AN ABSTRACT OF THE DISSERTATION OF

Qiwei Wang for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on May 31, 2019.

Title: Multi-channel Stochastic Resource Allocation and Dynamic Access Scheduling

Abstract approved: ____

Thinh Nguyen Bella Bose

Modern communication systems often have the ability to transmit signals on multiple communication mediums (e.g., RF, visible light) or interfaces (e.g., MAC layer protocols) at the same time. While each channel has different characteristics, a centralized controller with channel condition information will be able to schedule the resource allocated to each channel to achieve various optimization criteria. In this thesis, we focus on two usage scenarios: Indoor hybrid free space optical (FSO)-WiFi femtocells and multi-channel satellite communication (SATCOM). For the Indoor hybrid free space optical (FSO)-WiFi femtocells, a smart network controller is designed to determine which channel/interface to use for a specific user/time slot combination to maximize some pre-specified objectives such as load balance. In particular, this problem is modeled as a dynamic scheduling problem, which is a Markov decision process problem that is solved using a deep-Q reinforcement learning (RL) framework. For the SATCOM scenario, a smart network controller is proposed to transmit information securely on different channels to mitigate jamming and eavesdropping attacks. The proposed approaches combine

elements from game theory and information theory to provide provably secure protocols from an information theoretic viewpoint.

Multi-channel Stochastic Resource Allocation and Dynamic Access Scheduling

by

Qiwei Wang

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented May 31, 2019
Commencement June 2019

Doctor of Philosophy dissertation of Qiwei Wang presented on May 31, 2019.

APPROVED:

Co-Major Professor, representing Electrical and Computer Engineering

Co-Major Professor, representing Electrical and Computer Engineering

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Qiwei Wang, Author

ACKNOWLEDGEMENTS

First I would like to thank my major advisors, Dr. Thinh Nguyen and Dr. Bella Bose. We worked together for more than 5 years on several challenging but interesting projects. The accomplishment showed in this thesis will never become reality without their professional guidance and support. I would also thank my Dr. Alan X. Wang and Dr. Arun Natarajan for being my committee members. We did some very creative work together for the past a few years. I also learned a lot from my minor advisor Dr. Liang Huang, who is definitely among one of the best professors in the field of Computer Science that I know. My greatest appreciation is for my family, especially my parents Hongyan Zheng and Limin Wang, for their lifelong support to my growth and learning. It is never easy to let their only child to study in a country that is 8000 miles away, and their sacrifice is beyond imaginary. I just hope the completion of this thesis will bring them some joy, but I can never thank them enough. Last but not least, I would thank all my dearest friends I meet here at Corvallis. Your existence makes this already beautiful journey even more wonderful.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Figure Page

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

Modern communication systems often have the ability to transmit signals on multiple communication mediums (e.g., RF, visible light) or interfaces (e.g., MAC layer protocols) at the same time. Even in the same medium, it is very likely that a single transmitter has access to multiple frequency bands using technologies like Orthogonal frequency-division multiplexing (OFDM) or Multiple-Input Multiple-Output (MIMO) [33] [20]. As a result, a centralized controller can choose single medium or multiple mediums simultaneously to transmit the signals on. However, there has not been much literature on the optimal strategies that utilize the multi-medium in a resource efficient way while maximizing users' experiences. In this thesis, we present a scheduling framework to optimize the resource allocation for a multi-channel system on the background of several application scenarios.

1.1 Multi-channel model and smart scheduler

Communication and networking technologies have advanced tremendously over the past several decades. Notably, the original "end-to-end" argument [48] for designing "dumb" and "fast" physical and link layer devices (e.g., routers and switches) are now less applicable with the declining cost of silicon. In fact, the trend in recent years has been to design fast and sophisticated hardware to efficiently support various networking abstractions for many emerging networking technologies such as network virtualization and Software Defined Network (SDN) [49]. For example, SDN with OpenFlow allows for fast construction of logical networks that are decoupled from physical networks, to provide many benefits such as network isolation, flexible routing, workload orchestration, or application specific Quality of Service (QoS). To provide all of these benefits, sophisticated software and hardware and the information across various OSI layers must be optimized jointly. In this thesis, we generalize the diversity available to an end transmitter as "abstract channels". The abstracted diversities could be spatial, frequency, link, or protocol diversity. The spatial diversity may include the different paths of an end-to-end transmission; the frequency diversity may include different RF bands or/and optical frequencies; the link diversity may include difference network interfaces like WiFi or Ethernet; the protocol diversity may include the different transmission protocols for each link. Overall, in the proposed framework, any abstracted channel can be parameterized by a set of environmental variables, which are typically related to the packet delivery rate (PDR), power consumption or security grades. And those environmental parameters will be fed to a software-controlled smart scheduler, which is an important component of the virtualization technologies. Typically, a smart scheduler is connected to a high-speed data source, and has access to control the data delivery of the multi-channel communication system. According to a policy, the smart scheduler will be able to determine which abstract channel should be used at which time slot in order to achieve QoS requirements for various applications under resource constraints. The policy is determined based on the environmental parameters of the available abstract channels, and the specific QoS requirements as well. Note that:

- The QoS requirements can be user specific, or centralized as an overall optimization objective;
- The policy can be deterministic, or randomized to achieve some stochastic perfor-

mance requirements;

- The policy can be static to simplify the allocation process, or dynamic to have the ability to adapt to the change of abstract channels;
- The policy can be determined with all prior available knowledge of the multichannel model, or learned gradually during the transmission process.

For example, in a typical indoor wireless communication usage scenario, the smart scheduler can be implemented as a router/access point(AP). It is connected to a fast Internet backbone, and provides internet access to multiple users with different applications. The channel characteristics for each user can be very different, and may be rapidly changing as well. As a result, the access point should take advantage multi-channel availability, and schedule the transmission accordingly. A smart scheduler should be able to gather information of all users', including channel condition (SNR), incoming traffic pattern, current backlogging length and so on. Based on those information, instead of using a fixed transmission schedule, the scheduler should be adjusting transmission schedules in real time to satisfy all application's requirements.

1.2 Application problems for study

Next, we will briefly introduce three application problems that the proposed framework will be applied to:

1. Smart scheduler for WiFi-FSO hybrid indoor communication femtocell (WiFO): In this problem, we propose a smart scheduler that utilizes the channel options to maximize the QoS for each users in a WiFO femtocell system. The WiFO system aims to overcome the WiFi capacity overload problem by (1) enhancing wireless

capacity using the complement free-space optics (FSO) technology which does not interfere with WiFi transmissions; (2) providing larger *per user* bandwidth; (3) providing mobility via a novel architecture that enables wireless devices to seamlessly receives data simultaneously from FSO and WiFi channels. The WiFi and FSO channels are abstracted as channels with different throughput, packet delivery rate and power consumption. Since when a specific user moves among optical and RF transmitters will result significant changes of all those parameters, we approach this problem as a dynamic decision-making problem. A dynamic decision-making problem utilizes the observable state at each time slot, and tries to find the time/state-varying optimized action based on certain observation at certain time slot. We model this problem as a Markov decision process (MDP) problem. We try to find an optimized policy, i.e., a mapping between all possible observable states of the abstract channels and a scheduling decision. If the smart scheduler follows this policy for long enough, the QoS requirement of all applications will be satisfied to the maximized extent. While most existing algorithms solving MDP problem are model-based, in Chapter 2 we propose a online learning algorithm that is model free and applicable to large scale problems. A smart scheduler will be able to learn a dynamic policy by observing the current state and the outcome of a certain action. The performance and robustness of this online unsupervised learning algorithm is shown in simulation result. Due to the generalization, this learning algorithm can also be applied to other last-mile indoor packet delivery systems.

2. Smart scheduler for mitigating SATCOM eavesdropping and fading

   A number of hurdles must be overcome before a high bandwidth and low-latency

SATCOM Internet can be realized. SATCOM links are inherently broadcast, and hence is susceptible to jamming [12, 34, 31, 54] and eavesdropping attacks [11, 21]. They are less reliable as compared to fiber optic lines, due to severe fading caused by unexpected clouds, rains, and electrical storms. The eavesdropping and fading problem is addressed in Chapter 3. In this problem, the abstract channels are the diversities of space and frequency of the satellite communication paths. We propose a novel multi-user transmission scheduler that aims to alleviate these challenges via time and channel diversity. In particular, the proposed scheduler is a low complexity randomized algorithm that multiplexes user data over multiple channels and time slots to combat eavesdropping, and environmental fading while satisfying the requested throughput of individual users with high probability. Analysis and simulation results demonstrate effectiveness of the proposed scheduler. We assume the abstract channel parameters are available at the smart scheduler, and intended to find a schedule to maximize the security of the transmission in one shot. We formulate the problem as convex an optimization problems with the background of information theory. Generally speaking, we make the usage of multiple channels as even as possible by maximizing the entropy function, so that an eavesdropping attacker cannot focus its power/time on a specific channel. The convexity of the problem is proved and the algorithm to solve them as convex optimization problems are well established. We also provide analytical bound of the optimality results.

3. Smart scheduler for mitigating frequency hopping (FH) jamming attack

   Jamming attack is anther possible threat to a secure SATCOM network. By transmitting high-power noise at the communication channel, a jammer can severely degrade the channel's SNR thus negatively impact the QoS. In Chapter 4, we analyze one of the most common techniques to mitigate jamming, that is, frequency hopping (FH). In this problem, the abstract channels are the physical channels used for FH with different carrier frequencies. We analyze the behavior of the transmission scheduler in a game theory perspective, assuming that both the jamming attacker and defender acting optimally. Importantly, we present some closed forms of the Nash Equilibrium (NE) solutions, and analyze several scenarios with different information available for the attacker.

Note that although in this thesis we focus our discussion on the scenarios above, the application of the presented framework is not limited to them. For example, recent literature on channel bonding use a similar idea where a high level connection such as TCP can be bonded simultaneously to many underlying physical interfaces such as WiFi or Ethernet. Thus, our proposed framework can be used in these settings as well.

Chapter 2: Application I: Packet scheduling with WiFO hybrid communication femtocell

2.1 Description

Over the past several years, WiFi has become an indispensable wireless access technology. However, its limited wireless capacity is increasingly becoming a critical issue with an increase in network access from smart phones and tablets. For example, consider the popular WiFi system 802.11g which has a theoretical maximum rate of 54 Mbps. However, typical WiFi networks operate at only a fraction of the maximum capacity, e.g., 15-20 Mbps due to a number of factors such as the MAC protocol overhead and the distances between the wireless devices and the access point (AP). A simple calculation shows that such limited wireless capacities fail to provide adequate bandwidth for many scenarios. For example, consider a typical conference venue consisting of 40 attendees in a room or hotel lobby. With so many people in a small area, the overall bandwidth for 802.11g including the MAC protocol overhead is only a few megabits, e.g., 10 Mbps. Thus, each user will have an average of 250kbps which is unacceptable for video streaming applications. Another example is the boarding area in an airport terminal where the current WiFi networks fail to provide adequate Internet access when there are many passengers.

We focus on improving the downlink bandwidth of the current WiFi systems in which, data is transmitted from the Internet to the mobile devices via the AP. This scenario is the common scenario that causes the bandwidth overload in WiFi networks since the downlink traffic is often orders of magnitude larger than that of uplink traffic. That said, the uplink scenarios where the wireless devices send data to the Internet, can be handled using the existing WiFi mechanisms without the assisted FSO transmissions.

In most scenarios, users are often stationary, e.g., sitting on terminal benches at airports or lounges in hotel lobbies. As such, a network of LEDs with the high-speed Ethernet infrastructure can be deployed directly above the appropriate spots to provide local high rate FSO transmissions, in addition to the WiFi transmission. The current FSO technologies are inexpensive with the transmitters and receivers using LEDs and silicon photodiodes (PDs) that cost less than $20. In addition, they operate around 20mW with good SNR and well within the eye safety (850nm). Importantly, FSO can provide 50 Mbps for typical transmission range (3 to 5 meters), without interfering with the WiFi transmissions. This implies that a 10 Gigabit backbone Ethernet can theoretically modulate 200 LEDs at 50 Mbps each. Following are some salient features of the proposed WiFO systems.

Closed-Loop Architecture. Fig. 2.1 shows the architecture of the proposed WiFO system for the downlink scenarios. All the data from the Internet to the devices in a WiFi network is first traversed through the AP. For an IP packet of a given flow, the AP will decide whether to send the data on the WiFi or FSO channels. If it decides to send the data on the FSO channel for a particular device or user, the data will be encoded appropriately, and broadcast on the Gigabit Ethernet network with the appropriate information to allow the right FSO transmitter to receive the data. Upon receiving the data, the transmitter relays the data to the intended device below it. If the AP decides to send the data on the WiFi channel, then it just directly broadcast the data through the usual WiFi protocol. Upon receiving the data from the FSO channel, the

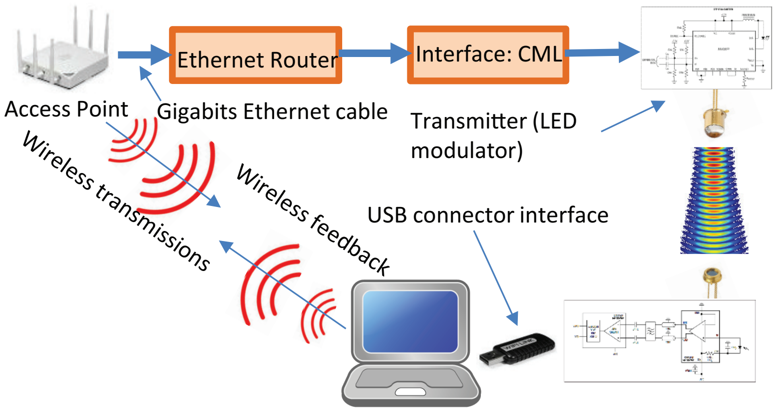


Figure 2.1: WiFO architecture

receiver decodes the data, and sends a feedback/ACK to the AP via the WiFi channel, completing a closed-loop transmission.

AP-centric Design. The AP will handle all the sophisticated functions, e.g., which channels to send the data on, how to encode the data, as well as setting the FSO and WiFi transmission parameters. Such an architecture will allow better overall performance through the joint optimization of multiple flows while keeping the wireless devices simple. Specifically, the AP-centric architecture is designed based on the cross-layer optimization approach with minimal modifications to the existing WiFi mechanisms. An important consequence of this design is that all existing applications are agnostic to the implementations of the lower OSI layers, and thus they will operate seamlessly in the WiFO system.

Channel Feedback. Feedback channel is critical to the WiFO's bandwidth and mobility. Unfortunately, sending the feedback from a receiver via the FSO channel is problematic due to difficult deployment and interferences with the forward channel. Therefore, we employ a novel feedback scheme that uses WiFi transmissions to report the channel conditions on the FSO channel (see Fig. 2.1). Specifically, the WiFO system implements a control channel that is used for sending periodic SNR feedback, as well as the ACK messages for the FSO transmissions, from a mobile device to the AP using high-priority WiFi control channels.

Mobility. In the existing WiFi infrastructure, at any moment, a mobile device is associated with an AP via a beacon signal. As a mobile device moves from the coverage of one AP into another, its association changes to the new AP with higher SNR level. Similarly, the WiFO system provides mobility as a user moves from one light cone to another.

Cross Layer Optimization. Given the current conditions of the FSO and WiFi channels, the performance of the WiFO system depends on the joint optimization that produces optimal system parameters and policies. The main challenge is how to allocate the transmission rates to FSO and WiFi channels optimally. This rate allocation problem directly dictates the designs of various algorithmic components and parameters in the WiFO system. Specifically, at the physical layer, the choices of the modulation schemes, channel coding, and transmission power levels must be jointly optimized for both WiFi and FSO channels. At the link layer, packet scheduling policy is to allocate packets appropriately to FSO or WiFi queues, effectively performing rate allocation over the two channels. Here, we take a more general way to model this problem mathematically. The WiFi/FSO channels are generalized as channels with different parameterized characteristics,and multiple users are generalized as applications with different incoming data throughput and quality of service (QoS) requirements. Depend on the location and mobility of users, channel characteristic and application QoS can vary. All kinds of this information are available are the centralized AP, and a smart AP should constantly examining the network environment and make dynamic decisions to allocation channel resources. Now, we present a MDP based mathematical framework that takes the advantage of multiple channels and can deal with changing network environment in large scale.

2.2 Related works

There is rich literature on resource and scheduling algorithms. In particular, the packet scheduler problem can be formulated as a Markov Decision Process (MDP) problem [44]. An MDP problem is well studied as a stochastic dynamic programming problem with many algorithmic solutions such as Backward Induction and Value Iteration or Policy Iteration. MDP solutions are purely model based. That is, given a state s and action a, the precise model of the transition probability $T(s, a, s')$ and the instant reward $r(s, a)$ must be provided to the algorithm which this approach less useful in the real world scenarios where models cannot be accurately determined. A more popular approach is to online Reinforcement Learning (RL). One popular algorithm is the Q-learning algorithm [35] to be described shortly. Using the Q-learning algorithm, an RL agent learns an optimal state-action pair by interacting with real environment without any modeling knowledge. At each time step the learning agent examines the current state s and takes action a that maximizes $Q(s, a)$. This mechanism makes it easy to implement an online learning algorithm that gradually improves the agent's performance. As a result, RL has been applied to network/communication control optimization. For example, routing scheme in [22] employed multi-agent RL to improve delivery time and avoid congestion. And in [42], the scheduling-admission problem of time varying channels is formulated as a constrained MDP, then an improved online learning algorithm was shown to outperform traditional Q-learning. While RL has been widely applied, many challenges (to be described shortly) need to be further studied to make the RL algorithms useful for real-world problems with large state space and fast changing environment. In particular, the combination of RL and deep learning utilizes neural network as a non-linear function approximator, and make it possible for the learning agent to deal with complicated Q functions and enormous size of state spaces. For example, in [37][36], convolutional neural network (CNN) were used to train a learning agent to play multiple Atari games. With no prior knowledge of the game, an agent is learned to perform equally or better than human, based only on the screen images. The well known Alpha Go [50] further confirm the possibility of applying RL on problems of very large size. Similarly, we propose a DQ framework for designing packet schedulers that utilizes benefits of both deep learning architecture and Q-Learning.

2.3 Mathematical model of the problem

2.3.1 Problem model overview

Based on the WiFO system, or any other indoor packet delivery system that has multiple channel access, we now present the mathematical model of the channel scheduling problem as an MDP. We look into several algorithms, apply those to this specific problem and analyze the results. From now on, we treat the network controller/scheduler/AP as a learning agent, and those terms are used interchangeably.

We assume there are several users running total of N applications with different QoS requirements (e.g., data rate, packet loss rate and delay). The links between each user and the AP are characterized by different channel conditions. Due to the limitation of processing power and channel capacity available for the AP, it can only serve limited number of users/applications in a fixed amount of time. The goal of the AP is to dynamically allocate its resources to satisfy all users QoS requirements. Since the network conditions can change quickly, the AP's resource allocation policy should adapt to the new network conditions timely. Fig. 2.2 shows the components involving the AP's operations.

1. *External environment:* The environment is modeled as a black box to the AP. The parameters of environment can be the channel quality, data rates, movement speed and direction of each user, and the like. The AP can only infer the external environment by interacting with it and observing the outputs, in RL terminology, the immediate rewards.

2. *Observable states:* The observable state is the internal states of the AP that can be observed and used to infer about the environment and make a good decision. States can be pending packets to be transmitted, current packet loss count, and the like.

3. *Resources:* Example of resources are computational power or total bandwidth.

4. *QoS requirements:* Examples of QoS are minimum bandwidth or maximum delay requested by an application. The AP needs to find the policy that satisfies those requirement.

5. *Policy:* A policy is a mapping between the observable states and an action such at sending a packet from a particular application given the current backlogs of all other applications.

By observing the states and actions, together with the corresponding rewards over time, the goal of the smart AP is to learn a policy that optimally allocates its resources while satisfying the user's QoS requirements.

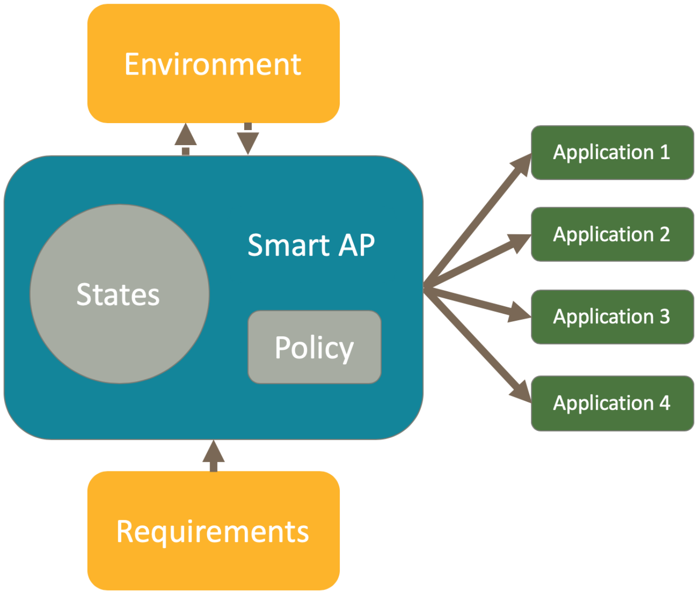


Figure 2.2: Problem framework

We now focus on the particular problem of designing a packet scheduler to provide QoS for different applications. A packet scheduler uses multiple queues for different applications as shown in Fig. 2.3. The en-queue and de-queue rates control the data rates, delay, and packet loss rates of the applications. Assuming that each user/application is associated with a buffer of length $L_i, i = 0, 1, ..., N-1$. Time is discretized into small time slots T_0. To simplify the analysis, in a single time slot, we assume the AP will take one action to transmit first, then new packet arrives.

Departure of a packet: At the beginning of each time slot, the AP observes the current network state, and decides which packet from the queues (applications) to serve, i.e., dequeue a packet from a queue and transmits. Depend on the QoS requirements, the AP can also do nothing if necessary (e.g., to save energy). The channel associated to each application is modeled as the packet delivery rate (PDR), denoted as $q_i, i = 0, 1, ..., N-1$. If a transmission attempt fails, the transmitted packet has to be put back in the queue for re-transmission.

Arrival of a packet: At each time slot, a packet might arrive, and it will be added to the end of the appropriate queue. The probability of a packet arrival for each application is denoted as $p_i, i = 0, 1, ..., N - 1$. If the time slot length T_0 is small enough, the arrival model is approximately Poisson, and p_i can be found by the average throughput of an application. Specifically, if on average application i requires incoming data throughput of m bytes/s, and each packet has K bytes, then p_i can be found by:

$$p_i = \frac{mT_0}{K}. \tag{2.1}$$

After a packet arrives, if its destination queue/buffer is full, the packet is dropped. The packet scheduler is trained to minimize a certain objective such as the probability of a packet drop due to full queue as a result of channel conditions. In this paper, the target of a smart packet scheduler is to find a policy that minimizes the packet loss rates. We note that the AP can only observe the current backlog length of each queue (applications) at the beginning of a time slot. The environmental parameters p_i and q_i are not available.

