Game-Theoretic Frameworks and Strategies for Defense Against Network Jamming and Collocation Attacks

2019

Ahmed Hemida
University of Central Florida

GAME-THEORETIC FRAMEWORKS AND STRATEGIES FOR DEFENSE AGAINST NETWORK JAMMING AND COLLOCATION ATTACKS

by

AHMED HOSSAM ANWAR ABDO HEMIDA
B.Sc. Alexandria University, Egypt 2011
M.Sc. Nile University, Egypt 2013

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2019

Major Professor: George Atia

ABSTRACT

Modern networks are becoming increasingly more complex, heterogeneous, and densely connected. While more diverse services are enabled to an ever-increasing number of users through ubiquitous networking and pervasive computing, several important challenges have emerged. For example, densely connected networks are prone to higher levels of interference, which makes them more vulnerable to jamming attacks. Also, the utilization of software-based protocols to perform routing, load balancing and power management functions in Software-Defined Networks gives rise to more vulnerabilities that could be exploited by malicious users and adversaries. Moreover, the increased reliance on cloud computing services due to a growing demand for communication and computation resources poses formidable security challenges due to the shared nature and virtualization of cloud computing. In this thesis, we study two types of attacks: jamming attacks on wireless networks and side-channel attacks on cloud computing servers. The former attacks disrupt the natural network operation by exploiting the static topology and dynamic channel assignment in wireless networks, while the latter attacks seek to gain access to unauthorized data by co-residing with target virtual machines (VMs) on the same physical node in a cloud server. In both attacks, the adversary faces a static attack surface and achieves her illegitimate goal by exploiting a stationary aspect of the network functionality. Hence, this dissertation proposes and develops counter approaches to both attacks using moving target defense strategies. We study the strategic interactions between the adversary and the network administrator within a game-theoretic framework.

First, in the context of jamming attacks, we present and analyze a game-theoretic formulation between the adversary and the network defender. In this problem, the attack surface is the network connectivity (the static topology) as the adversary jams a subset of nodes to increase the level of interference in the network. On the other side, the defender makes judicious adjustments of the transmission footprint of the various nodes, thereby continuously adapting the underlying network

topology to reduce the impact of the attack. The defender's strategy is based on playing Nash equilibrium strategies securing a worst-case network utility. Moreover, scalable decomposition-based approaches are developed yielding a scalable defense strategy whose performance closely approaches that of the non-decomposed game for large-scale and dense networks. We study a class of games considering discrete as well as continuous power levels.

In the second problem, we consider multi-tenant clouds, where a number of VMs are typically collocated on the same physical machine to optimize performance and power consumption and maximize profit. This increases the risk of a malicious virtual machine performing side-channel attacks and leaking sensitive information from neighboring VMs. The attack surface, in this case, is the static residency of VMs on a set of physical nodes, hence we develop a timed migration defense approach. Specifically, we analyze a timing game in which the cloud provider decides when to migrate a VM to a different physical machine to mitigate the risk of being compromised by a collocated malicious VM. The adversary decides the rate at which she launches new VMs to collocate with the victim VMs. Our formulation captures a data leakage model in which the cost incurred by the cloud provider depends on the duration of collocation with malicious VMs. It also captures costs incurred by the adversary in launching new VMs and by the defender in migrating VMs. We establish sufficient conditions for the existence of Nash equilibria for general cost functions, as well as for specific instantiations, and characterize the best response for both players. Furthermore, we extend our model to characterize its impact on the attacker's payoff when the cloud utilizes intrusion detection systems that detect side-channel attacks. Our theoretical findings are corroborated with extensive numerical results in various settings as well as a proof-of-concept implementation in a realistic cloud setting.

Dedicated to my wife Salwa, and my beloved daughter Talia

ACKNOWLEDGMENTS

First, and foremost, all praise and thanks are to Allah for his generosity and blessings. I would like to express my sincere gratitude and appreciation to Dr. George Atia for his guidance. I would like to thank him for his constant support, encouragement, and inspiring advice. His help and guidance throughout my study at UCF are invaluable. I am also very grateful to Dr. Mina Guirguis for his amazing guidance and support. I would like to thank him and his group at Texas State University for their productive collaboration.

I want to thank my doctoral committee members, Prof. Marwan Simaan, Dr. Azadeh Vosoughi, and Dr. Gita Sukthankar for their time, and also, for providing insightful comments and stimulating thoughtful discussion. My deepest gratitude to Diana Camerino for her kind help throughout my course of study.

Many thanks to all my brilliant DSP lab-mates. I would like to express special thanks to my lab-mate A. Aldahab, and my dearest friends A. Zewail, A. Ibrahim, and M. Ashour at PSU for sharing my memories, my sorrows, and critical situations. The help all of you offered, the thoughts we exchanged, and the interests we shared made these Ph.D. years unforgettable.

I would like to express my deepest gratitude and appreciation to my parents and my sister who have offered everything to me to reach this point in my life. Their encouragement and support are unbounded. I am truly lucky to have my wife Salwa, and my beautiful daughter, Talia, who always sacrifice their happiness for me. I am very lucky to have such a loving and respectful person in my life. I love you.

TABLE OF CONTENTS

LIST OF FIGURES xiii

LIST OF TABLES xx

CHAPTER 1: INTRODUCTION 1

1.1 Science of Cybersecurity 2

1.2 Attack Surface 4

1.3 Defense Strategies 5

1.4 Game Theory 5

1.4.1 Normal Form Game 7

1.4.2 Nash Equilibrium 8

1.5 Contributions 8

1.6 Organization 12

CHAPTER 2: PINBALL ATTACKS AGAINST DYNAMIC CHANNEL ASSIGNMENT IN WIRELESS NETWORKS 13

2.1 Introduction 13

2.2 Related Work 16

2.2.1 Channel Assignment Techniques 16

2.2.2 Security Implications of Channel Assignment 17

2.3 System Model 18

2.3.1 Channel Assignment Technique 19

2.3.2 Decoy Pinball Attack 22

2.3.3 MDP Formulation 23

2.3.4 Optimal Attack Policy 27

2.3.5 Suboptimal Attack Strategy 29

2.3.6 Feature Selection 30

2.3.7 Dimensionality Reduction: Big Graphs 32

2.4 The Dual Problem 33

2.4.1 Equivalence Formulation 33

2.4.2 Attacking Isomorphic Graphs 38

2.5 Experimental Results 39

2.5.1 Average Path Reward 43

2.5.2 Conflict Analysis 47

2.5.3 Node Attack Frequency Analysis 50

2.5.4 Conflict Decay Analysis 54

2.5.5 Conflict Size Distribution Analysis 55

2.5.6 Conflict Induction and Accumulation Analysis 56

2.5.7 Graph Connectivity Analysis 56

2.5.8 Attacking Networks of Repeated Structures 58

2.5.9 Attacking Networks of Arbitrary Structures 64

CHAPTER 3: ADAPTIVE TOPOLOGIES AGAINST JAMMING ATTACKS IN WIRELESS NETWORKS 66

3.1 Related Work 68

3.1.1 Security in Wireless Channel Allocation Techniques 68

3.1.2 Game-Theoretic Formulations in Security Games 69

3.2 System Model 71

3.2.1 Dynamic Channel Assignment 72

3.2.2 Jamming Attacks 72

3.2.3 General Framework 73

3.3 Game Formulation 74

3.3.1 Discrete Game 75

3.3.1.1 Decomposition-based approach 78

3.3.1.2 Progressive decomposition approach 80

3.3.2 Discrete Game with Explicit Constraints 81

3.3.3 Continuous Game 86

3.3.4 Extension to Multiple Bands 93

3.4 Experimental Results 94

3.5 Stochastic Games 108

3.5.1 Mixed strategy, state evolution and objective function 111

3.5.2 Q-minmax Algorithm 113

3.6 Approximate Policy 114

3.7 Experimental Results 115

3.7.1 Learning step 116

3.7.2 Testing step 119

CHAPTER 4: VIRTUAL MACHINES MIGRATION TIMING PROBLEM 121

4.1 Introduction 121

4.2 Related work 123

4.2.1 Cross-VM side channel attacks and mitigation strategies 123

4.2.2 Cloud security using game-theoretic techniques 124

4.3 System Model 126

4.3.1 The cloud 126

4.3.2 The game 127

4.3.2.1 Defender's action space 127

4.3.2.2 Attacker's action space 128

4.3.2.3 Attacker's utility 129

4.3.2.4 Defender's utility 130

4.4 Theoretical Analysis 131

4.4.1 General reward functions 132

4.4.2 Special instantiation analysis 136

4.5 Generalization: Game Model with IDS 141

4.6 Numerical Analysis 143

4.6.1 Utility functions 146

4.6.2 Cost effect and monotonicity 148

4.6.3 Best response curves 151

4.6.4 Different reward scaling regimes 155

4.6.5 Game simulation and implementation 157

4.6.6 Extended model with IDS 158

CHAPTER 5: CONCLUSION 162

LIST OF REFERENCES 164

LIST OF FIGURES

2.1 Step 1: Initial channel assignments, the network has 3 conflicts. Step 2: After an AP switches from channel 9 to 2, the network has 1 conflict. Step 3: After an AP switches from channel 7 to 10, the network is conflict-free. 21

2.2 Step 1: System starts with 3 conflicts. Step 2: Attacker jams its victim node with jamming channel 3 and the system switched the node using channel 9 to the non-interfering channel 2. However, due to the jammer the new frequency is now suffering interference. Step 3: The system is not only unable to resolve conflicts in 3 steps, but also three nodes may not find any vacant non-interfering channels, resulting in perpetual switching among other nodes as well. 21

