time found above is the cooperative sensing time when all users in a subset S cooperate. However, as we previously discussed, increasing the number of cooperating sensors does not necessarily improve the sensing time, and the optimal set of users can be a subset of the users. Therefore, the optimal sensing time for channel m can be given by $\tau_m^* = \min\limits_{\mathcal{S}\subseteq\{1,2,\cdots,N\}} \tau_{m,\mathcal{S}}$, where $\mathcal{S}$ is any subset of N users, and the cooperative sensing time for this subset is given by equation (5.12). The complexity of finding the best subset can be decreased from $O(2^N)$ to $O(N^2)$ by the fact that it is not required to check all subsets. The user with the best SNR should necessarily be one of the cooperating sensors, so if we sort the vector of SNRs in a descending order and rename the users accordingly, only the sensing time of N subsets $\{1\}, \{1,2\}, \ldots \{1,2,\ldots,N\}$ should be found and compared. Starting from $\{1\}$ towards larger subsets, we continue the search until the cooperative sensing time increases by adding the next user. We can see that this is equivalent to finding an SNR threshold and only have users who have an SNR greater than this threshold participate to the cooperative sensing.

Since the complexity of solving equation (5.12) depends on the solver and the algorithm used, we call it $E(N)$. From now on, we assume that the we know the optimal sensing time and the optimal sensing set for any channel, found with $E(N) + O(MN^2)$ complexity, and the cooperative sensing time for any subset of users found with an $E(N) + O(M2^N)$ complexity. Using those pieces of information, we now discuss how the sequential, parallel and sequential-parallel sensing strategies can be designed.

5.4.1 Sequential Sensing with Heterogeneous Sensors

As discussed, for any channel m, we could find the optimal set of users and the optimal sensing time τ_m^*. We use this optimal sensing time to find the optimal channel sensing order by sorting the channels in the decreasing value order of $\frac{C_m(1-u_m)}{\tau_m^*}$, and each channel is cooperatively sensed by its optimal set of users. Therefore, the only change compared to the case with homogeneous sensors is that a channel is not sensed by all users and a user will not thus necessarily sense all channels. The results will be called "Seq (Opt.)" in the numerical results figures.

As solving equation (5.12) and finding the optimal sensing time may be cumbersome for a large number of users, we propose the following average-based heuristic called "Seq-Heuristic-Avg" in the figures. For each channel m, we assume that the channel is sensed only by the users who have a detection SNR greater than the average SNR of all users. The rationale behind this heuristic is that, for a large number of users, there will also be a large number of users in the sensing set. As discussed previously, the cooperative sensing time gain is significant mostly for the first users. Therefore, discarding a few users will not have a major impact on the performance. We then assume that all users who have a SNR higher than the average have the same average SNR and we find the cooperative sensing time from equation (5.1) denoted by τ_m^{Avg}. Channels are sensed in descending order of $\frac{C_m(1-u_m)}{\tau_m^{\text{Avg}}}$.

5.4.2 Parallel Sensing with Heterogeneous Sensors

We saw that each channel has an optimal set of users to cooperatively sense that channel. If those sets were disjoint, it would be optimal to sense all channels in parallel, each one with its optimal set. Since the sets are not necessarily disjoint, we propose the following assignment strategies which follow similar concepts to the ones proposed for homogeneous sensors. Using the cooperative sensing time for any channel and by any set of users given from equation (5.12), we could have a dynamic programming model to find the optimal assignment, and a sub-optimal scheme with a lower complexity similar to the one proposed in Algorithm 4. Note that since the sensors are heterogeneous, the latter approach is not the solution of the DP (as it was for homogeneous sensors) and results in a sub-optimal assignment (i.e., the order of user assignment to channels is important here). We call the optimal and heuristic approaches respectively "Par-DP (Opt.)" and "Par-DPSim-HetSens" in the numerical results figures.

We also propose a classical greedy scheme, using the average sensing time introduced for sequential strategy. The channels are sorted in decreasing order of $\frac{C_m(1-u_m)}{\tau_m^{\text{Avg}}}$. The best channel is selected and the user with the best SNR for this channel is assigned to the channel and removed from the list of users. The algorithm is then continued for other channels. If one user is assigned to all channels and there are still unassigned users (i.e., $M < N$), the assignment is restarted from the first channel to add another sensor. This approach is called "Par-Avg-Greedy-HetSens" in the numerical results. As can be seen, the objective of this heuristic is to explore the most channels in parallel with their best user sets.

5.4.3 Sequential-Parallel Sensing with Heterogeneous Sensors

Similar to the Section 5.3.4 with homogeneous channels, we can find the expected throughput which can be obtained from a set of channels sensed cooperatively by a set of users in a selective manner. This is equivalent to using the sequential approach proposed above for any set of users and channels. Recall that due to heterogeneity in sensing, all users may not sense all channels. Forming a matrix with an $O(2^M 2^N)$ complexity, we could have a DP model to find the optimal assignment. It is called "Seq-Par-DP (Opt.)" in the numerical results figures. The complexity will however be too high. In the following, we thus propose a heuristic with a very low complexity.

In this heuristic, called "Seq-Par-Heuristic" in the numerical results figures, channels are sorted based on the obtainable throughput divided by the number of sensors, given the optimal sensing time. The first channel which is selected thus has the maximum $\frac{R_m(\tau_m^*)}{|\mathcal{S}_m^*|}$ where $R_m(\tau_m^*)$ is the throughput obtainable from channel m if it is sensed by its optimal set of users. After selecting the first channel, any other channel i whose optimal set of sensors is a subset of the optimal set of the selected channel (i.e., $\mathcal{S}_i^* \subseteq \mathcal{S}_m^*$) is also selected to be sensed by the same set of sensors in a sequential manner. All these channels are removed from the list and a new decision is made for the remaining channels and users. Naturally, the optimal set of users and optimal sensing time for all remaining channels must then be recalculated based on the remaining users.

Table 5.2: Memory space and computational complexity of heterogeneous sensing strategies.

Strategy	Memory	Computation
Sequential (Opt)	$\mathcal{O}(MNE(N))$	$\mathcal{O}(M^2)$
Seq-Heuristic-Avg	$\mathcal{O}(M)$	$\mathcal{O}(M^2)$
Parallel (DP,Opt)	$\mathcal{O}(M2^N E(N))$	$\mathcal{O}(M2^{2N})$
Parallel-DPSim-Het	$\mathcal{O}(M2^N E(N))$	$\mathcal{O}(MN^2)$
Parallel-Avg-Greedy	$\mathcal{O}(MN)$	$\mathcal{O}(M^2 N)$
Sequential-Parallel (DP,Opt)	$\mathcal{O}(2^M 2^N E(N))$	$\mathcal{O}(2^M 2^{2N})$
Sequential-Parallel-Heuristic	–	$\mathcal{O}(M^2 NE(N))$

5.4.4 Memory Usage and Computational Complexity Discussion with Heterogeneous Sensors

The complexity comparison of different solutions for the heterogeneous cases is summarized in Table 5.2. The main difference between homogeneous and heterogeneous cases from complexity point of view is the extra calculation for equation (5.12) for each channel, which is denoted by $E(N)$. In other words, after finding the optimal sensing time and optimal sets for each channel, the complexity is similar to homogeneous scenarios.

5.5 Robust Spectrum Sensing Scheduling Design

Observing equation (5.3), it can be seen that three system parameters, i.e., PUs duty cycle u, detection SNR γ, embedded in sensing time, and received SNR of SU transmission, embedded in capacity, have a random nature while in previous sections, we assumed a perfect knowledge of those parameters. A robust optimization, maximizing the throughput while keeping its variation below a threshold, can be provided for those parameters to take into account their random nature. Since in this work we focused on the sensing without taking the actual transmission into account, we continue to use the assumption of full knowledge of the channel capacity and discuss how the variation of the two other random parameters affect the decision made by different sensing strategies.

We now separately relax the assumptions that we have full knowledge for the duty cycle u and the detected SNR γ, i.e., we first analyze the case with imperfectly known duty cycle while the detected SNR is still assumed perfectly known, and then the inverse case is investigated. In the actual system, we need to estimate those parameters accurately so that the estimates would not degrade the system performance tremendously. In the following robust system design, we first propose a low-complexity average estimator to estimate the duty cycle u on a single channel, based on the discrete-time Markov chain (DTMC) assumption for PU transmission traffic, and then we analyze the statistics for the proposed estimator. Second, we formulate two uncertainties, i.e., primary traffic and channel uncertainty, and combine them with our proposed cooperative spectrum sensing scheduling schemes as a joint robust optimization problem. The primary traffic uncertainty comes from the estimation errors of the duty cycle, and the channel uncertainty comes from the detected SNR, which is not a constant but a random variable following a certain distribution.

5.5.1 Primary Traffic Estimation: Average Estimator and Its Performance Analysis

The PU traffic on a single channel is modeled as a DTMC [GLP13a], where z represents the current state of the PU (also termed as a PU traffic sample). States $z = 0$ and $z = 1$ indicate the PU is absent and present, respectively within one slot time T. This traffic is characterized by the steady-state distribution and the transition probabilities. The probability of PU absence is denoted as $P_0 = \Pr\{z = 0\} = 1 - u$ and of PU presence as $P_1 = \Pr\{z = 1\} = u$. The transition probabilities from state x to state y is denoted as P_{xy} with four probabilities $\{P_{00}, P_{01}, P_{10}, P_{11}\}$ with $u = \frac{P_{01}}{P_{01}+P_{10}}$, $P_{00} + P_{01} = 1$, and $P_{10} + P_{11} = 1$.

Assume we obtain W PU traffic samples to form a vector $\mathbf{z} = (z_1, z_2, \cdots, z_W)$, $z_i \in \{0, 1\}$, $1 \leq i \leq W$ under perfect sensing from a PU channel. Our goal is to estimate the duty cycle u using these observed traffic samples z_i. We adopt a low-complexity average estimator, i.e., $\hat{u} = \frac{1}{W} \sum_{i=1}^{W} z_i$ [GLP13a]. Its expected value is shown to be $\mathbb{E}\{\hat{u}\} = \frac{1}{W} \sum_{i=1}^{W} \mathbb{E}\{z_i\} = u$, and therefore it is an unbiased estimator. To derive its variance, we apply the results in [GLP13a, Eq. (15)] to obtain

$$\mathrm{Var}\{\hat{u}\} = \frac{u(1-u)}{W} + \frac{2u(1-u)r(r^W - Wr + W - 1)}{W^2(1-r)^2}, \tag{5.13}$$

where $r = P_{11} - P_{01}$[4]. Note that the asymptotic value for the variance of $\hat{u}$ is $\lim_{W\to\infty} \mathrm{Var}\{\hat{u}\} = 0$, which means that as we increase the number of PU traffic samples, the variance for the estimator will go to zero to make a perfect estimation.

5.5.2 Robust Optimization

In this section, we formulate the robust optimization problem regarding the primary traffic uncertainty and channel uncertainty, i.e., the estimation errors for duty cycle and statistical behavior from the detected SNR, respectively. Note that we relax one variable at a time and keep the other one fixed so that we can observe their effect separately.

5.5.2.1 Primary Traffic Uncertainty

By using the estimates of duty cycle $\hat{u}_i$ in the proposed system, where the channel activity is assumed independent among channels, with given traffic samples W_i, $1 \leq i \leq M$ for all channels, we first formulate the robust optimization problem by maximizing the expected estimated throughput, subject to the constraints that the variance for the estimated throughput should be no greater than a given threshold η [RG08], i.e.,

Robust Optimization Problem 1

$$\begin{aligned} &\max_{\mathcal{S}} \mathbb{E}\{\hat{R}(\mathcal{S})\} = \sum_{i=1}^{M} \frac{(T - T_I^{(i)}(\mathcal{S}))C_i(1 - \mathbb{E}\{\hat{u}_i\})}{T}, \\ &s.t.\ \mathcal{S} \in \mathcal{A},\ \ Var\{\hat{R}(\mathcal{S})\} = \sum_{i=1}^{M} \frac{(T - T_I^{(i)}(\mathcal{S}))^2 C_i^2\, Var\{\hat{u}_i\}}{T^2} \leq \eta, \end{aligned} \tag{5.14}$$

[4]Note that [GLP13a, Eq. (15)] is the variance for u assuming the traffic samples follow a continuous-time Markov chain (CTMC). However, if we constraint the CTMC by using uniform sampling, it would turn into the DTMC, where we can simply replace Γ_u in [GLP13a, Eq. (15)] with r in (5.13). The detailed derivation can be found in Appendix A.12

where $\hat{R}(\mathcal{S})$ is the estimated throughput and $\mathcal{S}$ is an element in the sensing strategy set $\mathcal{A}$. Since the estimator for the duty cycle is an unbiased estimator, i.e., $\mathbb{E}\{\hat{u}_i\} = u_i$, we have $\mathbb{E}\{\hat{R}(\mathcal{S})\} = R(\mathcal{S})$. In addition, we can obtain $\mathrm{Var}\{\hat{R}(\mathcal{S})\}$ by substituting the equation (5.13) into equation (5.14). To solve this robust optimization problem (ROP), we search for all the possible elements $\mathcal{S}$ in $\mathcal{A}$, sort the corresponding estimated throughput in a descending order, and then adopt this order to search for the variance constraint until we have the variance being less or equal to η. This optimal solution is summarized in Algorithm 7.

From a practical system design point of view, we do not have the information of how many traffic samples should we use in advance. Therefore, we need to have another optimization problem formulation to obtain an efficient system design. As observed in ROP 1, the objective function $\mathbb{E}\{\hat{R}(\mathcal{S})\}$ does not depend on W_i, which means that we can always achieve the optimal solution without considering the variance constraint. However, since $\mathrm{Var}\{\hat{R}(\mathcal{S})\}$ is a decreasing function in terms of W_i, as we keep increasing W_i, we can achieve any arbitrary variance constraint if we have sufficient traffic samples. Hence, we propose to obtain the minimum number of traffic samples if we have the variance constraint, or to obtain the minimum variance of estimated throughput if we have a sensing energy constraint, which are listed as two ROPs as follows respectively.

Robust Optimization Problem 2

$$\min_{W_i} \sum_{i=1}^{M} W_i, \;\; s.t. \;\; Var\{\hat{R}(\mathcal{S})\} \leq \eta, \tag{5.15}$$

and

Robust Optimization Problem 3

$$\min_{W_i} Var\{\hat{R}(\mathcal{S})\}, \;\; s.t. \;\; \sum_{i=1}^{M} W_i \leq \epsilon. \tag{5.16}$$

Here we show that the above two ROPs are equivalent by the following Lemma.

Lemma 3 *ROP 2 and ROP 3 are equivalent, i.e., the optimal solution for both problems are the same.*

Proof: See Appendix A.13.