2.3.2 Markov Decision Problem

Now we model our problem as an MDP problem based on the background knowledge introduced in Chapter 1, as follows: *States:* At a time slot t, the observable state is an

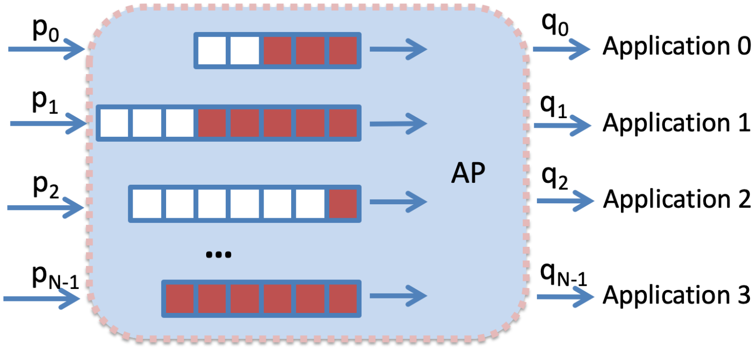


Figure 2.3: Application flows are modeled as queues

N tuple, denoting the backlog length of each application.

$$s^t = (l_0^t, l_1^t, ..., l_{N-1}^t), \tag{2.2}$$

where l_i^t is the backlog length of application i at time slot t.

Actions: For the simplicity, we assume only deterministic policy. That is to say, at each time slot, the agent decides to send one packet for one of the applications, or not to send at all, with probability of 1. The total number of possible action is $N + 1$. The action at time t, a^t, is determined by the policy. After an action is taken, the agent interact with the environment and observe the state of next time slot, s^{t+1}.

Instant Rewards: Given s^t, a^t, s^{t+1}, the AP will observe the instant reward R^t, which is the sum of instant rewards from all applications:

$$R^t = \sum_{i=0}^{N-1} r_i(s^t, a^t, s^{t+1}). \tag{2.3}$$

For each application, its QoS requirements can be modeled by choosing an appropriate $r_i(\cdot)$. For this paper, as an example, we consider the packet loss due to a full backlog buffer, so a negative constant C_i is assigned to each application as a penalty of packet loss, that is:

$$r_i(s,a,s') = \begin{cases} C_i, & \text{if a packet is lost.} \\ 0, & \text{elsewise.} \end{cases}$$

Remark 1. By changing C_i for each application, the controller is able to distribute the network resources unevenly to some of the applications. Thus, the priority for each application can be manually set by the assignment of C_i. A larger C_i means smaller penalty, indicating the corresponding application has higher priority when network is congested. Speaking in a more generalized way, a typical $r_i(\cdot)$ can be a function that is non-increasing in backlog length, packet drop rate, power consumption of each transmission, and so on. It should also be a function that is non-decreasing in packet delivery and so on. One application can submit custom-designed $r_i(\cdot)$ to describe its specific QoS requirement.

Transition probability: The transition probability, $p(s'|s,a)$, is not available to the learning agent due to the unknown arrival probabilities p_i and q_i. But still, at each time slot, given the action taken and packet arrival probability, the transition probability from $l_0^t, l_1^t, ..., l_{N-1}^t$ to $l_0^{t+1}, l_1^{t+1}, ..., l_{N-1}^{t+1}$ can be uniquely determined.

Optimal policy: The optimal policy is a policy that maximizes the estimated discounted reward, that is, a mapping $\pi^*(s \rightarrow a)$ such that the total discount reward:

$$R_{total} = E[\sum_t \beta^t R_t], \tag{2.4}$$

is maximized. β is a discount parameter between 0 and 1. Given the definition of instant reward, the maximization of total discounted reward can results a policy that has a specific long-term optimized performance (minimized delay, minimized power consumption, maximized throughput or even the combinations of a few). Since the instant reward is the penalty of packet loss. Maximizing R_{total} will minimize the expectation of packet loss.

2.4 Approaches

Given the environment model, this problem can be solved in several ways, depending on the availability of information and the size of the problem. To be exact, the well-known algorithms such as the value iteration (VI) can be used if all the model parameters are known. Model-free online learning algorithms such as the Q-learning algorithm can solve the problem at a smaller scale. To that end, we also present a neural network based Q-learning algorithm that can be applied to large size problem.

2.4.1 Model-based approach

For the model-based approach, we assume that the transition probability $T(s, a, s')$ and instant reward function $r(s, a)$ are both instantly available to the agent. In an infinite discounted reward problem, given a policy π, the value of a state s, $V_\pi(s)$ is defined as:

$$V_\pi(s) = E_{\pi,T}[\sum_{i=0}^{\infty} \beta^i r(s_i, \pi(s_i))], \tag{2.5}$$

in which β is a discount factor between 0 and 1, s_i is the state after i steps, and $r(s_i, \pi(s_i))$ is the instant reward at s_i and taking action according to policy π. $V_\pi(s)$ can be found by solving the following Bellman equation:

$$V_\pi(s) = r(s, \pi(s)) + \sum_{s' \in S} T(s, \pi(s), s')V_\pi(s'). \tag{2.6}$$

Bellman equation can be solved by stochastic dynamic programming with arbitrary initial value of $V(s)$ for any s. The value iteration algorithm is as follows:

Algorithm 1 Value Iteration

```
Randomly initialize $V(s)$ for all $s \in S$ (Usually all 0)
while $|V(s) - V_{prev}(s)| > \epsilon$ do
    for $s$ in $S$ do
        $V_{prev}(s) = V(s)$;
        $V(s) = \max_{a \in A}(r(s,a) + \sum_{s' \in S} T(s,a,s')V_{prev}(s'))$;
    end for
end while
```

$V(s)$ converges to the solution for Bellman Equation according to [44]. After a ϵ-optimal $V(s)$ is found, the policy is:

$$\pi(s) = argmax_{a \in A}(r(s,a) + \sum_{s' \in S} T(s,a,s')V_{prev}(s')). \tag{2.7}$$

Remark 2. Value iteration algorithm is not applicable for implementing a real network controller as the model parameters are often not timely available. Instead, the VI algorithm is applied to small problems to find the true value of each state and we use them to verify the correctness of other algorithms.

2.4.2 Q-learning with noisy temporal differential updates

The learning agent tries to learn the optimal policy by interact with the environment without knowing the actual system model, i.e., the transition probability and instant reward as a function of current state s, action a and next state s'. Similar to value iteration, one way to find the estimated value of $V(s)$ is by performing temporal differential updates as follows:

$$V(s) \leftarrow V(s) + \alpha(r(s,a) + V(s') - V(s)), \tag{2.8}$$

or equivalently:

$$V(s) \leftarrow (1-\alpha)V(s) + \alpha(r(s,a) + V(s')). \tag{2.9}$$

In fact, temporal differential updates is a noisy version of value iteration. It replaces the reward expectation term in Eq. 2.6, $\sum_{s' \in S} T(s,a,s')V_{prev}(s')$, with a sampled version with the action a.

At any state s, the action a is chosen by the following exploration/exploit policy:

$$a = \begin{cases} argmax_{a \in A}(r(s,a) + \sum_{s' \in S} T(s,a,s')V(s')), \text{ w.p. } \epsilon, \\ \text{randomly chosen from action space } A, \text{ w.p. } 1-\epsilon. \end{cases} \tag{2.10}$$

After finding a, Eq. 2.8 is used to update the value of current state s. It is obvious that temporal differential update of the value function still needs a model to perform. Usually, this model can be acquired by estimating the transition probability from the agents' decision trajectory at a cost of time and space complexity. Instead, a model-free learning agent can use temporal differential to update the Q value:

$$Q(s,a) = Q(s,a) + \alpha(r(s,a) + max_a' Q(s',a') - Q(s,a)). \tag{2.11}$$

At any state s, given the current $Q(s,a)$ mapping, use the following exploration/exploit policy without any knowledge of the model:

$$a = \begin{cases} argmax_{a \in A} Q(s,a), \text{ w.p. } \epsilon, \\ \text{randomly chosen from action space } A, \text{ w.p. } 1-\epsilon. \end{cases} \tag{2.12}$$

The pseudo code for the Q-learning algorithm is shown below:

Algorithm 2 Q-LEARNING BY TEMPORAL DIFFERENTIAL UPDATES

```
Randomly initialize $Q(s,a)$ for all $s \in S$ and $a \in A$ (Usually all 0);
Set agent at initial state $s$;
while $|Q(s,a) - Q_{prev}(s,a)| > \epsilon$ do
  Agent takes action $a$ by Eq. 2.12;
  Observe next state $s'$, instant reward $r$;
  Update Q value: $Q(s,a) = Q_{prev}(s,a) + \alpha(r + max_a' Q(s',a') - Q_{prev}(s,a))$;
  $s = s'$;
end while
```

To make sure the Q-learning algorithm converge(w.p.1), the learning rate α should satisfy [35]:

$$\sum_{n=0}^{\infty} \alpha_n = \infty, \tag{2.13}$$

$$\sum_{n=0}^{\infty} \alpha_n^2 < \infty, \tag{2.14}$$

where α_n is the learning rate at iteration n. A commonly used learning rate is $\alpha = \frac{k}{k+n}$, where k is a positive number that can be tuned for convergence speed.

2.4.3 Q-learning with function approximation

In this specific problem setting, the state and action space can get very large. For example, 10 users with maximum queue length of 20 will result 20^{10} states. With limited computation resources at the AP, it is not realistic to store a huge table for all (s, a) pairs and to visit all (s, a) pairs. Instead, a practical approach is to use function approximation which extracts a feature vector $\phi(s, a)$ from s, and the real $Q(s, a)$ table can be approximated by a function $F(\cdot)$:

$$\hat{Q}(s, a) = F(\phi(s, a)). \tag{2.15}$$

If $F(\cdot)$ is convex and can be parameterized by a vector θ, the optimal $\hat{Q}(s, a)$ can be found by minimizing the square error loss function:

$$||\hat{Q}(s, a) - Q(s, a)||_2, \tag{2.16}$$

using a gradient method over θ. Since the true value of $Q(s, a)$ in Eq. 2.16 is not immediately available during online training, we use a sampled version to replace $Q(s, a)$:

$$Q_{sample}(s, a) = r(s, a) + \beta max_{a'} \hat{Q}(s', a'). \tag{2.17}$$

Thus, the stochastic gradient update rule for θ will be:

$$\theta \leftarrow \theta + \alpha(r(s, a) + max_{a'} \hat{Q}(s', a'|\theta) - \hat{Q}(s, a|\theta))\nabla_\theta \hat{Q}(s, a|\theta). \tag{2.18}$$

The simplest approximator is the linear combination of $\phi(s,a)$ and θ:

$$\hat{Q}(s,a) = \sum_{n} \phi_n(s,a)\theta_n, \tag{2.19}$$

in which n is the dimension of the feature vector. The performance of linear approximator is acceptable in small size problem. Fig. 2.4 shows the experimental result of a small problem with 2 applications (A and B) and 3 available actions (send A, send B, send nothing). The true value of a state when A has 2 packets and B has 3 packets is calculated by value iteration. It is compared to the result of classic Q-learning and the result of linear function approximator. It shows that the approximated Q value is close to the truth but with some approximation error. While in a problem with such small size this error may not be critical, linear approximator is not suitable for problems with larger size.

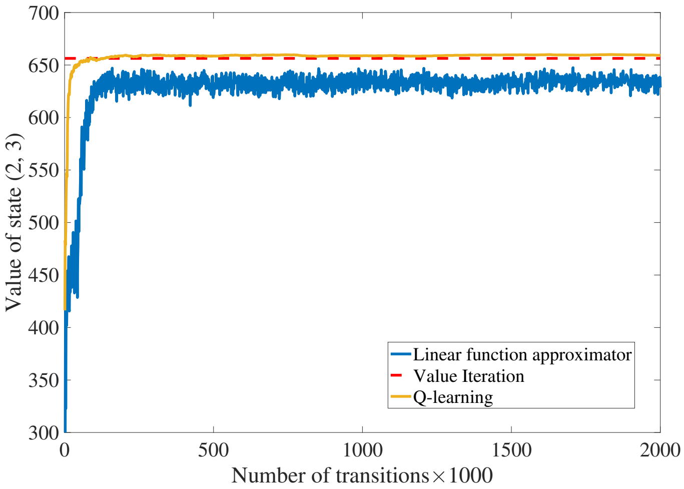


Figure 2.4: Linear function approximation compared to true Q value

2.4.4 Non-linear function approximation with neural network (NN)

In this section, we describe a non-linear function approximator based on the DQN framework introduced in [37]. Due to the large number of states, we use a neural network with multiple hidden layers to approximate the $Q(s, a)$.

2.4.4.1 Features and NN model

We model this problem as a regression problem. Instead of extracting the feature from (s, a) pair, only the state is used to generate the input feature. The output layer of the NN contains $N + 1$ neurons and each of them is associate with an action, as shown in Fig. 2.5. By doing this, we take advantage of the smaller action space size (number of applications, usually way less than the number of states), so we can obtain $\hat{Q}(s, a)$ values for all actions and find the best action in just one pass of forward propagation.

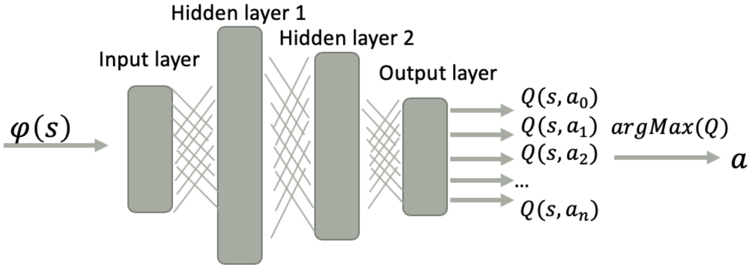


Figure 2.5: An example of the NN model

The input feature vector is the vector presentation of the current backlog lengths of all applications at the beginning of each epoch. The backlog lengths are normalized over the queue size such that $0 \leq \phi_i(s) \leq 1$:

$$\phi(s) = [l_0/L_0, l_1/L_1, ..., l_{N-1}/L_{N-1}]. \tag{2.20}$$

2.4.4.2 Experienced replay

Since for the most of the time, the action is taken by choosing a corresponding to the maximum output of the NN, a slight change of NN parameters may cause a total different policy and alters the learning trajectory. As a result, updating the NN parameters just by one sampled $Q(s, a)$ is very risky and may result in very unstable performance.

To counter this effect, we employ experienced replay [36][37] that uses a memory pool to memorize the newest M transitions. At the beginning of each time slot, the agent takes action based on the output of the NN, and the whole transitions (s, a, s', r) are recorded in the memory pool. The oldest transition is deleted if the pool size is larger than M. Then, a mini-batch is randomly chosen from the pool to perform a gradient descent updates of the NN parameters θ, rather than using just one sampled $Q(s, a)$.

2.4.4.3 Loss function

Similar to the linear case, $Q_{sample}(s, a)$ is used to estimate the true value of $Q(s, a)$, as in Eq. 2.17. Then stochastic gradient descent method is used to to carry out one step of update to minimize the mean square error:

$$f_c = \frac{||Q_{sample}(s,a) - \hat{Q}(s,a)||^2}{\text{size of mini-batch}}. \tag{2.21}$$

2.4.4.4 Delayed update of target NN

Note that in Eq. 2.17, we still need one pass of forward propagation of the NN to find the $\hat{Q}(s', a')$. When the NN parameters are updated, the sample itself changes too. In the meantime, with one update with SGD in Eq. 2.18, only one biased sample of transition based on current θ(one mini-batch if using experienced replay) is used to update the whole neural network, which will approximate the value of millions of $Q(s,a)$ in the future steps. This update can be very inaccurate and may negatively impact the future learning trajectory. To further improve the stability of the algorithm, an extra target NN is introduced. The decision NN is used to find the current best action, and is updated by the mini-batch every time slot. The target NN is used to find the $Q_{sample}(s,a)$. The target NN is only updated every T_{target} time slots by copying the current decision NN to it. Thus, the target NN is updated with all transactions (or multiple mini-batches) that are sampled during T_{target} time. For most of the time, the decision NN is optimized towards a "fixed target" instead of a target that keeps changing. If the decision NN and target NN are denoted as θ and θ', the new update rule is:

$$\theta \leftarrow \theta + \alpha(r(s,a) + max_{a'}\hat{Q}(s',a'|\theta') - \hat{Q}(s,a|\theta))\nabla_{\theta}\hat{Q}(s,a|\theta). \tag{2.22}$$

2.4.4.5 Learning procedure

The complete learning algorithm is shown in Fig. 2.6 and Algorithm 3.

- Decision Phase: At the beginning of each time slot, the agent observes the state and feeds the input feature vector into the decision NN. Based on the output, the agent either does random exploration, or takes the action a associated with the maximum NN output. At the end of the time slot, the agent observes the state s' and instant reward r, records the transition (s, a, s', r) in the memory.

- Learning Phase: A mini-batch is randomly chosen from the memory and is used to update the decision NN's parameters by the stochastic gradient descent algorithm.

If the number of time slots is a multiple of T_{target}, then copy the decision NN to the target NN.

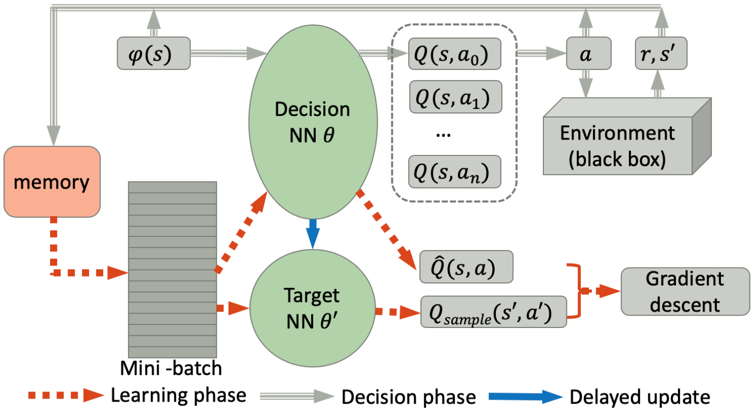


Figure 2.6: Graph of FC-NN approximation algorithm

2.4.4.6 Improvement of Stochastic Gradient Method (SGM)

We use ADAM optimizer [24] as a replacement of SGM for better convergence speed. An ADAM optimizer is a combination of gradient with momentum and RMSprop. It shows better convergence performance in many other deep reinforcement learning algorithms [4][14]. ADAM's performance in this problem is evaluated in Section 2.5.

2.5 Results

In this section, we show the performance evaluation of the presented learning algorithm. We assume the AP has a total capacity of 12 Gbps for all the users. While most commonly used wireless routers have smaller capacity, AP with larger capacity is expected in the future. Thus, if a packet has a fixed size of 1500 bytes, the time slot length will be 1 μs.

Algorithm 3 Non-linear Q function approximation

```
-Random initialize decision NN parameters $\theta$;
-Set the target NN parameters $\theta' = \theta$;
for $i$ in $[0, \max$ number of epochs$]$ do
    -Random initialize backlog lengths for all buffers: $(l_0, l_1, ..., l_{N-1})$;
    for $j$ in $[0, \max$ number of transitions$]$ do
        -Find feature vector $\phi(s)$;
        -$i = random([0, 1])$:
        if $i < \epsilon$ then
            $a = random([a_0, a_1, ...a_{N-1}])$ ;
        else
            $a = argmax_a \hat{Q}(s, a|\theta)$;
        end if
        -Take action $a$, observe $r, s'$ and add $(s, a, r, s')$ to history pool;
        -Randomly sample a mini-batch from history pool;
        -Find $Q_{sample}(s, a|\theta') = r + max'_a \hat{Q}(s', a'|\theta')$ with target NN;
        -Find $Q(s, a|\theta)$ with decision NN;
        -update $\theta$ by Eq. 2.22;
    end for
    -Update target NN: $\theta' = \theta$.
end for
```

Based on this, application's incoming data rate can be translated to arrival probability by Eq. 2.1. A scenario with 10 applications is simulated using various channel conditions and traffic patterns. Hyperparameters of the NN are shown in Table 2.1.

First, the applications are associated with a randomly generated p_i and q_i for $i = 0, 1, ..., 9$ to model network conditions and data rates. We also make $\sum_i p_i$ close to the mean of q_i so the total incoming data rate is close to AP's total channel capacity. Thus, the algorithm is running on a slightly congested environment.

Fig. 2.7 shows the convergence curve of the total discounted reward in each epoch, up to 5000 epochs (equivalent to 5 seconds in real time). The plain SGD is very noisy especially at the beginning since most of the states are not visited. Due to a fixed learning rate, the total discounted rate converges very slowly when compared to others.

Table 2.1: Hyperparameters of the neural network

Hidden layers	64×32
Mini-batch size	64
History pool size	100000
Parameter initializer	Xavier initializer
Delayed update frequency T_{target}	$10000T_0$
Learning rate α	0.0001
Discount rate β	0.9999

The ADAM optimizer converges with a constant step size converges faster, however it diverges from the optimal soon after reaching the optimal due to its instability. A carefully chosen shrinking step-size can deal with the instability, but it requires fine tuning of the parameters and a longer converging time. The delayed update of target NN handles the instability well and it maintains a better converge time (< 1000 epochs) as shown in Fig. 2.7. Fig. 2.8 shows the average packet loss of last 500 epochs of the 5000 epochs. Again, ADAM with delayed target NN update outperforms others with a much lower packet loss rate of 0.55%, while the packet loss rate of the other three are 22%, 19% and 3%.

Now we evaluate the robustness of the algorithm by introducing a sudden change of the arrival data rates. In Fig. 2.9, we randomly choose an application and assign it a large $p_i = 0.35$ while other 9 applications have $p_i = 0.05$. After training for 2000 epochs, the large incoming data rate is re-assigned to another application. In a real scenario, this can happen when some applications are newly started/reconfigured. It can be seen that the algorithm adapts to the new environment very well within a short period of fluctuation (less than 500 epochs, in our set up that is about 0.5 seconds). The packet

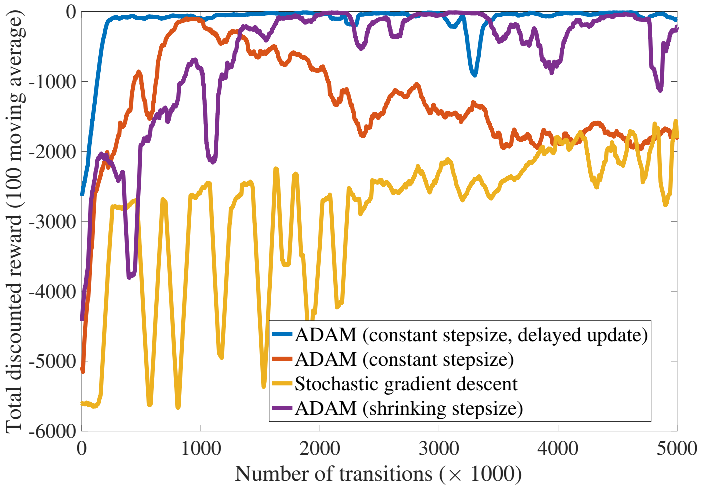


Figure 2.7: Comparison of performance

loss rate changes with the total discounted reward accordingly.