2.3 Approximate policy and exact policy path reward comparison. 31

2.4 Edge attack topology. 37

2.5 Edge attack topology mapped to node attack problem. 38

2.6 A 6-node chain network topology. 41

2.7 5-node interconnected network topology. 41

2.8 A 6-node ring network topology. 42

2.9 A 7-node tree network topology. 42

2.10 Comparing different attack policies on the 5-node chain topology. 43

2.11 Comparing different attack policies on the 5-node interconnected network topology. 43

2.12 Comparing different attack policies on the 5-node ring topology. 45

2.13 Comparing different attack policies on the 7-node tree topology. 45

2.14 A 6-node topology representing the network of the tested building. 46

2.15 Plan view of the tested building. 46

2.16 Comparing different attack policies on commercial building APs. 47

2.17 Accumulation of conflicts in the 7-node tree network for different attack policies. 48

2.18 Accumulation of conflicts in the 5-node interconnected network for different attack policies. 49

2.19 Accumulation of conflicts in the 8-node mixed network topology. 49

2.20 An 8-node mixed network topology. 50

2.21 Frequency of attack actions for the interconnected network topology. 50

2.22 Frequency of attack actions for the tree topology. 51

2.23 Frequency of attack actions for the 8-node mixed network topology. 51

2.24 Frequency of attack actions for the 8-node dense network topology 52

2.25 An 8-Node dense network topology. 52

2.26 Conflicts resolution over time for the 8-node mixed network topology. 54

2.27 Frequency of different conflict sizes in the 8-node mixed network topology. . 54

2.28 Path reward gained over time when 8-nodes mixed network topology starts from a zero-conflict state 55

2.29 A 10-node graph with $\delta_{\text{avg}} = 1.8$. 57

2.30 A 10-node graph with $\delta_{\text{avg}} = 4$. 57

2.31 Comparing the performance of different attack policies as we increase the connectivity of a 10-node graph. 58

2.32 Graph with repeated ring structure. 59

2.33 Graph with repeated interconnected subgraph structure. 60

2.34 Graph with repeated tree structure. 60

2.35 Comparing attack policies for graph with repeated ring structure. 61

2.36 Comparing attack policies for graph with repeated interconnected subgraph structure. 61

2.37 Comparing attack policies for graph with repeated tree structure. 62

2.38 Partial view of an 800-node network with a repeated structure. 62

2.39 Comparing attack policies for the 800-node network. 63

2.40 A 37-node network with node degree distribution. 63

2.41 A 32-node network of a repeated structure 64

2.42 The gap due to approximating a graph with a repeated structure graph 65

3.1 Building block topologies, (a) 7-node tree, (b) 5-node star, (c) 5-node interconnected network. 94

3.2 Defender reward for the 7-node tree network topology versus attack cost. . . . 96

3.3 Defender reward for different attack costs for the 5-node star network. 96

3.4 Defender reward for the 5-node interconnected network topology vs attack cost. 97

3.5 Defender reward for topology (a) vs number of conflicts. 98

3.6 Defender reward for topology (a) vs attack cost against two colluding jammers. 99

3.7 (Left) Mixed strategy for power assignment of 28 nodes in 3-floor building, (Right) The top plot shows the defender reward vs the attack cost, and the bottom plot displays the time evolution of the conflicts with and without defense. 100

3.8 Run time in seconds. 100

3.9 Defender's reward for a randomly generated 64-node topology. 101

3.10 Defender's strategies in a 3-floor building. 2D projection of the 3D footprint of each access point, without coverage constraints (left) and with coverage constraints (right). 101

3.11 3D wireless footprints based on marginal strategy. 103

3.12 Tradeoff between the number of non-conflicting APs and the minimum network coverage. 103

3.13 Power assignment in 3-floor building over two different wireless bands to enhance coverage 104

3.14 Defender reward for the 7-node tree network topology versus attack cost. . . . 106

3.17 Power radii for different minimum coverage constraints. 106

3.15 Defender reward for different attack costs for the 5-node star network. 107

3.16 Defender reward for the 5-node interconnected network topology vs attack cost. 108

3.18 A 3-node chain network topology 116

3.19 Convergence of the state value function for a 3-node chain network. 117

3.20 The defender strategy at a sample state for a 3-node chain network converges to a mixed NE strategy. 117

3.21 The attack strategy at a sample state for a 3-node chain network converges to a mixed NE strategy. 118

3.22 The defender's NE average path reward for a 3-node chain network against an attacker that plays best response and one that uses a random attack. Comparison to no-defense is also shown. 118

3.23 A 5-node interconnected network topology. 119

3.24 The defender's average path reward for a 5-node interconnected network compared with other attack and defense policies. 120

4.1 System model illustration for different placement events. 126

4.2 Attacker evades IDS by early stopping of malicious activity. 142

4.3 Attacker detected by the IDS. 143

4.4 For the shown action space, $\mathcal{A} = \mathcal{A}_d \times \mathcal{A}_a$, the game admits a NE in pure strategies. 144

4.5 At $\tau_d = 0.1$, the attacker payoff is monotonically decreasing, but not for $\tau_d = 2$, in agreement with the bound on C_a in Theorem 15. 145

4.6 Defender's utility versus migration time τ_d 147

4.7 Attacker's utility versus attack rate λ_a 148

4.8 Defender's utility versus migration time τ_d for $C_d = 0.03$ 149

4.9 Defender's utility versus migration time τ_d for $C_d = 10$ 149

4.10 Attacker's utility versus attack rate λ_a for $C_a = 0.01$ 150

4.11 Attacker's utility versus attack rate λ_a for $C_a = 6$. 151

4.12 Players best response curves for $C_d = 2.5$ and $C_a = 1$ 152

4.13 Players best response curves for $C_d = 0$ and $C_a = 0$ 153

4.14 Players best response curves for $C_d = 10$ and $C_a = 0$ 153

4.15 Players best response curves for $C_d = 0$ and $C_a = 6$ 154

4.16 Defender's best response curves for different reward scaling regimes. 154

4.17 Attacker's best response curves for different reward scaling regimes. 155

4.18 Attacker's payoff with and without IDS. 159

4.19 Attacker's utility vs stopping time s at different costs of detection. 160

4.20 Comparison between the proposed migration approach and no-migration. . . 160

4.21 Comparing the defender's utility with the proposed and the random migration policies at several migration cost values. 161

LIST OF TABLES

4.1 Players' Payoff For Several Attack and Defense Strategies. 159

CHAPTER 1: INTRODUCTION

Networks are now densely connected and heterogeneous to enable more services to an ever-increasing number of users. For instance, modern networks connect computers with different operating systems and protocols, wireless networks accommodate different access technologies and are able to provide services through a Wi-Fi LAN or through a cellular network as in heterogeneous wireless networks. Moreover, an increasing number of devices are being added to networks everyday. For example, the deployment of wireless-enabled devices (e.g., Internet of Things, wearables, etc.) has made our networks larger and denser. Hence, networks are prone to higher levels of interference, which makes them more vulnerable to jamming attacks. To accommodate such growth under spectrum constraints, significant research efforts focused on developing spectrum management techniques through the use of Software Defined Networks (SDNs), Network Function Virtualization (NFV), and Cognitive Radios (CR) to improve the spectrum utilization. These techniques rely on sensing the current conditions (e.g., channels used, transmission power, interference levels, etc.) and dynamically making the appropriate allocation decisions. Current Access Points (APs) can dynamically select their channels and their transmission ranges to minimize interference with surrounding APs in wireless mesh networks (e.g., Cisco's Transmit Power Control) [1]. The shared nature of wireless communication, however, has made many of the above techniques susceptible to attacks.

SDNs seek to separate network control functions from network forwarding functions, while NFV seeks to abstract network forwarding and other networking functions from the hardware on which it runs. Thus, both depend heavily on virtualization to enable network design and infrastructure to be abstracted in software and then implemented by the underlying software across hardware platforms and devices [2].

When SDNs execute on an NFV infrastructure, SDNs forward data packets from one network device to another. At the same time, SDNs' networking control functions for routing, policy definition and applications run in a virtual Machine (VM) somewhere on the network. Thus, NFV provides basic networking functions, while SDNs control and orchestrate them for specific uses. SDNs further allow configuration and behavior to be programmatically defined and modified. These new technologies are now replacing the infrastructure of traditional communication networks. However, it has led to a new set of vulnerabilities and security holes due to the reactive and dynamic nature of frequency assignment and power management protocols. Therefore, we shed more light on attackers' techniques first and how they exploit such vulnerabilities, then we propose effective remedies to those attacks.

Moreover, the growing usage of virtualization have increased the number of security issues in future networks. Therefore, security challenges in cloud servers are getting more attention. Virtualization is one of the main components of a cloud. But this poses major security risks. Ensuring that different instances running on the same physical machine are isolated from each other is a major task of virtualization which is not guaranteed completely in existing platforms [3]. In this thesis, we focus on the issue of collocation between neighboring VMs in public cloud servers and how to overcome possible collocation attacks. Before we delve into the detailed contributions of this dissertation, we first put our work into the context of cyber science.

1.1 Science of Cybersecurity

The annual Data Breach Investigations Report (DBIR) issued by Verizon Inc reported more than 53,000 security related incidents and 2,216 confirmed data breaches in the year 2018. Although 76% of the crimes were financially motivated, 58% of the victims were small businesses and 25% of the targets were healthcare organizations. Since people are becoming increasingly aware of

unsafe emails, attacks caused by malware went down from 50% in 2017 to 30% in 2018. However, the total number of security incidents and data breaches is increasing which poses more challenges to network and system administrators regarding how to secure their database servers and in case of being under attack, how to mitigate the effects of such attacks.