From the above Lemma, we can solve either ROP 2 or ROP 3 to have an efficient system design by minimizing both sensing energy and variation for the throughput.

Algorithm 7 Pseudo Algorithm for Solving ROP 1

---

procedure ROP SOLUTION
    for $i = 1 : |\mathcal{A}|$
    $Q_i = \mathbb{E}\{\hat{R}(\mathcal{S}_i)\}$.
    end for
    Sort Q_i in a descending order with the corresponding strategy $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_{|\mathcal{A}|})$, $\mathcal{E}_i \in \mathcal{A}$.
    $i = 1$.
    while $\mathrm{Var}\{\hat{R}(\mathcal{E}_i)\} > \eta$ **do**
        if $i \geq M$ **then**
            Return no solution.
        else
            $i = i + 1$.
        end if
    end while
    Return optimal strategy $\mathcal{E}_i^*$.
end procedure

5.5.2.2 Channel Uncertainty for PU Detection

In this section, we relax the assumption that the detected SNR γ is a constant value for all SUs. Instead, we model γ as a random variable for all SUs and we would like to discuss how its variation will affect the sensing scheduling optimization and thus our system performance. First, assume the channel gain g from PU to SU receiver follows a Rayleigh distribution. We can easily show that $\gamma = \frac{\Psi g^2}{\sigma^2}$ follows an exponential distribution $f(\gamma) = \beta e^{-\beta\gamma}$ if $\gamma \geq 0$, else $f(\gamma) = 0$ with a parameter β. Second, we make an assumption that the random variable γ should be constrained within a certain lower bound and upper bound, i.e., $\gamma \in (\phi_L, \phi_U)$. The lower bound is due to the fact that the PU detector suffers from some channel uncertainty effects (e.g. frequency offset and noise uncertainty), hence it can not detect PU signal below a SNR value which is called the *SNR wall* [CTB06]. The upper bound comes from the fact that the PU detector shown in equation (5.1) only targets the range of low SNR much less than 0 dB. Hence, the truncated probability density function (PDF) for γ can be derived as $f(\gamma) = \frac{\beta e^{-\beta\gamma}}{\int_{\phi_L}^{\phi_U} f(\gamma)d\gamma} = \frac{\beta e^{-\beta\gamma}}{e^{-\beta\phi_L} - e^{-\beta\phi_U}}$, $\gamma \in (\phi_L, \phi_U)$, else $f(\gamma) = 0$. The mean for this truncated γ can be derived as $\mathbb{E}\{\gamma\} = \frac{1}{\beta} + \frac{\phi_L e^{-\beta\phi_L} - \phi_U e^{-\beta\phi_U}}{e^{-\beta\phi_L} - e^{-\beta\phi_U}}$. Followed by the above assumptions, we can formulate a similar optimization problem as shown in ROP 1 which is to maximize the expected throughput given the variance constraint under the SNR variation, i.e.,

Robust Optimization Problem 4

$$\max_{\mathcal{S}} \mathbb{E}\{R(\mathcal{S})\} = \sum_{i=1}^{M} \frac{\left(T - \mathbb{E}\left\{\frac{A^{(i)}(\mathcal{S})}{\gamma}\right\}\right) C_i(1-u_i)}{T},$$

$$s.t.\ \mathcal{S} \in \mathcal{A},\ \ Var\{R(\mathcal{S})\} = Var\left\{\frac{1}{\gamma}\right\} \left[\sum_{i=1}^{M} \frac{\{A^{(i)}(\mathcal{S})\}C_i(1-u_i)}{T}\right]^2 \leq \eta, \tag{5.17}$$

since the sensing time can be written as $T_I^{(i)}(\mathcal{S}) = \frac{A^{(i)}(\mathcal{S})}{\gamma}$, where $A^{(i)}(\mathcal{S})$ is a constant value depending on the sensing strategy and the fusion rule. In order to obtain the first and second moments for the throughput, we apply the following Lemma.

Lemma 4 *Given a random variable γ with its truncated exponential PDF $\bar{f}(\gamma)$ for $\gamma \in (\phi_L, \phi_U)$ with the shape parameter β, we can obtain the first and second moments for its inverse random variable respectively as $\mathbb{E}\left\{\frac{1}{\gamma}\right\} = \frac{\beta(Ei(\beta\phi_L) - Ei(\beta\phi_U))}{e^{-\beta\phi_L} - e^{-\beta\phi_U}}$, and $\mathbb{E}\left\{\frac{1}{\gamma^2}\right\} = \frac{\beta}{e^{-\beta\phi_L} - e^{-\beta\phi_U}}[\frac{e^{-\beta\phi_L}}{\phi_L} - \frac{e^{-\beta\phi_U}}{\phi_U} - \beta(Ei(\beta\phi_L) - Ei(\beta\phi_U))]$, where $Ei(x) = \int_x^{\infty} \frac{e^{-t}}{t}dt$ is the exponential integral.*

Proof: For the first moment of $\frac{1}{\gamma}$, it can be derived by calculating it directly by the definition and with change of variable technique. For the second moment of $\frac{1}{\gamma}$, it can be derived using integral by parts combining with the result of its first moment.

From Lemma 4, we can derive the variance for $\frac{1}{\gamma}$ and therefore the variance for the throughput. In addition, to solve ROP 4, we can apply the same Algorithm 7 by simply replacing the objective and subjective functions accordingly.

5.6 Numerical Results

In this section, numerical results evaluating the expected normalized throughput from spectrum opportunities under different strategies are presented and discussed.

5.6.1 Throughput Comparisons for All Strategies

In Fig. 5.6, it can be observed that, among the parallel strategies, the constraint relaxation achieves the largest throughput because the solution in equation (5.5) is not necessarily integer hence it provides an upper bound for the parallel strategies. Second, there exists a performance gap between the proposed greedy heuristic strategy and the optimal parallel strategy. This can be explained by the fact that in the heuristic, we only consider the traffic and channel information, but we ignore the impact of the fusion model and thus sensing time. Third, the sequential strategy outperforms the parallel strategies when the number of users is small. This is due to the fact that all channels can be sequentially sensed in a time slot with small sensing time for each channel. In addition, the number of channels is large compared to the number of users so that the sequential strategy would not waste as many spectrum opportunities as the parallel strategies. Finally, as expected, the sequential-parallel strategy performs better than pure sequential and parallel strategies, and the proposed iterative parallel strategy can not outperform the sequential-parallel. The proposed greedy heuristic for the sequential-parallel does not perform well for the selected parameter values. Note that in Fig. 5.6 we consider the *OR* rule, but for the *AND* rule we also have similar results.

5.6.2 Throughput Comparisons for Parallel and Sequential Strategies

To have a better insight on all the strategies, we vary different parameters in Fig. 5.7 to show their impact on the optimal throughput. We assume that all channels have the same parameters to simplify the results interpretation.

In Fig. 5.7(a), we first vary various Q_d. A higher target Q_d implies a longer sensing time, and we thus observe a performance degradation for all strategies. When the sensing

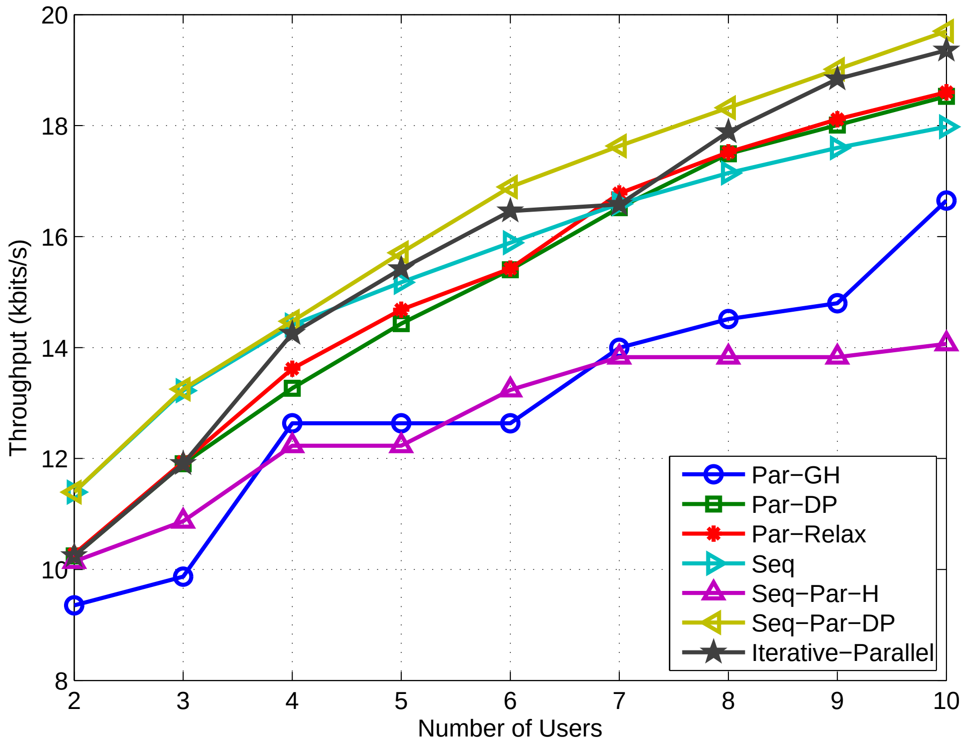


Figure 5.6: Performance comparison for parallel strategy with greedy heuristic (Par-GH), dynamic programming (Par-DP), constraint relaxation (Par-Relax), sequential strategy (Seq), sequential-parallel strategy with greedy heuristic (Seq-Par-GH), sequential-parallel strategy with dynamic programming (Seq-Par-DP), and iterative parallel strategy (Iterative-Parallel) under OR fusion rule versus the number of users. Simulation parameters are $T = 5$ ms, $\Gamma = 10$ dB, $Q_d = 0.9$, $Q_f = 0.15$, $M = 6$, $\gamma = -5$ dB, $B = 0.5 f_s = (1, 1.5, 2, 2.5, 3, 5)$ kHz, and $u = (0.1, 0.2, 0.3, 0.4, 0.5, 0.3)$ for each channel.

time is short, the sequential strategy guarantees that all channels will be sensed. On the other hand, the parallel strategy senses a maximum of N channels which is less than M. Hence it can be seen that the parallel strategy performs better than the sequential when the sensing time is long since the spectrum opportunities in the last channels in the sequential schedule will be very short or null. The sequential-parallel strategy naturally outperforms the sequential and parallel strategies. It can be seen that when the sensing time is very short, the performance of the sequential strategy and the sequential-parallel is the same. This means that the optimal decision by the sequential-parallel strategy also senses all channels by all users. When sensing time is very long, the sequential-parallel results are close to the parallel strategy.

It can be seen that analytical results provided in Section 5.3.3 are matched well with the numerical results except for the parallel strategy for very large values of Q_d. For the selected simulation parameters, sensing time with a few number of sensors to satisfy a large Q_d is larger than the time slot. We saw that the parallel analytical approach tries to distribute users as much as possible with less cooperation, so the throughput will be zero for large values of Q_d. The optimal solution however selects the maximum cooperation on a single channel.

Comparing the *AND* and *OR* rules, the performance is similar for the parallel strategy. Except for very large values of Q_d, the parallel strategy, for both *AND* and *OR* rules, distributes the users evenly among channels, implying that each user will sense one channel and no cooperation takes place for $N \leq M$. Therefore, the fusion model is irrelevant. For the sequential strategy, we can see that the *AND* rule outperforms the *OR* rule when the sensing time is short. It can be explained by the fact that the sensing time is shorter for the *AND* rule. However for $Q_d = 1 - Q_f = 0.85$, they have the same performance and after that the *OR* rule performs better since the sensing time becomes shorter for the *OR* rule. It is worth noting that decreasing Q_f or decreasing the sampling frequency has a similar impact for both rules, as the sensing time increases.

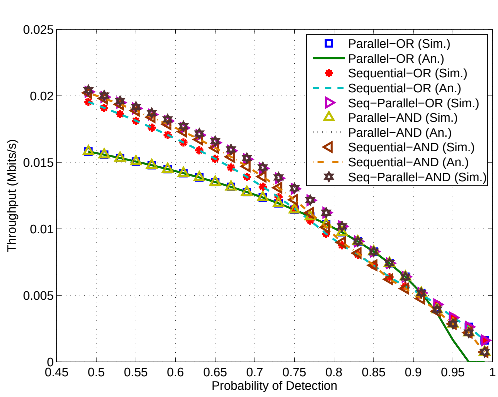
In Fig. 5.7(b), the number of users is fixed to three. When the number of channels M increases, it is expected to have some performance gain. However, the parallel strategy can not sense more than three channels and thus its throughput saturates at three channels. The sequential strategy performance increases with more channels until no more channels can be sensed. Given the sensing time equal to τ_N, maximum $\lfloor \frac{T}{\tau_N} \rfloor$ channels may be sensed in one time slot, which is $\lfloor \frac{5\text{ ms}}{1.8\text{ ms}} \rfloor = 2$ channels in this figure. The sequential-parallel strategy may have three user subsets and in each subset, maximum $\lfloor \frac{T}{\tau_1} \rfloor = \lfloor \frac{1\text{ ms}}{3.4\text{ ms}} \rfloor = 1$ channels can be sensed. The saturation thus occurs at $M = 3$. Discussions related to comparing *AND* and *OR* fusion rules are similar to the previous figure.

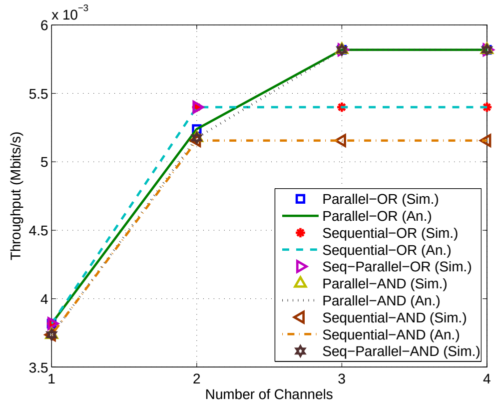
When we increase the number of users, it is expected to observe that when $N < M$, the parallel strategy can not sense all channels, so that the sequential strategy outperforms the parallel strategy. When the number of users increases, the parallel strategy is able to sense all channels, so it becomes superior. However in Fig. 5.7(c), the sequential scheme can not sense more than two channels when $N \geq 2$, and three channels when $N \geq 4$. So its performance improves mostly by the decrease in the cooperative sensing time. However, the parallel scheme will be able to sense four channels when the number of users increases from 2 to 4, and then for larger number of users benefits from cooperative sensing. Its performance is thus always superior to the sequential scheme.

The other interesting point to observe is that as the number of users increases, the gap between *OR* and *AND* fusion rules increases because cooperative sensing becomes more likely and for the given Q_d and Q_f, the *OR* fusion model has shorter cooperative sensing times.