Now we evaluate the algorithm's performance under a sudden change in channel conditions by changing the PDR. As shown in Fig. 2.10, at the beginning, the channel PDR associated with a particular application is $q_i = 0.9$. After training for 1500 epochs (1.5 seconds), the PDR is changed to $q_i = 0.65$. This could be a result of this user moving away from the transmitter or behind some obstacles. The channel condition is bad such that the expectation of number of transitions to finish all packets in 1000 time slots is about 1030. In other words, on average it takes about 1030 transmissions to send all arriving packets in an epoch. We can see that the algorithm converges quickly to a point where the packet loss rate of 0.3%, close to the best it can do. After 3500 epochs (3.5 seconds), the channel PDR is changed to a very good value, $q_i = 0.99$, and the packet loss result is back to close to 0.

Next we show the impact of the instant reward C_i assigned to a specific user. In this scenario, we set $p_i = 0.091$ and $q_i = 0.9$ for all applications to eliminate the bias

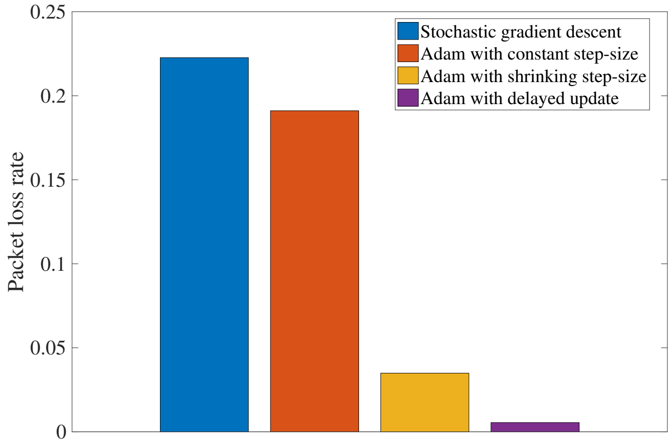


Figure 2.8: Comparison of average packet loss rate

from the environment. Also, we make $\sum_i p_i$ slightly *larger* than average q_i to simulate a more congested network conditions. Fig. 2.11 shows the average packet loss when the instant reward is uniformly assigned, that is, $C_i = -10$ for all applications. The packet loss rate for each application is around 2.5 per 1000 time slots. In Fig. 2.12, application 3's instant reward is increase to -1 to decrease its priority. Because the AP does not have enough resources to fulfill all applications' requirement, it serves fewer packets from application 3. As a result, the packet loss rate of application 3 is increased by a large amount to give the other 9 applications a better performance. This can be verified by the decrease of the average packet loss rate from 2.62 to 1.57 per 1000.

Remark 3. : Importantly, due to the proposed DQN architecture, the one pass forward/backward propagation (computing the output for a given input to the DNN) can be very fast. Also, since the algorithm does not require any hand-labeled data, the agent can be continuously trained and make decision simultaneously in real time in the practical settings where traffic characteristics and channel conditions change frequently.

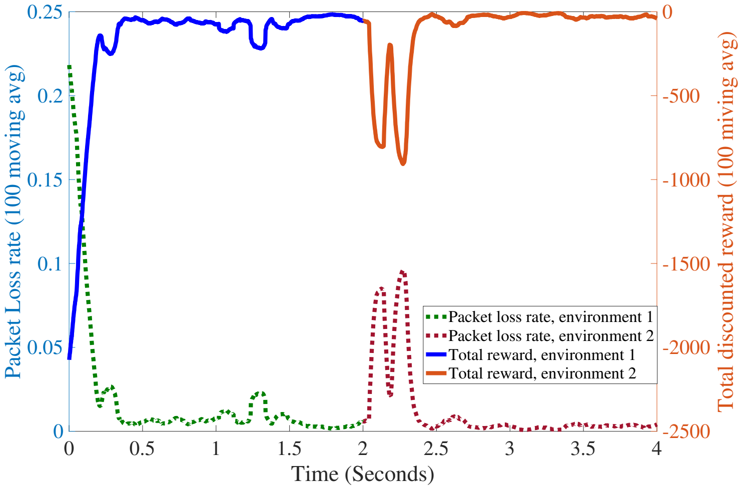


Figure 2.9: Dynamic policy adjustment with a sudden change in traffic rates

2.6 Summary

In this Chapter, we describe an adaptive packet schedulers that can be applied to a WiFi-FSO hybrid indoor communication system, or any last-mile packet delivery system that has multiple channel access. We present the scheduling policy that optimize for application specific quality of service (QoS) requirements. The framework models the problem as an MDP and integrates a deep neural network with online Q-learning algorithms that enables a DQ-based packet scheduler to learn a good packet transmission policy. Importantly, a DQ based packet scheduler can be deployed without any prior training or network traffic models. Rather, the DQ- based packet scheduler progressively learns a good policy in real-time, based directly on the available observations. Our simulation results indicate that the proposed DQ-based scheduler can adapt to the changes in network conditions and/or application requirements in real time to achieve various QoS.

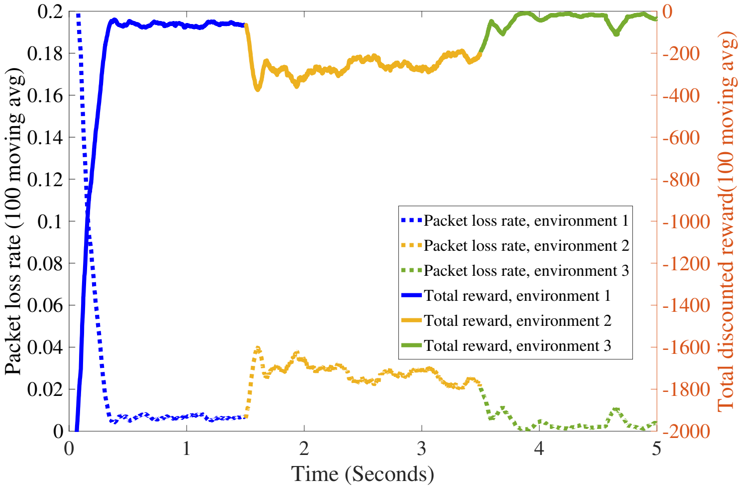


Figure 2.10: Dynamic policy adjustment with a sudden change of channel conditions

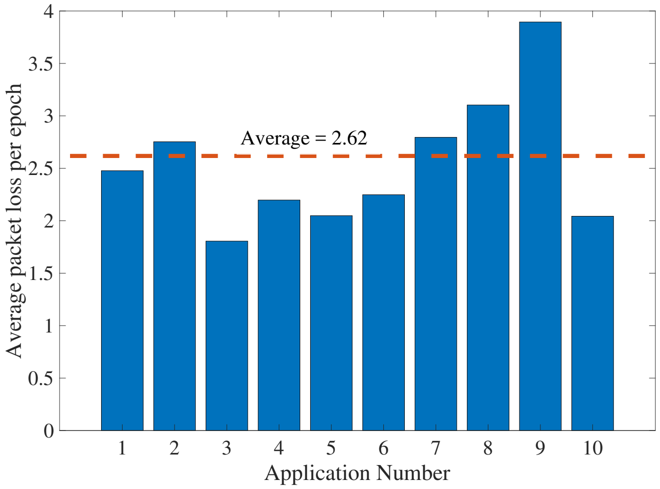


Figure 2.11: Packet loss comparison with uniformed instant reward

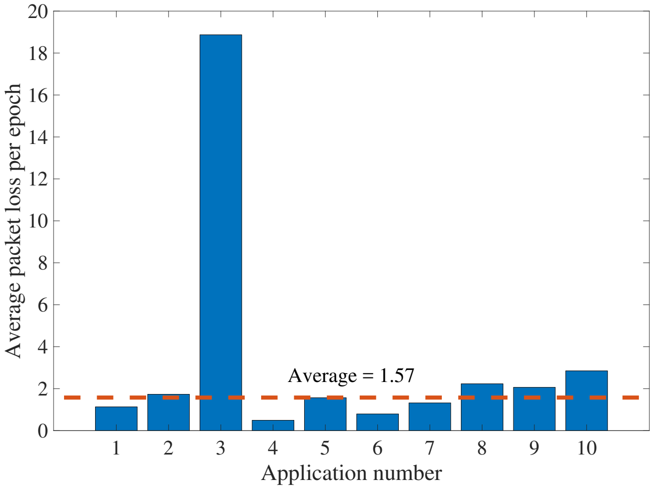


Figure 2.12: Packet loss comparison with biased instant reward

Chapter 3: Application II: Mitigating Eavesdropping and fading: Efficient resource scheduler for secure multi-channel satellite communication

3.1 Description

Based on the background of SATCOM introduced in Chapter 1, in this chapter, we describe a scenario where we can generalize the satellite links as multiple channels, and proposes a novel multi-user transmission scheduler that aims to alleviate eavesdropping/fading problems via time and channel diversity. In particular, the proposed scheduler is a low complexity randomized algorithm that multiplexes user data over multiple frequencies and time slots to combat eavesdropping and environmental fading while satisfying the QoS requirements of individual users with high probability. First, we present two typical usage examples: one shows the frequency diversity of the channels, the other shows the spatial diversity:

Example 1: Consider a simple scenario in which a satellite acts as a relay node between a sender(s) and a receiver(s) as shown in Fig. 3.1. The communication between the satellite and the sender (receiver) commonly takes place on some specified frequency band. However, communication on a fixed band for sufficiently long time is potentially vulnerable to jamming and eavesdropping attacks. Doing so provides an opportunity for an attacker to collect enough statistical information to enable it to infer the transmitting band. Knowing this, an attacker can launch a jamming attack by generating noise on the same transmission band and directs at the satellite to decrease the SNR. With sufficient noise power, an attacker can seriously degrade or cut off both satellite uplink and downlink. Knowing the transmission band, an eavesdropper can also listen and decode the information if the satellite transmission does not use sufficiently strong encryption. From the wireless communication perspective, transmitting information on a fixed band is also not optimal due to channel selected fading, depending on the unexpected clouds, rains, electrical storms [8] [28] that can create link burst losses or outages. As will be discussed shortly, the solution to both security and performance problem is to employ both time and frequency diversity. OFDM, for example, is a classical technique to combating fading by spreading the symbols over multiple frequencies. However, OFDM uses only small consecutive bands with no data scheduling and rate allocation among the bands. Our solution provides high level scheme that views frequency diversity as a special case. Before discussing our approach, we now describe the second scenario to highlight the problem that frequency diversity might not be able to address, and to outline how a future satellite constellation together with our solution can help.

Example 2: We consider a future satellite constellation as shown in Fig. 3.2. Each satellite acts as a router. Consequently, the information can be sent from any one satellite to another. A sender and a receiver on earth can communicate with each other via a number of relay satellites. A sender/receiver might have the capability to communicate to multiple satellites. Similar to scenario one, using a fixed path between a sender and a receiver is not only vulnerable to jamming/eavesdropping attacks but also is susceptible to burst losses and outages. However, the future satellite constellation can further increase the resiliency and robustness with additional spatial diversity. As seen, in Fig. 3.2, there are three paths (red, blue, and brown) that can be used simultaneously to transmit data from the sender to the receiver. Due to the different locations of the

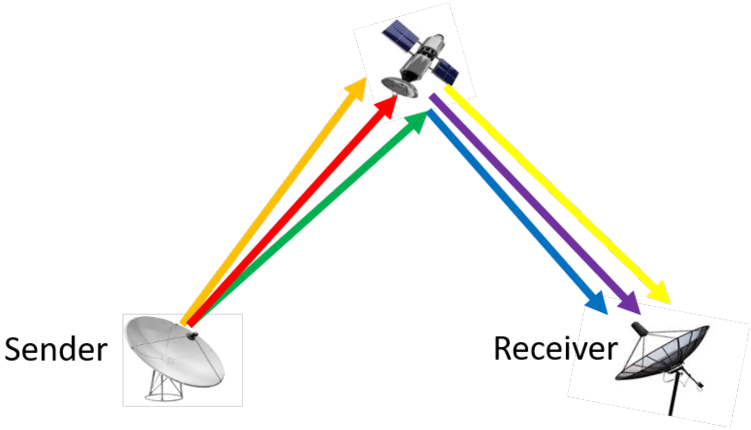


Figure 3.1: Frequency diversity: Sender transmits data over multiple frequency bands. Different colors denotes different frequencies. Frequency diversity helps to combat channel fading as well as eavesdropping and jamming attacks.

three last-mile satellites ($S1$, $S2$, and $S3$), the three satellite-earth links provide spatial diversity to combat potential cloud, rains, and electrical storms. Specifically, these last-mile satellites might transmit data to different ground stations ($B1$, $B2$, $B3$) that belong to a secure ground network. The receiver in the secure network can collect the data from these ground stations. Because of spatial diversity, the fading and burst losses caused by the local clouds/rain/storm near the receiver can be alleviated since data is also simultaneously transmitted on other unaffected satellite-earth links.

In this chapter, we still study a high level abstraction framework that employs the principle of diversity to enhance communication performance over multiple available abstract channels described in Chapter 1. Instead of the indoor communication scenario in Chapter 2, we study the problem in the background of security data delivery of SATCOM. The main issue we want to address is that, given a number of users with QoS requirements, together with a number of channels and their qualities (e.g., packet loss rates, SNR), how can we optimally transmit the data for these users in such a way to

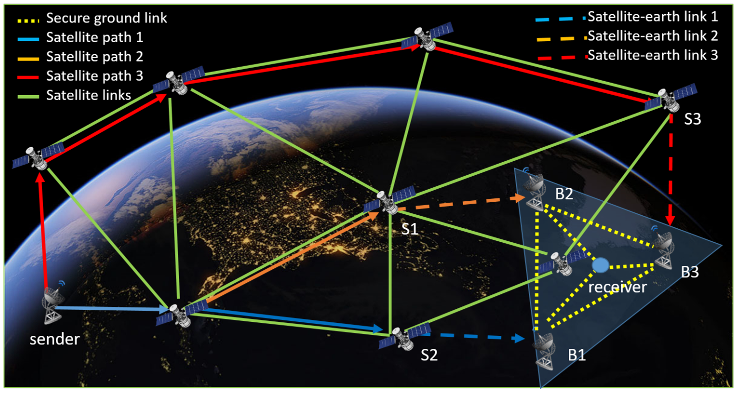


Figure 3.2: Spatial diversity: Sender simultaneously transmits data on multiple paths to a receiver. Receiver belongs to secure ground network and collect data from multiple satellite links. This path diversity architecture increase security and robustness against channel failure.

satisfy each user's QoS requirement while minimizing the probability of the transmitted data to be compromised.

3.2 Related works

Our work in this chapter shares similar flavor with rich literature in wireless communication that uses spatial and frequency diversity [51] to increase capacity and combat fading. On the other hand, our work aims to propose a high level model optimization where each channel does not need to be a frequency band. Our work also proposes a randomized algorithm to send data of multiple users simultaneously onto these abstract channels in such a way to satisfy each user's QoS requirement.

Also, our work in this chapter is similar to many works on resource and scheduling algorithms [29, 3, 30, 40]. However, a majority of these algorithms are deterministic which lead to higher complexity. Some of these algorithms aim to find approximate solutions to hard problems such as the job scheduling under deadline constraints [43, 6], which have been shown to be NP-complete. On the other hand, our proposed algorithm is probabilistic in nature that results in low complexity which can be implemented for SATCOM. Even though the proposed algorithm is probabilistic, the approximate solution is proved to be bounded within the optimal solution with high probability.

3.3 Mathematical model of the problem

3.3.1 Convex Optimization Formulation

In this section, we formulate the multi-user scheduling problem as a convex optimization problem. The solution to the convex optimization problem is then used in a randomized algorithm to ensure security while satisfying all the QoS requirements. The multi-user scheduling problem is as follows. Given a number of users with the pre-specified QoS requirements (e.g., throughputs), a number of available channels and their qualities (e.g., packet loss rates), prior knowledge about how secure each channel is, and the number of time slots for transmitting the data, we want to decide, for any given time slot, which user's data and which channel it should be sent on, in order to satisfy each user's QoS requirement and maximizing the ”security” level of data.

The decision on which user's data will be transmitted at a particular time slot and channel is probabilistic. We emphasize on the importance of probabilistic methods, i.e., randomized scheduling algorithms for security. First, a typical deterministic algorithm poses higher security risk than a typical randomized scheduling algorithm. Consider a following simple example in which there is only a single user, and two available channels for transmitting its data. At any time slot, using a deterministic algorithm, we can either to transmit the data on channel 1 all the time or on channel 2 all the time. In this case, an attacker/eavesdropper can listen to these channels over multiple time slots and determine the transmitting schedule correctly, and therefore designs an effective attack, e.g., jamming on the transmitted channel only. On the other hand, suppose that at any time slot, we flip an unbiased coin. If the head occurs, then data is transmitted on channel 1. Otherwise, data is transmitted on channel 2. Using such a randomized scheduling algorithm, the attacker would have only 50% chance of predicting the correct channel regardless of the number of time slots it observes. Note that the throughput of both schemes are the same. Our proposed algorithm uses the same randomized approach in which for a given time slot and a channel, the scheduling algorithm will send the data from a user with some probability.

Before mathematically formulating the problem, we introduce the notion of entropy and Kullback-Leibler (KL) distance [9] in order to quantify our notion of security. Intuitively, entropy characterizes the amount of information or amount of uncertainties in the outcomes of an i.i.d (independently identically distributed) random variable. For a discrete random variable X, it is defined as:

$$H(X) = -\sum_i p(x_i) \log \left(p(x_i)\right), \tag{3.1}$$

where $p(x_i)$ denotes the probability mass function of X. On the other hand, KL distance is a measurement between the two distributions $p(x)$ and $q(x)$. It is defined as:

$$KL(P||Q) = \sum_i p(x_i) \log \frac{p(x_i)}{q(x_i)}. \tag{3.2}$$

Note that KL distance is not symmetric. Nevertheless, a smaller KL distance implies the two distributions are more similar, and a larger KL distance implies otherwise.

We now apply the notion of information theoretic security to our multi-channel setting as follows. Consider a single user and let $p(x_i)$ denote the probability that a piece of data from that user is sent on channel i at any time slot. Then, the best way to prevent a jammer/eavesdropper from making a good prediction about which channel is used to transmit the data is to make $p(x)$ a uniform distribution, i.e., $p(x_i) = 1/c$ where c is the number of channels. Furthermore, the probability of correctly predicting the transmit channel is $1/c$, regardless of the number of time slots used by the eavesdropper/jammer to collect the statistics due to the i.i.d distribution of $p(x)$. While using a uniform distribution is good for security, the QoS requirement might not be satisfied due to some low quality channels. Therefore, one approach to balance between security and performance is to find a distribution that is as close to the uniform distribution as possible while satisfying the QoS requirements. Specifically, using KL distance and let $q(x) = 1/c$, we want to find $p(x)$ such as $KL(p||q)$ is minimum while the QoS constraints are satisfied. We will discuss these constraints shortly. Now using Eq. 3.2, we have:

$$\begin{aligned} KL(p||q) &= \sum_i p(x_i) \log \frac{p(x_i)}{q(x_i)} && (3.3) \\ &= \sum_i p(x_i) \log p(x_i) - \sum_i p(x_i) \log \frac{1}{c} && (3.4) \\ &= -H(X) - \log \frac{1}{c}. \end{aligned}$$

Since $\log \frac{1}{c}$ is a constant, for a single user, the objective is to find the $p(x)$ that minimizes the negative entropy.

We are now ready to formulate the multi-user randomized scheduling problem. We will first describe a generic problem from which many scheduling problems with various

QoS metrics can be cast as one of its instances. The generic problem is as follows. There are $u-1$ $(1 \ldots u-1)$ actual users and one ”virtual” user u, c number of channels, t number of time slots. At every time slot i and for every available channel j, the scheduler will randomly choose a user k and send its data on the channel j with probability x_{ij}^{k}. x_{ij}^{u} denotes the probability that the scheduler does not send any data. A reward r_{ij} is rewarded to a user k if k is chosen for sending data in time slot i and on channel j. Furthermore, each user k requires at least an average reward R^k over the t time slots. The goal is to find the x_{ij}^{k*}, $i = 1, 2, \ldots, t, j = 1, 2, \ldots, c, k = 1, 2, \ldots, u$ that maximize the weighted entropy of the conditional probability distributions $y_j^k = \sum_i x_{ij}^k$ of sending data for the actual user k (i.e., $k \neq n$) given channel j while satisfying the average reward constraints R^k per time slot for all users.

Table 3.1: Chapter 2: Notations

u	Total number of users
c	Total number of available channels
t	Number of time slots in a round
r_{ij}	Reward for sending data at time slot i on channel j
R^k	Average of reward for user k per time slot
x_{ij}^k	Conditional probability of sending data for user k given time slot i and channel j, $x_{ij}^k \overset{\triangle}{=} P(k\|i,j)$
y_j^k	Conditional probability of sending data for user k given channel j, $y_j^k = \sum_i x_{ij}^k$
y_j	$\overset{\triangle}{=} (y_j^1, y_j^2, \ldots, y_j^u)$

Now, let y_j denote the probability mass vectors of length u whose elements are $y_j^k = \sum_i x_{ij}^k$. y_j is a valid probability mass function that specifies the conditional probability of sending data for different users for given channel j. As discussed previously, we want y_j to be similar to a uniform distribution since if an attacker eavesdrops on channel j, it can only observe a fraction of the data of a user k since data for each user is spread out due to the effect of uniform-like distribution. Since we want y_j to be similar to a uniform distribution for every channel j, then based on previous discussion, we want to minimize the weighted sum of negative entropies of y_j as:

$$f(y_1, y_2, \dots y_c) = -\sum_{w_j} w_j H(y_j), \tag{3.5}$$

where $H(y_j)$ denotes the entropy of y_j and w_j denotes the pre-specified weights for each channel. When channel j is more secure, w_j is be smaller, implying that one can send a large fraction of a single user on channel j. Using the notations in Table 3.1, the multi-user scheduling problem then can be cast as the following convex optimization problem:

Problem P1

Minimize: $-\sum_{w_j} w_j H(y_j)$

Subject to:

$$\sum_k x_{ij}^k = 1, i = 1, \dots, t, j = 1, \dots, c, \tag{3.6}$$

$$y_j^k = \sum_i x_{ij}^k, j = 1, 2, \dots, c, k = 1, \dots, u, \tag{3.7}$$

$$\frac{1}{t} \sum_{i,j} x_{ij}^k r_{ij} \geq R^k, k = 1, \dots, u-1, \tag{3.8}$$

$$x_{ij}^k \geq 0, i = 1, \dots, t, j = 1, \dots, c, k = 1, \dots, u, \tag{3.9}$$

$$x_{ij}^k \leq 1, i = 1, \dots, t, j = 1, \dots, c, k = 1, \dots, u. \tag{3.10}$$

Equality constraint Eq. 3.6 ensures that the probabilities of sending data of different users for a given time slot and channel must add to 1. The inequality constraints (3.9) and (3.10) arise by definition of probability. Constraint (3.7) is simply a calculation of marginal distribution, and constraint (3.8) enforces the minimum reward for each user. Note that (a) the reward in constraint Eq. 3.8 is just an average reward per time slot and (b) the constraints are not enforced for the virtual user u. The virtual user u is used to allow the algorithm not to send any data for any user. Specifically, when $x_{ij}^{u} > 0$ then there is a non-zero probability that the system is not sending any data on channel j at time slot i. This happens when there is enough system resource to accommodate every user.