The domain of science of cybersecurity comprises phenomena that involve malicious agents used to hack and damage a network of computing devices to perform actions desired by the attacker and generally contrary to the intent (the policy) of the network administrator (the defender). More specifically, the objects of research in cybersecurity can be summarized into:

- An attacker or (a set of colluding attackers) along with the attacker's tools and techniques (set of actions)
- A network defender with a set of defensive tools and actions
- The operational assets, networks and systems, which represent the target of the attacker.
- A set of defender's assertions or requirements about what events should and should not happen. To simplify, we may focus on cyber incidents and events that should not occur [4].

The science of cybersecurity develops a coherent family of models of relations between attributes, structures and dynamics of: violations of cybersecurity policy; the network of computing devices under attack; the defenders' tools and techniques; and the attackers' tools and techniques where malicious software plays the central role.

For example, in the **intrusion detection** problem we seek to derive the best response actions for the defender given the system status, actions taken by the adversary and the set of unwanted events. In other words, the goal is to find defense techniques that detect certain types of malicious activities which affect certain networked system, and result in some undesired cyber incidents.

Another field of study within the science of cybersecurity is to assess the network **vulnerabilities**. In this case, the goal is to identify the weaknesses of the system and the structures of the network with respect to the defender given the possible actions that can be taken by an attacker.

In this thesis we focus mainly on deriving defensive techniques and securing actions to protect wireless networks and cloud servers against jamming and collocation attacks, respectively.

1.2 Attack Surface

In the domain of software security, the attack surface of a software environment is the collection of points through which an unauthorized user (the attacker) can try to enter data to, or extract data from, an environment. Hence, security experts aim to reduce the attack surface. In this thesis, we use the same notion of attack surface to represent the main points that the attacker is exploiting within the network to disrupt the natural operation of the network. Specifically, we investigate two attack scenarios that have the same attack surface nature as explained next.

We study two types of attacks: jamming attacks on wireless networks and side-channel attacks on cloud networks. The former attacks disrupt the natural network operation by exploiting the static topology and dynamic channel assignment in wireless networks, while the latter attacks seek to gain access to unauthorized data by co-residing with target VMs on the same physical node in a cloud network. In both attacks, the adversary exploits a fixed or static attack surface and achieves her illegitimate goal by exploiting a stationary aspect of the network functionality. Hence, this dissertation proposes and develops counter approaches to both attacks using Moving Target Defense (MTD) strategies to shift and modify the attack surface periodically. We study the strategic interactions between the adversary and the network administrator within a game-theoretic framework.

1.3 Defense Strategies

The most common and widely implemented reactive defense approach is the Intrusion Detecction and Prevention system (IDPS). Such systems analyze the behavior of the users online and capture anomalies and malicious activities as a reactive defense mechanism. However, the network administrator needs to develop defense strategies in order to prevent and mitigate the effects of those attacks and keep hold of the network resources until the IDPS identifies the malicious user and knocks her out of the network. To defend against an attacker who is exploiting the static nature of the network's attack surface, we rely on Moving Target Defense (MTD) strategies as a proactive defense technique to change the attack surface of the system over time. In this case, the defender continuously moves the attack surface and controls the network resources in order to mitigate the damage of the attack and alter any prior knowledge that the attacker may have obtained. As a result, the adversary attacks the network based on outdated information, or instead spends additional time to monitor the system before launching another attack. In order to understand the strategic interactions between the attacker and the system's defender we develop a game-theoretic framework between the two players. This thesis studies the trade-offs between the direct impact of the aforementioned attacks and the attack cost incurred by the adversary.

Next, we introduce the standard definition of normal form games which are commonly used to model security games in networks and systems.

1.4 Game Theory

Game theory can be defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers. It provides general mathematical techniques for analyzing situations in which two or more individuals make decisions that will influence one another's

welfare. As such, it offers insights of fundamental importance for researchers in several branches of science as well as for practical decision-making [5]. Game theory is widely adopted by the engineering community as a distributed optimization and control framework for networked systems, partly for taking into account preferences of individual users, who share and compete for system resources. Resting upon a rich mathematical foundation, game theoretical approaches, especially strategic (non-cooperative) games, have been valuable for the analysis and design of various resource allocation protocols in wireless and wired networks. Problems such as rate control, interference management, and power control (e.g. in wireless and optical networks), for example, have been investigated using game theoretical methods.

A non-cooperative game is a game with competition between individual players and in which only self-enforcing alliances are possible due to the absence of external means to enforce cooperative behavior. This is in contrast to cooperative games, which allow for external enforcement of cooperative behavior. Cooperative games focus on predicting which coalitions will form, what the possible joint group actions are, and the resulting collective payoffs. On the other hand, non-cooperative game theory focuses on predicting individual players' actions and payoffs. Players are aware of the fact that the outcome of the game is affected by each individual in the game. One of the principal assumptions in game theory is that all players are rational, which means that they have well defined objectives over the set of all possible actions of the game and would implement the best available strategy to pursue those objectives.

To study the strategic interactions between attackers and defenders, game theory has provided a rigorous framework to model the strategies of each player and has been instrumental in advancing the state-of-the-art in various security areas. In the following chapters of this dissertation, we present different classes of games. These games are classified according to the type of the players' action space (e.g, continuous or discrete). Also, those games can fall into one of two categories, static or dynamic games depending on whether they are played in one shot or sequentially.

1.4.1 Normal Form Game

Also called Strategic Form Game, a normal formal game is defined as a triple $\Gamma(\mathcal{K}, \mathcal{A}, \mathcal{R})$

- $\mathcal{K}$ is a finite set of players.
- $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_{|\mathcal{K}|}$ is the action profile of the game defined as the product of the action space of all players.
- $\mathcal{R} = \{R_1, R_2, ..., R_{|\mathcal{K}|}\}$ is the reward (payoff) function, where the reward of the i^{th} player is $R_i : \mathcal{A} \to \mathbb{R}$.

A joint action is $a = (a_1, a_2, ..., a_K)$, where $K = |\mathcal{K}|$ is also known as strategy profile. A best response is the strategy which maximizes the outcome of a player in response to certain actions taken by his opponents. In other words, for a fixed collection of other players strategies a_{-i}, a strategy a_i played by the i^{th} player is a best response if,

$$R_i(a_i, a_{-i}) \geq R_i(a'_i, a_{-i}) \ , \ \forall \ a'_i \in \mathcal{A}_i. \tag{1.1}$$

Definition 1. *In an N-person nonzero sum game, let $u_i(a_1, \ldots, a_i, \ldots, a_N)$ be the utility function of player i. For each player $i \in \{1, \ldots, N\}$, assume that the maximum of u_i with respect to $a_i \in \mathcal{A}_i$ can be attained for any players' action profile $a_{-i} \in \mathcal{A}_{-i}$, where $a_{-i} := \{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N\}$ and $\mathcal{A}_{-i} \equiv \mathcal{A}_1 \times \ldots \times \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \ldots \times \mathcal{A}_N$. Then, the set $R_i(a_{-i}) \subset \mathcal{A}_i$ defined by*

$$u_i(a_{-i}) = \{\zeta \in \mathcal{A}_i : u_i(\zeta, a_{-i}) \geq u_i(a_i, a_{-i}), \ \forall a_i \in \mathcal{A}_i\},$$

is called the optimal (or best) response of player i. If u_i is a singleton for every $a_{-i} \in \mathcal{A}_{-i}$, then it

is called the reaction curve [6, 7].

1.4.2 Nash Equilibrium

Nash equilibrium (NE) is a strategy profile such that each strategy is a best response (maximizes payoff) to all the other strategies. An outcome $a^* = (a_1^*, a_2^*, ..., a_K^*)$ is a NE if for each player $i \in \mathcal{K}$,

$$R_i(a_i^*, a_{-i}^*) \geq R_i(a_i, a_{-i}^*) \ \forall \ a_i \in \mathcal{A}_i \tag{1.2}$$

Accordingly, it follows from the definition of a NE (in that no player can gain by a unilateral change of strategy if the strategies of the other players remain unchanged) that the intersection points of the best responses are NE. Hence, NE is considered self-enforcing in the sense that no player has an incentive to deviate unilaterally.

1.5 Contributions

Finally, we state the main contributions of our dissertation:

- In a wireless network setting, we characterize a new type of jamming attacks and propose defense strategies.
  - We characterize a new type of attack strategy (aka pinball attacks) that aims to induce a cascading channel switching behavior exploiting the adaptive nature of Dynamic Channel Assignment (DCA) algorithms and the connectivity of the wireless nodes. We show that the attacker can solve an MDP to maximize the expected long-term reward.
  - We develop suboptimal attack policies by judiciously choosing representative features

in an approximate dynamic programming framework to deal with the explosion of the state space.

- We devise approximations to scale the policies learned for simple topologies to larger size networks with repeated structures.
- We study two general setups (node and edge centric) and establish a duality principle, namely, pinball attack policies learned for one setup can be directly mapped to attack policies on the other.
- We conduct an extensive numerical evaluation to study the impact of pinball attacks on various network topologies in different scenarios, including a wireless network of APs deployed in a commercial building. Our results demonstrate that pinball attacks can cause significantly more damage than other attack policies even on large scale networks.
- We develop a unified game-theoretic formulation in which the defender simultaneously adapts the footprint of N wireless nodes to maximize his utility against a jamming adversary who selects a node (or a subset of nodes) to jam to increase the degree of interference between nodes. We consider various cases of the action spaces of the players (e.g., binary, discrete based on multiple power levels and continuous). In all these cases, we consider a zero-sum game parameterized by an "attack-cost" metric that captures the risk of exposure for the adversary, a "coverage" metric or constraint that captures the level of service that needs to be maintained by the defender (e.g., preventing nodes from having too small of footprints that they do not become useful), and a measure of the "attack-impact" on the network due to the action of the jammer. This formulation is very different from previous work in this area that typically consider the defender as a single sender-receiver pair along with a jamming attacker [8, 9]. In particular, our formulations capture the dependencies between neighboring nodes

resulting in coupled optimization problems that we solve efficiently.