5.6.3 Performance Comparison for Heterogeneous Sensors

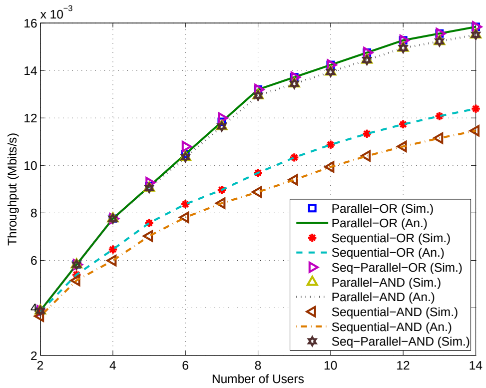
In this section, we discuss the performance of the proposed solutions for the scenarios with heterogeneous sensors. Due to space limit, we only provide the heterogeneous version of Fig. 5.7(c) since the results are similar for other scenarios. Further, the results are only for the *AND* fusion rule because with the *OR* rule, mixing any set of SNRs does not worsen the performance and the cooperative sensing time will not be worse than the non-cooperative; therefore, the notion of optimal set of sensors is not applicable. Instead of a fixed detection SNR equal to -5 dB, in this section users have an exponentially distributed random SNR with an average of -5 dB but limited to the bound $(-15, 0)$ dB. Fig. 5.8 is the heterogeneous version of Fig. 5.7(c) comparing sequential, parallel and sequential-parallel strategies together and with the proposed heuristics. We ran the simulation for 500 runs for all schemes except the sequential-parallel DP for $N = 6$, which was repeated 100 times. However, it can be seen that the confidence intervals are acceptable for 90% confidence. It should be noted that even though the second heuristic proposed for the parallel strategy does not perform





(a) Performance comparison versus probability of detection Q_d. Simulation parameters are $Q_f = 0.15$, $N = 3$, $M = 4$, $\gamma = -5$ dB, $f_s = 2B = 5$ kHz, $u = 0.3$, $T = 5$ ms, and $\Gamma = 10$ dB.

(b) Performance comparison versus number of channels. Simulation parameters are $Q_d = 0.9$, $Q_f = 0.15$, $N = 3$, $\gamma = -5$ dB, $f_s = 2B = 5$ kHz, $u = 0.3$, $T = 5$ ms, and $\Gamma = 10$ dB.



(c) Performance comparison versus number of users. Simulation parameters are $Q_d = 0.9$, $Q_f = 0.15$, $M = 4$, $\gamma = -5$ dB, $f_s = 2B = 5$ kHz, $u = 0.3$, $T = 5$ ms, and $\Gamma = 10$ dB.

Figure 5.7: Throughput comparisons for using the *OR* fusion rule of parallel (Par-OR), sequential (Seq-OR), sequential-parallel (Seq-Par-OR), and the *AND* fusion rule of parallel (Par-AND), sequential (Seq-AND), sequential-parallel (Seq-Par-AND) strategies with homogeneous channels. Simulation results (Sim.) are shown to be matched with analytical results (An.).

well, it is provided to reveal the fact that when two users with different SNRs cooperate, the performance can be worse. For this heuristic, there are four channels, so up to $N = 4$, each user has one channel. For $N = 5$, two users cooperate, but it may degrade the performance since those cooperating users are not selected optimally in the second heuristic. Compared to Fig. 5.7(c), we can also see that throughput is higher here with heterogeneous sensors. This can be explained by Jensen's inequality. Considering throughput as a function of detection SNR, i.e., $R(\gamma)$, we observed that this function is convex in the range of $(-15, 0)$ dB (but concave for higher values of SNR). Therefore, it is expected to have $R(\mathbb{E}\{\gamma\}) \leq \mathbb{E}\{R(\gamma)\}$.

5.6.4 Performance Comparison for Primary Traffic and Channel Uncertainty

In this section, we compare the throughput performance under primary traffic and channel uncertainty by solving *the Robust Optimization Problem 1* and *Robust Optimization Problem 4*, respectively. To demonstrate the results, we choose the proposed parallel sensing strategy as an example with fusion *OR* rule. For primary traffic uncertainty design in *Robust Optimization Problem 1*, by applying the parallel sensing strategy, the strategy constraint, i.e., $\mathcal{S} \in \mathcal{A}$, would be $\sum_{i=1}^{N} k_i = N$, and the sensing time $T_I^{(i)}(\mathcal{S})$ is equal to τ_{i,k_i}, as shown in equation (5.5). For channel uncertainty design in *Robust Optimization Problem 4*, by applying the parallel sensing strategy, the strategy constraint is also $\sum_{i=1}^{N} k_i = N$, while the constant factor is $A^{(i)}(\mathcal{S}) = \frac{[Q^{-1}(1-\sqrt[k_i]{1-Q_f}) - Q^{-1}(1-\sqrt[k_i]{1-Q_d})]^2}{f_s}$, from equation (5.2). Fig. 5.9 shows the normalized throughput loss with the standard deviation of the throughput constraint for primary traffic uncertainty case given a constant number of traffic samples W for all channels, and for the channel uncertainty case. Note that they are compared with the same average detection SNR γ in -5 dB. The normalized throughput loss is defined as the throughput loss from maximal throughput without variance constraint and normalized by it. From the figure, first we observe that, as we increase the standard deviation of the throughput constraint, which means that we have relaxed the robust optimization constraint, we have a lower throughput loss. This is due to the fact that we have more search space for the optimal allocation k_i, and hence we can achieve higher throughput. Second, as we increase

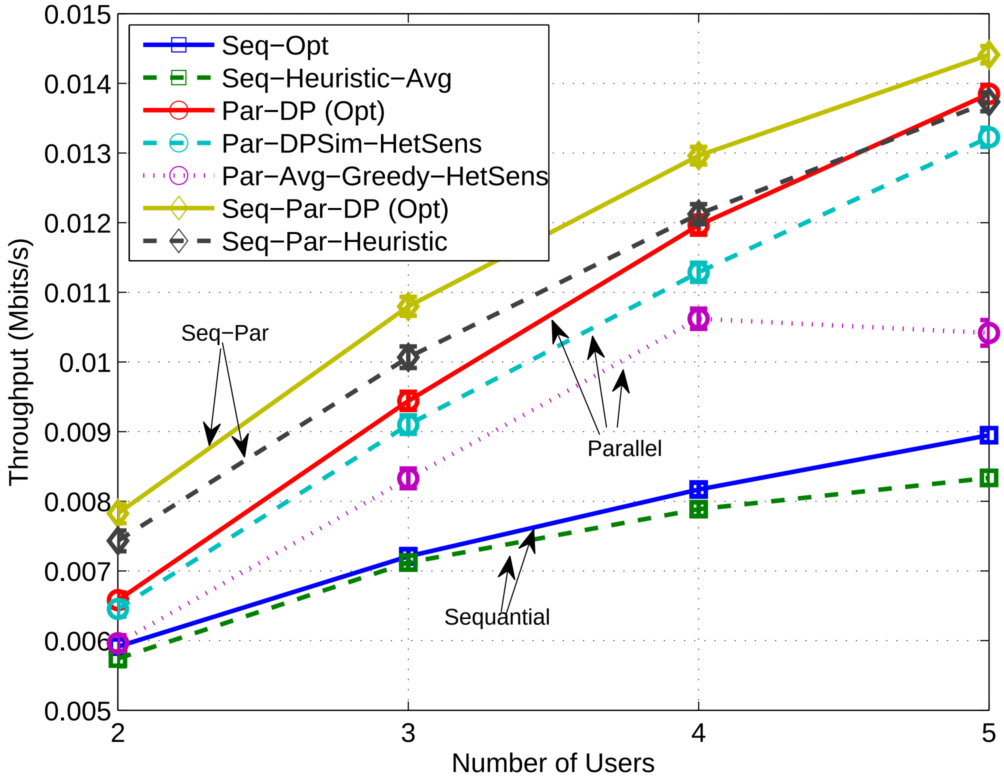


Figure 5.8: Throughput comparisons for using *AND* fusion rule for the scenario with heterogeneous sensors. Optimal results and proposed heuristics for Sequential (Seq), Parallel (Par) and Sequential-Parallel (Seq-Par) strategies are compared. The figure is the heterogeneous version of Fig. 5.7(c) with the same parameter values except the detection SNR γ, which is random here. For the noise power, we set its variance $\sigma^2 = 10^{-8}$ W. The confidence interval for each simulation points is set to be 90%.

the number of traffic samples W, the throughput loss decreases. This is because as we use more traffic samples to estimate the channel busy rate u_i, we have less uncertainty on the estimation. Therefore we can have lower variance for the estimator, which results in lower variance for the throughput. In other words, we can achieve larger throughput with the same variance constraint. Third, we can observe that the decreasing rate for the channel detection uncertainty case is smaller than for the primary traffic uncertainty. This means that if we have channel detection uncertainty, it will be difficult to achieve low throughput loss compared with traffic uncertainty. This is important since in practice, accurately estimating the instantaneous detection SNR can be difficult. Finally, as we increase the constraint, the optimal allocation solution k_i will also change accordingly. Note that as the standard deviation is large enough, the normalized throughput loss will eventually go to zero, i.e., to provide the same optimal allocation as the optimization without constraint, e.g., $\mathbf{k}^* = (0, 2, 2, 2, 2, 2)$ when $W = 20$. In addition, since we have a discrete solution space for k_i, if the variance threshold does not increase significantly, the solution may stay the same hence resulting in the ladder type curve as shown for channel detection uncertainty case.

5.6.5 Design Examples

In this section, we elaborate two design examples for primary traffic uncertainty by solving ROP 2 assuming that we do not have knowledge of primary traffic statistics. To achieve the maximal throughput without variance constraint with the optimal allocation k_i^*, we substitute it in ROP 2 to solve for the minimum number of traffic samples that we need to use to achieve the minimum variance for the throughput. We propose two designs, i.e., homogeneous and heterogeneous estimation, shown as *Design 1:* $\min_{W} W$, s.t. $\text{Var}\{\hat{R}(\mathcal{S})\} \leq \eta$, and *Design 2:* $\min_{W_i} \sum_{i=1}^{M} W_i$, s.t. $\text{Var}\{\hat{R}(\mathcal{S})\} \leq \eta$. *Design 1* represents the case where we estimate the traffic duty cycle for all channels with the same number of traffic samples, while *Design 2* is when we estimate the traffic duty cycle on each channel with its own necessary number of traffic samples. Fig. 5.10 shows the total number of samples needed to achieve a certain standard deviation of throughput. First, as we relax the threshold to

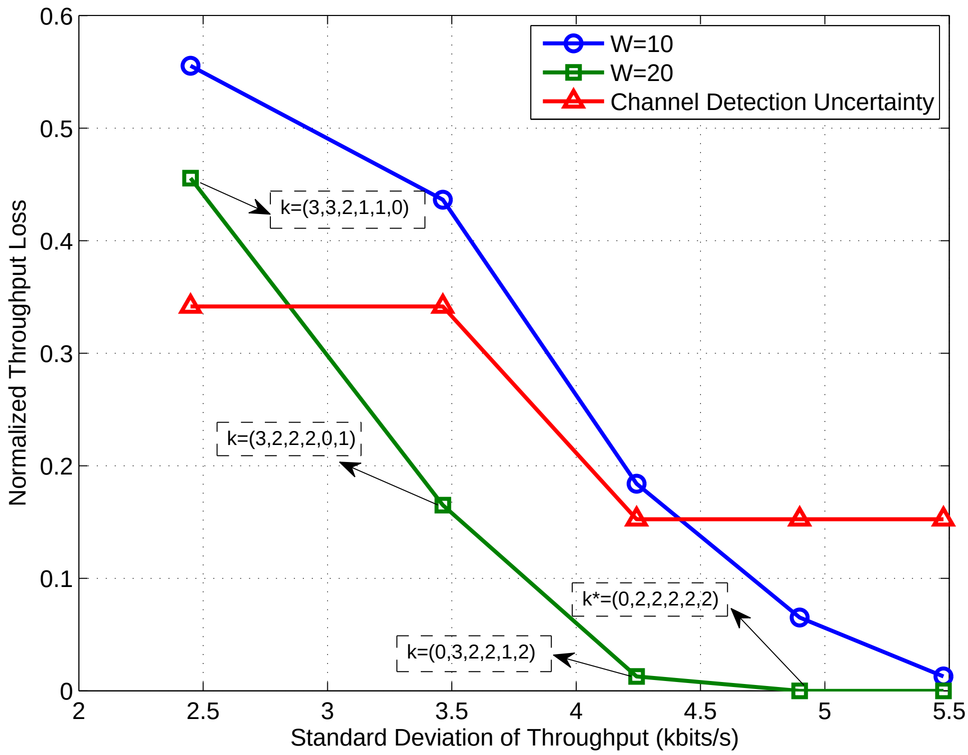


Figure 5.9: Normalized throughput loss comparison for traffic and channel uncertainty with parallel sensing strategy under OR fusion rule versus the standard deviation of throughput. Simulation parameters are $T = 5$ ms, $\Gamma = 10$ dB, $Q_d = 0.9$, $Q_f = 0.25$, $N = 10$, $M = 6$, $\gamma = -5$ dB for the traffic uncertainty, $(\phi_L, \phi_U) = (-15, -1)$ dB, $B = 0.5f_s = (1, 1.5, 2, 2.5, 3, 5)$ kHz, $u = (0.1, 0.2, 0.3, 0.4, 0.5, 0.3)$ and $P_{00} = 0.9$ for each channel.

have less robustness, we need less number of samples for estimation. Second, since *Design 2* is a general optimization process considering traffic characteristics for each channel, it can achieve the same standard deviation threshold using less number of samples. Finally, from the system design point of view, for example, if we want to achieve a robust design for the optimal throughput with standard deviation less than 3.55 kbits/s, we need to use at least 100 samples for traffic estimation.

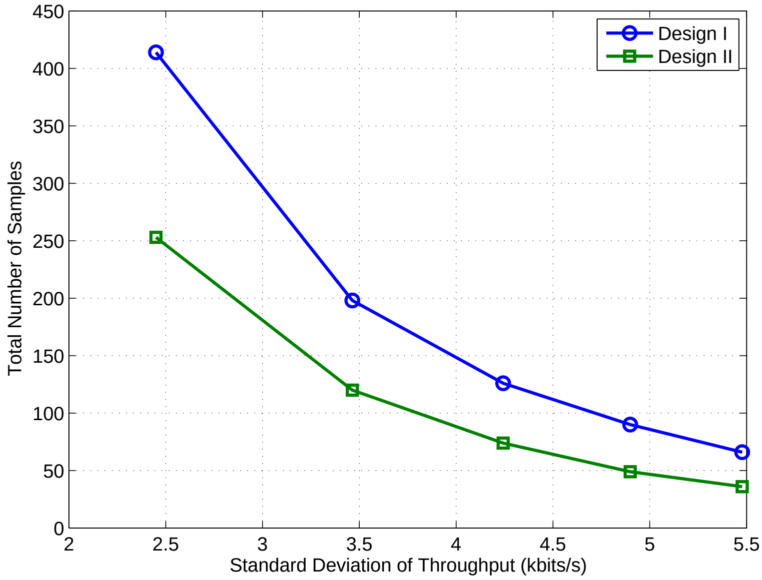


Figure 5.10: Total number of samples comparison for Design I and Design II with parallel sensing strategy under OR fusion rule versus the standard deviation of throughput. Simulation parameters are $T = 5$ ms, $\Gamma = 10$ dB, $Q_d = 0.9$, $Q_f = 0.25$, $N = 10$, $M = 6$, $\gamma = -5$ dB, $B = 0.5 f_s = (1, 1.5, 2, 2.5, 3, 5)$ kHz, $u = (0.1, 0.2, 0.3, 0.4, 0.5, 0.3)$ and $P_{00} = 0.9$ for each channel.