P1 is a convex optimization problem because the objective function is convex and all the constraints are linear. The objective function is convex because any entropy $H(y_j)$ is a concave function in the distribution y_j [9] and a positive linear combination of concave functions is also a concave function [5], and therefore $-\sum_{w_j} w_j H(y_j)$ is a convex function.

We note that many multi-channel communication settings with QoS requirements can be cast as an instance of problem **P1**. As an example, we consider a simple scenario for which the notion of rewards can be used to model the QoS requirements. In this scenario, suppose each channel j has a packet loss rate p_j (independent of time slot), some fixed sending rate s_j packets per second, and the duration of a time slot is d seconds. Furthermore, each user k requires an average throughput of at least T_k packets per second. The objective and all the constraints except constraint (3.8) in **P1** can be used for this scenario. Constraint (3.8) can be easily modified to reflect the current scenario by letting

$$r_{ij} = p_j s_j, R^k = \frac{T_k}{d}.$$

Note that the throughput is not the only applicable QoS criteria. Other metrics/constraints such as power, packet losses, or even the combination of them, can be modeled using appropriate reward functions. We also note that there exists many efficient methods for solving convex optimization problems such as gradient descent algorithms [5]. In practice, **P1** is small and can be solved in real time. Therefore, we will not discuss the algorithmic solution for **P1**.

3.3.2 Analysis

When the optimal x_{ij}^{k*} is found, the scheduling algorithm is simple because for a given time slot and a channel, it just simply picks the data from a user based on x_{ij}^{k*}. However, since the algorithm is probabilistic, the empirical average reward $\hat{R}_t^k$ for user k per time slot over t time slots is a random variable:

$$\hat{R}_t^k = \frac{\sum_{i,j}^{t,c} R_{ij}^k}{t},$$

where R_{ij}^k are i.i.d Bernoulli random variables, i.e.

$$R_{ij}^k = \begin{cases} r_{ij} & \text{with probability } x_{ij}^{k*} \\ 0 & \text{with probability } 1 - x_{ij}^{k*}. \end{cases}$$

By the weak law of large number and if $r_{ij} = r_j$ for all i, we would expect that

$$\lim_{t \to \infty} \hat{R}_t^k \to R^k = \sum_{i,j}^{t,c} x_{ij}^{k*} r_{ij}.$$

In practice, we are interested in how close $\hat{R}_t^k$ to R^k for a given number of channels and a finite number of time slots. Specifically, we consider the probability that $\hat{R}_t^k < (1-\epsilon)R^k$ which is the probability that the algorithm produces an empirical average reward that is smaller than the required reward by a factor of ϵ for a given number of channels c over a finite number of time slots t. We want $P(\hat{R}_t^k < (1-\epsilon)R^k)$ as small as possible. We have the following Proposition 1:

Proposition 1. *Let x_{ij}^{k*} be the optimal solution to problem* **P1**, *$\mathcal{A}_k = \{(i,j)|x_{ij}^k > 0\}$, and define $\mu^k = \sum_{i,j}^{t,c} x_{ij}^{k*}, r_{max}^k = \max_{(i,j)\in\mathcal{A}_k} r_{ij}, r_{min}^k = \min_{(i,j)\in\mathcal{A}_k} r_{ij}$, then*

1.

$$P\left(\hat{R}_t^k < (1-\epsilon)R^k\right) \leq \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{(1-\epsilon)\frac{r_{max}^k}{r_{min}^k}}}\right)^{\mu^k}. \tag{3.11}$$

2. If $\frac{r_{max}^k}{r_{min}^k} > \frac{2}{2-\epsilon}$ then $\epsilon - (\epsilon - \frac{\epsilon^2}{2})\frac{r_{max}^k}{r_{min}^k} \geq 0$. Let $C = \epsilon - (\epsilon - \frac{\epsilon^2}{2})\frac{r_{max}^k}{r_{min}^k}$, then

$$P\left(\hat{R}_t^k < (1-\epsilon)R^k\right) \leq e^{-C\mu^k}. \tag{3.12}$$

Furthermore, for a common case where $r_{ij} = r_j, \forall i, j$, i.e, the rewards only depend on channel and not time, and let $\lambda^k = \sum_j^c x_{1j}^{k*}$, then:

$$P\left(\hat{R}^k < (1-\epsilon)R^k\right) \leq e^{-C\lambda^k t}. \tag{3.13}$$

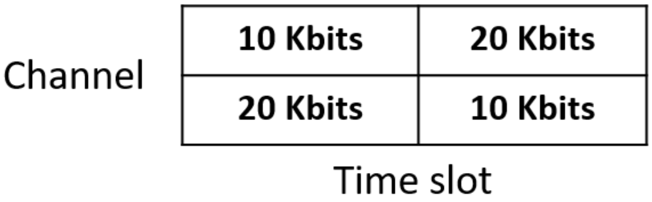
Proof. See the Appendix A. □

The main intuition of Proposition 1 is that when running the proposed randomized algorithm, the probability of the empirical reward (e.g. throughput) is less than the specified reward requirement by a factor of ϵ is exponentially decreased with the numbers of time slots and channels. To see this, Eq. 3.11 is simplified into an exponential form in Eq. 3.12. We note that $\mu^k = \sum_{ij}^{t,c} x_{ij}^{k*}$ is proportional to the number of time slots t for a given number of channels. In typical cases, $c > 0$ and therefore the probability that the empirical reward is less than the required reward by a factor ϵ is exponentially decreased with the number of time slots used. This can be seen by Eq. 3.13 where r_{ij} is assumed to be constant (again a typical case) with respect to time slot i. Given a time slot duration is typically small, the randomized algorithm will most certainly produce results that satisfy the average reward requirement (e.g. throughput) over a reasonable short duration. Next, we discuss the results for some real world scenarios.

3.4 Discussions and Results

We present the simulation results for five different scenarios to illustrate the benefits of the proposed randomized algorithm.

Scenario 1: We consider a simple scenario to illustrate the intuitive result of the algorithm. Specifically, we consider only two real users, two channels, and two time slots. The time slot duration is 1 ms. During each time slot and for a given channel, the sender can transmit an amount of data of 10 Kbits or 20 Kbits, depending on the time slot and channel as shown in Fig. 3.3. Assuming that each user requests 15 Mbps. Now, there are more than one way to schedule the transmission that satisfy the user's requests. A simple scheduler would transmit user one's data on channel 1 and user 2's data on channel 2 all the time. Clearly, this assignment satisfies the users requests since each channel can support 30 Kbits/2 ms = 15 Mbps. However, this scheme is vulnerable since a jammer/eavesdropper can attack/eavesdrop on user 1 by focusing on channel 1 (assuming that an attacker with limited resource can only attack/eavesdrop on only one channel). On the other hand, a scheme that randomly transmits the data of user 1 (or 2) on channel 1 at 50% of the time and the other 50% on channel 2 is much more secure since an eavesdropper will not be able to have access to the entire data of a single user. Indeed, the proposed algorithm produces such results as shown in Fig. 3.4.



10 Kbits	20 Kbits
20 Kbits	10 Kbits

Channel
Time slot

Figure 3.3: Reward structure for scenario 1

Fig. 3.4(a) shows the conditional probability that the data is sent on each channel given that user k' is selected for sending. The two bars indicates the two channels. Note that the third user in Fig. 3.4(a) is a virtual user that models the scenario where the algorithm does not send any data. As seen, for both users, there is a 50% chance that their data will be sent on channel 1 or 2, making this scheme robust to eavesdropping and jamming. Fig. 3.4(b) shows the conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending. For an eavesdropper that has access to channel 1, it can see data from user 1 50% of the time, and similarly for user 2. Fig. 3.4(c) shows the average throughput for two users as a function of time. As seen, the average throughput is around the requested throughput of 15 Mbps and the variances of these throughputs reduce as the number of time slots increases as predicted by Proposition 1.

Scenario 2: In scenario 1, the total requested throughput is equal to the total system throughput. In scenario 2, the reward structure is the same as that of scenario 1 except the requested throughputs are reduced to 13.5 Mbps for each user. Fig. 3.5(a) shows the conditional probability that the data is sent on each channel given that user k is selected for sending. Again, for a given user, the data is uniformly distributed over the two channels. Fig. 3.5(b) shows the conditional probability that the data sent on channel j belongs to user k given that channel j is selected for sending. For an eavesdropper that has access to channel 1, it can see data from user 1 45% of the time, data from user 2 45% of the time, and 10% of time it sees no data transmitted. The third (yellow) bar represents the probability that the "data" come from the virtual user, i.e., no data being transmitted. This is plausible since the total requested throughput is 10% below the system throughput. Fig. 3.5(c) shows the average throughput for two users as a function of time. As seen, the average throughput is around the requested throughput of 13.5 Mbps and the variances of these throughputs reduce as the number of time slots increases as predicted.

Scenario 3: In this scenario, we consider a scenario consisting of 5 users, 5 channels, and 1 time slot. In this case, $r_{ij} = r_j = 15$ Kbits. In other words, the reward only depend on channel, not time, and all the channels have the reward of 15 Kbits per time slot (1 ms). Or equivalently, the average throughput is 15 Mbps. The requested throughput for user 1 to user 5 are: 25 Mbps, 20 Mbps, 15 Mbps, 10 Mbps, and 4 Mbps. In this case, the system is operating at near capacity of 75 Mbps (5×15 Mbps). Fig. 3.6(a) shows the conditional probability that the data is sent on each channel given that user k is selected for sending. As seen, for a given user, the data is uniformly distributed over the channels. This makes sense since all the channels have the same capacity and it makes no difference to distribute the data of any user to any channel. Note that user 6 is the virtual user. On the other hand, Fig. 3.6(b) shows the conditional probability that the data sent on channel j belongs to user k given that channel j is selected for sending.

Since the requested throughput for each user is different, we can see that the probability of sending data for different users on a given channel is different. The probability of sending data for user 1 is the highest since it requests 25 Mbps and the probability of sending data for user 5 is lowest since it requests 4 Mbps. Fig. 3.6(c) shows the average throughput for different users as a function of time. As seen, the average throughput is around the requested throughputs for each user. Furthermore, the variances of these throughput reduce as the number of time slots increases as predicted.

Scenario 4: This scenario is similar to that of scenario 3 except the reward structure is changed. In particular, channels 1 to 5 have rewards of 5 Kbits, 10 Kbits, 15 kbits, 20 Kbits, and 25 Kbits. The time slot duration is still 1 ms, and the requested throughput for user 1 to user 5 are still: 25 Mbps, 20 Mbps, 15 Mbps, 10 Mbps, and 4 Mbps. Fig. 3.7(a) shows the conditional probability that the data is sent on channel given that user k is selected for sending. As seen, for a given user, the data is no longer uniformly distributed over the channels. This is due to the fact that the channels do not have the same capacity as in scenario 3. For each user, it is not possible to send data on each channel evenly to maximize the entropy like in previous scenarios, otherwise the throughput requirement will not be satisfied. Instead, while still maintaining the distribution close to the uniform distribution, the users with smaller throughput requirement use less of those channels with high capacity in order to save bandwidth for those users with larger requirement. This is confirmed in 3.7(a). On the other hand, Fig. 3.7(b) shows the conditional probability that the data sent on channel j belongs to user k given that channel j is selected for sending. Again, it shows that a greater amount of better channels will be used to serve those users with higher throughput requirements. Due to the varying capacities of different channels, the channels are no longer uniformly used. Fig. 3.7(c) shows the average throughput for different users as a function of time. As seen, the average throughput is around the requested throughputs for each user and the variances of these throughput reduce as the number of time slots increases.

Scenario 5: In this scenario, one of total 5 channels is less secure than other channels. As a result, the weight w_j of each channel is not the same anymore. The weight parameter is changed from $[1, 1, 1, 1, 1]^T$ to $[0.5, 0.5, 0.5, 3, 0.5]^T$, indicating that the 4th channel is significantly more vulnerable to attack than other channels. Throughput requirement and all 5 channels' rewards are the same as in Scenario 3. Fig. 3.8(a) shows the conditional probability that the data is sent on a channel given a specific user. Notice that overall the usage of channel 4 is decreased as compared to Scenario 3. (i.e., the usage of channel 4 by the virtual user 6 is increased, which means the "idle" time for channel 4 is increased.) In order to compensate its vulnerability, channel 4's conditional distribution is closer to a uniform one as compared to other channels, so that it would be more difficult for the potential attackers to get information from this channel. This result is shown in Fig. 3.8(b). Still, Fig. 3.8(c) shows all users throughput requirements are satisfied, and the variances decrease as the number of time slots increases.

Scenario 6: In this scenario, we show that different QoS constraints can be applied to this framework, and the combination of them can solve a more sophisticated real world scenario problem. First, we assume that each channel has a fixed transmitting power, and each user has an energy budget. The time slot is still 1 ms. Channel 1 to 5's transmission powers are 2 mW, 2 mW, 2 mW, 1 mW, 1 mW, respectively, and user's power budgets are 2 mW, 1.8mW, 1.6 mW, 1.4 mW, 1.2 mW. Now, the throughput constrains in Eq. 3.8 is replaced by the energy reward for each channel: -2 μJ, -2 μJ, -2 μJ, -1 μJ, -1 μJ. A negative reward indicates the energy is consumed when a channel is chosen. For each user, the power it consumed in each time slot should be less than its power budget. Fig. 3.9(a) shows the simulation results of channel usage and Fig. 3.9(b) is the actual power consumption. Notice that all users' power consumption is less than the budget, and users 1 to 4 do not consume the maximum power. By using less power, it makes user's data distribution closer to the uniform distribution. So it would be more difficult for a potential attacker to have access to the user's data on a particular channel. Thus, extra power is not necessary in this case.

Fig. 3.10(a), Fig. 3.10(b) and Fig. 3.10(c) show a more realistic case when both throughput and power consumption are taken into account. As a result, there are two kinds of rewards for each channel/time slot: a) A positive reward indicates throughput gain. b) A negative reward indicates the power it consumed. These combined rewards can be found in in Table 3.2(a). Both kinds of reward should be constrained by the resource available for each user, shown in Table 3.2(b). Thus, similar to Eq. 3.8, two constrains corresponding to throughput and power can to be added to Problem P1. Fig. 3.9(c) shows the optimized transmission schedule given channel 1 to 5 is chosen. Fig. 3.9(d) and Fig. 3.9(e) show both power and throughput requirements are satisfied.

Table 3.2: Rewards and constrains for Scenario 6

Channel	1	2	3	4	5
Throughput reward (Kbits)	30	30	20	15	15
Energy cost (μJ)	-2	-2	-2	-1	-1

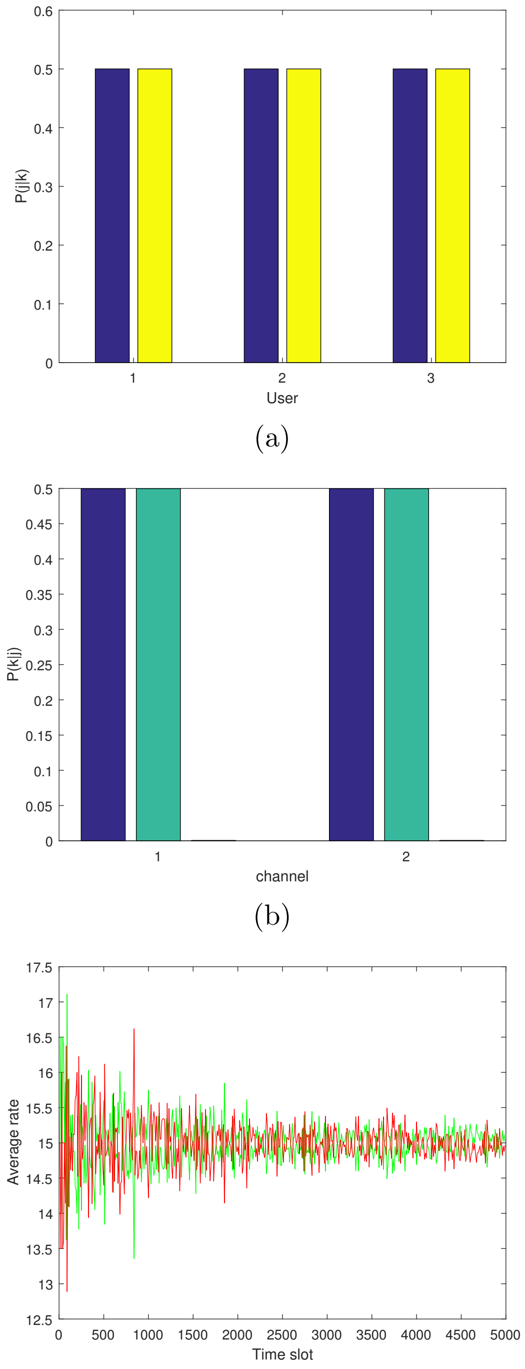
(a)

User	1	2	3	4	5
Throughput requirement (Mbps)	25	25	20	20	15
Power budget (mW)	2	1.8	1.6	1.4	1.2

(b)

3.5 Summary

In this chapter, we proposes a novel multi-user transmission scheduler that aims to alleviate eavesdropping/environmental fading issues of SATCOM via time and channel diversity. In particular, the proposed scheduler is a low complexity randomized algorithm that multiplexes user data over multiple frequencies and time slots to combat those issues while satisfying the Quality of Service (QoS) requirements of individual users with high probability. Analysis and simulation results demonstrate effectiveness of the proposed scheduler.

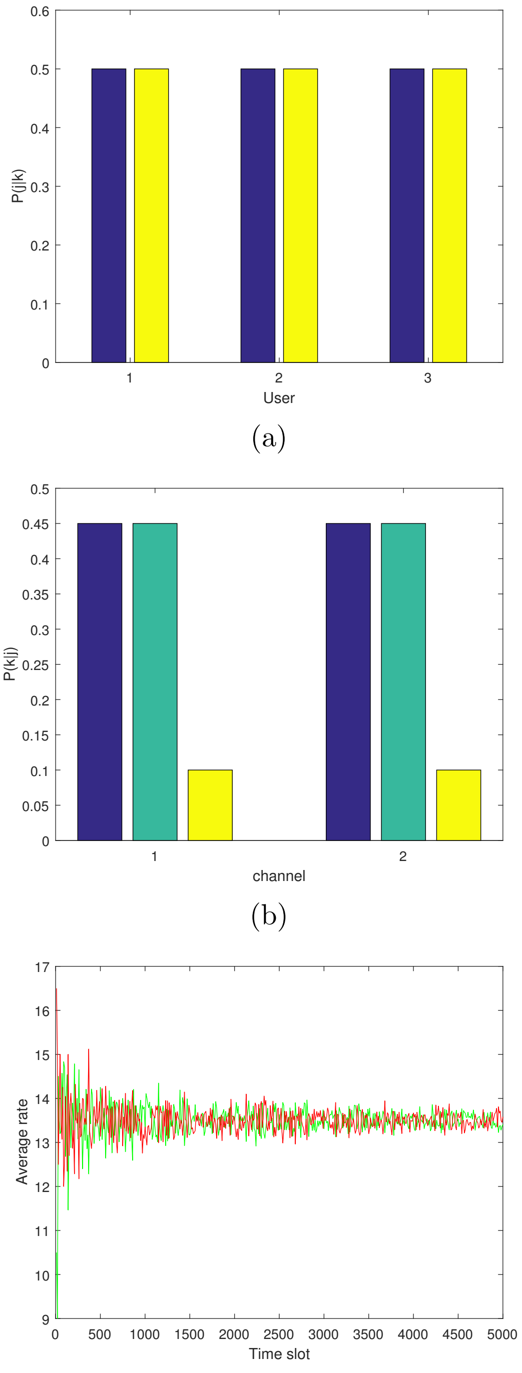


(a)

(b)

(c)

Figure 3.4: Scenario 1: (a) Conditional probability that the data is sent on a channel j, given that user k is selected for sending; (b) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending; (c) Average throughput of users as a function of time slots

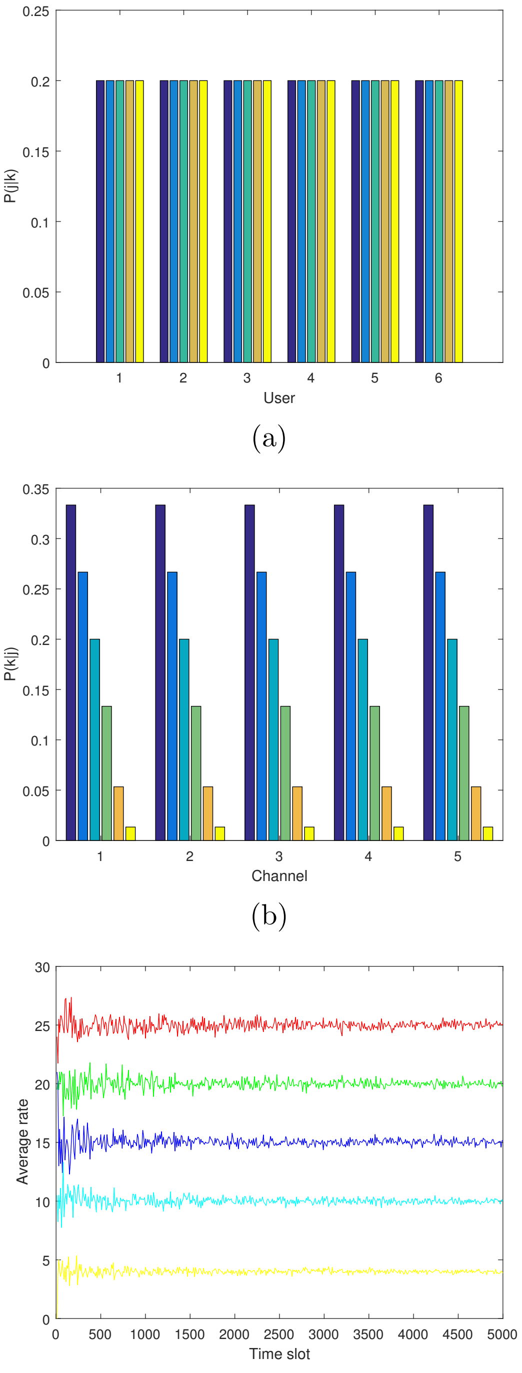


(a)

(b)

(c)

Figure 3.5: Scenario 2: (a) Conditional probability that the data is sent on a channel j, given that user k is selected for sending; (b) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending; (c) Average throughput of users as a function of time slots