– In addition we develop concrete decomposition approaches (c.f. Section 3.3.1.1, 3.3.1.2) to obtain scalable defense strategies which are applicable to large networks. Due to the combinatorial nature of the action space for the defender (e.g., selecting the footprint of N nodes based on just two transmission power levels is 2^N), finding optimal strategies for large networks is computationally prohibitive for the discrete control set settings considered. In the decomposition-based approach, we developed two novel techniques:

  1. We decompose the game into N sub-games, one for each AP. We solve a sub-game for each AP assuming a worst-case scenario for the defender based on the state of the surrounding nodes. Solutions for the sub-games are combined to yield a global tractable and scalable mixed-strategy that well approximates the optimal best response strategy based on the non-decomposed game.
  2. We consider progressive games in which we sequentially solve small sub-games per node according to a predefined ordering of the nodes. The footprints obtained for a node and its neighbors are leveraged and fed to subsequent sub-games until all nodes are exhausted to obtain a solution for the game.
  3. We consider a discrete action space with multiple power levels and explicit coverage constraints. To address complexity, we optimize over marginal strategies that satisfy local and global coverage constraints to ensure network dependability (c.f. Section 3.3.2). Then, we show that the obtained marginalized strategy is implementable, i.e, can be expressed as a convex combination of the pure actions of the defender.

– Moreover, we consider a continuous version of the game wherein the utility function is redefined in terms of the continuous domain of the node and jamming power distribution. We prove that this setting admits a unique pure strategy NE.

  - We support our findings with results from a real network topology consisting of 28 nodes deployed in a 3-floor academic building, in addition to results from extensive simulation studies based on "building-block" topologies. We evaluate the performance of our defense strategies played against various adversaries (e.g., random, rational, computationally unrestrained, greedy) and demonstrate their effectiveness.

- In a cloud network setting, we develop timely migration defense approach to secure VMs running on public clouds against side channel attacks. The proposed approach is based on moving target defense strategy.

  - We provide a new game-theoretic formulation for the VM collocation timing problem. We do not assume the defender has prior knowledge of the exact location of the attacker, thereby allowing for realistic threat and defense models. The defender has to migrate the VMs at the right time(s) to defend against malicious collocating users.

  - We analytically characterize the NE for the studied game model and derive sufficient existence conditions.

  - We study the behavior of the adversary when the defender adopts an intrusion detection system (IDS). In this case, the adversary not only takes attack actions, but also decides when to stop her attack to reduce the risk of being detected.

  - We provide extensive numerical experiments to support our theoretical findings. In our numerical evaluation, we consider several reward functions to reflect the degree of the attack and the severity of the data breach. As a proof of concept, we also implement the migration defense approach on a realistic cloud setup using the Xen hypervisor.

1.6 Organization

This dissertation is organized as follows: A motivational security problem is proposed in Chapter 2 to explore the security challenges in wireless networks [10]. In Chapter 3 we present a static security game to secure wireless networks against jamming attacks [11]. We also present a dynamic version of the game [12]. In Chapter 4 we present a timing game to secure public clouds against side-channel attacks, [13]. In Chapter 5 we conclude our work.

CHAPTER 2: PINBALL ATTACKS AGAINST DYNAMIC CHANNEL ASSIGNMENT IN WIRELESS NETWORKS

2.1 Introduction

Spectrum shortage and spectrum interference are two major challenges that face the evolution and innovation of wireless technologies. On one hand, we are witnessing a tremendous growth in the numbers and types of wireless devices owing to the emergence of the Internet of Things (IoT) with different (often high) bandwidth requirements and provisioned Quality of Service (QoS) that supports mobility. By 2021, it is envisioned that there will be up to 27.1 billion networked devices putting strain on the available wireless resources [14]. On the other hand, the available wireless spectrum is not growing as fast as it is tightly regulated by the Federal Communications Commission (FCC). These challenges have prompted notable research efforts in the areas of Cognitive Radio (CR) Networks [15], Software Defined Networks (SDNs) [16] and femtocells [17] so that the frequency spectrum can be efficiently utilized among devices. Improving the utilization of the wireless spectrum requires the wireless nodes to monitor the spectrum (e.g., channels, noise, load) and adapt their own usages (e.g., choice of channel, power).

To minimize interference among wireless nodes (e.g., access points (APs) in a building), each radio interface should be assigned a different frequency channel from those assigned to other adjacent radio interfaces. Due to spectrum shortage, the number of available channel is limited. For example, only 11 channels are allowed in the US in the 802.11 specifications and only channels 1, 6 and 11 do not overlap [18]. In general, signal interference is classified into two types:

- *Co-Channel Interference (CCI):* CCI occurs between two or more radio interfaces that use the same frequency channel and are within each others' interference radius. CCI increases

transmission delays due to medium contention as nodes must wait for an idle medium before transmission.

- *Adjacent-Channel Interference (ACI):* ACI occurs between two or more radio interfaces that use adjacent overlapping frequency channels. In ACI, a node's transmission is perceived as a noise signal to nearby nodes on adjacent overlapping channels. ACI causes poor quality of service and packet loss to the involved nodes.

To address CCI and ACI, channel assignment (CA) techniques carefully assign frequencies to radio interfaces in a non-overlapping non-interfering manner, whenever possible. CA techniques are classified into two groups: Static Channel Assignment (SCA) and Dynamic Channel Assignment (DCA) techniques. While SCA is typically used in smaller networks with limited mobility, DCA allows nodes to dynamically select frequencies based on the network state and the sensed assignments of their neighbors. Wireless networks employing DCA techniques to provide the nodes with this self-resolving capability in software, however, are vulnerable to pinball attacks introduced in this work. An example is SDNs, where the DCA function has been pushed to the software layer [19], making them susceptible to such exploits.

The goal of an RF jammer is to increase the interference on the radio channel to degrade the signal quality at the receiver [20]. On the other hand, MAC layer attacks typically target specific protocols that are responsible for power management and/or disrupt the association process of stations with the APs [21]. The adaptive nature of DCA techniques makes them susceptible to attacks [22, 23]. An attacker can easily trigger unnecessary switching behavior by inducing a conflicting channel that causes nearby victims to switch their channels (with a possible cascading effect of damage). A channel switch involves some overhead. For example, an AP must perform availability check prior to switching, then broadcast an 802.11h channel switch announcement, and switch channels at the end. This process takes 224 microseconds in hardware, but at Layer 2 or 3 it can take more

than 320ms [24, 25]. Excessive channel switching behavior based on induced conflicts increases the network latency and could potentially prevent the network from reaching a resolving state. Furthermore, an intelligent attacker can tailor their attack to cause nodes to switch to an overall configuration that has a higher number of conflicts, exploiting the way nodes are connected and their node degrees. Understanding the effect of attacks that exploit software-based network activity is crucial to improving the security of various wireless networks (e.g., SDNs) and is the focus of this chapter.

In this chapter, we identify a class of stealthy attacks – which we coin pinball attacks – that target DCA algorithms with the goal of inducing conflicting channels to lure the network into a cascading channel switching behavior that increases the levels of interference and instability in the network[1]. Pinball attack policies are the solutions to various Markov Decision Process (MDP) problems in which the attacker aims to maximize their long-term net reward through choosing the right action (e.g., which node to attack and what channel to use for interference). The rewards are composed of a damage component that reflects the degree of interference induced by the attack and a cost component that captures the cost incurred in mounting a pinball attack. Due to the large state space representation, we develop approximation methods that rely on a well-chosen set of features to approximate the value of being in any state of the network. We study two general setups. In the first setup, we consider pinball attacks that induce interference in wireless networks composed of APs that each should be assigned a different frequency from its neighbors. This setup appears in campus buildings, apartment complexes and residential neighborhoods. In the second setup, we consider pinball attacks that induce interference in networks composed of nodes running multiple radio interfaces where every two interfaces on every neighboring nodes should be assigned the same frequency to communicate. This setup arises in various sensor networks and

[1]The attack resembles the pinball arcade game in which the player strikes different targets with a ball using flippers and tries to keep the ball in play by having each target induce a cascading effect of striking other targets.

other applications with automatic resource assignment that can be intentionally triggered to cause instability.

2.2 Related Work

The problem to be investigated in this chapter is related to a large body of research on CA techniques in wireless networks and their security implications. We discuss representative pieces of work that are most related to our work.

2.2.1 Channel Assignment Techniques

CA is instrumental in the design and operation of wireless networks and is done through static and dynamic techniques that seek to minimize interference and improve the wireless spectrum (e.g., [26, 27, 28, 29, 30, 18]). In general, the assignment of channels amounts to a graph coloring problem so channels can be reused efficiently in a non-interfering manner [26]. In [27], the authors develop a CA algorithm for maximizing the signal-to-interference ratio for users connected to access points in a wireless local area network (WLAN). In [28], the authors develop an interference estimation technique based on a multi-radio conflict graph to assign channels to minimize interference within a mesh network. The authors in [29] cast the CA problem in a multi-radio interface setting as an optimization problem. They develop centralized and distributed algorithms through semidefinite and linear programs to derive lower bounds on the overall channel interference. A greedy heuristic CA algorithm in multi-radio mesh networks is developed in [30] to identify connected and low interference topologies. The authors in [18] surveyed DCA algorithms that seek to minimize interference in IEEE 802.11-based WLANs.

The use of SDNs and CRs has allowed for a more efficient spectrum assignment, specially with

the proliferation of the Internet of Things (IoT) and Cyber-Physical Systems (CPS) [31, 32, 33]. For example, vacant channels can be dynamically allocated to secondary users when they are not in use by primary ones.