5.7 Summary

In this work, we proposed and compared several cooperative spectrum sensing strategies for homogeneous and heterogeneous sensors, i.e., sequential, parallel, and sequential-parallel to schedule users to sense multiple channels in order to achieve the optimal throughput. For each strategy, we introduced several solutions including low-complexity heuristic and dynamic programming methods. In addition, we proposed a robust scheduling design in terms of both primary traffic and channel detection uncertainty, and a design guideline was also provided for primary traffic estimation given the throughput variation constraint. In terms of throughput performance, we showed that with longer sensing time, such as when we have a stringent constraint on probability of detection, smaller number of channels, or larger number of users, the parallel sensing strategy is recommended. Otherwise the sequential sensing strategy should be adopted. A hybrid sequential-parallel sensing strategy has the benefits of both approaches and perform better in almost all scenarios while it suffers from a high complexity limiting its implementation and usage.

CHAPTER 6

Conclusions

6.1 Research Contributions

To summarize the contributions of this dissertation, first we proposed and analyzed traffic distribution classification algorithms in DSA networks. In addition to the algorithm design, we considered the effect of blind and imperfect information of traffic parameters on classification algorithms. A design guideline was provided in order to use minimum energy or time for spectrum sensing in order to achieve a certain correct classification rate. The classification works are listed in the publications [J4] and [C3].

Second, we proposed traffic prediction algorithms based on traffic statistics. The prediction of PU traffic states was achieved by constructing Markov chain using traffic ON/OFF distributions. We also considered the case if we have blind knowledge of traffic parameters, and we proposed prediction regions to link prediction performance and the traffic parameter estimation errors. The prediction works are listed in the publications [J3] and [C5].

Finally we proposed several multi-user sensing and transmission scheduling strategies to optimize the SU throughput. For each strategy, we not only derived its optimal solution, but also provided the low-complexity heuristic solution. A robust optimization problem was formulated in order to maximize the system throughput when we had both traffic and channel uncertainties. The scheduling works are listed in the publications [J1], [J2], [J5], and [C2].

6.2 Future Work

In this dissertation, we limited the traffic distribution as the gamma distribution (including exponential and Erlang-2 distributions) with independent and identically distributed properties for ON/OFF traffic durations. However, current literature has shown that other famous distributions, such as log-normal, Weibull, and hyper-Erlang distributions are also measured in certain frequency bands at various locations. Since these distributions have much different characteristics than the gamma distribution, new classification/prediction frameworks or algorithms should be designed and analyzed in order to have more general solutions with respect to the realistic traffic scenarios. Moreover, current literature also showed that the ON/OFF traffic durations may not be independent, i.e., they are correlated under certain circumstances. Therefore the new classification/prediction frameworks should also take the correlated ON/OFF durations into account which exploit the properties of joint ON/OFF distributions.

When designing the OSA protocols, in this dissertation we limited the PUs transmitting in a time-slotted fashion. Note that there are still many other communication protocols that adopt the non-slotted transmission such as carrier sense multiple access (CSMA) protocol for PUs. As we consider the non-slotted PU transmission, not only should SU throughput being analyzed, but also collisions would occur with PUs. The collisions can be analyzed according to different traffic distributions which will serve as an important metric to evaluate the OSA protocol. Therefore, sensing scheduling protocols should be redesigned by including the collision metric to form a general OSA network.

APPENDIX A

Appendix

A.1 Derivation of Mean and Variance for the Distribution of $y_i^{(j,k)}$

To derive the mean and variance for $y_i^{(j,k)}$, we need to derive its PDF first. Here we ignore the index i for convenience since all $y_i^{(j,k)}$ have the same distribution. Since we know the PDF for x under hypothesis $\mathcal{H}_j$, we can apply the change of variable technique to derive the PDF for $y^{(j,k)}$ as

$$\begin{aligned} f_{Y^{(j,k)}}(y^{(j,k)}) = & \left| \frac{\partial}{\partial y^{(j,k)}} h^{-1}\left(y^{(j,k)}\right) \right| \\ & \times f_j\left(h^{-1}\left(y^{(j,k)}\right) | \mathbf{\Theta}_j\right), \end{aligned} \tag{A.1}$$

where $|\cdot|$ is the absolute value function, $h(x) = \alpha_{j,k} \log x - \beta_{j,k} x$, and h^{-1} is the inverse function of h. To find h^{-1}, we introduce first the following Lemma.

Lemma 5 *The inverse function for $h(x) = \alpha \log(x) - \beta x$ is*

$$h^{-1}(y) = -\frac{\alpha}{\beta} W\left(0, e^{\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)}\right), \ \forall \alpha \neq 0, \ \textit{if } h^{-1}(y) \leq \frac{\alpha}{\beta}, \tag{A.2}$$

and

$$h^{-1}(y) = -\frac{\alpha}{\beta} W\left(-1, e^{\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)}\right) \ \forall \alpha \neq 0, \ \textit{otherwise}, \tag{A.3}$$

where $W(k, y)$ is a Lambert W function of branch k, where k is an integer for complex y and $k \in \{0, -1\}$ for real y (refer to MATLAB's `lambertw` *function implementation description) [CGH96, Eq. (1.5)].*

Proof: Consider the Wright omega function, $\omega(y)$ [CJ00, Eq. (1)], which is defined as the unique solution to $y = \log(x) + x$, which can be also written recursively as

$$y = \log(\omega(y)) + \omega(y), \tag{A.4}$$

where $W(0, e^y) = \omega(y)$. Embedding $x = -\frac{\alpha}{\beta}\omega\left(\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)\right)$ to the expression $\alpha\log(x) - \beta x$ we can show that

$$\begin{aligned}
&\alpha \log\left(-\frac{\alpha}{\beta}\omega\left(\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)\right)\right) \\
&\quad - \beta\left(-\frac{\alpha}{\beta}\omega\left(\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)\right)\right) && \text{(A.5)} \\
&\quad = \alpha\left(\log\omega\left(\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)\right) + \omega\left(\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)\right)\right) \\
&\qquad + \alpha\log\left(\frac{-\alpha}{\beta}\right) && \text{(A.6)} \\
&\quad = \alpha\left(\frac{y}{\alpha} + \log\left(\frac{-\beta}{\alpha}\right)\right) + \alpha\log\left(\frac{-\alpha}{\beta}\right) = y, && \text{(A.7)}
\end{aligned}$$

where (A.7) stems directly from (A.4). Therefore we know x is an inverse function.

Now, note that the function $h(x)$ is a concave function as $\alpha \geq 0$, and convex otherwise. Therefore, for $\alpha \geq 0$, there are two possible real-value solutions for $h(x) = y$: (i) one is located on the left hand side of the peak value for $h(x)$, i.e., $x = \frac{\alpha}{\beta}$, and (ii) another located on its right hand side. By definition of a Lambert W function, these two solutions are shown to be located on $k = 0$ and $k = -1$ branches. For $\alpha < 0$, there is only one solution on $k = 0$ branch since $h(x)$ is a decreasing function.

By applying the derivative of the Lambert W function, i.e., $\frac{\partial W(k,s)}{\partial s} = \frac{W(k,s)}{s(1+W(k,s))}$, and Lemma 5 to (A.1), we can derive the PDF for $y^{(j,k)}$ as

$$
\begin{aligned}
f_{Y^{(j,k)}}(y^{(j,k)}) = & \left| \frac{W\left(0, e^{B^{(j,k)}}\right)}{\beta_{j,k}\left(1 + W\left(0, e^{B^{(j,k)}}\right)\right)} \right| \\
& \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(0, e^{B^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right) \\
& + I\left(\frac{\alpha_{j,k}}{\beta_{j,k}}\right) \left| \frac{W\left(-1, e^{B^{(j,k)}}\right)}{\beta_{j,k}\left(1 + W\left(-1, e^{B^{(j,k)}}\right)\right)} \right| \\
& \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(-1, e^{B^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right),
\end{aligned} \tag{A.8}
$$

where $B^{(j,k)} \triangleq \frac{y^{(j,k)}}{\alpha_{j,k}} + \log\left(\frac{-\beta_{j,k}}{\alpha_{j,k}}\right)$ (defined for presentation compactness), and $I(a) = 1$ if $a \geq 0$ and $I(a) = 0$ otherwise.

We can finally derive the mean and variance using (A.8) as

$$
\mu_{j,k} = \int_{-\infty}^{\infty} y^{(j,k)} f_{Y^{(j,k)}}\left(y^{(j,k)}\right) dy^{(j,k)}, \tag{A.9}
$$

$$
\begin{aligned}
\sigma_{j,k}^2 = & \int_{-\infty}^{\infty} \left(y^{(j,k)}\right)^2 f_{Y^{(j,k)}}\left(y^{(j,k)}\right) dy^{(j,k)} \\
& - \left(\mu_{j,k}\right)^2,
\end{aligned} \tag{A.10}
$$

respectively, through numerical integration.

A.2 Derivation of Squared Hellinger Distance between Two Gamma Distributions

The SH distance for two probability distributions is defined as [Van98, Ch. 14.5, pp. 211]

$$
\begin{aligned}
H^2(f_j(x|\boldsymbol{\Theta}_j), & f_k(x|\boldsymbol{\Theta}_k)) \\
& \triangleq 1 - \int_{-\infty}^{\infty} \sqrt{f_j(x|\boldsymbol{\Theta}_j) f_k(x|\boldsymbol{\Theta}_k)} dx,
\end{aligned} \tag{A.11}
$$

(again, note that the 0.5 constant is omitted for convenience as remarked in [Pol02, Ch. 3.3, pp. 61]). Before calculating the closed-form expression of SH distance for two gamma distributions we introduce the following integral

$$\int_a^\infty x^\rho e^{-\mu x} dx = \frac{\Gamma(\rho+1, a\mu)}{\mu^{\rho+1}}, \tag{A.12}$$

where $\Gamma(\rho+1, x) = \int_x^\infty t^\rho e^{-t} dt$ is the incomplete gamma function. Integral (A.12) can be derived through calculating the incomplete gamma function by the change of variable technique.

From the definition of (A.11) the SH distance for two distributions $f_j(x|\mathbf{\Theta}_j)$ and $f_k(x|\mathbf{\Theta}_k)$ can be derived as

$$\begin{aligned} &H^2(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k)) \\ &\quad = 1 - C(\mathbf{\Theta}_j, \mathbf{\Theta}_k) \int_0^\infty x^{\frac{\alpha_j+\alpha_k}{2}-1} e^{-\left(\frac{\beta_j+\beta_k}{2}\right)x} dx, \end{aligned} \tag{A.13}$$

where $C(\mathbf{\Theta}_j, \mathbf{\Theta}_k) = \sqrt{\frac{\beta_j^{\alpha_j}\beta_k^{\alpha_k}}{\Gamma(\alpha_j)\Gamma(\alpha_k)}}$. Applying (A.12) with $\rho = \frac{\alpha_j+\alpha_k}{2} - 1$ and $\mu = \frac{\beta_j+\beta_k}{2}$ to (A.13) the SH distance in (A.13) can be simplified to

$$\begin{aligned} &H^2(f_j(x|\mathbf{\Theta}_j), f_k(x|\mathbf{\Theta}_k)) \\ &\qquad\qquad = 1 - C(\mathbf{\Theta}_j, \mathbf{\Theta}_k) \frac{\Gamma(\frac{\alpha_j+\alpha_k}{2})}{\left(\frac{\beta_j+\beta_k}{2}\right)^{\frac{\alpha_j+\alpha_k}{2}}}. \end{aligned} \tag{A.14}$$

Note that the average SH distance with ALF, which is used to represent the average distance among hypotheses in Table 2.2, can be calculated by using (2.33) to replace $f_j(x|\mathbf{\Theta}_j)$ in (A.11). Also note that the average SH distance with ALF has no closed-form expression and it can only be computed through numerical methods.

A.3 Derivation of Expected Number of PU Traffic Samples under Sampling

The expected average number of PU traffic samplings for one period T_{on} under hypothesis $\mathcal{H}_j$ can be calculated as

$$\mathbb{E}\{N|\mathcal{H}_j\} = \mathbb{E}\left\{\left\lfloor \frac{T_{\text{on}}}{T_s} \right\rfloor\right\} + 1, \tag{A.15}$$

where $\lfloor \cdot \rfloor$ is the floor function. To calculate (A.15) we first need to derive the following conditional probability, i.e.,

$$
\begin{aligned}
\Pr\left\{\left\lfloor \frac{T_{\text{on}}}{T_s} \right\rfloor = k|\mathcal{H}_j\right\} &= \Pr\left\{\frac{T_{\text{on}}}{T_s} - 1 < k \leq \frac{T_{\text{on}}}{T_s}|\mathcal{H}_j\right\} \\
&= \Pr\{kT_s \leq T_{\text{on}} < (k+1)T_s|\mathcal{H}_j\} \\
&= G((k+1)T_s|\mathbf{\Theta}_j) - G(kT_s|\mathbf{\Theta}_j), \qquad \text{(A.16)}
\end{aligned}
$$

where $G(\cdot|\mathbf{\Theta}_j)$ is the CDF function for gamma distribution with parameters α_j and β_j. Applying (A.16) to (A.15) we have

$$
\begin{aligned}
\mathbb{E}\left\{\left\lfloor \frac{T_{\text{on}}}{T_s} \right\rfloor\right\} &= \sum_{k=1}^{\infty} k \Pr\left\{\left\lfloor \frac{T_{\text{on}}}{T_s} \right\rfloor = k|\mathcal{H}_j\right\} \\
&= \lim_{L\to\infty} \sum_{k=1}^{L} k[G((k+1)T_s|\mathbf{\Theta}_j) - G(kT_s|\mathbf{\Theta}_j)] \\
&= \lim_{L\to\infty} (L+1)G((L+1)T_s|\mathbf{\Theta}_j) - \sum_{k=1}^{L+1} G(kT_s|\mathbf{\Theta}_j) \qquad \text{(A.17)} \\
&= \lim_{L\to\infty} -L\frac{\Gamma(\alpha_j,(L+1)\beta_j T_s)}{\Gamma(\alpha_j)} + \sum_{k=1}^{L} \frac{\Gamma(\alpha_j, k\beta_j T_s)}{\Gamma(\alpha_j)} \qquad \text{(A.18)} \\
&= \sum_{k=1}^{\infty} \frac{\Gamma(\alpha_j, k\beta_j T_s)}{\Gamma(\alpha_j)}, \qquad \text{(A.19)}
\end{aligned}
$$

by applying $G(kT_s|\mathbf{\Theta}_j) = \frac{\Gamma(\alpha_j)-\Gamma(\alpha_j,k\beta_j T_s)}{\Gamma(\alpha_j)}$, and the left hand part in (A.18) can be shown to be zero by L'Hopital's rule. Then we introduce the following Lemma as a step to prove (A.19) converges.