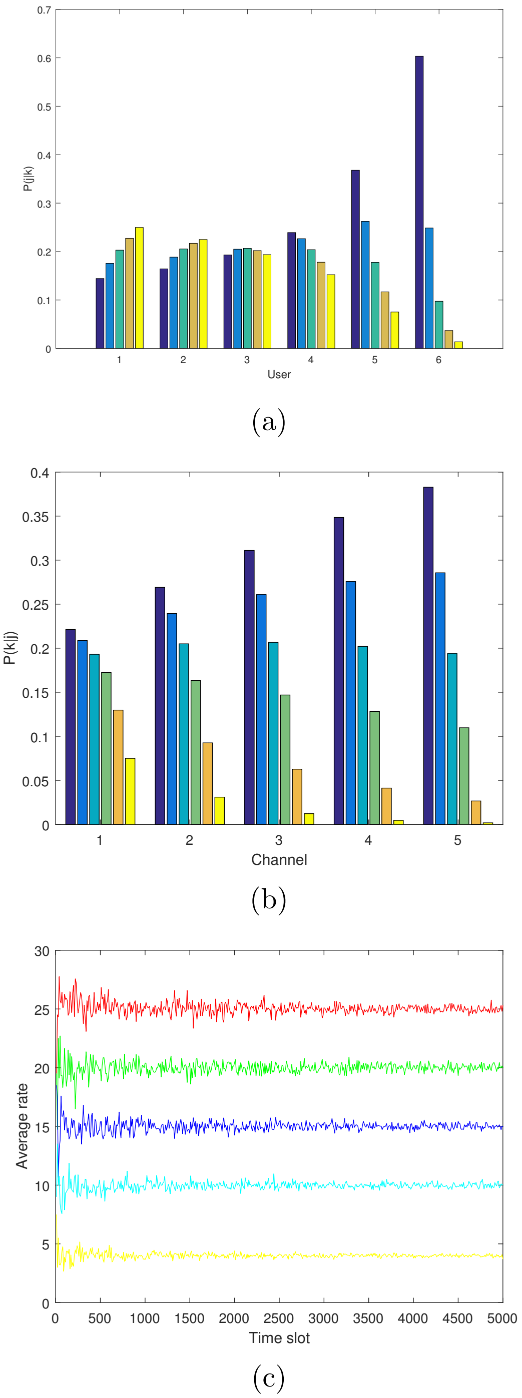


(a)

(b)

(c)

Figure 3.6: Scenario 3: (a) Conditional probability that the data is sent on a channel j, given that user k is selected for sending; (b) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending; (c) Average throughput of users as a function of time slots

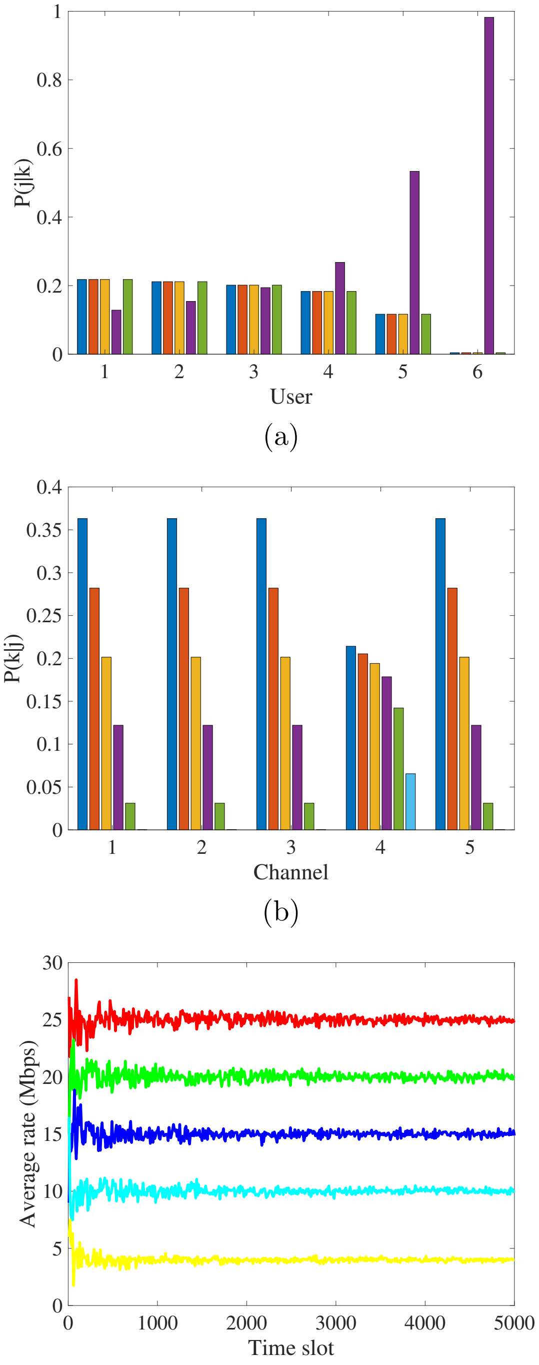


(a)

(b)

(c)

Figure 3.7: Scenario 4: (a) Conditional probability that the data is sent on a channel j, given that user k is selected for sending; (b) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending; (c) Average throughput of users as a function of time slots

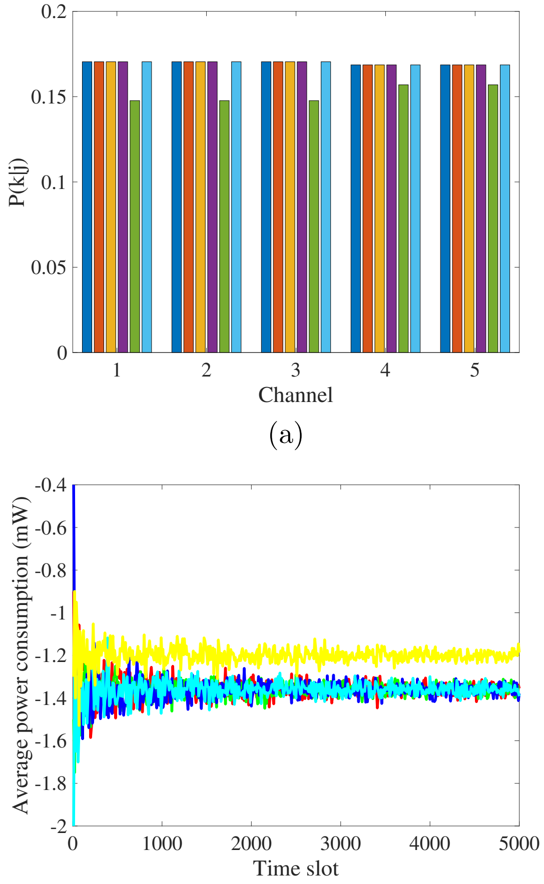


(a)

(b)

(c)

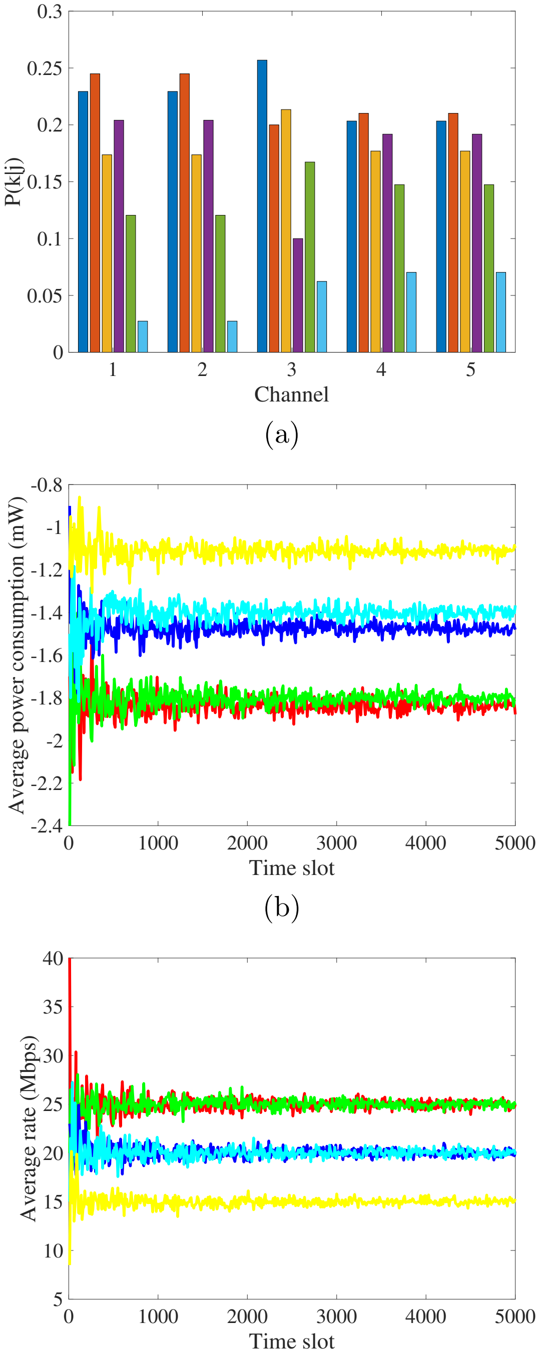
Figure 3.8: Scenario 5: (a) Conditional probability that the data is sent on a channel j, given that user k is selected for sending; (b) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending; (c) Average throughput of users as a function of time slots



(a)

(b)

Figure 3.9: Scenario 6: (a) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending, with power budget; (b) Average power consumption of users as a function of time slots, with power budget (c) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending, power and throughput constraints combined; (d) Average power consumption of users as a function of time slots, power and throughput constraints combined; (e) Average throughput of users as a function of time slots, power and throughput constraints combined



(a)

(b)

(c)

Figure 3.10: Scenario 6: (a) Conditional probability that the data sent on channel j belong to user k given that channel j is selected for sending, power and throughput constraints combined; (b) Average power consumption of users as a function of time slots, power and throughput constraints combined; (c) Average throughput of users as a function of time slots, power and throughput constraints combined

Chapter 4: Application III: Mitigating satellite jamming: A game theory prospective

4.1 Description

Satellite jamming has its roots in radio frequency (RF) jamming [46]. RF jamming is a simple idea. Its aim is to degrade the signal's integrity between a pair of senders and receivers by transmitting noise with sufficient power on the same communication band as the sender and receiver in order to lower the signal-to-noise ratio (SNR) of their transmission. Consequently, RF jamming can reduce or effectively cut off the communication link between the sender and receiver. Fig. 4.1 shows a typical scenario

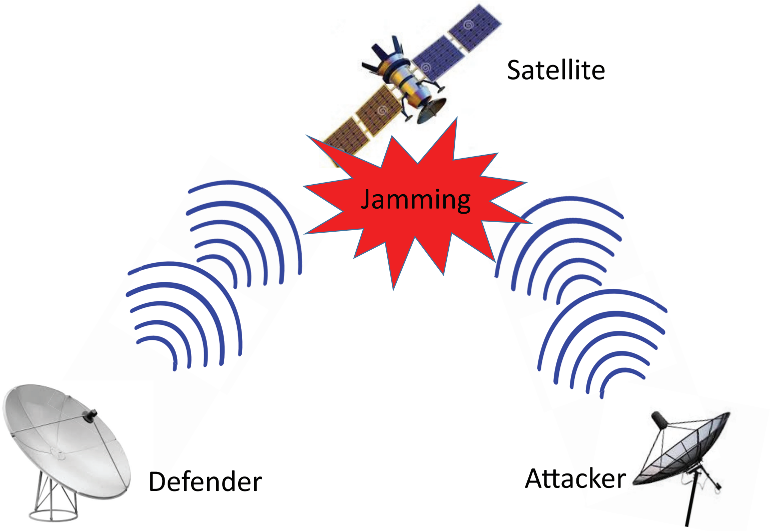


Figure 4.1: A typical scenario of a uplink radio interference and mitigation with potential satellite communications applications.

where an attack occurs at a satellite. The defender transmits information to the satellite using the spread-spectrum technique, where the transmitted signal is spread over multiple transmission bands. On the other hand, the attacker tries to reduce the information rate by transmitting noise via spread-spectrum techniques, i.e., jamming the defender's signal. A jamming attack is successful if the attacker is able to greatly reduce the defender's information rate. Central to a successful attack is the capability of the jammer, which includes the following: (1) transmission power and (2) information about the frequency on which the good signal is transmitted. The reason for this is clear because the noise generated by the jammer needs to have sufficient power and to be on the same band as the good signal in order to reduce the SNR of the good signal. For satellite communications, the transmission between earth-based terminals is relayed by a satellite. Thus, an effective way for the jammer to attack is through the relay, i.e., the satellite, since it is more difficult to attack the terminal. The difficulty comes from the fact that the jammer needs to be in proximity of the receiver, which it may know, or it might increase the potential of being detected. Thus, in this chapter, we will analyze the frequency hopping (FH) radio jamming and mitigation in which, both the jammer and the defender will employ their optimal strategies based on what they know from a zero-sum game theoretic setting. Specifically, our contributions include:

1. Formulate the problem of minimizing the damaging effect of satellite jamming attacks using the two-player asymmetric zero-sum game framework. The payoff is modeled as the channel capacity of the defender under white additive Gaussian noise. The defender and attacker are capable of spreading their signals over a pre-specified frequency band.

2. Provide performance analysis for the *Perfect Information Game.* In this scenario,

both attacker and defender are rational and intelligent entities with perfect knowledge of the game. We show that there exists an optimal Nash equilibrium (NE) strategy for each player. Furthermore, we obtain a closed-form for the NE strategies that turns out to be a *modified* version of the well-known water-filling problem [10]. Any deviation from their own NE strategy would reduce their payoffs.

3. Provide performance analysis for the *Defender-biased game (typical cases)*. In this scenario, an attacker has partial information about the game, while the defender has perfect information about the game. We show that the defender will take advantage of this lack of knowledge and play an optimal strategy to obtain a payoff that is higher than the rate obtained if the attacker would play the NE strategy with perfect information. This is the important property of a game that has NE.

4. Provide performance analysis for the *Attacker-biased game (rare cases)*. We analyze the special case when the attacker knows the defender's strategies, but the defender does not know the attacker's strategy due to imperfection information. We provide an algorithm to find the corresponding payoffs.

4.2 Related works

There exists rich literature on secure SATCOM. To prevent eavesdropping, standard cryptographic techniques for transmissions in insecure environments [32] can be directly used in SATCOM. Cryptographic techniques encrypt the transmitted information such that even when an eavesdropper obtained the encrypted data, it cannot recover the transmitted information without a secret key. Cryptographic techniques encrypt the transmitted information such that even when an eavesdropper obtained the encrypted data, it cannot recover the transmitted information without a secret key. Cryptographic techniques however require key exchange protocols, tend to be computational expensive, and lead to consume more satellite resources. Furthermore, cryptographic techniques cannot defend against the jamming attack. Jamming attacks aim to degrade the signal's integrity between a pair of sender and receiver by transmitting noise with sufficient power on the same communication band as the sender and receiver in order to lower the signal-to-noise ratio (SNR) of their transmission. Consequently, jamming attack can reduce or effectively cut off the communication link between the sender and receiver. To mitigate jamming attack in SATCOM, frequency-hopping and spread spectrum techniques [54] have been proposed recently. The main idea is to spread the transmitted information across different frequency bands so that if one frequency band is jammed, information can be recovered from other unjammed frequency bands. Our work is similar to these works in the sense that we use channel and time diversity to combat eavesdropping and jamming. On the other hand, ours is the first to quantify the diversity via information theoretic quantity entropy.

RF jamming has been used to disrupt radar systems that guide aircraft and missiles. It is also used to disrupt radio broadcast stations in wartime or during tense periods in enemy countries [45]. Currently, there also has been a rise in the number of cases in which RF jamming techniques are used to launch denial of service (DoS) attacks in WiFi and cellular networks [57]. Notably, wireless sensor networks are most vulnerable to RF jamming attacks due to their limited transmission power and capability of mitigating attacks [55], [27]. Early literature on defense techniques against RF jamming attacks typically focused on narrow band jamming. Specific techniques such as transversal filters [26] or the singular-value decomposition (SVD)-based method [52] are proposed to suppress single-tone attacks. On the other hand, when information on jamming frequency is not known at the defender, defense schemes using channel codes such as convolutional codes, or Bose-Chaudhuri-Hocquenghem (BCH) codes, have been shown to be highly effective [31]. However, these techniques introduce extra latency and bandwidth. Recently, a number of adaptive anti-jamming techniques have been proposed for global positioning system (GPS) satellites [25]. For example, these schemes include adaptive antenna array [53] and frequency/time domain filtering [7]. Another type of anti-jamming technique uses spread-spectrum methods such as frequency hopping [23], [18], [19] to evade the jammer. Specifically, in [18], a detect/transmit mode switch mechanism is proposed to identify the jamming frequency statistics, and an optimized frequency hopping strategy is proposed based on the Markov decision process [18]. In [19], the defender observes the jamming statistics and, based on this, generates a frequency hopping pattern to minimize the error rate caused by jamming. Other spread-spectrum-based techniques such as a scheme using notch filters on the base band [12] are also shown to be effective against jamming attack. More recently, many novel approaches have been proposed on jamming/anti-jamming attacks. [15] improves the attack efficiency towards a wireless smart grid network by dynamically implementing spoofing and jamming. The optimality is found by dynamic programming. A security-aware efficient data transmission scheme for Intelligent Transportation System (ITS) is introduced in [16] by cloud-based server using dynamic server selection methodology.

All of the aforementioned techniques assume that attackers are not sufficiently knowledgeable about the defender. On the other hand, a sophisticated attacker can employ different jamming strategies adaptively to reduce the effectiveness of a defense strategy. Essentially, both the defender and attacker play a game in which the defender tries to maximize some payoff, e.g., throughput, and the attacker tries to minimize it. Therefore, the game theory approach [38] is often employed to study channel security as well as spectrum allocation [47], [41]. Work based on game theory in the context of FH jamming [60], [17], [1] has also been done. For example, in [60], the NE of an uncoordinated frequency hopping (UFH) scenario is characterized by showing a mixed strategy for the transmitter, receiver, and jammer. A more sophisticated scenario, namely quorum-based FH rendezvous, is analyzed in [1]. Unlike the other FH techniques that simply randomly pick a frequency band, a quorum-based FH rendezvous uses a quorum rule to pick the transmission channel, and the jammer chooses the attack channel in the same way. In this case, the NE of a three-player game is shown not to exist, but does exist for a simplified two-player game. Additionally, recent research has focused on the gaming analysis of a timing channel [56], [59], [58], [13]. In [56], W. Xu et. al described the timing channel anti-jamming technique. The timing channel is able to transmit data encoded by the time duration of a signal. The power allocation game between the transmitter and the defender in a timing channel can be modeled as a non-zero-sum Stackelberg Game. Unlike Nash equilibrium, a Stackelberg equilibrium assumes that one player is leading while the other is following. In the game analyzed in [59], [58], [13], the transmitter is modeled as the leader and the attacker is the follower. It is proved that a Nash equilibrium and a Stackelberg equilibrium both exist, and the latter performs better for the transmitter. While the analysis is thorough for a timing channel with specific payoff functions, our work in this chapter takes a more generalized approach by defining the payoff as the total capacity.

4.3 Mathematical model of the problem

Table 4.1 shows the notations that are used.

We assume a free space path loss model, as shown in Fig. 4.1, because both the

Table 4.1: Chapter 3: Notations

N	Number of discrete frequency bins
B	Bandwidth per frequency bin
P_D	Total power received at satellite relay from the defender
P_A	Total power received at satellite relay from the attacker
P_N	Average noise power over all frequency bins
$\mathbf{n} \in \mathbb{R}_+^N$	Vector whose ith element denotes the average additive white noise power on frequency bin i
$\mathbf{x} \in \mathbb{R}_+^N$	Strategy of defender, received power on certain frequency bin i is $\mathbf{x}_i$
$\mathbf{y} \in \mathbb{R}_+^N$	Strategy of attacker, defined similarly as $\mathbf{x}$
$\mathcal{X}$	Feasible set of defender strategies. $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}_+^n \| \sum_i^N \mathbf{x}_i \leq P_D\}$.
$\mathcal{Y}$	Feasible set of attacker strategies. $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}_+^n \| \sum_i^N \mathbf{y}_i \leq P_A\}$.
$\mathbf{x}^*$, $\mathbf{y}^*$	Optimal strategy used by defender and attacker, respectively
$p, p^* \in \mathbb{R}$	Expected payoff and optimal payoff, respectively

defender and the attacker are typically in the line of sight (LOS) to the satellite relay. In addition, this paper considers a decoder and forward type relay satellite, and does not include the typical satellite channel characteristics, e.g., non-linearity in a satellite transponder, a rain loss, etc. This paper focuses on the effects on the data rate of the channel from a transmitter to a relay satellite under a jamming attack environment. The power at the relay satellite received from the defender and the attacker can be simplified as $\mathbf{x}_i = P_{DTi}\left(\frac{\sqrt{G_{DT}G_R}\lambda_i}{4\pi d_{DR}}\right)^2$ and $\mathbf{y}_i = P_{ATi}\left(\frac{\sqrt{G_{AT}G_R}\lambda_i}{4\pi d_{AR}}\right)^2$, respectively, at a certain frequency bin i, where G_{DT} and G_{AT} are the transmit antenna gain of the defender and attacker, respectively; d_{DR} and d_{AR} are the distance from the defender and the attacker to the relay satellite, respectively; G_R is the receiver antenna gain at the relay satellite; and λ_i is the wavelength at the hopping frequency i. Therefore, if $\mathbf{x}_i$ and $\mathbf{y}_i$

are determined, then the corresponding transmit power P_{DTi} and P_{ATi} at frequency bin i can be computed using the other known parameters. Hence, this paper focuses on the computation of $\mathbf{x}_i$ and $\mathbf{y}_i$ using game theory. The satellite jamming game is modeled as a zero-sum game. The objective of the defender is to maximize the information rate, while the objective of the attacker is to minimize this rate. Assuming that the channels have white additive Gaussian noise, if the defender plays strategy $\mathbf{x}$ and the attacker plays strategy $\mathbf{y}$, then the information rate, i.e., the maximum bit rate [9] that can be transmitted by the defender is

$$f(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} B \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right). \tag{4.1}$$

We note that the maximum bit rate, or the channel capacity, is widely used as game payoff in literature. In [2], an OFDM transmitter's payoff is modeled as the sum of the capacity of all sub-channels, taking into account of fading channel gains as well as possible power costs. Here in Eq. 4.1 we suppose a more general representation. We now begin with the scenario where both attacker and defender know each other's strategies and the payoff matrix. This is called the Perfect Information Game.

4.3.1 Perfect Information Game

Similar to the classic zero-sum game discussed above, if the defender knows the attacker's strategy, and vice versa, then the goal for the defender is to find the optimal strategy $\mathbf{x}^*$ that maximizes the payoff, in particular the information rate, which is

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}).$$

Similarly, the goal for the attacker is to find its optimal strategy $\mathbf{y}^*$ that minimizes the information rate which is:

$$\min_{\mathbf{y}\in\mathcal{Y}} \max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x},\mathbf{y}).$$

It is not immediately clear whether this game has a NE as the classic zero-sum game . The NE is obtained when there exists a pair $(\mathbf{x}^*,\mathbf{y}^*)$ such that $\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y}) = \min_{\mathbf{y}\in\mathcal{Y}} \max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x},\mathbf{y})$.