2.2.2 Security Implications of Channel Assignment

The adaptive nature of DCA techniques has been shown to be vulnerable to various attacks (e.g., [34, 22]. The work in [34] investigates three types of attacks that can potentially lead to a break-down of the CA protocol of multi-interface and multi-channel (MIMC) networks thereby reducing the overall network throughput. Security vulnerabilities in the 802.11 standard were identified in [22], where the authors have shown that a one-minute denial of service (DOS) attack can be achieved with a single message if the attacker forges the channel switch information. Other attacks such as radio interference jamming attacks on wireless networks were investigated in [35], and several intelligent jamming attacks were studied and analyzed in [36]. The work in [37] exposes three types of attacks against CA algorithms that capitalize on attacking the channels with the highest loads. One of those attacks is the Low-Cost Ripple Effect Attack (LORA) that aims to force the network in a quasi-stable state by continuously inducing channel conflicts. On the defense front, the authors in [38, 39, 40] address the control channel jamming attacks and propose a randomized distributed scheme in which nodes can reestablish the control channel using frequency hopping techniques. Furthermore, in [39] the authors introduce two methods for identifying the jammers whether acting individually or colluding.

Within SDNs and CRs, various works have investigated related security issues (e.g., [41, 42, 23]). In [43], the authors show a number of vulnerabilities in MAC protocols due to greedy CR users that attempt to gain more than their fair share of resources. The authors in [44] propose a stochastic game between cognitive attackers and secondary users and develop a minimax-Q learning method

for the users to learn an optimal strategy.

Unlike previous work, we formulate the interference problem as an MDP problem, in which the attacker seeks to find an optimal interference policy – exploiting the topology and the channels used in the network. We apply approximation methods to obtain suboptimal policies due to the intractable nature of the problem [45]. We believe this work shows inherent susceptibility of DCA methods that are becoming critical components in SDNs and CRs.

2.3 System Model

In this section, we present a model of a wireless network employing DCA techniques. In this setup, it is desired that each node (representing an AP in a static topology) uses a channel that does not conflict with any of its neighbors, if possible.

An N-node network is represented by an interference graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i\}, i = 1, \ldots, N$, is the vertex set of APs, and $\mathcal{E} = \{e_{ij}\}, i, j \in \{1, \ldots, N\}$, is the edge set. Each node $v \in \mathcal{V}$ is assigned a frequency channel $c_v \in \mathcal{C}$, where $\mathcal{C}$ is the set of all usable frequencies. For example, in the United States there are 11 usable channels in the 2.4 GHz band for the 802.11n network specifications [46], in which case $\mathcal{C} = \{1, 2, \ldots, 11\}$. The network topology is described by a standard $N \times N$ adjacency matrix, such that any APs connected by an edge cannot use the same or adjacent channels without experiencing interference. Therefore, in this model two nodes are considered adjacent, thus connected by an edge, if they are within each other's interference radius. Later in Section 2.4, we use different graph representations in which edges represent communication links between the nodes.

2.3.1 Channel Assignment Technique

We present a discrete-time interference-aware DCA technique to reduce channel conflicts for a given network topology. We let $\mathcal{N}_v$ denote the neighbors of node $v \in \mathcal{V}$,

$$\mathcal{N}_v = \{u \in \mathcal{V} | e_{vu} \in \mathcal{E}\}\,. \tag{2.1}$$

We also define $\delta_v, v \in \mathcal{V}$, as the degree of node v, i.e., the cardinality $|\mathcal{N}_v|$ of the neighboring set.

In the event of a channel conflict with one or more neighbors, we assume an AP can determine which channels are available to switch to so that it is no longer in conflict with its neighbors. An available channel is one that is in the set of usable channels but not in any of the interference sets of the AP's neighbors. An interference set includes all channels that overlap with the assigned channel based on a channel separation constant. For an AP, $v \in \mathcal{V}$, on channel $c_v \in \mathcal{C}$, the AP's interference set denoted $\mathcal{I}_v$ consists of adjacent channels, i.e.,

$$\mathcal{I}_v = \{\max(c_v - \Delta, 1), \ldots, c_v - 1, c_v, c_v + 1, \ldots, \min(c_v + \Delta, |\mathcal{C}|)\}, \tag{2.2}$$

where Δ is the channel separation constant. For example, for a separation value of 2, a channel c_v will overlap with channels $c_v - 2, c_v - 1, c_v + 1$ and $c_v + 2$. For all examined test cases, we used a channel separation constant $\Delta = 2$. The set of channels an AP v can freely switch to, denoted $\mathcal{L}_v$, is thus the set difference of usable channels and the union of the interference sets of all its neighbors, i.e.,

$$\mathcal{L}_v = \mathcal{C} \setminus \bigcup_{u \in \mathcal{N}_v} \mathcal{I}_u\ , \quad u, v \in \mathcal{V}. \tag{2.3}$$

We also define the conflict set $\mathcal{D}_v$ as the set of nodes in the neighborhood of v that use channels in

its interference set, i.e.,

$$\mathcal{D}_v = \{u \in \mathcal{N}_v : c_u \in \mathcal{I}_v\}. \tag{2.4}$$

Since we can count how many times a given channel appears in an interference set, we can build a histogram $\forall c \in \mathcal{C}$ to find the least interfering channel.

Without loss of generality, we follow the following procedure to assign a new channel to a certain node v, albeit pinball attacks apply to other DCA algorithms. If $\mathcal{L}_v$ is non-empty, i.e., $\mathcal{L}_v \neq \emptyset$, then v is assigned any channel drawn randomly from $\mathcal{L}_v$. Otherwise, v is assigned the *least interfering channel* according to the histogram. Graph coloring is generally NP-hard. Therefore, we do not attempt to obtain a state of minimum conflict. Instead, we adopt a greedy graph coloring approach to channel assignment as described in Algorithm 1.

Before randomly assigning channels to any node, the nodes are sorted in decreasing order according to their node degrees, thereby giving priority to the hub nodes with the higher degrees. Let $\mathbf{v}_{\text{sorted}}$ denote the ordered sequence of nodes.

Algorithm 1 CA based on Greedy Graph Coloring

```
1:  procedure CHANNEL ASSIGNMENT(v_sorted, C)
2:      for v in v_sorted do
3:          Find: L_v, N_v and I_v.
4:          Update: Channel Histogram.
5:          if L_v ≠ ∅ then
6:              Assign v a channel c_v = c, c ∈ L_v, drawn randomly.
7:          else
8:              Choose c_v as the least interfering channel.
9:          end if
10:     end for
11: end procedure
```

Algorithm 1 is a finite step greedy algorithm as it terminates once every node is assigned either a

non-interfering channel or a least interfering channel. At any time stage, the system may trigger this algorithm to resolve a conflict. Based on this algorithm, no channel switching would occur unless it resolves or reduces the network interference as in the DCA protocol applied in the Cisco Radio Resource Management (RRM) system [1]. In the RRM system, the wireless controller collects information from all the APs to estimate the interference and noise levels experienced by each channel, and makes channel switching decisions accordingly.

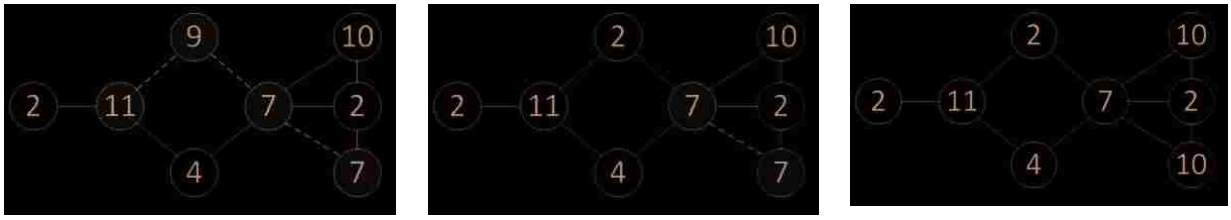

Figure 2.1: Step 1: Initial channel assignments, the network has 3 conflicts. Step 2: After an AP switches from channel 9 to 2, the network has 1 conflict. Step 3: After an AP switches from channel 7 to 10, the network is conflict-free.

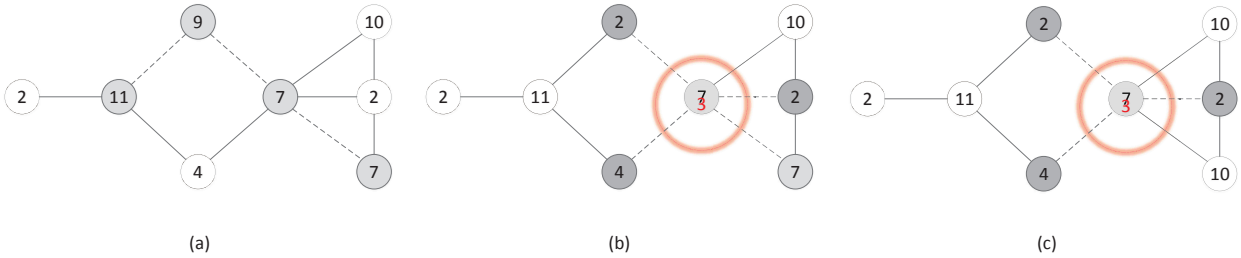

Figure 2.2: Step 1: System starts with 3 conflicts. Step 2: Attacker jams its victim node with jamming channel 3 and the system switched the node using channel 9 to the non-interfering channel 2. However, due to the jammer the new frequency is now suffering interference. Step 3: The system is not only unable to resolve conflicts in 3 steps, but also three nodes may not find any vacant non-interfering channels, resulting in perpetual switching among other nodes as well.