Lemma 6

$$
\int_0^{\infty} \Gamma(\alpha_j, k\beta_j T_s)dk = \frac{\alpha_j \Gamma(\alpha_j)}{\beta_j T_s}. \qquad \text{(A.20)}
$$

Proof: We can easily prove it by applying the change of variable technique. Since $\Gamma(\alpha_j, k\beta_j T_s)$ is a decreasing function with respect to k by definition and $\int_0^{\infty} \frac{\Gamma(\alpha_j,k\beta_j T_s)}{\Gamma(\alpha_j)}dk = \frac{\alpha_j}{\beta_j T_s}$, from the integral test, we know (A.19) converges. Therefore, using (A.19) we can derive the average expected number of PU traffic samples by taking the average for all possible hypotheses which results in (2.18).

A.4 PDF Derivation for Sum of the Gamma and Triangular Distributed Random Variables

By directly convolving the PDF of gamma distributed random variable x, i.e., $f_X(x|\mathbf{\Theta})$ where $\mathbf{\Theta} = (\alpha, \beta)$ with the PDF of triangular distributed random variable ϕ, i.e., $f_\Phi(\phi)$, we have the PDF for $\tilde{x} = x + \phi$ as

$$
\begin{aligned}
f_{\tilde{X}}(\tilde{x}) &= \int_{-\infty}^{\infty} f_X(\tilde{x} - x|\mathbf{\Theta}) f_\Phi(x) dx \\
&= \begin{cases}
\int_{-T_s}^{0} \frac{\beta^\alpha}{\Gamma(\alpha)} (\tilde{x} - x)^{\alpha-1} e^{\beta(\tilde{x}-x)} & \\
\quad \times \left(\frac{1}{T_s^2} x + \frac{1}{T_s} \right) dx & \\
\quad + \int_{0}^{T_s} \frac{\beta^\alpha}{\Gamma(\alpha)} (\tilde{x} - x)^{\alpha-1} e^{\beta(\tilde{x}-x)} & \\
\quad \times \left(\frac{-1}{T_s^2} x + \frac{1}{T_s} \right) dx, & \text{if } \tilde{x} \geq T_s, \\
\int_{-T_s}^{0} \frac{\beta^\alpha}{\Gamma(\alpha)} (\tilde{x} - x)^{\alpha-1} e^{\beta(\tilde{x}-x)} & \\
\quad \times \left(\frac{1}{T_s^2} x + \frac{1}{T_s} \right) dx & \\
\quad + \int_{0}^{\tilde{x}} \frac{\beta^\alpha}{\Gamma(\alpha)} (\tilde{x} - x)^{\alpha-1} e^{\beta(\tilde{x}-x)} & \\
\quad \times \left(\frac{-1}{T_s^2} x + \frac{1}{T_s} \right) dx, & \text{if } 0 \leq \tilde{x} < T_s, \\
\int_{-T_s}^{\tilde{x}} \frac{\beta^\alpha}{\Gamma(\alpha)} (\tilde{x} - x)^{\alpha-1} e^{\beta(\tilde{x}-x)} & \\
\quad \times \left(\frac{1}{T_s^2} x + \frac{1}{T_s} \right) dx, & \text{if } -T_s \leq \tilde{x} < 0, \\
0, & \text{otherwise.}
\end{cases}
\end{aligned} \tag{A.21}
$$

We now introduce the following Lemma.

Lemma 7

$$
\begin{aligned}
&\int_a^b \frac{\beta^\alpha}{\Gamma(\alpha)} (\tilde{x} - x)^{\alpha-1} e^{-\beta(\tilde{x}-x)} \left(\frac{1}{T_s^2} x + \frac{1}{T_s} \right) dx \\
&= \frac{\Gamma(\alpha+1, \beta(\tilde{x} - a)) - \Gamma(\alpha+1, \beta(\tilde{x} - b))}{\Gamma(\alpha)\beta T_s^2} \\
&\quad - \frac{(\tilde{x} + T_s)(\Gamma(\alpha, \beta(\tilde{x} - a)) - \Gamma(\alpha, \beta(\tilde{x} - b)))}{\Gamma(\alpha) T_s^2},
\end{aligned} \tag{A.22}
$$

$$
\begin{aligned}
&\int_a^b \frac{\beta^\alpha}{\Gamma(\alpha)}(\tilde{x}-x)^{\alpha-1}e^{-\beta(\tilde{x}-x)}\left(-\frac{1}{T_s^2}x+\frac{1}{T_s}\right)dx \\
&= \frac{-\Gamma(\alpha+1,\beta(\tilde{x}-a))+\Gamma(\alpha+1,\beta(\tilde{x}-b))}{\Gamma(\alpha)\beta T_s^2} \\
&\quad + \frac{(\tilde{x}-T_s)(\Gamma(\alpha,\beta(\tilde{x}-a))-\Gamma(\alpha,\beta(\tilde{x}-b)))}{\Gamma(\alpha)T_s^2}. \qquad \text{(A.23)}
\end{aligned}
$$

Proof: Expression (A.22) and (A.23) can be calculated directly from the definition of incomplete gamma function and through the integration by parts technique. Finally, applying Lemma 7 to (A.21) we obtain (2.25).

A.5 Derivation of Variance for $\tilde{y}_i^{(j,k)}$

We ignore the index i for notation convenience and denote $\tilde{x} = x + \phi$. We would like to find the variance of $\tilde{y}^{(j,k)} = \alpha_{j,k}\log\tilde{x} - \beta_{j,k}\tilde{x}$, where $x \sim f_j(x|\mathbf{\Theta}_j)$, $\phi \sim \Lambda(-T_s, T_s)$, and $\tilde{x} \sim f_j(\tilde{x}|\mathbf{\Theta}_j, T_s)$ given in Theorem 2. Since $\tilde{x}$ can be negative, $\tilde{y}^{(j,k)}$ may be a complex number. Therefore we define $\tilde{y}^{(j,k)} \triangleq \tilde{y}_R^{(j,k)} + j\tilde{y}_I^{(j,k)}$, where $\tilde{y}_R^{(j,k)} = \alpha_{j,k}\log\tilde{x} - \beta_{j,k}\tilde{x}$ and $\tilde{y}_I^{(j,k)} = 0$, if $\tilde{x} \geq 0$, and $\tilde{y}_R^{(j,k)} = \alpha_{j,k}\log(-\tilde{x}) - \beta_{j,k}\tilde{x}$ and $\tilde{y}_I^{(j,k)} = \pi\alpha_{j,k}$, otherwise. Note the PDF of $\tilde{y}^{(j,k)}$ can be represented as $f_{\tilde{Y}}^{(j,k)}\left(\tilde{y}^{(j,k)}\right) = f_R\left(\tilde{y}^{(j,k)}\right) + jf_I\left(\tilde{y}^{(j,k)}\right)$, where $f_R(\cdot)$ and $f_I(\cdot)$ are the PDFs with respect to the real part and imaginary part of $\tilde{y}^{(j,k)}$. Likewise, the variance for $\tilde{y}^{(j,k)}$, i.e., $\tilde{\sigma}_{j,k}^2$, is the sum of the variance of its real part $\tilde{\sigma}_{R,j,k}^2$ and imaginary part $\tilde{\sigma}_{I,j,k}^2$.

First we calculate the variance of the imaginary part. Noting that the first and the second moment for $\tilde{y}_I^{(j,k)}$, which are

$$
\mathbb{E}\left\{\tilde{y}_I^{(j,k)}\right\} = \pi\alpha_{j,k}\mathbb{E}\left\{\tilde{x} < 0\right\} = \pi\alpha_{j,k}\int_{-\infty}^{0} f_j(\tilde{x}|\mathbf{\Theta}_j, T_s)d\tilde{x} \qquad \text{(A.24)}
$$

and

$$
\mathbb{E}\left\{\left(\tilde{y}_I^{(j,k)}\right)^2\right\} = \pi^2\alpha_{j,k}^2\mathbb{E}\{\tilde{x} < 0\} = \pi^2\alpha_{j,k}^2\int_{-\infty}^{0} f_j(\tilde{x}|\mathbf{\Theta}_j, T_s)d\tilde{x}, \qquad \text{(A.25)}
$$

respectively, we can derive $\tilde{\sigma}_{I,j,k}^2$. The variance for the real part can be obtained through $f_R(\tilde{y}^{(j,k)})$. Using Lemma 5 and observing that there may be at most three solutions to

$\tilde{y}_R^{(j,k)} = h(\tilde{x})$, we can derive the PDF for the real part of $\tilde{y}^{(j,k)}$ as

$$f_R(\tilde{y}^{(j,k)}) = \begin{cases} I(\eta_1 - \tilde{y}^{(j,k)}) \left| \frac{W\left(-1, e^{B^{(j,k)}}\right)}{\beta_{j,k}\left(1+W\left(-1, e^{B^{(j,k)}}\right)\right)} \right| \\ \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(-1, e^{B^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right) \\ +I(\eta_1 - \tilde{y}^{(j,k)}) \left| \frac{W\left(0, e^{C^{(j,k)}}\right)}{\beta_{j,k}\left(1+W\left(0, e^{C^{(j,k)}}\right)\right)} \right| \\ \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(0, e^{C^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right) \\ + \left| \frac{W\left(0, e^{B^{(j,k)}}\right)}{\beta_{j,k}\left(1+W\left(0, e^{B^{(j,k)}}\right)\right)} \right| \\ \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(0, e^{B^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right), & \text{if } \frac{\alpha_{j,k}}{\beta_{j,k}} \geq 0, \\ \left| \frac{W\left(0, e^{B^{(j,k)}}\right)}{\beta_{j,k}\left(1+W\left(0, e^{B^{(j,k)}}\right)\right)} \right| \\ \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(0, e^{B^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right) \\ +I(\tilde{y}^{(j,k)} - \eta_2) \left| \frac{W\left(0, e^{C^{(j,k)}}\right)}{\beta_{j,k}\left(1+W\left(0, e^{C^{(j,k)}}\right)\right)} \right| \\ \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(0, e^{C^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right) \\ +I(\tilde{y}^{(j,k)} - \eta_2) \left| \frac{W\left(1, e^{C^{(j,k)}}\right)}{\beta_{j,k}\left(1+W\left(1, e^{C^{(j,k)}}\right)\right)} \right| \\ \times f_j\left(-\frac{\alpha_{j,k}}{\beta_{j,k}} W\left(1, e^{C^{(j,k)}}\right) |\boldsymbol{\Theta}_j\right), & \text{if } \frac{\alpha_{j,k}}{\beta_{j,k}} < 0. \end{cases} \tag{A.26}$$

where $B^{(j,k)}$ and $I(\cdot)$ are defined in Appendix A.1, $C^{(j,k)} \triangleq \frac{y^{(j,k)}}{\alpha_{j,k}} - \log\left(\frac{\alpha_{j,k}}{\beta_{j,k}}\right)$, $\eta_1 \triangleq \alpha_{j,k} \log\left(\frac{\alpha_{j,k}}{\beta_{j,k}}\right) - \alpha_{j,k}$, and $\eta_2 \triangleq \alpha_{j,k} \log\left(\frac{-\alpha_{j,k}}{\beta_{j,k}}\right) - \alpha_{j,k}$. Therefore we can obtain $\tilde{\sigma}^2_{R,j,k}$ by (A.26).

A.6 Derivation of the Correlation Coefficient

The correlation coefficient ρ in equation (3.26) is derived as

$$\begin{aligned} \rho &= \frac{\mathbb{E}\{\Delta\lambda_f \Delta\lambda_n\} - m_f m_n}{\sigma_f \sigma_n} \\ &= \frac{\mathbb{E}\{\hat{\lambda}_f \hat{\lambda}_n\} - \lambda_f \mathbb{E}\{\hat{\lambda}_n\} - \lambda_n \mathbb{E}\{\hat{\lambda}_f\} + \lambda_f \lambda_n - m_f m_n}{\sigma_f \sigma_n}. \end{aligned} \tag{A.27}$$

Since $\mathbb{E}\{\Delta\lambda_f\} = \mathbb{E}\{\hat{\lambda}_f\} - \lambda_f = m_f$ and $\mathbb{E}\{\Delta\lambda_n\} = \mathbb{E}\{\hat{\lambda}_n\} - \lambda_n = m_n$, we obtain $\mathbb{E}\{\hat{\lambda}_f^2\} = \sigma_f^2 + \mathbb{E}\{\hat{\lambda}_f\}^2 = \sigma_f^2 + (\lambda_f + m_f)^2$. Substituting $\mathbb{E}\{\hat{\lambda}_f^2\}$, $\mathbb{E}\{\hat{\lambda}_f\}$, $\mathbb{E}\{\hat{\lambda}_n\}$ in (A.27), we derive

$$\rho = \frac{(\frac{1-u}{u})(\sigma_f^2 + (\lambda_f + m_f)^2) - \lambda_f(m_n + \lambda_n) - m_f(m_n + \lambda_n)}{(\frac{1-u}{u})\sigma_f^2}$$

$$= 1 - \epsilon,$$

where $\epsilon = \frac{\lambda_f m_n + \lambda_n m_f + \lambda_f \lambda_n + m_f m_n - (\frac{1-u}{u})(\lambda_f + m_f)^2}{(\frac{1-u}{u})\sigma_f^2}$. If m_f and m_n are small enough, then

$$\epsilon \approx \frac{\lambda_f \lambda_n - (\frac{1-u}{u})\lambda_f^2}{(\frac{1-u}{u})\sigma_f^2} = \frac{\lambda_f(\lambda_n - (\frac{1-u}{u})\lambda_f)}{(\frac{1-u}{u})\sigma_f^2} = 0,$$

so that ρ approaches 1.