Our first result is that this game does indeed have a NE. Our proof relies on the following theorem from the work of J. Neumann [39].

Theorem 1. *Let $\mathbf{x}\in\mathcal{X}$ and $\mathbf{y}\in\mathcal{Y}$, if $f(\mathbf{x},\mathbf{y})$ is concave in $\mathbf{x}$ for any $\mathbf{y}$, and $f(\mathbf{x},\mathbf{y})$ is convex in $\mathbf{y}$ for any $\mathbf{x}$. Then:*

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x},\mathbf{y}) = \min_{\mathbf{y}\in\mathcal{Y}} \max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x},\mathbf{y}).$$

Definition 1. *$f(\mathbf{x})$ is a convex function if for $0\leq a\leq 1$ and for any $\mathbf{x}$, $\mathbf{y}$, the following applies:*

$$f(a\mathbf{x}+(1-a)\mathbf{y}) \leq af(\mathbf{x})+(1-a)f(\mathbf{y}).$$

Similarly, $f(\mathbf{x})$ is a concave function if

$$f(a\mathbf{x}+(1-a)\mathbf{y}) \geq af(\mathbf{x})+(1-a)f(\mathbf{y}).$$

We are now ready to prove the first result.

Proposition 2. *The spread-spectrum game where the information rate is the payoff has a Nash equilibrium.*

Proof. See Appendix B. □

Proposition 3. *Let P_D and P_A be the total powers of the defender signal and attacker signal received by the satellite, and let x^* be the optimal defender strategy; then the satellite hub's maximum information rate (payoff) is*

$$p^* = \sum_{i \in J} B \log\left(1 + \frac{\mathbf{x}_i^*}{(P_A + P_N^J)/|J|}\right) \\ + \sum_{i \in (K \setminus J)} B \log\left(1 + \frac{\mathbf{x}_i^*}{\mathbf{n}_i}\right), \tag{4.2}$$

in which J and K denote the set of index of bins used by the attacker and defender, respectively. P_N^J denotes the amount of noise power in those bins. $|J|$ denotes the cardinality of J. When the optimal attacker and defender use all the frequency bins ($|J| = N$),

$$p^* = NB \log\left(1 + \frac{P_D}{P_A + P_N}\right).$$

Proof. Consider the attacker's viewpoint. The attacker knows that the defender knows its strategy. Naturally, the defender would try to maximize the information rate based on the given attacker's strategy $\mathbf{y}$. Thus, from the attacker's viewpoint, it will try to minimize the information rate. In other words, the attacker will solve this problem:

$$p^* = \min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{N} B \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right). \tag{4.3}$$

First, consider the max problem from the defender's viewpoint given the attacker's strategy $\mathbf{y}$. The defender will play the optimal strategy $\mathbf{x}$ such that

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{N} \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right).$$

Note that B in the equation above can be omitted since it is a constant, so the optimal solution will not change. With a slight modification, this can be viewed as the well-known problem of capacity maximization of parallel Gaussian channels. Specifically, we now consider $\mathbf{n}_i + \mathbf{y}_i$ as the average power of background noise in bin i. In particular, the optimal $\mathbf{x}^*$ can be found using the Lagrange's multiplier method. To maximize a concave function $f(\mathbf{x})$ subject to a number of constraints $g_i(\mathbf{x}) \leq 0$, $i = 1, 2, \ldots, M$, the Karush-Kuhn-Tucker (KKT) conditions state that the optimal $\mathbf{x}^*$ must satisfy the following:

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} - \lambda_i \frac{\partial g_i(\mathbf{x})}{\partial \mathbf{x}_i}|_{\mathbf{x}=\mathbf{x}^*} = 0, i = 1, 2, \ldots, M. \tag{4.4}$$

Replacing $f(\mathbf{x}) = \sum_{i=1}^{N} \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right)$ for given $\mathbf{n}_i$ and $\mathbf{y}_i$, $g_1(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{x}_i - P_D$ into Eq. 4.4 yields

$$\mathbf{x}_i + \mathbf{n}_i + \mathbf{y}_i = \lambda^{-1}. \tag{4.5}$$

Now, summing up the left- and right-hand sides over i, with the total noise power $P_N = \sum_{i=1}^{N} \mathbf{n}_i$, yields

$$\lambda^{-1} = \frac{P_D + P_A + P_N}{N}. \tag{4.6}$$

From Eqs. (4.5) and (4.6), the optimal strategy $\mathbf{x}^*$ for the defender is

$$\mathbf{x}_i^* = \frac{(P_D + P_A + P_N)}{N} - \mathbf{y}_i - \mathbf{n}_i, i = 1, 2, \ldots, N. \tag{4.7}$$

Next, from the attacker's viewpoint, it will find $\mathbf{y}^*$ that minimizes Eq. 4.3. Substituting Eq. 4.7 into Eq. 4.3 yields the following:

$$\begin{aligned} p &= \sum_{i=1}^{N} B\log\left(1+\frac{\mathbf{x}_i}{\mathbf{n}_i+\mathbf{y}_i}\right) \qquad (4.8)\\ &= \sum_{i=1}^{N} B\log\left(1+\frac{\frac{P_D+P_A+P_N}{N}-\mathbf{n}_i-\mathbf{y}_i}{\mathbf{n}_i+\mathbf{y}_i}\right)\\ &= \sum_{i=1}^{N} B\log\left(\frac{P_D+P_A+P_N}{N}\right)\\ &- \sum_{i=1}^{N} B\log\left(\mathbf{n}_i+\mathbf{y}_i\right), \end{aligned}$$

which is minimized when $\sum_{i=1}^{N}\log\left(\mathbf{n}_i+\mathbf{y}_i\right)$ is maximized. Now, using the Lagrange method with μ as the multiplier yields

$$\mathbf{y}_i=\mu^{-1}-\mathbf{n}_i, i=1,2,\ldots,N. \qquad (4.9)$$

Summing the left- and right-hand sides of Eq. 4.9 yields

$$\mu^{-1}=\frac{P_A+P_N}{N}.$$

Therefore, the optimal strategy of the attacker $\mathbf{y}^*$ is

$$\mathbf{y}_i^*=\frac{P_A+P_N}{N}-\mathbf{n}_i. \qquad (4.10)$$

Since $\mathbf{y}_i^*\geq 0$ is required, if from Eq. 4.10, $\mathbf{y}_i^*<0$, then simply set $\mathbf{y}_i^*=0$, ignore bin i, and re-run the analysis with the remaining $N-1$ bins. Repeat this process to obtain a feasible solution. If J is used to denote the set of bin indexes used by the attacker with cardinality $|J|$, and P_N^J denotes the amount of noise power in those bins, then $\mathbf{y}_i^*$

can be expressed as

$$\mathbf{y}_i^* = \begin{cases} \frac{P_A + P_N^J}{|J|} - \mathbf{n}_i, i \in J \\ 0, otherwise. \end{cases} \tag{4.11}$$

Next, plug $\mathbf{y}^*$ into the payoff expression Eq. 4.7 to find $\mathbf{x}_i^*$. If $\mathbf{x}_i^* < 0$, then set $\mathbf{x}_i^* = 0$ and ignore bin i. If K is used to denote the set of bin indexes used by the defender with cardinality $|K|$, and P_N^K denotes the amount of noise power in those bins, then $\mathbf{x}_i^*$ can be expressed as

$$\mathbf{x}_i^* = \begin{cases} \frac{P_D + P_A + P_N^K}{|K|} - \mathbf{y}_i^* - \mathbf{n}_i, i \in K \\ 0, otherwise. \end{cases} \tag{4.12}$$

Now, notice that $|J| \leq |K| \leq N$(see Remark 4). Then the following is obtained:

$$\begin{aligned} p^* &= \sum_{i \in N} B \log\left(1 + \frac{\mathbf{x}_i^*}{\mathbf{y}_i^* + \mathbf{n}_i}\right) \\ &= \sum_{i \in K} B \log\left(1 + \frac{\mathbf{x}_i^*}{\mathbf{y}_i^* + \mathbf{n}_i}\right) \\ &= \sum_{i \in J} B \log\left(1 + \frac{\mathbf{x}_i^*}{(P_A + P_N^J)/|J|}\right) + \\ &\quad \sum_{i \in (K \setminus J)} B \log\left(1 + \frac{\mathbf{x}_i^*}{\mathbf{n}_i}\right). \end{aligned} \tag{4.13}$$

When $|J| = |K| = N$,

$$p^* = \sum_{i \in N} B \log \left(1 + \frac{\mathbf{x}_i^*}{\mathbf{y}_i^* + \mathbf{n}_i}\right) = NB \log \left(1 + \frac{P_D}{P_A + P_N}\right). \tag{4.14}$$

Now, by Proposition 2, $\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \mathbf{y})$; therefore, the payoff of the defender is $q^* = p^*$.

□

Remark 4. From Eq. 4.9, the optimal strategy for the attacker is essentially to try to fill every bin so that they have equal power. When this is not possible for some bins, it omits those bins and tries to make the power levels of the remaining bins equal. This strategy follows our intuition since any attacker's strategy that deviates from uniform distribution on the power levels, by symmetry, would allow the defender to take advantage of it. Also, note that in a low SNR scenarios where every frequency bin has low noise power compared to the total power of the receiver, then $|J| = N$ or the attacker will spread its power over all the frequency bins. Finally, the bins that will be used by the attackers will be those with the lowest noise power levels.

Fig. 4.2(a) illustrates two cases: (1) every frequency bin is used, and (2) some frequency bins are not used in the attack. In both cases, it is noted that the jammer tries to spread the power over the bins as evenly as possible. In turn, the defender also tries to spread the power evenly over every bin. These are optimal strategies for both cases.

Remark 5. In the real world, it is true that the jammer usually has limited information about the channel. However, one should not take a myopic view that all information is secure. Communication parameters can be leaked through other means (e.g., espionage). Furthermore, many educated guesses can be made about the satellite hardware and algorithms since most of this information is public. The point is that a sophisticated attacker, e.g., nation with large resources can potentially acquire this information. Thus, there is a need to analyze the perfect information scenario, i.e., the worst case scenario for the defender. Importantly, the existence of Nash equilibrium guarantees that the payoff for the attacker under this perfect information scenario is the best it can ever hope for. Thus, the defender can quantify the degree of damage for given communication parameters.

4.3.2 Defender-Biased Game

In this game, the attacker does not know the strategy of the defender. On the other hand, the defender knows the attacker's strategy and it knows that the attacker does not know its strategy. Being rational, the defender does not have to play the strategy $\mathbf{x}^* = \text{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in]\mathcal{Y}} f(\mathbf{x}, \mathbf{y})$, since the strategy $\mathbf{x}^*$ is optimized for the worst case. Indeed, by knowing the attacker's strategy $\mathbf{y}$, the defender can achieve a higher payoff by playing the strategy as follows:

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{N} B \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right).$$

As previously derived in Eq. 4.7,

$$\mathbf{x}_i^* = \frac{(P_D + P_A + P_N)}{N} - \mathbf{y}_i - \mathbf{n}_i, i = 1, 2, \ldots, N. \tag{4.15}$$

If $\mathbf{x}_i^* < 0$ for some i, then set $\mathbf{x}_i^* = 0$, ignore the frequency bin i, and re-run the

Lagrange multiplier method for the remaining $N-1$ frequency bins.

Now, consider the scenario when the attacker has no knowledge of the background noise and the defender's strategy. In this scenario, we propose the following:

Proposition 4. *If the attacker has no knowledge of the background noise and the defender's strategy, then the defender's optimal payoff is*

$$q_1^* = \sum_{i=1}^{|K|} B \log \left(1 + \frac{\frac{P_D + P_N^K}{|K|} - \mathbf{n}_i}{\mathbf{n}_i + \frac{P_A}{N}} \right),$$

where K is the set of bin indexes used in the optimal strategy $\mathbf{x}^*$, $|K|$ is the cardinality of K, and P_N^K is the total noise power in $|K|$ bins used by the defender.

Proof. Without any information regarding the SNRs of the frequency bins or the defender's strategy, by the principle of insufficient reasons, the attacker would spread its power equally among N frequency bins by playing the strategy $\mathbf{y}_i = P_A/N$. Consequently, from Eq. 4.15, the defender will play the strategy that maximizes the payoff given $\mathbf{y}_i = P_A/N$ as

$$\begin{aligned} \mathbf{x}_i^* &= \frac{(P_D + P_A + P_N)}{N} - \frac{P_A}{N} \\ &- \mathbf{n}_i, i = 1, 2, \ldots, N. \\ &= \frac{P_D + P_N}{N} - \mathbf{n}_i, \end{aligned} \tag{4.16}$$

assuming that $\frac{P_D+P_N}{N} - \mathbf{n}_i > 0$. If for some frequency bin i, the positive power constraint is not satisfied, then the defender will ignore frequency bin i and re-run the optimization for the other remaining bins. Plugging in $\mathbf{x}^*$ and $\mathbf{y}_i = P_A/N$ into the payoff function, the optimal payoff of the defender can be obtained as

$$
\begin{aligned}
q_1^* &= \sum_{i=1}^{|K|} B \log\left(1 + \frac{\mathbf{x}_i^*}{\mathbf{n}_i + \frac{P_A}{N}}\right) \qquad (4.17)\\
&= \sum_{i=1}^{|K|} B \log\left(1 + \frac{\frac{P_D + P_N^K}{|K|} - \mathbf{n}_i}{\mathbf{n}_i + \frac{P_A}{N}}\right).
\end{aligned}
$$

. □

Fig. 4.2(b) illustrates the power allocation of the jammer and the defender's strategies. In this scenario, the jammer simply allocates power uniformly at random over all frequency bins. On the other hand, the defender will try to equalize the power across all frequency bins as much as possible, given its power budget.

If the jammer does not know as much information as the defender, then it will not be able to play the Nash equilibrium strategy correctly. Thus, the defender will be able to take advantage of this and improve its own strategy to get a higher payoff. For instance, if the attacker does not have the exact information of noise distribution of the frequency bins, it randomly chooses some bins to allocate with more power, and other bins with less. As a result, the defender will be able to allocate more power to those less-corrupted channels and actually have a higher payoff. It turns out that, in the Defender-biased scenario, it is actually reasonable for the attacker to split its power budget evenly. Both our theoretical and simulation results show that a less-uniform noise channel (i.e, channel whose distribution noise power on the frequency is far from uniform distribution in terms of Kullback-Leibler (KL) distance will be more advantageous to the defender. That is, if the attacker's action makes the noise distribution over the channel less uniform, then the defender gains by putting more of its power in the less-corrupted frequency bins.

As a result, the defender gets a better payoff. Intuitively, without any knowledge of the noise distribution, any non-uniform distribution of the attacker's power will be unwise, because it is very likely to bring more variance to the existing noise. A uniform power distribution, however, at least does not increase the difference of each channel. These cases are illustrated in Section 4.5.

The game with Perfect Information scenario and the game with Defender Biased scenario are similar to the case of channel state information (CSI) being available at both transmitter (TX) and receiver (RX) and the case of CSI being available at only the RX in a multiple-input and multiple-output (MIMO) system. In this game, both the defender and the attacker can apply the water-filling strategy simultaneously when perfect information is available, whereas in the MIMO system, only the TX can apply the water-filling assuming the known ocean bottom level (i.e., the attacker plus noise level $\mathbf{y}_i + \mathbf{n}_i$ known to the MIMO TX). In the Defender-Biased game, only the defender can apply the water-filling strategy, whereas the attacker uses the equal power strategy. This is similar to the MIMO without CSI at TX, where the TX uses the equal power strategy because it has no information on the ocean bottom level.

4.3.3 Attacker-Biased Game

We now consider the Attacker-biased game. Here the attacker knows the defender's strategy, and it knows that the defender does not know its strategy. Similar to Section 4.3.2, suppose the defender uses a strategy $\mathbf{x}$ that is known to the attacker; then a rational attacker will try to minimize the payoff, i.e., information rate using

$$\mathbf{y} = \mathrm{argmin}_{\mathbf{y}\in\mathcal{Y}} \sum_{i=1}^{N} B \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right).$$

Since B is a constant, it is equivalent to minimizing the function

$$f(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{N} \log\left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right). \tag{4.18}$$

Using the Lagrange multiplier method, similar to Section 4.3.1 yields

$$\frac{\mathbf{x}_i}{(\mathbf{n}_i + \mathbf{y}_i + \mathbf{x}_i)(\mathbf{n}_i + \mathbf{y}_i)} = \lambda. \tag{4.19}$$

Letting $\mathbf{z}_i = \mathbf{n}_i + \mathbf{y}_i$ and solving for $\mathbf{z}_i$ yields

$$\begin{aligned} \mathbf{z}_i &= \mathbf{n}_i + \mathbf{y}_i \qquad (4.20) \\ &= \frac{-\lambda\mathbf{x}_i + \sqrt{\lambda^2\mathbf{x}_i^2 + 4\mathbf{x}_i\lambda}}{2\lambda}. \end{aligned}$$

Equivalently, the optimal strategy for the attacker is

$$\mathbf{y}_i' = \frac{-\lambda\mathbf{x}_i + \sqrt{\lambda^2\mathbf{x}_i^2 + 4\mathbf{x}_i\lambda}}{2\lambda} - \mathbf{n}_i. \tag{4.21}$$

However, we do not know λ. The following procedure is used to search for λ using an upper and a lower bound computed as follows. First, we note that by summing the left-

and right-hand sides of Eq. 4.19,

$$\begin{aligned}\lambda &= \frac{\sum_i \mathbf{x}_i}{\sum_i\left((\mathbf{n}_i+\mathbf{y}_i+\mathbf{x}_i)(\mathbf{n}_i+\mathbf{y}_i)\right)} && (4.22)\\ &\geq \frac{P_D}{\sum_i(\mathbf{n}_i+\mathbf{y}_i+\mathbf{x}_i)\sum_i(\mathbf{n}_i+\mathbf{y}_i)} && (4.23)\\ &= \frac{P_D}{(P_A+P_D+P_N)(P_N+P_A)}, && (4.24)\end{aligned}$$

where Eq. 4.23 is due to Schwartz's inequality. Now an upper bound for λ can be found as follows:

$$\begin{aligned}\lambda &= \frac{\sum_i \mathbf{x}_i}{\sum_i\left((\mathbf{x}_i+\mathbf{n}_i+\mathbf{y}_i)(\mathbf{n}_i+\mathbf{y}_i)\right)} && (4.25)\\ &\leq \frac{P_D}{\sum_i(\mathbf{n}_i+\mathbf{y}_i)^2} && (4.26)\\ &\leq \frac{P_D}{\sum_i \mathbf{n}_i^2+\sum_i \mathbf{y}_i^2} && (4.27)\\ &\leq \frac{P_D}{\frac{(\sum_i \mathbf{n}_i)^2}{N}+\frac{(\sum_i \mathbf{y}_i)^2}{N}} && (4.28)\\ &= \frac{NP_D}{P_A^2+P_N^2}, && (4.29)\end{aligned}$$

where N is the number of the frequency bin used for jamming. We note that Eq. 4.28 is due to the well-known bound for l_1-norm and l_2-norm.

Next, an algorithm that performs the search for λ over these bounds is proposed. For each value of λ, $\mathbf{y}'$ is computed using inequality (4.21); then $\mathbf{y}'$ is checked to see if it satisfies all power constraints.

Proposition 5. *Let $f(\lambda)=\sum_{i=1}^{N}\mathbf{y}_i$. Then $f(\lambda)$ is monotonically decreasing in λ within the interval specified by inequality (4.29) and inequality (4.24).*

Proof. See Appendix C. □

Based on Proposition 5, a binary search algorithm with the complexity of $\log(n)$ can be used to find λ efficiently, where n is the number of partition in the search space. In our specific scenario, since we are searching for the right value of λ, if we want the value of λ to be within ϵ of the optimal value, then we can set $n = O(1/\epsilon)$. The algorithm is show in Algorithm 4:

Algorithm 4 BINARY SEARCH FOR λ

```
while $|P_A - f(\lambda)| > \epsilon$ do
    if $P_A - f(\lambda) > 0$ then
        $\lambda_{upper} = \lambda$
    else
        $\lambda_{lower} = \lambda$
    end if
    $\lambda = (\lambda_{lower} + \lambda_{upper})/2$
end while
```

Then, the best strategy for the attacker $\mathbf{y}_i'$ can be found by Eq. 4.21.

4.4 Extension to Continuous Spread-Spectrum Jamming

In this section, we extend the satellite jamming attack settings from a setting consisting of finite discrete frequency bins to the setting where signals are spread in a continuous spectrum, i.e., an uncountable infinite number of frequency bins. In particular, we will focus on the problem from the defender's perspective in a Defender-biased game.

4.4.1 Problem Formulation

To provide a brief background, we first consider the discrete time i.i.d. white Gaussian channel modeled as:

$$\mathbf{r}_i = \mathbf{s}_i + \mathbf{w}_i,$$

where $\mathbf{w}_i \sim N(0, \sigma_N^2)$ denotes the noise with average power σ_N^2, $\mathbf{s}_i$ denotes the transmitted signal, and $\mathbf{r}_i$ denotes the received signal. Furthermore, we assume that

$$\sum_{i=1}^{N} \mathbf{s}_i^2 \leq NP,$$

where P denotes the average power of a transmitted signal. It is well-known that the capacity for this channel is

$$C = \frac{1}{2} \log \left(1 + \frac{P}{\sigma_N^2} \right).$$

Furthermore, the capacity is achieved when

$$\mathbf{s}_i \sim N(0, P). \tag{4.30}$$

Consequently, we also have

$$\mathbf{r}_i \sim N(0, P + \sigma_N^2). \tag{4.31}$$

Now, we turn our attention to modeling the game. From the defender's point of view, the sum of the attacker signal and the background noise signal can be treated as one single noise signal with the total power as

$$\sigma_N^2 = P_A + P_N.$$

The strategies of the defender and the attacker are no longer sets of discrete power levels on each frequency bin. Rather, their strategies are to find the power spectrum of their signal so that the payoff functions are maximized. Specifically, in a Defender-biased game, if we denote the power spectral density of the defender and the total noise as $S(\omega)$ and $Z(\omega)$, respectively, the defender intends to find $S(\omega)$ to maximize the following payoff:

$$C = \frac{1}{2}\int_{-B/2}^{B/2} \log\left(1 + \frac{S(\omega)}{Z(\omega)}\right) d\omega. \tag{4.32}$$

4.4.2 Defender-Biased Game with Continues Spread-Spectrum

Proposition 6. *To maximize the payoff described in Eq. 4.32, the power spectral density of the defender $S(\omega)$ is*

$$S(\omega) = \max(P + \sigma_N^2 - Z(\omega), 0),$$

where $Z(\omega)$ are the power spectral density of the total noise. Thus, the optimal payoff is

$$C = \frac{1}{2}\int_{-B/2}^{B/2} \log\left(1 + \frac{\max(P + \sigma_N^2 - Z(\omega), 0)}{Z(\omega)}\right) d\omega.$$

Proof. Consider in the discrete case, the covariance matrices of the transmitted signal $\mathbf{s}_i$ and the noise $\mathbf{z}_i$, which is sum of the attacker's signal and noise signal. We will show that these covariance matrices are directly related to the spectrum of the transmitted and noise signals when $N \longrightarrow \infty$. Let $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$ be the vector denoting the transmitted signal, $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ be the vector denoting the sum of noise and the attacker signal, and $\mathbf{r} = \{r_1, r_2, \ldots, r_N\}$ be the vector denoting the received signal.