The neighborhood sets $\mathcal{N}_v, v \in \mathcal{V}$, capture the interference map of the network since only neigh-

boring nodes assigned adjacent frequencies interfere. While the generation of the interference map (hence, the underlying topology) depends on the physical distances between the nodes, the channel attenuation, and the used power levels, only the emerging adjacency structure and the actual channel assignment are relevant to the occurrence of conflicts and to channel switching decisions[2]. For example, in DCA protocols, channel switching occurs when the interference levels exceed certain thresholds – and the topology already specifies which nodes interfere at levels that warrant switching. In turn, the adjacency structure and the channel assignment information are sufficient for optimal policy design by an attacker aiming to increase the number of conflicts and induce a cascading channel switching behavior. Therefore, we henceforth use the distance in hops as our distance measure.

2.3.2 Decoy Pinball Attack

An attacker mounting a decoy attack aims to degrade network performance by increasing network instability and delays with unnecessary channel switching, increasing signal interference and congestion by causing more conflicts, and preventing the system from reaching a conflict-free channel assignment if one exists. Since each AP relies on the state of its neighbors to make switching decisions, an attacker can travel to an AP in the network (e.g., by driving to, and taking a place, near a home in a neighborhood) and broadcast a high-power signal that overwhelms the victim AP. Neighboring APs detect and make switching decisions off the fake channel as well. It is also worth mentioning that an attacker who can hijack an AP and switch its channel to a conflicting one can achieve a similar attack behavior and outcome. We do not explicitly consider those hijacking attacks since it is generally easier to broadcast a channel than to hijack an AP. An example of the progression of a system that initially suffers 3 conflicts with $\Delta = 2$ is shown in Fig. 2.1. The

[2]This is not the case if we also consider system dynamics, such as due to power control, or time-varying channel conditions through fading, in which case the topology is time-variant. However, this is not considered in this work.

system can fully resolve all conflicts within 3 steps using Algorithm 1. On the other hand, Fig. 2.2 shows that under pinball attacks, the system fails to reach a zero conflict state within the same number of steps.

Launching an attack does not guarantee that the system will enter into a worse state. One AP may switch its channel per state transition, so even if an attacker creates a fake conflict, an AP involved in a real conflict elsewhere in the network may be selected to switch instead of one of the APs affected by the attack. Additionally, if an affected AP is selected to switch, it will not necessarily switch to a conflicting channel. For this reason, the attacker must weigh the potential benefit of successfully causing damage against the definite cost of launching an attack. We consider the attack cost to be the risk of exposure, which means that the attack cost scales with the network density as measured by node degree. Attacking an AP with a high node degree could possibly impact a large number of surrounding APs, but the attacker runs a higher risk of being caught (e.g., see [47, 48] on localizing jammers based on jamming effect analysis).

2.3.3 MDP Formulation

To maximize the damage caused in comparison to the costs incurred, the attacker must programmatically find a damaging attack pattern by maximizing (resp. minimizing) a reward (resp. cost) function. When attacking a victim, a single attacker must travel to be within an attacking radius from the desired victim AP's location in order to launch the attack. The possible victims available to the attacker at each step depend on the attacker's location, so the travel distance is the limiting factor in possible victims. We use the distance in hops between APs to represent the travel distance for the attacker.

For a decoy attack, an attacker has two possible course of actions: (1) do nothing and (2) broadcast channel $c \in \mathcal{C}$ at the location of an AP to make it appear as though the AP is using that channel.

In this system model, an attacker has knowledge of the network state, i.e., the topology and the currently assigned channels, and can compute additional information such as node degrees and hop distances between the APs.

For an N-node network, the state of the system s_k at time k is represented by the tuple

$$s_k = (\underline{\mathbf{c}}_k\ , n_k,\ b_k),$$

where $\underline{\mathbf{c}}_k$ is a vector of length N containing the channels assigned to each AP at time k, n_k is the AP most recently attacked (if any), and b_k is the number of steps since the last attack. Also, let $a_k = [v, c]$ denote the action of the attacker at time k, which consists of a victim node v and an attack channel $c \in \mathcal{I}_u$, for some $u \in \mathcal{N}_v \cup \{v\}$. The system transitions from s_k to a new state s_{k+1} when an AP switches channels, thus modifying the assignment vector. Being in state s_k, the system evolves to the next state s_{k+1} with a transition probability $p(s_{k+1}|s_k, a_k)$, which is uniform over the set of reachable future states under action a_k taken by the attacker at time k.

As the system transitions from s_k to s_{k+1}, the attacker collects an immediate reward $r(s_k, a_k, s_{k+1})$, which is the sum of conflicts in the reached state s_{k+1},

$$r(s_k, a_k, s_{k+1}) = \sum_{v \in \mathcal{V}} m_{k+1}(v) \tag{2.5}$$

where the sum is over all the nodes in the graph and $m_{k+1}(v)$ is the number of channel conflicts at node v at time $k+1$ defined as

$$m(v) = |\mathcal{D}_v|. \tag{2.6}$$

The cost $\sigma(s_k, a_k)$ of an attack action $a_k = [v, c]$ is

$$\sigma(s_k, a_k) = \begin{cases} h \cdot d(n_k, v) + \delta_v & \text{if } d(n_k, v) \leq b_k + 1 \\ \infty & \text{otherwise} \end{cases} \tag{2.7}$$

where $d(n_k, v)$ is the number of hops between the last attacked AP n_k and the victim node v, h is a constant cost per hop, and δ_v is the node degree of the AP. Thus, mounting an attack incurs a cost that takes into account the distance to the node (it is more costly to attack further away nodes), and the importance of the node being attacked. Hence, this cost model captures the relative importance of the different nodes. In practice, much traffic goes through central APs (hubs), hence disruption of their operation could have a lager impact on the entire network. Therefore, associating a higher cost with vital nodes (having larger node degrees) reflects the fact that attacking such nodes is generally more costly to the attacker. The plausibility of this cost abstraction is evidenced by a good number of recent studies showing that jammers can be better localized based on the effect of their jamming [47, 48], or the increased difficulty in hijacking a central hub normally readied with a more secure defense system. Also, note that the distance to the victim AP must be less than or equal to the number of steps b_k since the last attack plus 1. To clarify, we note that if the attacker can only travel a distance of one hop at each transition, then at each time instant the reach of the attack extends only to the neighboring nodes of its last visited node. Therefore, if the attacker chooses not to mount an attack in a given stage and to save his jamming power, then the set of reachable victim nodes will grow to encompass those nodes within a two-hop distance, and so on. Thus, the infinite cost in (2.7) ensures that infeasible actions (i.e., attacking far away nodes that are not within the reach of the attacker) will not be selected by the attacker. An alternative justification for the cost in (2.7), is that if the attacker chooses not to attack for a number of steps (thereby saving power), then an attack can be launched at future steps at higher power levels (to reach nodes that are further away) without violating his power budget constraint. In other words, the attacker can increase the

radius of the attack based on the time spent without launching an attack. In this case, the infinite cost concerns attacking nodes that cannot be reached without violating the power constraint.

The net reward, denoted $g(s_k, a_k, s_{k+1})$, is the difference between the immediate reward and the cost of the attack. Hence,

$$g(s_k, a_k, s_{k+1}) = r(s_k, a_k, s_{k+1}) - \sigma(s_k, a_k). \tag{2.8}$$

The expected net reward $g(s_k, a_k)$ when transitioning from state s_k to s_{k+1} is

$$g(s_k, a_k) = \sum_{s_{k+1}} p(s_{k+1}|s_k, a_k) g(s_k, a_k, s_{k+1}).$$

The more conflicts an attacker causes over the transition, the higher the path reward, which is the sum of rewards as the system transitions along a discrete Markov chain. The reward earned during a transition from s_k to s_{k+1} is weighted by the probability $p(s_{k+1}|s_k, a_k)$ of transitioning from s_k to s_{k+1}.

In order to find a favorable tradeoff between reward and attack cost, the attacker must identify a sequence of attack policies $\pi = \{\mu_0, \mu_1, \dots, \}$, where $\mu_k : \mathcal{S} \to \mathcal{A}$, is the attack policy at time k, i.e., a mapping from the state space $\mathcal{S}$ to the control space $\mathcal{A}$. An attack policy describes the sequence of actions the attacker should take over some time frame for all sequences of system states. An optimal policy should optimize the tradeoff between the reward and attack cost so the attacker can inflict maximum damage to the system at little expense. This is a discrete-time system with observable state, hence the attacker can solve an MDP problem to identify an attack policy. Since this channel switching model has an infinite horizon, we use a discount factor $0 < \gamma < 1$ to

weight the potential rewards and bias the attacker towards closer rewards. In particular, let

$$J_\pi(s_0) = \sum_{k=0}^{\infty} \gamma^k \, \mathbb{E}\left[g(s_k, \mu_k(s_k), s_{k+1})\right], \tag{2.9}$$

where s_0 is the initial state, $\mathbb{E}[.]$ the expectation w.r.t. the future states (which are unknown at the times of the decisions), and $\pi = \{\mu_0, \mu_1, \ldots, \}$ the attack strategy. The attacker aims to solve for the optimal strategy $\pi^* = \{\mu_0^*, \mu_1^*, \ldots\}$ that maximizes the total discounted expected reward, i.e., solve the optimization problem

$$\underset{\pi}{\text{maximize}} \quad \sum_{k=0}^{\infty} \gamma^k \, \mathbb{E}\left[g(s_k, \mu_k(s_k), s_{k+1})\right] \, . \tag{2.10}$$

The optimal solution $J_{\pi^*}(s_0)$ is the optimal value function $J^*(s_0)$.