A.7 Sequence Interchanging Algorithm

Assume we have two sequences $A = (a_1, a_2, \cdots, a_M)$ and $B = (b_1, b_2, \cdots, b_M)$, where A and B are two different permutations of the set $\mathbf{M}$. Denote a_j as the j-th element in A and b_j as the j-th element in B. In order to interchange A to B, we can follow Algorithm 8. Note in each step of interchanging (a_l, a_m) we always follow the rule of $l < m$.

A.8 Cramér-Rao Bound Derivation

We derive the CR bound as given by (4.12) and (4.13) by deriving the Fisher information matrix. Let $\theta = [P_{00}, P_{10}]^T$. The bounds of the estimators are defined as the diagonal terms of the reciprocal of the 2×2 Fisher information matrix, which is written as

$$I_N(P_{00}, P_{10}) = \mathbb{E}_{\mathbf{z}} \left\{ \frac{-\partial^2}{\partial \theta^2} \log L(\mathbf{z}_N) \right\}. \quad \text{(A.28)}$$

The subscript N implies the Fisher information is calculated from N number of traffic samples, and $\mathbb{E}_{\mathbf{z}}\{\cdot\}$ represents the expectation calculated over all permutations of $\mathbf{z}_N$. The bounds are expressed as

$$V_0 = \frac{I_N(P_{00}, P_{10})[2,2]}{|I_N(P_{00}, P_{10})|}, \quad \text{(A.29)}$$

$$V_1 = \frac{I_N(P_{00}, P_{10})[1,1]}{|I_N(P_{00}, P_{10})|}, \quad \text{(A.30)}$$

Algorithm 8 Sequence Interchanging Algorithm

procedure INTERCHANGE(A, B)
    $i \leftarrow 1$
    while $i \leq M$ **do**
        Search k for $a_k == b_i$
        $j \leftarrow k$
        while $j - i \geq 1$ **do**
            $X_j \leftarrow a_j$ ▷ Interchange a_j and a_{j-1} in A
            $X_{j-1} \leftarrow a_{j-1}$
            $a_j \leftarrow X_{j-1}$
            $a_{j-1} \leftarrow X_j$
            $j \leftarrow j - 1$
        end while
        $i \leftarrow i + 1$
    end while
    return A
end procedure

where $|\cdot|$ represents determinant of a matrix. In order to derive the elements in the Fisher information matrix, we need to introduce Lemma 8 beforehand.

Lemma 8 *The expectation of transition numbers with $n \geq 1$ PU traffic samples given a discrete-time Markov chain $\mathbf{z}_n$ are expressed as*

$$\mathbb{E}_{\mathbf{z}}\{n_0(\mathbf{z}_n)\} = P_0 P_{00}(n-1), \tag{A.31}$$

$$\mathbb{E}_{\mathbf{z}}\{n_1(\mathbf{z}_n)\} = P_0 P_{01}(n-1), \tag{A.32}$$

$$\mathbb{E}_{\mathbf{z}}\{n_2(\mathbf{z}_n)\} = P_1 P_{10}(n-1), \tag{A.33}$$

$$\mathbb{E}_{\mathbf{z}}\{n_3(\mathbf{z}_n)\} = P_1 P_{11}(n-1). \tag{A.34}$$

Proof: Consider the transition number $n_0(\mathbf{z}_n)$. Note that $n_0(\mathbf{z}_n)$ is a function of the state variable and its expectation can be expressed recursively as

$$\mathbb{E}_{\mathbf{z}}\{n_0(\mathbf{z}_n)\} = \mathbb{E}_{\mathbf{z}}\{n_0(\mathbf{z}_{n-1})\} + P_0 P_{00}, \ \forall n \geq 3, \tag{A.35}$$

with the initial condition $\mathbb{E}_{\mathbf{z}}\{n_0(\mathbf{z}_2)\} = P_0 P_{00}$. By solving (A.35), we derive $\mathbb{E}_{\mathbf{z}}\{n_0(\mathbf{z}_n)\} = P_0 P_{00}(n-1)$. Similarly, we apply the same procedure for $\mathbb{E}_{\mathbf{z}}\{n_1(\mathbf{z}_n)\}$, $\mathbb{E}_{\mathbf{z}}\{n_2(\mathbf{z}_n)\}$, $\mathbb{E}_{\mathbf{z}}\{n_3(\mathbf{z}_n)\}$ using its own recursive equation, and we verify the results in (A.32)–(A.34), respectively. Note that Lemma 8 also shows that the ML estimators for the transition probabilities are unbiased, i.e., $\mathbb{E}\{\hat{P}_{00}\} = P_{00}$ and $\mathbb{E}\{\hat{P}_{10}\} = P_{10}$. Therefore, we can apply CR bound as our lower bound for the variance of the estimators. The elements in the Fisher information matrix are written as

$$\begin{aligned} I_N(P_{00}, P_{10})[1,1] &= \mathbb{E}_{\mathbf{z}}\left\{\frac{-\partial^2}{\partial P_{00}^2}\log L(\mathbf{z}_N)\right\} \\ &= \mathbb{E}_{\mathbf{z}}\left\{\frac{n_0(\mathbf{z}_N)}{P_{00}^2} + \frac{n_1(\mathbf{z}_N)}{(1-P_{00})^2}\right\} \\ &= \frac{P_0(N-1)}{P_{00}(1-P_{00})}, \end{aligned} \tag{A.36}$$

$$\begin{aligned} I_N(P_{00}, P_{10})[2,2] &= \mathbb{E}_{\mathbf{z}}\left\{\frac{-\partial^2}{\partial P_{10}^2}\log L(\mathbf{z}_N)\right\} \\ &= \mathbb{E}_{\mathbf{z}}\left\{\frac{n_2(\mathbf{z}_N)}{P_{10}^2} + \frac{n_3(\mathbf{z}_N)}{(1-P_{10})^2}\right\} \\ &= \frac{P_1(N-1)}{P_{10}(1-P_{10})}, \end{aligned} \tag{A.37}$$

$$I_N(P_{00}, P_{10})[1,2] = \mathbb{E}_{\mathbf{z}}\left\{\frac{-\partial^2}{\partial P_{00}P_{10}}\log L(\mathbf{z}_N)\right\} = 0, \tag{A.38}$$

$$I_N(P_{00}, P_{10})[2,1] = \mathbb{E}_{\mathbf{z}}\left\{\frac{-\partial^2}{\partial P_{10}P_{00}}\log L(\mathbf{z}_N)\right\} = 0, \tag{A.39}$$

by applying (A.31)–(A.34) into (A.36)–(A.37) with N traffic samples, and the determinant is

$$\begin{aligned} |I_N(P_{00}, P_{10})| &= I_N(P_{00}, P_{10})[1,1] I_N(P_{00}, P_{10})[2,2] \\ &= \frac{P_0 P_1 (N-1)^2}{P_{00}(1-P_{00})P_{10}(1-P_{10})}. \end{aligned} \tag{A.40}$$

Finally, applying (A.36), (A.37) and (A.40) into (A.29)–(A.30), we obtain the CR bound as

$$V_0 = \frac{P_{00}(1-P_{00})}{P_0(N-1)}, \tag{A.41}$$

$$V_1 = \frac{P_{10}(1-P_{10})}{P_1(N-1)}. \tag{A.42}$$

A.9 Proof of Concavity of $\kappa_{\mathbf{s}}^{[o]}$

We want to prove that $\kappa_{\mathbf{s}}^{[o]} = \prod_{i=1}^{M-1} Q\left(\frac{\epsilon_i P_{b_{i+1}0}^{(a_{i+1})} - P_{b_i0}^{(a_i)}}{\sqrt{\sigma_{a_i}^2 + \epsilon_i^2 \sigma_{a_{i+1}}^2}}\right)$ is a concave function. Define $g_i \triangleq \frac{P_{b_i0}^{(a_i)}\left(1-P_{b_i0}^{(a_i)}\right)}{P_{b_i}^{(a_i)}\left(P_{b_i0}^{(a_i)} - \epsilon_i P_{b_{i+1}0}^{(a_{i+1})}\right)^2} \geq 0$, $h_i \triangleq \frac{\epsilon_i^2 P_{b_{i+1}0}^{(a_{i+1})}\left(1-P_{b_{i+1}0}^{(a_{i+1})}\right)}{P_{b_{i+1}}^{(a_{i+1})}\left(P_{b_i0}^{(a_i)} - \epsilon_i P_{b_{i+1}0}^{(a_{i+1})}\right)^2} \geq 0$, and $P_{b_i}^{(a_i)}$ as the absence/presence probability of the a_i-th channel such that $Q\left(\frac{\epsilon_i P_{b_{i+1}0}^{(a_{i+1})} - P_{b_i0}^{(a_i)}}{\sqrt{\sigma_{a_i}^2 + \epsilon_i^2 \sigma_{a_{i+1}}^2}}\right) = Q\left(\frac{-1}{\sqrt{\frac{g_i}{N_i - 1} + \frac{h_i}{N_{i+1}-1}}}\right) \triangleq f(N_i, N_{i+1})$. To prove $f(N_i, N_{i+1})$ is a concave function we need to introduce the following Theorem.

Theorem 5 *Any real function in the form $f(x,y) = Q\left(\frac{-1}{\sqrt{\frac{g}{x}+\frac{h}{y}}}\right)$ is a concave function, given $g, h, x, y \geq 0$.*

Proof: The sufficient and necessary condition for $f(x,y)$ to be concave is $\nabla^2 f(x,y) \leq 0$, i.e., a negative semidefinite matrix. Therefore, we need to calculate all the elements in $\nabla^2 f(x,y)$. Note that $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{m^2}{2}} dm = \int_{-\infty}^{-x} \frac{1}{\sqrt{2\pi}} e^{-\frac{m^2}{2}} dm$, thus $f(x,y) = \int_{-\infty}^{\left(\frac{g}{x}+\frac{h}{y}\right)^{-\frac{1}{2}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{m^2}{2}} dm$. By the first fundamental theorem of calculus, the first partial derivative of $f(x,y)$ is expressed as

$$\frac{\partial f}{\partial x} = \frac{g}{2\sqrt{2\pi}} x^{-2} \left(\frac{g}{x}+\frac{h}{y}\right)^{-\frac{3}{2}} e^{-\frac{1}{2}\left(\frac{g}{x}+\frac{h}{y}\right)^{-1}}, \tag{A.43}$$

$$\frac{\partial f}{\partial y} = \frac{h}{2\sqrt{2\pi}} y^{-2} \left(\frac{g}{x}+\frac{h}{y}\right)^{-\frac{3}{2}} e^{-\frac{1}{2}\left(\frac{g}{x}+\frac{h}{y}\right)^{-1}}, \tag{A.44}$$

and the second partial derivative of $f(x,y)$ is expressed as

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} = {} & \frac{g}{2\sqrt{2\pi}} x^{-4} \left(\frac{g}{x}+\frac{h}{y}\right)^{-\frac{7}{2}} e^{-\frac{1}{2}\left(\frac{g}{x}+\frac{h}{y}\right)^{-1}} \\ & \times \left(-2x\left(\frac{g}{x}+\frac{h}{y}\right)^2 + \frac{3}{2}\left(\frac{g}{x}+\frac{h}{y}\right) g - \frac{1}{2} g\right),\end{aligned} \tag{A.45}$$

$$\begin{aligned}\frac{\partial^2 f}{\partial y^2} = {} & \frac{h}{2\sqrt{2\pi}} y^{-4} \left(\frac{g}{x}+\frac{h}{y}\right)^{-\frac{7}{2}} e^{-\frac{1}{2}\left(\frac{g}{x}+\frac{h}{y}\right)^{-1}} \\ & \times \left(-2y\left(\frac{g}{x}+\frac{h}{y}\right)^2 + \frac{3}{2}\left(\frac{g}{x}+\frac{h}{y}\right) h - \frac{1}{2} h\right),\end{aligned} \tag{A.46}$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{gh}{2\sqrt{2\pi}} x^{-2} y^{-2} \left(\frac{g}{x} + \frac{h}{y}\right)^{-\frac{7}{2}} e^{-\frac{1}{2}\left(\frac{g}{x}+\frac{h}{y}\right)^{-1}}$$
$$\times \left(\frac{3}{2}\left(\frac{g}{x} + \frac{h}{y}\right) - \frac{1}{2}\right). \tag{A.47}$$

Thus, we obtain $\nabla^2 f(x, y) = \frac{1}{2\sqrt{2\pi}}(\frac{g}{x} + \frac{h}{y})^{-\frac{7}{2}}\mathbf{W}$, where $\mathbf{W}$ is a 2×2 matrix with W_{ij} representing its i-th row and j-th column element, $i, j \in \{1, 2\}$. The elements W_{ij} are calculated to be

$$W_{11} = g x^{-4}\left(-2x\left(\frac{g}{x} + \frac{h}{y}\right)^2 + \frac{3}{2}\left(\frac{g}{x} + \frac{h}{y}\right) g - \frac{1}{2} g\right), \tag{A.48}$$

$$W_{22} = h y^{-4}\left(-2y\left(\frac{g}{x} + \frac{h}{y}\right)^2 + \frac{3}{2}\left(\frac{g}{x} + \frac{h}{y}\right) h - \frac{1}{2} h\right), \tag{A.49}$$

$$W_{12} = W_{21} = gh x^{-2} y^{-2}\left(\frac{3}{2}\left(\frac{g}{x} + \frac{h}{y}\right) - \frac{1}{2}\right). \tag{A.50}$$

To prove $\nabla^2 f(x, y) \le 0$ is equivalent to prove $\mathbf{W} \le 0$. We introduce the following Lemma which gives a sufficient and necessary condition of a 2×2 matrix being positive semidefinite.

Lemma 9 *Given a 2×2 matrix $\mathbf{W}$, $\mathbf{W}$ is positive semidefinite if and only if $W_{11} \ge 0$ and $|\mathbf{W}| \ge 0$.*

Proof: See [Lau05, Ch. 10]. Note that

$$-W_{11} = g x^{-4}\left(\left(\frac{g}{x} + \frac{h}{y}\right)\left(\frac{2xh}{y} + \frac{g}{2}\right) + \frac{g}{2}\right) \ge 0, \tag{A.51}$$

$$\begin{aligned} |-\mathbf{W}| &= W_{11}W_{22} - W_{12}W_{21} \\ &= gh x^{-4} y^{-4}\left(\frac{g}{x} + \frac{h}{y}\right)\left((xh + yg)\left(\frac{g}{x} + \frac{h}{y}\right)^2\right. \\ &\quad \left. + (xh + yg)\left(\frac{g}{x} + \frac{h}{y}\right) + 1\right) \ge 0, \end{aligned} \tag{A.52}$$

from Lemma 9 we have $-\mathbf{W} \ge 0$, which implies $\mathbf{W} \le 0$. Therefore $\nabla^2 f(x, y) \le 0$ completes the proof.