Then we have the following channel:

$$\mathbf{r} = \mathbf{s} + \mathbf{z}, \tag{4.33}$$

where $\mathbf{z}_i$ may not be independent.

Given the covariance matrix $\mathbf{K}_{zz}$ of the noise signal, the defender will try to find the covariance matrix $\mathbf{K}_{ss}$ of the transmitted signal such that it maximizes the capacity. We proceed as follows. We have

$$\begin{aligned}\mathbf{K}_{zz} &= E[\mathbf{z}\mathbf{z}^T] \\ \mathbf{K}_{ss} &= E[\mathbf{s}\mathbf{s}^T].\end{aligned}$$

Since $\mathbf{K}_{zz}$ is a symmetric matrix, performing eigenvalue decomposition yields

$$\mathbf{K}_{zz} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T,$$

where $\mathbf{D}$ is a diagonal matrix, whose diagonal entries are non-zero eigenvalues, and $\mathbf{Q}$ is the matrix whose columns are eigenvectors; thus, $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. Next, multiplying Eq. 4.33 by $\mathbf{Q}^T$ yields

$$\mathbf{Q}^T\mathbf{r} = \mathbf{Q}^T\mathbf{s} + \mathbf{Q}^T\mathbf{z}. \tag{4.34}$$

Let $\mathbf{w} = \mathbf{Q}^T\mathbf{z}$, then

$$\begin{aligned}\mathbf{K}_{ww} &= E[\mathbf{w}\mathbf{w}^T] \\ &= E[\mathbf{Q}^T\mathbf{z}\mathbf{z}^T\mathbf{Q}] \\ &= \mathbf{Q}^T\mathbf{K}_{zz}\mathbf{Q} \\ &= \mathbf{D}.\end{aligned}$$

Therefore, $\mathbf{w}_i$ are independent. We also note that the power constraint of the signal $\mathbf{Q}^T\mathbf{s}$ and $\mathbf{s}$ are the same since

$$\begin{aligned}tr(E[\mathbf{Q}^T\mathbf{s}\mathbf{s}^T\mathbf{Q}]) &= tr(\mathbf{Q}^T\mathbf{K}_{ss}\mathbf{Q}) \\ &= tr(\mathbf{K}_{ss}\mathbf{Q}\mathbf{Q}^T) \\ &= tr(\mathbf{K}_{ss}) \\ &= NP.\end{aligned} \tag{4.35}$$

Since $\mathbf{w}_i$ are independent, based on the well-known capacity of additive white noise channel (Eqs. (4.30) and (4.31)), each component of $\mathbf{v} = Q^T\mathbf{s}$ must have independent Gaussian distribution. Similarly, each component of the corresponding $\mathbf{u} = Q^T\mathbf{r}$ must also have independent Gaussian distribution. Using this condition, multiplying Eq. 4.34 by $\mathbf{u}^T$, and taking the expectation on both sides yields

$$\begin{aligned}E[\mathbf{u}\mathbf{u}^T] &= E[(\mathbf{Q}^T\mathbf{s} + \mathbf{w})(\mathbf{Q}^T\mathbf{s} + \mathbf{w})^T] \\ &= \mathbf{Q}\mathbf{K}_{ss}\mathbf{Q}^T + \mathbf{K}_{ww} \\ &= \mathbf{Q}\mathbf{K}_{ss}\mathbf{Q}^T + \mathbf{D}.\end{aligned} \tag{4.36}$$

Thus,

$$\mathbf{K}_{ss} = \mathbf{Q}^T(\mathbf{K}_{uu} - \mathbf{D})\mathbf{Q}. \tag{4.37}$$

Since $\mathbf{u}_i$ are independent, or $\mathbf{K}_{uu}$ is a diagonal matrix, choosing

$$\mathbf{K}_{uu} = \left(P + \frac{tr(\mathbf{D})}{N}\right)\mathbf{I},$$

yields $tr(\mathbf{K}_{ss}) = NP$, which satisfies the power constraint. Therefore, the optimal transmitted signal $\mathbf{s}$ for the defender should have its covariance matrix as

$$\mathbf{K}_{ss} = \mathbf{Q}^T\left(\left(P + \frac{tr(\mathbf{D})}{N}\right)\mathbf{I} - \mathbf{D}\right)\mathbf{Q}.$$

Here we assume that the defender knows the covariance matrix of the sum of background noise and the attacker, and thus can compute the optimal $\mathbf{K}_{ss}$.

Now, we turn our attention to the continuous spectrum. If $\mathbf{z}$ is wide-sense stationary, then for $N \longrightarrow \infty$, the diagonal entries of $\mathbf{D}$ are indeed the power spectrum of $\mathbf{z}$. Thus, using the water-filling argument discussed in the previous section, the optimal spectrum of $\mathbf{s}$ would be

$$S(\omega) = \max(P + W(\omega) - Z(\omega), 0),$$

where $S(\omega)$, $W(\omega)$, and $Z(\omega)$ are the power spectral densities of $s(t)$, $w(t)$, and $z(t)$, respectively. Note that $W(\omega) = \sigma_N^2$. The corresponding optimal payoff (transmission rate) is

$$C = \frac{1}{2}\int_{-B/2}^{B/2} \log\left(1 + \frac{\max(P + W(\omega) - Z(\omega), 0)}{Z(\omega)}\right) d\omega.$$

□

4.5 Simulation results

In this simulation, ten bins with an identical bandwidth of 3 kbps are used. Assume that the defender can deliver -120 dBW at the satellite relay, and the jamming attacker can deliver half of the power, i.e., -123 dbW. The total noise power is described by $SNR = P_D/P_N$.

In a real-world scenario, the attacker can use any strategy it wants. However, the attacker will inevitably do less damage to the defender with any strategy that deviates from its Nash equilibrium strategy, one that assumes both defender and attacker have perfect information about the game. To illustrate this point, Fig. 4.3 shows what happens if one player decides to change to some other strategy. In this case, SNR is fixed at $10dB$, and noise distribution is quantified by the *concentration index (CI)*. One bin is randomly picked, and the CI indicates the average percentage of P_N that is confined in this specific bin. In this case, when $N = 10$, $CI = 0.1$ indicates a "flat" distribution, while a higher CI indicates a more highly-concentrated one. If the defender decides to move to some other strategy while the attacker plays its NE strategy, the defender reduces its payoff as shown by the red curve. Similarly, if the attacker changes its strategy while the defender stays with its NE strategy, the payoff to the defender becomes higher as shown by the green curve, i.e., the attacker reduces its payoff since this is a zero-sum game. As a result, at the NE, both players have no motivation to move to other strategies, and the payoff is shown by the blue curve.

Fig. 4.4 shows the payoff comparison for all scenarios. Five different scenarios are considered here:

- Perfect information Nash equilibrium (NE)
- Defender-biased scenario with the attacker uniformly distributing its power (DB-

U): Attacker distributes its power equally to each frequency bins.

- Defender-biased scenario with the attacker randomly distributing its power (DB-R): Attacker distributes its power for each bin following the uniform distribution in $[0.25\frac{P_A}{N}, 1.75\frac{P_A}{N}]$.

- Attacker-biased scenario (AB): Defender does not know the existence of the attacker. As a result, the defender allocates its power by maximizing its capacity only according to the noise distribution.

- Random scenario (R): Defender's power allocation follows uniform distribution $[0.25\frac{P_N}{N}, 1.75\frac{P_N}{N}]$, and the attacker's power allocation follows uniform distribution $[0.25\frac{P_A}{N}, 1.75\frac{P_A}{N}]$

Comparing the DB-U and the NE scenario, the noise powers in the channels become less uniform as the CI increases, and the defender will have more advantage since it can adjust its power allocation accordingly. On the other hand, the attacker wants to make the channel as even as possible in order to "cancel out" the advantage of the defender. As expected, Fig. 4.4 shows that in these two scenarios, the defender can obtain a higher rate when the noise powers in the channels become less uniform. Furthermore, the information rate in the Defender-biased scenario is always higher than that of the Perfect Information scenario, as expected. The curve is flat when $CI < 0.6$, indicating the filling effect introduced by the attacker. When the channel noise is more uniform across the frequency bins, the attacker is able to fill the gap between channels with its limited power budget. Thus, the defender's gain stays constant by the attacker's filling. However, if the attacker has no idea about how the channel noise is distributed, an evenly distributed jamming power will hardly decrease the extent of variation in power for each channel. As a result, in the DB-U case, the payoff keeps increasing as the CI increases.

Comparing the AB and the NE scenarios, it is assumed that the defender always knows the channel conditions and distributes its power accordingly, but the defender does not know the existence of the attacker. As the CI increases, the attacker becomes more effective because of the increase in variation of power across the frequency bins. The attacker can distribute power according to the defender's action in order to achieve optimality. However the attacker's gain stops increasing at some value of CI. This is because from this point on, both attacker and defender discard the most noisy channel, and there will be no further change of variation.

Comparing the DB-U and DB-R scenarios, it is obvious that without the channel condition, if the attacker decides to use a randomly distributed power allocation, then the defender will be able to gain a large advantage. Even in the R scenario, where the defender uses a random strategy instead of an optimized strategy, the attacker may still get a payoff that is worse than the DB-U case. As we discussed in Section 4.3.2, this result is not surprising. When the channel condition is not available, a random strategy of the attacker will be very likely to make the noise distribution in the frequency bins less uniform. As a result, the defender is able to take advantage of this and applies more power on those less noisy bins. If the attacker allocates its power evenly, then it at least will not increase the power differences in each bin.

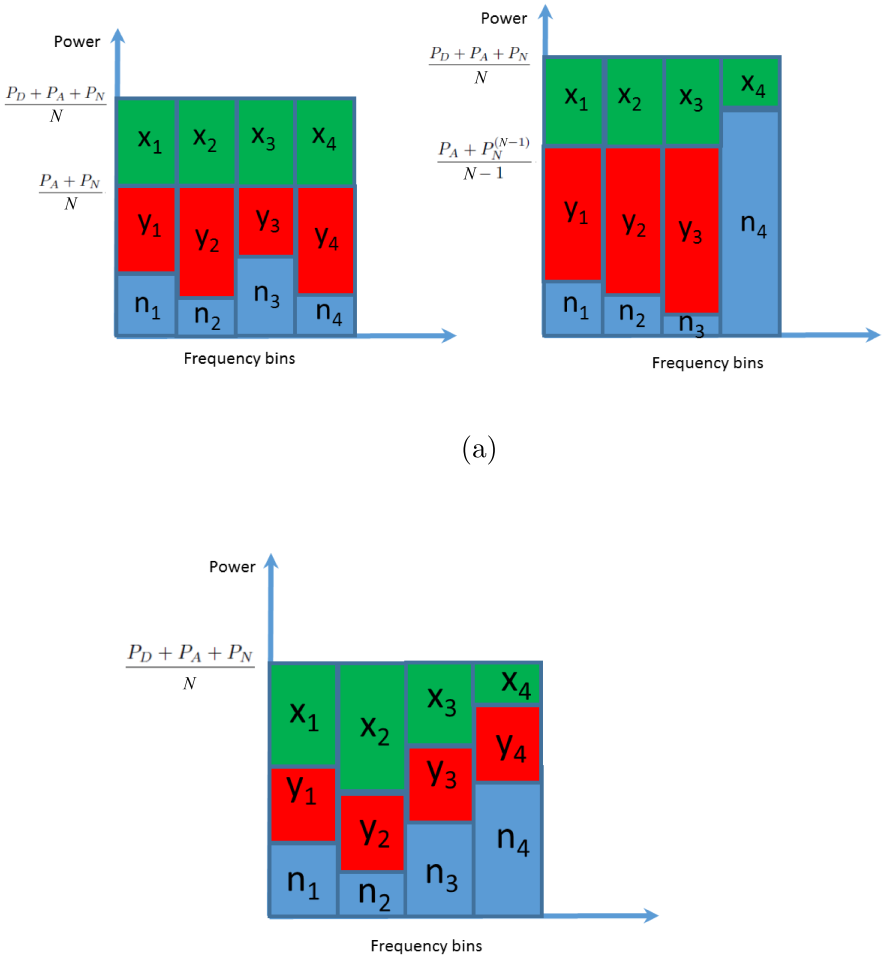
Fig. 4.5 further illustrates the attacker's ability to flatten the noise distribution among the bins, thus eliminating the advantage of variation for defenders. In this case, P_A and P_D are fixed, while the noise power increases from $SNR = 12.5dB$ to $5dB$. In both the perfect information and Defender-biased scenarios, the overall performance decreases as the SNR decreases. In Fig. 4.5(a), when the noise power is sufficiently small ($12.5dB$), the attacker is able to fill the gaps, regardless of the channel condition variation. In this case, the defender has no benefit, even when CI is large. As the noise power increases, the attacker will no longer be able to cover the variation at some points, and the benefit of the defender happens earlier when CI increases. However, as shown in Fig. 4.5(b), in the Defender-biased scenario, the attacker never flatten the channel. As a result, benefits for the defender always exists.

Fig. 4.6 compares two different kinds of defender behavior in the Attacker-biased case. A "smart defender" distributes its power among the bins according to channel conditions, while a "non-smart defender" simply distributes power evenly. Notice that the AB scenario defined above is actually a smart defender. The result is plotted when $CI = 0.2$. It is interesting to see that in this specific scenario, a "non-smart defender" always performs better! When the "smart defender" distributes the power according to channel condition, the result is more varied. As a result, the attacker's advantage becomes even larger. In fact, Eq. 4.21 shows that for a "non-smart defender", the attacker can do nothing more than "flatten" the variation of the channel noise, which is the same as in the perfect information case.

Fig. 4.7 shows the defender payoff when the noise distribution is fixed among ten bins. In this case, 75% of the noise power is concentrated in three bins. As expected, in all scenarios, performance increases as the SNR increases. At any SNR, the two Defender-biased scenarios (DB-R and DB-U) always have better performance compared to the other two. It is obvious that the DB-R scenario will provide more advantages to the defender because of the extra variance due to the randomness of the attacker. The NE scenario comes third, and the AB scenario is the worst. Differences among the DB-U, NE, and AB scenarios decrease when the SNR increases. Given a fixed distribution of noise across the bins, the absolute difference between a bad channel and a good channel is small when the SNR is large. As a result, the defender/attacker does not have a huge benefit when given the information advantage, and their power is distributed almost evenly at the end.

4.6 Summary

In this chapter, the FH satellite jamming attack is modeled as a zero-sum game. The spread-spectrum attack is introduced, and the existence of NE is shown. Furthermore, analytical results on the perfect information game, Defender-biased game, and Attacker-biased game are provided. Both theoretical analysis and intuitions agree with the simulated performance results of each scenario under different channel conditions.



(a)

(b)

Figure 4.2: Optimal power allocations of defender and attacker: (a) Perfect Information; (b) Defender-Biased.

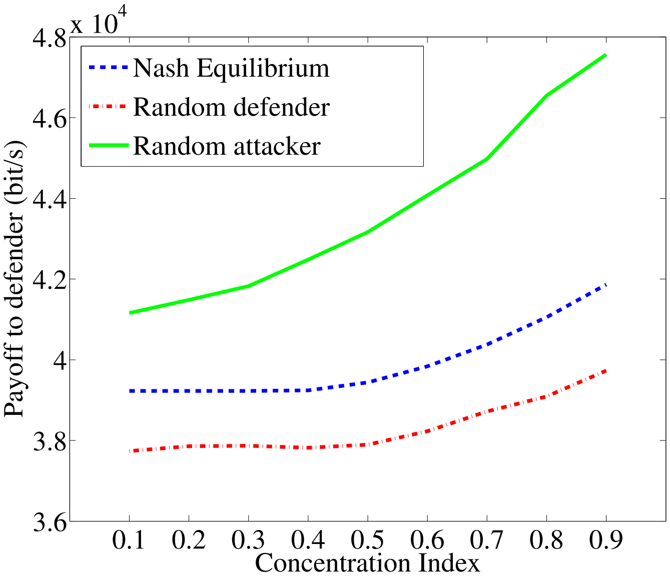


Figure 4.3: Illustration of Nash equilibrium

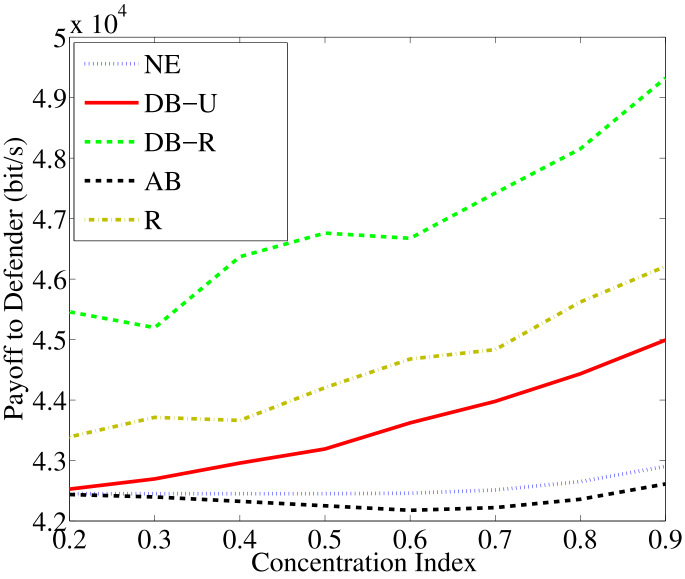
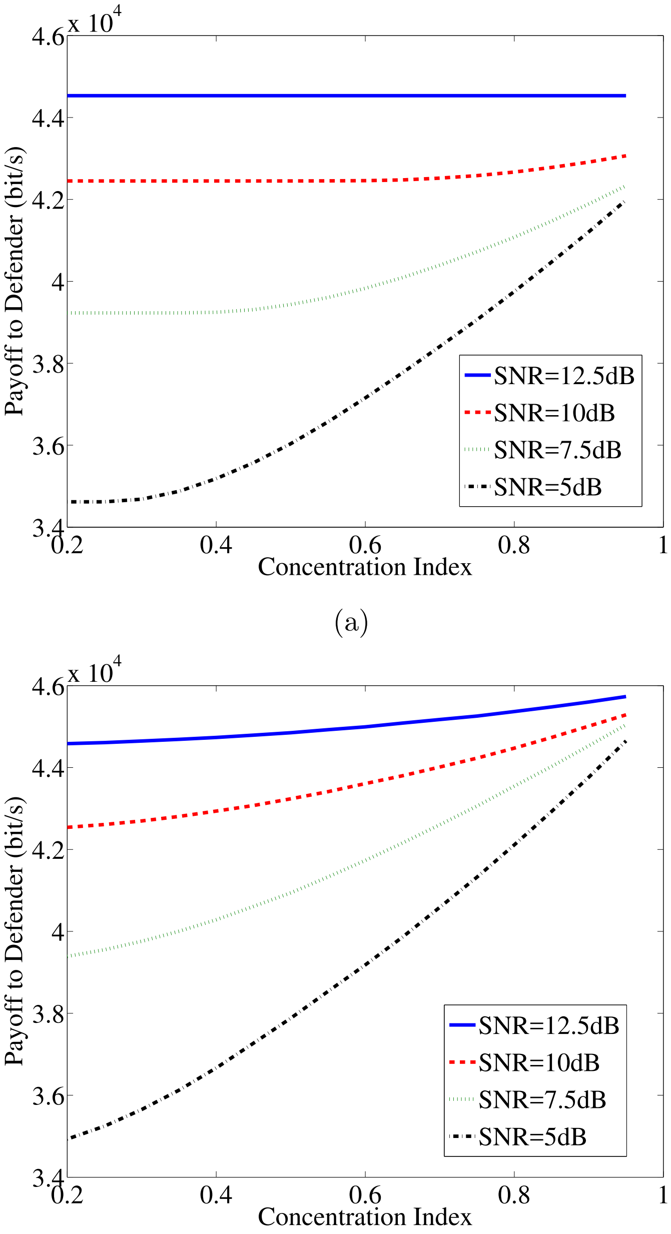


Figure 4.4: Payoff (rate) for scenarios under different noise distributions



(a)

(b)

Figure 4.5: (a) Defender payoff (rate) in perfect information scenario; (b) in Defender-biased scenario.

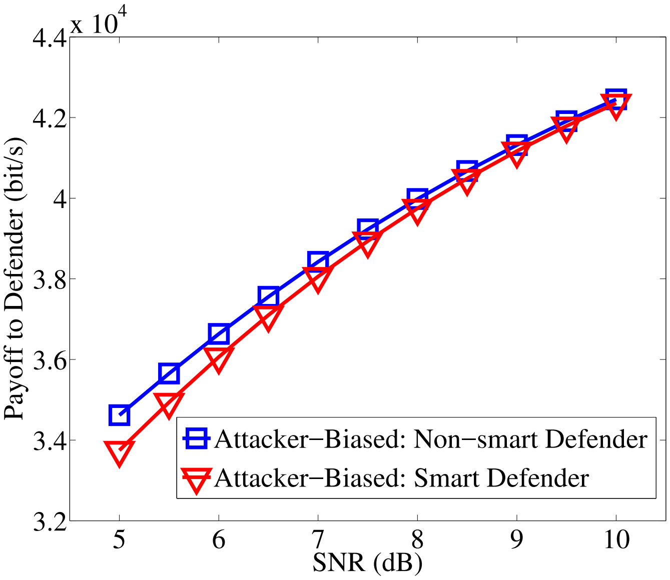


Figure 4.6: Performance comparison of different types of defenders.

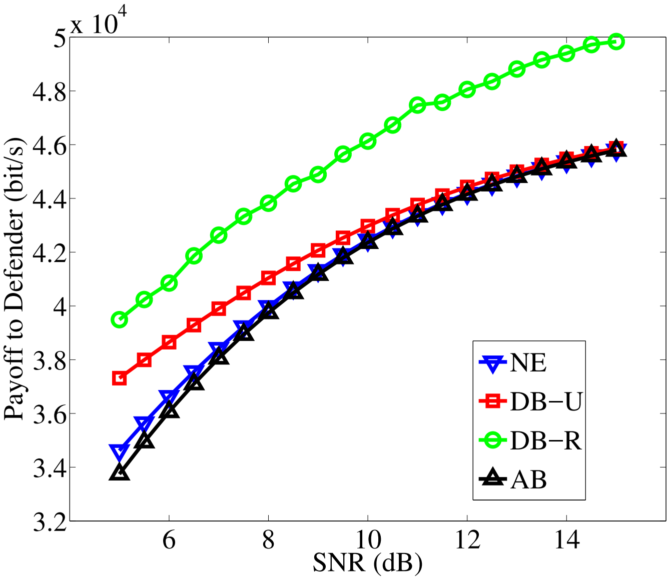


Figure 4.7: Comparison of all scenarios for different SNRs.