Remark 1. *Our formulation assumes that the state of the network is observable to the attacker, wherefore the problem is modeled as a fully observed MDP. The assumption of full observability implicitly assumes worst case damage due to a powerful attacker capable of acquiring global network information by listening to the channels before deciding on the next move. If the attacker only has access to partial state information, then the problem can be modeled as a Partially observable MDP (POMDP) with the main difference that the state is replaced with a posterior probability distribution over the state space (so-called belief) given all past information. In this case, the belief plays the role of the state and the state evolution is replaced by a belief evolution. However, a POMDP formulation does not lead to fundamental changes from a conceptual standpoint.*

2.3.4 Optimal Attack Policy

We have modeled our system as an MDP in which an attacker uses sequential decision-making to navigate discrete system states by defined state transitions. The optimal solution to this problem is

a time-invariant policy μ^* [49] defining the optimal action for the attacker to take at each state, and can be obtained as a solution to the Bellman equation,

$$J(s_k) = \max_{a_k \in \mathcal{A}(s_k)} \{g(s_k, a_k) + \gamma \, \mathbb{E}\left[J(s_{k+1})\right]\}, \tag{2.11}$$

where $\mathcal{A}(s_k)$ denotes the set of allowable controls given the current state s_k.

In order to obtain the optimal attack policy μ^*, the attacker would need to solve a set of linear equations to evaluate the expected reward J_μ of a policy μ

$$J_\mu(s) = \sum_{s'} p(s'|s, \mu(s)) \cdot \left[g(s, \mu(s), s') + \gamma J_\mu(s')\right], \tag{2.12}$$

for $s = 1, \ldots, |\mathcal{S}|$. The summation above is over all states s' reachable from s. Then, the attacker would iteratively improve the policy by solving

$$\bar{\mu}(s) = \arg \max_{a \in \mathcal{A}(s)} \sum_{s'} p(s'|s, a) \cdot \left[g(s, a, s') + \gamma J_\mu(s')\right], \tag{2.13}$$

and then evaluate $J_{\bar{\mu}}$ as in equation (2.12). The attacker iterates over policy evaluation in (2.12) and policy improvement in (2.13) until convergence to the optimal policy that maximizes the reward[3].

The optimal policy obtained as an exact solution describes the optimal action the attacker should choose given any state. For a very small network, an attacker could possibly enumerate every state and select actions that lead to states with the best reward, but since we focuses on a state space similar to a real-world network with N nodes and $|\mathcal{C}|$ channels, the exact solution is intractable. Even if the maximum number of successive no-attack actions is restricted to K, the state space would be of size $|\mathcal{C}|^N \times N \times K$, which grows exponentially with N. Thus, it is computationally

[3]Convergence is guaranteed in this case since the transformation defined through policy iteration is a contraction mapping, hence the iterations converge to a unique fixed point.

intractable for the attacker to evaluate the expected state reward for every possible state and action combination.

2.3.5 Suboptimal Attack Strategy

Instead, we develop suboptimal attack policies using Approximate Policy Iteration (API) based on estimations of the state reward. Rather than examining every possible state, API uses representative states and features. Representative states are used as training states during policy improvement, so they must be selected in such a way that they capture most possibilities for the system and give wide coverage of useful regions in the state space. Recalling the definition of a state s_k, the representative states have to capture all important combinations of the three variables, $\underline{\mathbf{c}}_k$, n_k, and b_k. Since, the initial distribution of conflicts among the network is a function of $\underline{\mathbf{c}}_k$ for a given network topology, we selected representative states containing a spectrum of possible conflict states, from minimal conflicts or a conflict-free assignment to maximum conflicts with all APs assigned the same channel. All APs may start on any default channel, so an experimental run may begin with any number of conflicts. The minimal size of a single conflict is 2 and the maximal size is $\max_{v \in \mathcal{V}} \delta_v$, the largest degree of an AP in the network.

From the representative states, a set of representative features is extracted that capture the characteristics of the state and can be weighted and used to estimate its value. In API, the attacker evaluates an approximate parametric expected state rewards using a weight vector $\underline{\mathbf{r}}$,

$$\tilde{J}_{\underline{\mathbf{r}}}(s) = \sum_{j=1}^{M} r_j \phi_j(s), \tag{2.14}$$

where M is the total number of features, $\phi_j(s)$ denotes the j-th feature for state s, and $r_j, j = 1, \ldots, M$ (the j-th entry of vector $\underline{\mathbf{r}}$) the weight of the j-th feature. Instead of iteratively evaluating

and improving the policy until the iteration loop naturally terminates, API uses Monte Carlo simulations to evaluate the feature weights over a number of independent trajectories from these representative states. More specifically, the weight vector $\underline{\mathbf{r}}$ is obtained as a solution to a least-squares problem minimizing the square error between the empirical average reward of these trajectories and the parametric approximation $\tilde{J}_{\underline{\mathbf{r}}}$ in (2.14). Then, to compute an improved action for state s, we solve

$$\tilde{\mu}(s) = \arg\max_{a} \sum_{s'} p(s'|s,a) \left[g(s,a,s') + \gamma \, \tilde{J}_{\underline{\mathbf{r}}}(s') \right], \tag{2.15}$$

where the value function is replaced by the approximate parametric form $\tilde{J}_{\underline{\mathbf{r}}}$.

2.3.6 Feature Selection

API relies on a set of representative features to capture the characteristics of each state and approximate the state value. For topologies with a diverse range of node degrees, we used the following features:

- ϕ_1 = Number of APs in conflict with one or more neighbors in the current state
- ϕ_2 = Ratio of maximum number of APs involved in the same conflict to degree of the network graph
- ϕ_3 = Average number of APs involved in the same conflict
- ϕ_4 = Average number of channels unavailable to an AP
- ϕ_5 = Average conflict size of the highest degree APs
- ϕ_6 = Last attacked AP
- ϕ_7 = Steps since last attack

- ϕ_8 = Flag for whether attacker is at most complex node (MCN), i.e., node with highest degree
- ϕ_9 = Degree of last attacked location
- ϕ_{10} = Conflicts at last attacked location
- ϕ_{11} = Available channels of last attacked location
- ϕ_{12} = Degree of largest neighbor
- ϕ_{13} = Fraction of APs within hop distance

We also fine tune the features used depending on the network topology. For a ring topology, for example, we omitted features 8, 9, 12 since all APs in a ring have the same degree. To evaluate how well the selected features capture the state characteristics, we compared the estimated path reward computed based on the features to the actual path reward of an attacker over 35 steps and found them to be fairly close. An example is shown in Fig 2.3.

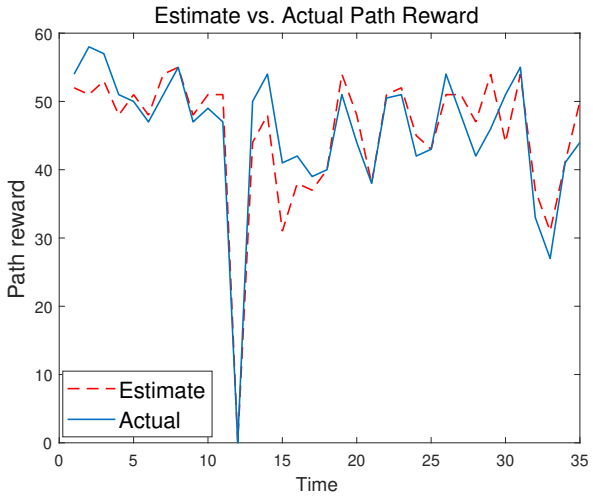

Figure 2.3: Approximate policy and exact policy path reward comparison.

2.3.7 Dimensionality Reduction: Big Graphs

In this subsection, we address the scalability of pinball attacks to larger size graphs. Obtaining an exact solution to the formulated MDP is intractable due to the dimensionality of the state space (increases exponentially in the number of nodes), which motivated the aforementioned development of suboptimal attack strategies based on feature selection and API. While the strategy that describes the mapping of states into actions need only be computed offline, the run time for the offline approximation still depends on the problem size, i.e., the number of nodes of the network. We extend pinball attacks to large size graphs by considering large graphs with repeated patterns.

Attacking Networks of Repeated Structures: We consider a special set of large graphs that have repeated patterns, namely, comprising those graphs with repeated structures of other smaller unit graphs. Since the attacker's control set is limited to the set of nodes in its current neighborhood, exploiting the graph structure and the attacker's limitations leads to a simplified approach to attacking larger graphs.

In this case, we propose to obtain the pinball policy (offline) for the small graph, which is then used by the attacker to launch his attack on the full scale graph. As such, the attacker obtains a pinball strategy by solving a small-size MDP for the unit graph in lieu of solving an MDP with a large state space for the full graph. Then, the attacker uses this policy to attack the full graph taking only the local neighborhood into consideration at every time step.

We have tested this approach on some synthesized large graphs with repeated structures of the topologies considered initially in Section 2.3. The numerical results using this approach yield larger rewards for the attacker compared to those obtained by other attack policies as shown in Section 2.5.8. This approach has also shown to be work well when tested on large size networks with arbitrary structures that can be approximated to large networks with repeated structures as in

the example discussed in Section 2.5.9.

2.4 The Dual Problem

In this section, we consider a dual problem to the one presented in Section 2.3. In particular, we consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in which each node is equipped with multiple radio interfaces. An edge between two nodes represents a communication link between their radio interfaces and a certain frequency is assigned to this edge. Assuming link separation[4] (i.e., a link only sees its assigned frequency), interference occurs when two interfaces on a single node use the same channel frequency, i.e., two adjacent edges are assigned the same frequency. This link separation could be achieved, for example, by using directional antennas to establish communication links between the radio interfaces in a wireless setting. Here, we show that both problems are in fact equivalent, wherefore pinball attack policies identified in the former setup can be mapped into attack policies in this setup.

2.4.1 Equivalence Formulation

For each edge e_{ij}, we define a set of neighboring edges, $\mathcal{N}_{e_{ij}}$, that share any of the end points of e_{ij},

$$\mathcal{N}_{e_{ij}} = \{e_{iu} | i, u \in \mathcal{V}\} \bigcup \{e_{vj} | v, j \in \mathcal{V}\} .$$

For simplicity we refer to any arbitrary edge in the network as e with no subscripts. Similar to

[4]This assumption could be relaxed by expanding on the notion of neighboring and interference sets to account for other sources of interference.

the first setup, we let the set $\mathcal{L}_e$ denote the channels available for edge e and let $\mathcal{I}_e$ denote its interference set as for the primary node attacking problem in Section 2.3. DCA uses the procedure described earlier in Algorithm 1 but with respect to the set of edges $\mathcal{E}$.