Since $f(N_i, N_{i+1})$ is a concave function by Theorem 5, $\log f(N_i, N_{i+1})$ is also a concave function by the composition rule [BV04]. Hence, the sum of concave functions $\sum_{i=1}^{M-1} \log f(N_i, N_{i+1}) =$

$\log \prod_{i=1}^{M-1} f(N_i, N_{i+1})$ is a concave function. Taking the exponential of $\log \prod_{i=1}^{M-1} f(N_i, N_{i+1})$ we can derive $\prod_{i=1}^{M-1} f(N_i, N_{i+1})$, showing it is a concave function again by the composition rule, which completes our proof.

A.10 Derivation of Transition Probabilities Considering Spectrum Sensing Errors

First we note that the probability vectors for future state $\hat{s}$ and current state s can be written as

$$\begin{bmatrix} \Pr\{\hat{s}=0\} \\ \Pr\{\hat{s}=1\} \end{bmatrix} = \begin{bmatrix} P_{00} & P_{10} \\ P_{01} & P_{11} \end{bmatrix} \begin{bmatrix} \Pr\{s=0\} \\ \Pr\{s=1\} \end{bmatrix}. \tag{A.53}$$

Furthermore, the probability vector under sensing errors for estimated current state $\tilde{s}$ can be shown as

$$\begin{bmatrix} \Pr\{\tilde{s}=0\} \\ \Pr\{\tilde{s}=1\} \end{bmatrix} = \begin{bmatrix} 1-P_f & P_m \\ P_f & 1-P_m \end{bmatrix} \begin{bmatrix} \Pr\{s=0\} \\ \Pr\{s=1\} \end{bmatrix}. \tag{A.54}$$

Now, consider a reverse Markov chain describing transition from $\tilde{s}$ to s. The probability vectors for this reverse Markov chain are

$$\begin{bmatrix} \Pr\{s=0\} \\ \Pr\{s=1\} \end{bmatrix} = \begin{bmatrix} 1-P_f & P_m \\ P_f & 1-P_m \end{bmatrix} \begin{bmatrix} \Pr\{\tilde{s}=0\} \\ \Pr\{\tilde{s}=1\} \end{bmatrix}. \tag{A.55}$$

Embedding (A.55) into (A.53) we can derive the probabilistic relation for $\hat{s}$ and $\tilde{s}$ as

$$\begin{aligned} \begin{bmatrix} \Pr\{\hat{s}=0\} \\ \Pr\{\hat{s}=1\} \end{bmatrix} &= \begin{bmatrix} P_{00} & P_{10} \\ P_{01} & P_{11} \end{bmatrix} \begin{bmatrix} 1-P_f & P_m \\ P_f & 1-P_m \end{bmatrix} \begin{bmatrix} \Pr\{\tilde{s}=0\} \\ \Pr\{\tilde{s}=1\} \end{bmatrix} \\ &\triangleq \mathbf{Q} \times \begin{bmatrix} \Pr\{\tilde{s}=0\} \\ \Pr\{\tilde{s}=1\} \end{bmatrix}, \end{aligned} \tag{A.56}$$

where $\mathbf{Q} \triangleq [Q_{\tilde{s}\hat{s}}]_{2\times 2}$ is the observed transition probability matrix given as

$$\mathbf{Q} = \begin{bmatrix} (1-P_f)P_{00} + P_f P_{10} & P_m P_{00} + (1-P_m)P_{10} \\ (1-P_f)P_{01} + P_f P_{11} & P_m P_{01} + (1-P_m)P_{11} \end{bmatrix}. \tag{A.57}$$

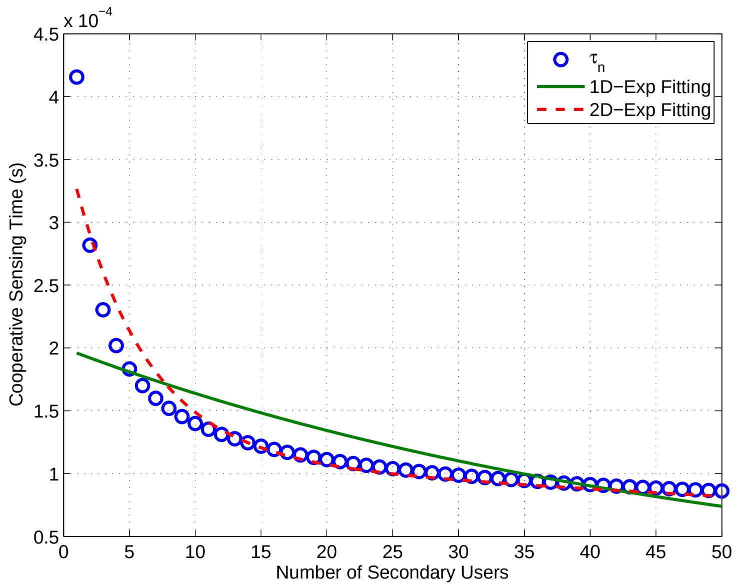


Figure A.1: Fitting τ_n with a 1-degree ae^{-bx} and a 2-degree $ae^{-bx} + ce^{-dx}$ exponential functions using emphOR fusion rule with used parameters $Q_d = 0.9$, $Q_d = 0.1$, $SNR = -5$ dB, and $f_s = 5$ kHz. Fitting results are $a \approx 0.0022$ and $b \approx 0.0265$ for 1-D exponential, and $a \approx 0.0030$, $b \approx 0.3088$, $c \approx 0.0015$ and $d \approx 0.0136$ for the 2-D exponential fitting.

Therefore, $\Pr\{\hat{s} = 0\} = Q_{10} = P_m P_{00} + (1 - P_m)P_{10}$ if $\tilde{s} = 1$, and $\Pr\{\hat{s} = 0\} = Q_{00} = (1 - P_f)P_{00} + P_f P_{10}$ if $\tilde{s} = 0$.

A.11 Analytical Approximation for the General Parallel Strategy

We provided an analytical analysis for the case with homogeneous channels. For a more general case, it can be seen that the convexity of the throughput function of the parallel strategy depends on τ_{i,k_i} values, so it should be investigated whether the function of sensing time versus the number of collaborators is a convex function or not. For this aim, we define $f(x) = Q^{-1}(\sqrt[k]{1 - Q_f}) - Q^{-1}(\sqrt[k]{1 - Q_d})$ (please see equation (5.2)). Examining the Hessian of this function shows that $f(k)$ is not convex unless if there are enough users. The exact threshold for the number of users to have a convex function depends on the other parameters such as Q_d and Q_f. Therefore, for a large number of users, if we approximate $f(k)$ by an exponential function in the form $ae^{-bk}, a > 0, b > 0$, the optimal user allocation can be found analytically by convex optimization, which is equal to:

$$k_i^* = \frac{\ln(abC_i(1-u_i))}{b} - \frac{1}{Mb}\sum_{j=1}^{M} abC_j(1-u_j) + \frac{bN}{M}, \tag{A.58}$$

where $1 \leq i \leq M$. Fig. A.1 confirms that fitting with an exponential function is not accurate when the number of users is low, but is acceptable when there are a large number of users. In practical scenarios, it is expected to have a large number of users, otherwise the problem can easily be solved by a Brute-Force method, so that the exponential approximation and consequently the analytical derivation for optimal assignment are very helpful.

A.12 Derivation of the Variance for the Average Estimator

The variance of the proposed $\hat{u}$ can be written as

$$\begin{aligned} \mathrm{Var}\{\hat{u}\} &= \mathbb{E}\{[\hat{u} - \mathbb{E}\{\hat{u}\}]^2\} \\ &= \mathbb{E}\left\{\left(\frac{1}{W}\sum_{i=1}^{W} z_i\right)^2\right\} - u^2 \\ &= \frac{1}{W^2}\sum_{i=1}^{W}\mathbb{E}\left\{z_i^2\right\} + \frac{2}{W^2}\sum_{i=1}^{W-1}\sum_{j=1}^{W-i}\mathbb{E}\left\{z_i z_{i+j}\right\} - u^2. \end{aligned} \tag{A.59}$$

The expression $\mathbb{E}\left\{z_i z_{i+j}\right\}$ in equation (A.59) is the correlation terms between z_i and z_{i+j}, and we denoted as $R_{i,i+j}$. For $R_{i,i+j}$, $\forall j \geq 0$, we can solve it by its recursive definition, i.e.,

$$\begin{aligned} R_{i,i+j} &= \mathbb{E}\{z_i z_{i+j}\} = \Pr\{z_i = 1, z_{i+j} = 1\} \\ &= \Pr\{z_i = 1, z_{i+j-1} = 1\} \times P_{11} \\ &+ \Pr\{z_i = 1, z_{i+j-1} = 0\} \times P_{01} \\ &= R_{i,i+j-1} \times P_{11} + (u - R_{i,i+j-1}) \times P_{01} \\ &= R_{i,i+j-1} \times r + u \times P_{01}, \end{aligned} \tag{A.60}$$

where $r = P_{11} - P_{01}$. The initial condition for the recursive equation (A.60) is $R_{i,i} = \mathbb{E}\{z_i^2\} = u$. Hence, solving the recursive equation (A.60) gives the result

$$R_{i,i+j} = \frac{uP_{01}(1-r^j)}{1-r} + ur^j. \tag{A.61}$$

By plugging equation (A.61) into equation (A.59), we can simplify it as the result shown in equation (5.13).

A.13 Proof of Lemma 3

First consider ROP 2. Define the objective function as $f(\mathbf{W}) = \sum_{i=1}^{M} W_i$, and the constraint function as $g(\mathbf{W}) = \mathrm{Var}\{\hat{R}(\mathcal{S})\}$. Assume we obtain the optimal solution $\mathbf{W}^* = (W_1^*, W_2^*, \cdots, W_M^*)$ to minimize the function $f(\mathbf{W})$. Define $\epsilon = f(\mathbf{W}^*)$. This means for any other feasible solutions $\mathbf{W} \neq \mathbf{W}^*$ satisfying $g(\mathbf{W}) \leq \eta$, we always have $f(\mathbf{W}) \geq \epsilon$. The above statement is equivalent to if we find a solution $\mathbf{W}$ such that $f(\mathbf{W}) \leq \epsilon$, we should always have the constraint being violated, i.e., $g(\mathbf{W}) \geq \eta$. Among all of these possible $\mathbf{W'}$**s**, we want to find the one with minimum $g(\mathbf{W})$, which forms the proposed ROP 3. We can see that since all the possible $\mathbf{W} \neq \mathbf{W}^*$ gives $g(\mathbf{W}) \geq \eta$, while we know $g(\mathbf{W}^*) \leq \eta$ by the original constraint, we can conclude that $\mathbf{W}^*$ is still the optimal solution for ROP 3 to minimize $g(\mathbf{W})$. Note that we can also proof the other equivalence by the same technique, i.e., given the optimal solution for ROP 3, we can show that it is also the optimal solution for ROP 2.

References

[AFS13] A. Azarfar, J. Frigon, and B. Sanso. "User-differentiated Channel Recovery in Multi-channel Cognitive Radio Networks." In *International Conference on DRCN*, pp. 242–249, March 2013.

[AK14] Peter Anker and Taco Kluver, April 24, 2014. private communication.

[Ber05] D. P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 2005. Vol. I, 3rd Edition.

[BGR14] P. Balina, K. Garg, and S. Rao. "Spectrum Occupancy Measurements and Analysis in a Rural Area Setting." In *IEEE International Conference on Parallel, Distributed and Grid Computing*, December 2014.

[BV94] C. W. Baum and V. V. Veeravalli. "A Sequential Procedure for Multihypothesis Testing." *IEEE Trans. Inform. Theory*, **40**(6):1994–2007, November 1994.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[BZ09] K. Ben Letaief and W. Zhang. "Cooperative Communications for Cognitive Radio Networks." *Proc. IEEE*, **97**(5):878–893, May 2009.

[Cas13] Rui Castro. "Lectures 12 and 13–Complexity Penalized Maximum Likelihood Estimation." Applied Statistics Course, TU Eindhoven, October 14–17, 2013.

[CGH96] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. "On the Lambert W function." *Advances in Computational Mathematics*, **5**(1):329–359, 1996.

[CJ00] Robert M. Corless and David J. Jeffrey. "On the Wright ω function." Department of applied mathematics, tr-00-12, University of Western Ontario, 2000.

[CRS07] R. I. C. Chiang, G. B. Rowe, and K. W. Sowerby. "A Quantitative Analysis of Spectral Occupancy Measurements for Cognitive Radio." In *IEEE Vehicular Technology Conference (VTC) Spring*, April 2007.

[CTB06] D. Cabric, A. Tkachenko, and R. W. Brodersen. "Spectrum Sensing Measurements of Pilot, Energy, and Collaborative Detection." In *IEEE Military Commun. Conf. (MILCOM)*, October 2006.

[CVF11] S. Contreras, G. Villardi, R. Funada, and H. Harada. "An Investigation into the Spectrum Occupancy in Japan in the Context of TV White Space Systems." In *Proc. of the IEEE Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, June 2011.

[CYZ09] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li. "Mining Spectrum Usage Data: A Large-scale Spectrum Measurement Study." In *Proc. of the 15th annual international conference on Mobile computing and networking (MobiCom)*, September 2009.

[CZ11] H. T. Cheng and W. Zhuang. "Simple Channel Sensing Order in Cognitive Radio Networks." *IEEE J. Select. Areas Commun.*, **29**(4):676–688, April 2011.

[DAB07] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su. "Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends." *IET Commun.*, **1**(2):137–156, April 2007.

[DB12] D. DePov and T. Bose. "Predictive Behavior Classification for Cognitive Radio: Introduction and Preliminary Results." In *Proc. of the IEEE Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, Stockholm, Sweden, June 18–20, 2012.

[DRW09] D. Datla, R. Rajbanshi, Alexander M. Wyglinski, and G.J. Minden. "An Adaptive Spectrum Sensing Architecture for Dynamic Spectrum Access Networks." *IEEE Trans. Wireless Commun.*, **8**(8):4211–4219, August 2009.

[GB11] M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 1.21.* CVX Research, Inc., 2011.

[GLP13a] W. Gabran, C.-H. Liu, P. Pawełczak, and D. Cabric. "Primary User Traffic Estimation for Dynamic Spectrum Access." *IEEE J. Select. Areas Commun.*, **31**(3):544–558, March 2013.

[GLP13b] Wesam Gabran, Chun-Hao Liu, Przemysław Pawełczak, and Danijela Cabric. "Primary User Traffic Estimation for Dynamic Spectrum Access." *IEEE J. Select. Areas Commun.*, **31**(3):544–558, March 2013.

[GPL13] W. Gabran, Przemysław Pawełczak, C.-H. Liu, and D. Cabric. "Blind Estimation of Primary User Traffic Parameters in Dynamic Spectrum Access Networks Under Sensing Errors." In *IEEE International Conf. on Communications (ICC)*, June 2013.