Chapter 5: Conclusion

A transmitter of a modern communication system often has access to more than one mediums. In this thesis, we generalize the diversity available to the transmitters as abstract channels, and present a smart scheduling framework that can be applied to multiple application scenarios. Specifically, we present the in-depth study of three application problems, designed and analyzed the optimal scheduling strategy for both indoor packet delivery system and SATCOM communication system. Taking advantage of the channel diversities, out presented algorithms are shown to have the ability to satisfy multiple users' QoS requirements. First, for indoor packet delivery system, we show the DQ-based packet scheduler progressively learns a good policy in real-time, based directly on the available observations. Second, in the context of SATCOM, we proposed scheduler is a low complexity randomized algorithm that multiplexes user data over multiple frequencies and time slots to combat eavesdropping and environmental fading. Last, to mitigate the jamming attack in a FH satellite communication system, we analyze the behaviours of both attacker and defender in a game theory perspective.

Bibliography

[1] M. J. Abdel-Rahman and M. Krunz. Game-theoretic quorum-based frequency hopping for anti-jamming rendezvous in dsa networks. In *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pages 248–258, April 2014.

[2] Eitan Altman, Konstantin Avrachenkov, and Andrey Garnaev. A jamming game in wireless networks with transmission cost. In *Proceedings of the 1st EuroFGI International Conference on Network Control and Optimization*, NET-COOP'07, pages 1–12, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Phil Whiting, and Rajiv Vijayakumar. Providing quality of service over a shared wireless link. *IEEE Communications magazine*, 39(2):150–154, 2001.

[4] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *CoRR*, abs/1611.02167, 2017.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[6] Peter Brucker and P Brucker. *Scheduling algorithms*, volume 3. Springer, 2007.

[7] P. T. Capozza, B. J. Holland, T. M. Hopkinson, and R. L. Landrau. A single-chip narrow-band frequency-domain excisor for a global positioning system (gps) receiver. *IEEE Journal of Solid-State Circuits*, 35(3):401–411, March 2000.

[8] J. P. Choi and V. W. S. Chan. Adaptive communications over fading satellite channels. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, volume 9, pages 2635–2639 vol.9, June 2001.

[9] Thomas Cover and Joy Thomas. *Elements of information theory*. Wiley-Interscience, 2 edition, 7 2006.

[10] Thomas M. Cover and Joy A. Thomas. *The Gaussian Channel*, pages 239–265. John Wiley and Sons, Inc., 2001.

[11] H. Dai, H. Wang, H. Xiao, X. Li, and Q. Wang. On eavesdropping attacks in wireless networks. In *2016 IEEE Intl Conference on Computational Science and Engineering*

(CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), pages 138–141, Aug 2016.

[12] W. Dai, C. Qiao, Y. Wang, and C. Zhou. Improved anti-jamming scheme for direct-sequence spread-spectrum receivers. *Electronics Letters*, 52(2):161–163, 2016.

[13] S. D'Oro, L. Galluccio, G. Morabito, S. Palazzo, L. Chen, and F. Martignon. Defeating jamming with the power of silence: A game-theoretic analysis. *IEEE Transactions on Wireless Communications*, 14(5):2337–2352, May 2015.

[14] Erik Ekstedt. A deep reinforcement learning framework where agents learn a basic form of social movement. 2018.

[15] K. Gai, M. Qiu, Z. Ming, H. Zhao, and L. Qiu. Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks. *IEEE Transactions on Smart Grid*, 8(5):2431–2439, Sept 2017.

[16] Keke Gai, Longfei Qiu, Min Chen, Hui Zhao, and Meikang Qiu. Sa-east: Security-aware efficient data transmission for its in mobile heterogeneous cloud computing. *ACM Trans. Embed. Comput. Syst.*, 16(2):60:1–60:22, January 2017.

[17] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz. Game theoretic anti-jamming dynamic frequency hopping and rate adaptation in wireless systems. In *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 247–254, May 2014.

[18] M. K. Hanawal, D. N. Nguyen, and M. Krunz. Jamming attack on in-band full-duplex communications: Detection and countermeasures. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[19] M. Hannon, S. Feng, H. Kwon, and K. Pham. Jamming statistics-dependent frequency hopping. In *IEEE Military Communications Conf.*, Nov 2016.

[20] Y. Hong, T. Wu, and L. Chen. On the performance of adaptive mimo-ofdm indoor visible light communications. *IEEE Photonics Technology Letters*, 28(8):907–910, April 2016.

[21] Ashkan Kalantari, Gan Zheng, Zhen Gao, Zhu Han, and Björn Ottersten. Secrecy analysis on network coding in bidirectional multibeam satellite communications. *IEEE Transactions on Information Forensics and Security*, 10(9):1862–1874, 2015.

[22] S. Khodayari and M. J. Yazdanpanah. Network routing based on reinforcement learning in dynamically changing networks. In *ICTAI'05*, pages 5 pp.–366, Nov 2005.

[23] Kikeun Kim, MinWoo Lee, and JaeSung Lim. Spreading technique of satellite beacon to avoid jamming attacks. In *Advanced Communication Technology (ICACT), 2012 14th International Conference on*, pages 778–781. IEEE, 2012.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[25] C. l. Chang. Multiplexing scheme for anti-jamming global navigation satellite system receivers. *IET Radar, Sonar Navigation*, 6(6):443–457, July 2012.

[26] Loh-Ming Li and L. Milstein. Rejection of narrow-band interference in pn spread-spectrum systems using transversal filters. *IEEE Transactions on Communications*, 30(5):925–928, May 1982.

[27] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 1307–1315, May 2007.

[28] W. Liu and D. G. Michelson. Fade slope analysis of ka-band earth-leo satellite links using a synthetic rain field model. *IEEE Transactions on Vehicular Technology*, 58(8):4013–4022, Oct 2009.

[29] Songwu Lu, Vaduvur Bharghavan, and Rayadurgam Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on networking*, 7(4):473–489, 1999.

[30] Supriya Maheshwari, Sridhar Iyer, and Krishna Paul. An efficient qos scheduling architecture for ieee 802.16 wireless mans.

[31] P. Martinelli, E. Cianca, M. De Sanctis, L. Di Paolo, A. Pisano, and L. Simone. Robustness of satellite telecommand links to jamming attacks. In *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, pages 1–6, Oct 2012.

[32] James L Massey. An introduction to contemporary cryptology. *Proceedings of the IEEE*, 76(5):533–549, 1988.

[33] D. Meenakshi, S. Prabha, and N. R. Raajan. Compare the performance analysis for fft based mimo-ofdm with dwt based mimo-ofdm. In *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, pages 441–445, March 2013.

[34] X. Mei and L. Yu. An adaptive hybrid spread spectrum system design and anti-jamming capability analysis. In *High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013 IEEE 10th International Conference on*, pages 226–229, Nov 2013.

[35] Francisco S. Melo and M. Isabel Ribeiro. Q-learning with linear function approximation. In *Proceedings of the 20th Annual Conference on Learning Theory*, COLT'07, pages 308–322, 2007.

[36] Volodymyr Mnih and et.al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[37] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[38] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[39] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[40] Dong Nguyen, Tuan Tran, Thinh Nguyen, and Bella Bose. Hybrid arq-random network coding for wireless media streaming. In *Communications and Electronics, 2008. ICCE 2008. Second International Conference on*, pages 115–120. IEEE, 2008.

[41] J. Park and M. van der Schaar. The theory of intervention games for resource sharing in wireless communications. *IEEE Journal on Selected Areas in Communications*, 30(1):165–175, January 2012.

[42] K. T. Phan, T. Le-Ngoc, M. van der Schaar, and F. Fu. Optimal scheduling over time-varying channels with traffic admission control: Structural results and online learning algorithms. *IEEE Transactions on Wireless Communications*, 12(9):4434–4444, Sep. 2013.

[43] Chris N Potts and Mikhail Y Kovalyov. Scheduling with batching: A review. *European journal of operational research*, 120(2):228–249, 2000.

[44] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[45] Hank Rausch. Jamming commercial satellite communications during wartime: An empirical study. In *Proceedings of the Fourth IEEE International Workshop on Information Assurance*, IWIA '06, pages 109–118, Washington, DC, USA, 2006. IEEE Computer Society.

[46] B. Reiffen and H. Sherman. Parametric analysis of jammed active satellite links. *IEEE Transactions on Communications Systems*, 12(1):102–103, March 1964.

[47] S. Roy, L. Wu, and M. Zawodniok. Spectrum management for wireless networks using adaptive control and game theory. In *2011 IEEE Wireless Communications and Networking Conference*, pages 1062–1067, March 2011.

[48] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end arguments in system design. *ACM Trans. Comput. Syst.*, 2(4):277–288, November 1984.

[49] H. Shimonishi and S. Ishii. Virtualized network infrastructure using openflow. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*, pages 74–79, 2010.

[50] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354 EP –, 10 2017.

[51] Weifeng Su, Z. Safar, and K. J. R. Liu. Towards maximum achievable diversity in space, time, and frequency: performance analysis and code design. *IEEE Transactions on Wireless Communications*, 4(4):1847–1857, July 2005.

[52] K. C. Teh, C. C. Teng, A. C. Kot, and K. H. Li. Jammer suppression in spread spectrum. In *Networks, 1995. Theme: Electrotechnology 2000: Communications and Networks. [in conjunction with the] International Conference on Information Engineering., Proceedings of IEEE Singapore International*, pages 220–224, Jul 1995.

[53] D. Wang, J. Li, W. Gong, and S. Wu. Attitude aided space-time multi-beamformer anti-jamming approach for satellite navigation receiver. In *2014 12th International Conference on Signal Processing (ICSP)*, pages 368–372, Oct 2014.

[54] Qiwei Wang, Thinh P Nguyen, Khanh Pham, and Hyuck M Kwon. Mitigating jamming attack: A game theoretic perspective. *IEEE Transactions on Vehicular Technology*, 2018.

[55] Wenyuan Xu, Ke Ma, W. Trappe, and Yanyong Zhang. Jamming sensor networks: attack and defense strategies. *IEEE Network*, 20(3):41–47, May 2006.

[56] Wenyuan Xu, Wade Trappe, and Yanyong Zhang. Anti-jamming timing channels for wireless networks. In *Proceedings of the First ACM Conference on Wireless Network Security*, WiSec '08, pages 203–213, New York, NY, USA, 2008. ACM.

[57] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57. ACM, 2005.

[58] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang. Coping with a smart jammer in wireless networks: A stackelberg game approach. *IEEE Transactions on Wireless Communications*, 12(8):4038–4047, August 2013.

[59] Dejun Yang, Jin Zhang, Xi Fang, A. Richa, and Guoliang Xue. Optimal transmission power control in the presence of a smart jammer. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 5506–5511, Dec 2012.

[60] B. Zhang and L. Lai. Optimal strategies in jamming resistant uncoordinated frequency hopping systems. In *2013 47th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2013.

APPENDICES

Appendix A: Proof of Proposition 1

The proof will be based on Chernoff bound. Let R_{ij}^k be a random variable denoting the reward obtained by user k at time slot i and channel j. Since at each time slot i and channel j, the probability of transmitting the data for user k is x_{ij}^k, then R_{ij}^k is a Bernoulli random variable:

$$R_{ij}^k = \begin{cases} r_{ij} & \text{with probability } x_{ij}^{k*} \\ 0 & \text{with probability } 1 - x_{ij}^{k*} \end{cases}.$$

Let $\hat{R}_t^k = \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{c} R_{ij}^k$, and $X_t^k = t\hat{R}_t^k$, we want to find an upper bound for $P\Bigg(\hat{R}_t^k < (1-\epsilon)R^k\Bigg) = P\Bigg(X_t^k < t(1-\epsilon)R^k\Bigg)$.

Using Chernoff bound, for any a, we have:

$$P\Bigg(X_t^k \le a\Bigg) \le \min_{s>0} \frac{\mathbb{E}(e^{-X_t^k s})}{e^{-as}}. \tag{A.1}$$

Now,

$$\mathbb{E}(e^{-X_t^k s}) = \mathbb{E}(e^{-s\sum\limits_{i,j}^{t,c} \hat{R}_{ij}^k}) = \mathbb{E}(e^{-s(R_{11}^k + R_{12}^k + \cdots + R_{tc}^k)}).$$

Since R_{ij}'s are independent, we have:

$$\mathbb{E}(e^{-X_t^k s}) = \mathbb{E}(e^{-sR_{11}^k})\mathbb{E}(e^{-sR_{12}^k})\ldots\mathbb{E}(e^{-sR_{tc}^k}). \tag{A.2}$$

Next,

$$\mathbb{E}(e^{-sR_{ij}^k}) = e^{-\frac{s}{r_{ij}}} x_{ij}^{k*} + 1 - x_{ij}^{k*} = 1 - x_{ij}^{k*}(1 - e^{-sr_{ij}}).$$

Using the well known inequality

$$e^{-x} \geq 1 - x$$

and let $x = x_{ij}^{k*}(1 - e^{-sr_{ij}})$, we have:

$$\mathbb{E}(e^{-sR_{ij}^k}) \leq e^{-x_{ij}^k(1-e^{-sr_{ij}})} \leq e^{x_{ij}^{k*}(e^{-sr_{min}^k}-1)}. \tag{A.3}$$

From Eq. A.2 and Eq. A.3, we have:

$$\mathbb{E}(e^{-X_t^k s}) \leq e^{(e^{-sr_{min}^k}-1)\sum_{i,j}^{t,c} x_{ij}^{k*}} = e^{(e^{-sr_{min}^k}-1)\mu^k}. \tag{A.4}$$

Now, using Eq. A.1, and let $a = st(1-\epsilon)R_t^k$, we have:

$$\begin{aligned} P\left(X_t^k \leq s(1-\epsilon)R^k\right) &\leq \min_{s>0} \frac{\mathbb{E}(e^{-X_t^k s})}{e^{-s(1-\epsilon)R^k}} \qquad \text{(A.5)} \\ &\leq \min_{s>0} \frac{e^{(e^{-sr_{min}^k}-1)\mu^k}}{e^{-s(1-\epsilon)R^k}}. \qquad \text{(A.6)} \end{aligned}$$

Taking derivative with respect to s of the right hand side of Eq. A.6 and set it to zero, we have:

$$-r_{min}^k \mu^k e^{s-r_{min}^k} + (1-\epsilon) r_{min}^k \mu^k = 0.$$

Or,

$$e^{-sr_{min}^k} = 1 - \epsilon,$$

$$s = -\frac{\ln(1-\epsilon)}{r^k_{min}}. \tag{A.7}$$

Plug in the optimal s into Eq. A.6, we have:

$$P\Bigg(X^k_t \leq t(1-\epsilon)R^k\Bigg) \leq \frac{e^{-\epsilon\mu^k}}{(1-\epsilon)^{(1-\epsilon)\frac{R^k}{r^k_{min}}}}. \tag{A.8}$$

Since

$$R_k = \sum_{i,j}^{t,c} x^{k*}_{ij} r_{ij} \leq \sum_{i,j}^{t,c} x^{k*}_{i,j} r^k_{max} = \mu^k r^k_{max},$$

Replace $R^k = \mu^k r^k_{max}$ in Eq. A.8, we have:

$$P\Bigg(X^k_t \leq t(1-\epsilon)R^k\Bigg) \leq \Bigg(\frac{e^{-\epsilon}}{(1-\epsilon)^{(1-\epsilon)\frac{r^k_{max}}{r^k_{min}}}}\Bigg)^{\mu^k}, \tag{A.9}$$

which conclude the proof of part one of Proposition 1.

To prove Eq. 3.12 in Proposition 1, we use the Taylor series expansion of $\ln(1-\epsilon)$:

$$\ln(1-\epsilon) = -\epsilon - \frac{\epsilon^2}{2} - \frac{\epsilon^3}{3} - \dots$$

$$(1-\epsilon)\ln(1-\epsilon) = -\epsilon + \frac{\epsilon^2}{2} + \delta,$$

where δ is the sum of all positive terms. Hence,

$$(1-\epsilon)\ln(1-\epsilon) \geq -\epsilon + \frac{\epsilon^2}{2}.$$

Thus,

$$(1-\epsilon)^{(1-\epsilon)\frac{r^k_{max}}{r^k_{min}}} = e^{\frac{r^k_{max}}{r^k_{min}}(1-\epsilon)\ln(1-\epsilon)} \leq e^{(-\epsilon+\frac{\epsilon^2}{2})\frac{r^k_{max}}{r^k_{min}}}.$$

Based on this and Eq. A.9, we have

$$P\left(X_t^k \leq t(1-\epsilon)R^k\right) \leq \left(e^{-\epsilon+(\epsilon-\frac{\epsilon^2}{2})\frac{r^k_{max}}{r^k_{min}}}\right)^{\mu^k}.$$

Since μ^k increases with t, to be useful, we want

$$-\epsilon + (\epsilon - \frac{\epsilon^2}{2})\frac{r^k_{max}}{r^k_{min}} < 0.$$

After a simple algebraic manipulation, this implies that

$$\frac{r^k_{max}}{r^k_{min}} > \frac{2}{2-\epsilon},$$

and Eq. 3.12 follows.

To prove Eq. 3.13 in Proposition 1, we note that if $r_{ij} = r_j$ for all channel i, then the optimal probability $x^{k*}_{ij} = x^{k*}_{1j}$ since time does not matter. Consequently,

$$\mu^k = \sum_{i,j}^{t,c} x^{k*}_{ij} = t\sum_{j}^{c} x^{k*}_{1j} = t\lambda^k.$$

Replacing $\mu^k = t\lambda^k$ in Eq. 3.12, Eq. 3.13 follows immediately.

Appendix B: Proof of Proposition 2

Proof. It will be shown that the Hessian $\nabla^2_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$ is a semi-definite positive matrix (equivalently, its eigenvalues are greater than or equal to 0) for any given $\mathbf{x}$; thus $f(\mathbf{x}, \mathbf{y})$ is convex in $\mathbf{y}$. Similarly, we will show that $\nabla^2_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ is a semi-definite negative matrix (equivalently, its eigenvalues are less than or equal to zero) for any given $\mathbf{y}$; thus $f(\mathbf{x}, \mathbf{y})$ is concave in $\mathbf{x}$. The proof of Proposition 2 immediately follows using Theorem 1.

First note that

$$\nabla^2_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial^2 \mathbf{y}_1} & \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}_1 \partial \mathbf{y}_2} & \cdots & \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}_1 \partial \mathbf{y}_N} \\ \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}_2 \partial \mathbf{y}_1} & \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial^2 \mathbf{y}_2}, & \cdots & \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}_2 \partial \mathbf{y}_N} \\ \cdots & \cdots & \cdots & \\ \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}_n \partial \mathbf{y}_1} & \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial \mathbf{y}_n \partial \mathbf{y}_{N-1}}, & \cdots & \frac{\partial^2 f(\mathbf{x},\mathbf{y})}{\partial^2 \mathbf{y}_N}. \end{bmatrix}$$

With $f(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^{N} B \log \left(1 + \frac{\mathbf{x}_i}{\mathbf{n}_i + \mathbf{y}_i}\right)$,

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{y}_i} &= -\frac{B \mathbf{x}_i}{2(\mathbf{n}_i + \mathbf{y}_i + \mathbf{x}_i)(\mathbf{n}_i + \mathbf{y}_i)} \\ \frac{\partial^2 f}{\partial \mathbf{y}_i \partial \mathbf{y}_j} &= \begin{cases} \frac{B \mathbf{x}_i (2\mathbf{n}_i + 2\mathbf{y}_i + \mathbf{x}_i)}{2((\mathbf{n}_i + \mathbf{y}_i + \mathbf{x}_i)(\mathbf{n}_i + \mathbf{y}_i))^2} & i = j \\ 0 & i \neq j. \end{cases} \end{aligned}$$

Since B and $\mathbf{x}_i$ are greater than or equal to zero, $\nabla^2_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$ is a diagonal matrix whose eigenvalues (diagonal entries) are greater than or equal to zero, or equivalently, $\nabla^2_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$ is a semi-definite positive matrix.

Similarly, for a fixed $\mathbf{y}$, it can be shown that

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{x}_i} &= \frac{B}{2(\mathbf{n}_i + \mathbf{y}_i + \mathbf{x}_i)} \\ \frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j} &= \begin{cases} -\frac{B}{2(\mathbf{n}_i + \mathbf{y}_i + \mathbf{x}_i)^2} & i = j. \\ 0 & i \neq j. \end{cases} \end{aligned}$$

Thus, $\nabla^2_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ is a diagonal matrix whose eigenvalues (diagonal entries) are less than or equal to zero. Equivalently, $\nabla^2_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$ is a semi-definite negative matrix.

□

Appendix C: Proof of Proposition 5

Proof. From Eq. 4.21, $f(\lambda)$ can be expressed as

$$f(\lambda) = \sum_{i=1}^{N} \left(\frac{-\lambda \mathbf{x}_i + \sqrt{\lambda^2 \mathbf{x}_i^2 + 4\mathbf{x}_i \lambda}}{2\lambda} - \mathbf{n}_i \right).$$

Denote

$$\begin{aligned} f(\lambda)_i = & \quad \frac{-\lambda \mathbf{x}_i + \sqrt{\lambda^2 \mathbf{x}_i^2 + 4\mathbf{x}_i \lambda}}{2\lambda} - \mathbf{n}_i \\ = & \quad \frac{-\mathbf{x}_i + \sqrt{\mathbf{x}_i^2 + 4\mathbf{x}_i / \lambda}}{2} - \mathbf{n}_i. \end{aligned}$$

Then,

$$\begin{aligned} \frac{\partial f(\lambda)_i}{\partial \lambda} &= \frac{\partial}{\partial \lambda} \left(\frac{-\mathbf{x}_i + \sqrt{\mathbf{x}_i^2 + 4\mathbf{x}_i / \lambda}}{2} - \mathbf{n}_i \right) \\ &= \frac{\partial}{\partial \lambda} \left(\frac{-\mathbf{x}_i + \sqrt{\mathbf{x}_i^2 + 4\mathbf{x}_i / \lambda}}{2} \right) \\ &= \frac{\partial}{\partial \lambda} \left(\frac{\sqrt{\mathbf{x}_i^2 + 4\mathbf{x}_i / \lambda}}{2} \right). \end{aligned}$$

It is obvious that $\sqrt{\mathbf{x}_i^2 + 4\mathbf{x}_i/\lambda}$ is monotonically decreasing in λ, that is

$$\begin{aligned}\frac{\partial f(\lambda)_i}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\left(\frac{\sqrt{\mathbf{x}_i^2 + 4\mathbf{x}_i/\lambda}}{2}\right) \\ &\leq 0.\end{aligned}$$

Thus,

$$\begin{aligned}\frac{\partial f(\lambda)}{\partial \lambda} &= \sum_{i=1}^{N} \frac{\partial f(\lambda)_i}{\partial \lambda} \\ &\leq 0\end{aligned}$$

□