[GS05] A. Ghasemi and E.S. Sousa. "Collaborative Spectrum Sensing for Opportunistic Access in Fading Environments." In *Proc. of the IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pp. 131–136, November 2005.

[Hay05] Simon Haykin. "Cognitive Radio: Brain-Empowered Wireless Communications." *IEEE J. Select. Areas Commun.*, **23**(2):201–220, February 2005.

[HLK13] M. Hoyhtya, J. Lehtomaki, J. Kokkoniemi, M. Matinmikko, and A . Mammela. "Measurements and Analysis of Spectrum Occupancy with Several Bandwidths." In *IEEE International Conf. on Commun. (ICC)*, June 2013.

[HPM10] M. Hoyhtya, S. Pollin, and A. Mammela. "Classification-based Predictive Channel Selection for Cognitive Radios." In *IEEE International Conf. on Commun. (ICC)*, Cape Town, South Africa, June 23–27, 2010.

[HSM11] M. Hoyhtya, H. Sarvanko, M. Matinmikko, and A. Mammela. "Autocorrelation-Based Traffic Pattern Classification for Cognitive Radios." In *IEEE Vehicular Technology Conference (VTC) Fall*, San Francisco, CA, USA, September 5–8, 2011.

[HSN12] L. Hesham, A. Sultan, M. Nafie, and F. Digham. "Distributed Spectrum Sensing with Sequential Ordered Transmissions to a Cognitive Fusion Center." *IEEE Trans. Signal Processing*, **60**(5):2524–2538, May 2012.

[IKO08] M. H. Islam, C.L. Koh, S. W. Oh, X. Qing, Y. Y. Lai, C. Wang, Y.-C. Liang, B. E. Toh, F. Chin, G. L. Tan, and W. Toh. "Spectrum Survey in Singapore: Occupancy Measurements and Analyses." In *Proc. of the IEEE Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWN-COM)*, May 2008.

[JL12a] E. Jung and X. Liu. "Opportunistic Spectrum Access in Multiple-Primary-User Environments Under the Packet Collision." *IEEE/ACM Trans. Networking*, **20**(2):501–514, April 2012.

[JL12b] Eric Jung and Xin Liu. "Opportunistic Spectrum Access in Multiple-Primary-User Environments Under the Packet Collision Constraint." *IEEE/ACM Trans. Networking*, **20**(2):501–514, April 2012.

[JLF09] H. Jiang, L. Lai, R. Fan, and V. Poor. "Optimal Selection of Channel Sensing Order in Cognitive Radio." *IEEE Trans. Commun.*, **8**(1):297–307, January 2009.

[KKP13] A. Kliks, P. Kryszkiewicz, J. Perez-Romero, A. Umbert, and F. Casadevall. "Spectrum Occupancy in Big Cities-Comparative Study-Measurement Campaigns in Barcelona and Poznan." In *Proc. of the International Symposium on Wireless Communication Systems*, August 2013.

[KS08a] H. Kim and K. G. Shin. "Efficient Discovery of Spectrum Opportunities with Mac-layer Sensing in Cognitive Radio Networks." *IEEE Trans. Mobile Comput.*, **7**(5):553–545, May 2008.

[KS08b] Hyoil Kim and Kang G. Shin. "Fast Discovery of Spectrum Opportunities in Cognitive Radio Networks." In *Proc. of the IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, October 2008.

[KYY12] Vinod Kone, Lei Yang, Xue Yang, Ben Y. Zhao, and Haitao Zheng. "The Effectiveness of Opportunistic Spectrum Access: A Measurement Study." *IEEE/ACM Trans. Networking*, **20**(6):2005–2016, December 2012.

[Lau05] A. J. Laub. *Matrix Analysis for Scientists & Engineers.* Society for Industrial and Applied Mathematics, 2005.

[LC11] M. Lopez-Benitez and F. Casadevall. "Discrete-Time Spectrum Occupancy Model based on Markov Chain and Duty Cycle Models." In *Proc. of the IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, May 2011.

[LC13] M. Lopez-Benitez and F. Casadevall. "Time-dimension Models of Spectrum Usage for the Analysis, Design and Simulation of Cognitive Radio Networks." *IEEE Trans. Veh. Technol.*, **62**(5):2091–2104, June 2013.

[LCL11] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen. "Cognitive Radio Networking and Communications: An Overview." *IEEE Trans. Veh. Technol.*, **60**(7):3386–3407, September 2011.

[LGC12] C.-H. Liu, W. Gabran, and D. Cabric. "Prediction of Exponentially Distributed Primary User Traffic for Dynamic Spectrum Access." In *IEEE Global Telecommunications Conference (GLOBECOM)*, Anaheim, CA, USA, December 3–7, 2012.

[LJX08] T. Luo, T. Jiang, W. Xiang, and H. H. Chen. "A Subcarriers Allocation Scheme for Cognitive Radio Systems Based on Multi-Carrier Modulation." *IEEE Trans. Wireless Commun.*, **7**(9):3335–3340, September 2008.

[Lop13] M. Lopez-Benitez. "Can Primary Activity Statistics in Cognitive Radio be Estimated under Imperfect Spectrum Sensing?" In *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, London, UK, September 8–11, 2013.

[LRP13] C.-H. Liu, E. Rebeiz, P. Pawelczak, and D. Cabric. "Primary User Traffic Classification in Dynamic Spectrum Access Networks." In *IEEE Global Telecommunications Conference (GLOBECOM)*, Atlanta, GA, USA, December 9–13, 2013.

[LS13] M. V. d. Lima and L. d. S. Mello. "Cognitive Radio Simulation Based on Spectrum Occupancy Measurements at One Site in Brazil." In *SBMO/IEEE MTT-S International Microwave & Optoelectronics Conference*, August 2013.

[LTP13] C.-H. Liu, Jason A. Tran, Przemysław Pawełczak, and Danijela Cabric. "Traffic-Aware Channel Sensing Order in Dynamic Spectrum Access Networks." *IEEE J. Select. Areas Commun.*, **31**(11):2312–2323, November 2013.

[LUC09] M. Lopez-Benitez, A. Umbert, and F. Casadevall. "Evaluation of Spectrum Occupancy in Spain for Cognitive radio Applications." In *IEEE Vehicular Technology Conference (VTC) Spring*, April 2009.

[LWC14] Q. Liu, X. Wang, and Y. Cui. "Robust and Adpative Scheduling of Sequential Periodic Sensing for Cognitive Radios." *IEEE J. Select. Areas Commun.*, **32**(3):503–515, March 2014.

[LZP08] Y.-C Liang, Y. Zeng, E. C. Y. Peh, and A. T. Hoang. "Sensing-Throughput Tradeoff for Cognitive Radio Networks." *IEEE Trans. Wireless Commun.*, **7**(4):1326–1337, April 2008.

[Mil11] R. B. Millar. *Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB.* Wiley, 2011.

[MM99] III Mitola, J. and Jr. Maguire, G.Q. "Cognitive Radio: Making Software Radios More Personal." *IEEE Personal Communications*, **6**(4):13–18, August 1999.

[MM14] J. Misic and V. B. Misic. "Probability Distribution of Spectral Hole Duration in Cognitive Networks." In *IEEE International Conf. on Computer Commun. (INFOCOM)*, May 2014.

[MRA12] M. Mehdawi, N. Riley, M. Ammar, and M. Zolfaghari. "Comparing Historical and Current Spectrum Occupancy Measurements in the Context of Cognitive Radio." In *Telecommunications Forum (TELFOR)*, November 2012.

[MT90] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations.* Wiley, 1990.

[MTM06] Mark A. McHenry, Peter A. Tenhula, Dan McCloskey, Dennis A. Roberson, and Cynthia S. Hood. "Chicago Spectrum Occupancy Measurements & Analysis and a Long-term Studies Proposal." In *Proceedings of the first international workshop on Technology and policy for accessing spectrum*, TAPAS '06, New York, NY, USA, 2006. ACM.

[NA08] T. T. T. Nguyen and Grenville Armitage. "A Survey of Techniques for Internet Traffic Classification using Machine Learning." *IEEE Communications Surveys & Tutorials*, **10**(4):56–76, Fourth Quarter 2008.

[NLW10] G. Noh, J. Lee, H. Wang, S. Kim, S. Choi, and D. Hong. "Throughput Analysis and Optimization of Sensing-Based Cognitive Radio Systems With Markovian Traffic." *IEEE Trans. Veh. Technol.*, **59**(8):4163–4169, October 2010.

[Pax98] Vern Paxson. "Empirically-Derived Analytic Models of Wide-Area TCP Connections." *IEEE/ACM Trans. Networking*, **2**(4):316–336, August 1998.

[Pen01] W. D. Penny. "Kullback-Leibler Divergences of Normal, Gamma, Dirichlet and Wishart Densities." Wellcome department of cognitive neurology, University of College London, 2001.

[PLG09] E. C. Y. Peh, Y.-C. Liang, Y. L. Guan, and Y. Zeng. "Optimization of Cooperative Sensing in Cognitive Radio Networks: A Sensing-Throughput Tradeoff View." *IEEE Trans. Veh. Technol.*, **58**(9):5294–5299, November 2009.

[PMP14] Marko Palola, Marja Matinmikko, Jarmo Prokkola, Miia Mustonen, Marjo Heikkilä, Tero Kippola, Seppo Yrjlä, Vesa Hartikainen, Lucia Tudose, Arto Kivinen, Jarkko Paavola, and Kari Heiska. "Live Field Trial of Licensed Shared Access (LSA) Concept using LTE Network in 2.3 GHz Band." In *Proc. of the IEEE*

Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), McLean, VA, USA, April 1–4, 2014.

[Pol02] David Pollard. *A User's Guide to Measure Theoretic Probability.* Cambridge University Press, 2002.

[PPS09] P. Pawelczak, S. Pollin, H. So, A. Bahai, R. Prasad, and R. Hekmat. "Performance Analysis of Multichannel Medium Access Control Algorithms for Opportunistic Spectrum Access." *IEEE Trans. Veh. Technol.*, **58**(6):3014–3031, July 2009.

[PPS11] K. Patil, R. Prasad, and K. Skouby. "A Survey of Worldwide Spectrum Occupancy Measurement Campaigns for Cognitive Radio." In *International Conference on Devices and Communications (ICDeCom)*, February 2011.

[RG08] A. Ribeiro and G. B. Giannakis. "Robust Stochastic Routing and Scheduling for Wireless Ad-Hoc Networks." In *IEEE Conference on Wireless Communications and Mobile Computing*, August 2008.

[RW13] Kui Ren and Qian Wang. "Opportunistic Spectrum Access: From Stochastic Channels To Non Stochastic Channels." *IEEE Wireless Commun. Mag.*, **20**(3):128–135, June 2013.

[SCL04] Duncan S. Sharp, Nikola Cackov, Nenad Lasković, Qing Shao, and Ljiljana Trajković. "Analysis of Public Safety Traffic on Trunked Land Mobile Radio Systems." *IEEE J. Select. Areas Commun.*, **22**(7):1197–1205, September 2004.

[SCZ10a] C. Song, D. Chen, and Q. Zhang. "Understand the Predictability of Wireless Spectrum: A Large-Scale Empirical Study." In *IEEE International Conf. on Commun. (ICC)*, May 2010.

[SCZ10b] Chengqi Song, Dawei Chen, and Qian Zhang. "Understand the Predictability of Wireless Spectrum: A Large-Scale Empirical Study." In *IEEE International Conf. on Commun. (ICC)*, pp. 1–5, May 2010.

[Sim02] M. K. Simon. *Probability Distributions Involving Gaussian Random Variables.* Kluwer Academic, 2002.

[SRK15] S. Subramaniam, H. Reyes, and N. Kaabouch. "Spectrum Occupancy Measurement: An Autocorrelation based Scanning Technique using USRP." In *IEEE Wireless and Microwave Technology Conference*, April 2015.

[SS14] M. Sharma and A. Sahoo. "Stochastic Model Based Opportunistic Channel Access in Dynamic Spectrum Access Networks." *IEEE Trans. Mobile Comput.*, **13**(7):1625–1639, July 2014.

[SZ08] H. Su and X. Zhang. "Cross-Layer Based Opportunistic MAC Protocols for QoS Provisionings Over Cognitive Radio Wireless Networks." *IEEE J. Select. Areas Commun.*, **26**(1):118–129, January 2008.

[TPC11] Michael Timmers, Sofie Pollin, and Francky Catthoor. "On the Value of Prediction in Opportunistic Radio Systems." In *IEEE International Conf. on Commun. (ICC)*, June 2011.

[TTZ12] P. Tehrani, L. Tong, and Q. Zhao. "Asymptotically Efficient Multichannel Estimation for Opportunistic Spectrum Access." *IEEE Trans. Signal Processing*, **60**(10):5347–5360, October 2012.

[Van98] Aad W. Van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998.

[VMB10] V. Valenta, R. Marsalek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert. "Survey on Spectrum Utilization in Europe: Measurements, Analyses and Observations." In *Proc. of the IEEE Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, June 2010.

[WM10] Matthias Wellens and Petri Mähönen. "Lessons Learned from an Extensive Spectrum Occupancy Measurement Campaign and a Stochastic Duty Cycle Model." *Mobile Networks and Applications*, **15**(3):461–474, June 2010.

[WMB08a] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. "Primary Users in Cellular Networks: A Large-Scale Measurement Study." In *Proc. of the IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, October 2008.

[WMB08b] Daniel Willkomm, Sridhar Machiraju, Jean Bolot, and Adam Wolisz. "Primary Users in Cellular Networks: A Large-scale Measurement Study." In *Proc. of the IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Chicago, IL, USA, October 14–17, 2008.

[WRM09] M. Wellens, J. Riihijarvi, and P. Mahonen. "Empirical Time and Frequency Domain Models of Spectrum Use." *Physical Communication*, **2**(1–2):10–32, March 2009.

[XFC13] J. Xue, Z. Feng, and K. Chen. "Beijing Spectrum Survey for Cognitive Radio Applications." In *IEEE Vehicular Technology Conference (VTC) Fall*, September 2013.

[Yan05] J. Yang. "Spatial Channel Characterization for Cognitive Radios." Technical Report UCB/ERL M05/8, EECS Department, University of California, Berkeley, January 2005.

[YCZ10] S. Yin, D. Chen, Q. Zhang, and S. Li. "Prediction-Based Throughput Optimization for Dynamic Spectrum Access." *IEEE Trans. Veh. Technol.*, **60**(3):1284–1289, March 2010.

[ZGT08] Qianchuan Zhao, Stefan Geirhofer, Lang Tong, and Brian M. Sadler. "Opportunistic Spectrum Access via Periodic Channel Sensing." *IEEE Trans. Signal Processing*, **56**(2):785–796, April 2008.

[ZTS07] Q. Zhao, L. Tong, A. Swami, and Y. Chen. "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework." *IEEE J. Select. Areas Commun.*, **25**(3):589–600, April 2007.