The attacker attacks the network edges by causing interference to the communication links, thereby prompting the respective radio interfaces of the end nodes to switch their communication channels. The edge attacking problem can also be formulated as an MDP with the objective of causing the maximum damage to the network.

Let s_k denote the state of the network at time k. Specifically, $s_k = (\underline{\mathbf{c}}_k, \ell_k, b_k)$, where $\underline{\mathbf{c}}_k$ is a $1 \times |\mathcal{E}|$ vector representing the channels assigned to every edge in the network at time k, ℓ_k is the last attacked edge, and b_k denotes the number of successive no-attack actions at time k as defined earlier. Being in state s_k, the system evolves to the next state s_{k+1} with a transition probability $p(s_{k+1}|s_k, a_k)$ under action a_k taken by the attacker at time k, where $p(s_{k+1}|s_k, a_k)$ is uniform over all possible future states. The attacker gains a reward and incurs a cost for taking this action a_k. For the sake of establishing the equivalence, let $m_k(e)$ also denote the number of conflicts at time k in the network with respect to edge $e \in \mathcal{E}$, and y_k be the cost of an action $a_k = [q, c]$ when the attacker attacks edge $q \in \mathcal{E}$ with jamming frequency c. Hence,

$$m(e) = |\{\bar{e}|c_e \in \mathcal{I}_{\bar{e}}, \bar{e} \in \mathcal{N}_e\}|, \ e \in \mathcal{E}, \tag{2.16}$$

where c_e is the channel assigned to link e at a given time. In other words, $m(e)$ is the number of edges adjacent to e that have c_e in their interference sets. The reward, $r(s_k, a_k, s_{k+1})$, is defined as the difference between the total number of conflicts and the cost of the attack,

$$r(s_k, a_k, s_{k+1}) = \sum_{e \in \mathcal{E}} m_{k+1}(e) - \sigma_k. \tag{2.17}$$

The Attack cost σ_k is defined as,

$$\sigma_k = \begin{cases} \infty & \text{if } d(\ell_k, q) > b_k + 1 \\ h \cdot d(\ell_k, q) + \zeta_q & \text{otherwise,} \end{cases} \tag{2.18}$$

where $d(\ell_k, q)$ is the number of hops between the current edge location ℓ_k and the victim edge q, and h is some constant capturing the cost per unit distance. The edge degree ζ_q is taken into consideration for cost computation since a higher degree implies a higher risk of attack. Hence, the expected reward over all possible future states is

$$\bar{r}(s_k, a_k) = \sum_{s_{k+1}} p(s_{k+1}|s_k, a_k) r(s_k, a_k, s_{k+1}). \tag{2.19}$$

Again, we consider a discounted cost MDP formulation, where the attacker aims to solve for the optimal policy that maximizes the total discounted expected reward [45]. In particular, let

$$J_\pi(s_0) = \sum_{k=0}^{\infty} \gamma^k \bar{r}(s_k, \mu_k(s_k)), \tag{2.20}$$

be the total discounted expected reward with equivalent definitions for the strategy π and the discount factor γ as before. The attacker aims to solve the optimization problem

$$\underset{\pi}{\text{maximize}} \quad J_\pi(s_0)\ . \tag{2.21}$$

We can readily state the following Theorem, which establishes equivalence between the node and edge attack formulations.

Theorem 1. *The edge attack problem for a network with graph $\mathcal{G}_a(\mathcal{V}_a, \mathcal{E}_a)$ is equivalent to the node attack problem for the network graph $\mathcal{G}_b(\mathcal{V}_b, \mathcal{E}_b)$ for ${}^{\mathrm{e}}\mathbf{H}_a = \mathbf{H}_b$, where $\mathbf{H}$ is the node adjacency matrix of a graph and ${}^{\mathrm{e}}\mathbf{H}$ the edge adjacency matrix.*

Proof. If ${}^{\mathrm{e}}\mathbf{H}_a = \mathbf{H}_b$, then $\mathcal{V}_b = \mathcal{E}_a$, which means that $\mathcal{G}_b$ is the line graph of the original graph $\mathcal{G}_a$ [50]. Therefore, the line graph of a graph preserves the adjacency between the edges. We start by showing that $\forall\theta \in \mathcal{V}_b$ and $\forall e \in \mathcal{E}_a$, the neighboring set $\mathcal{N}_\theta = \mathcal{N}_e$ for $\theta = e$. Since the two graphs are connected, then $\mathcal{N}_\theta$ and $\mathcal{N}_e$ are non-empty. Given the condition in the statement of Theorem 1, the line graphs of $\mathcal{G}_a$ and $\mathcal{G}_b$ are isomorphic.

Hence, $\forall\theta \in \mathcal{V}_b$ we can define the neighboring set $\mathcal{N}_\theta$ as

$$\mathcal{N}_\theta = \{\lambda | e_{\theta,\lambda} \in \mathcal{E}_b\}. \tag{2.22}$$

Moreover, $\delta_\theta = \zeta_e$ since $|\mathcal{N}_\theta| = |\mathcal{N}_e|$, where δ_θ is the node degree and ζ_e the edge degree for $\theta \in \mathcal{V}_b$ and $e \in \mathcal{E}_a$.

From equations (2.5), (2.6) and (2.7), the attacker's reward depends only on the neighboring sets. Noting that Algorithm 1 for both graphs uses random assignment for frequencies drawn from the set of available channels, its randomness does not affect the number of conflicts associated with any node or edge. Thus, both the node and edge attack problems have the same control set. Therefore, the reward maximization problem associated with the attack of the nodes of $\mathcal{G}_b$ is equivalent to the one associated with the attack of edges of $\mathcal{G}_a$.

□

As an example of the mapping between both versions of the problem, consider the network de-

picted in Fig. 2.4, whose node and edge adjacency matrices $\mathbf{H}_a$ and ${}^{\mathbf{e}}\mathbf{H}_a$ are given by

$$\mathbf{H}_a = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad {}^{\mathbf{e}}\mathbf{H}_a = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix},$$

giving rise to the topology depicted in Fig. 2.5. Hence, the edge attack problem based on the topology in Fig. 2.4 is mapped to an equivalent node attack problem with the topology in Fig. 2.5 that can be solved using the framework described earlier in Section 2.3.

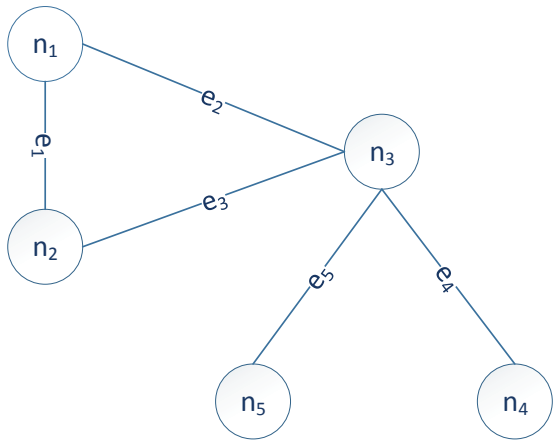

Figure 2.4: Edge attack topology.

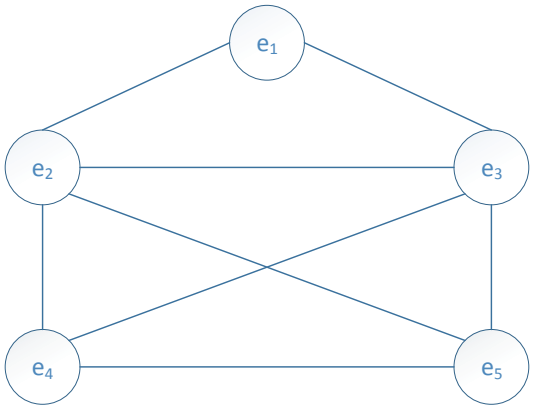

Figure 2.5: Edge attack topology mapped to node attack problem.

2.4.2 Attacking Isomorphic Graphs

It follows that we can establish equivalence of the attack problem on any two isomorphic graphs since the attacker's reward is only dependent on the network adjacency. We state the following result.

Theorem 2. *Let* $\mathbf{G} = \{\mathcal{G}_i(\mathcal{V}_i, \mathcal{E}_i)\}$ *be a set of isomorphic graphs. If* $\mathcal{G}_a(\mathcal{V}_a, \mathcal{E}_a)$ *and* $\mathcal{G}_b(\mathcal{V}_b, \mathcal{E}_b) \in \mathbf{G}$*, then the node attack problems on* $\mathcal{G}_a$ *and* $\mathcal{G}_b$ *are equivalent.*

Proof. The proof follows the same lines of the proof of Theorem 1. The graphs $\mathcal{G}_a$ and $\mathcal{G}_b$ are isomorphic, hence by definition there exists a bijective function $f : \mathcal{G}_a \mapsto \mathcal{G}_b$ such that two nodes u and v are adjacent in $\mathcal{G}_a$ iff $f(u)$ and $f(v)$ are adjacent in $\mathcal{G}_b$. Therefore, $\mathcal{N}_u = \mathcal{N}_{f(u)}$. The attacker's reward (i.e, the objective function) is thus common to both graphs since it is function of the neighboring sets $\mathcal{N}_u$. Also, the set of available controls depends on the adjacency of the node of the attacker's location, hence the control set is the same for both graphs $\mathcal{G}_a$ and $\mathcal{G}_b$. Therefore, the reward maximization problems for both graphs are equivalent. □

In the next section, we present our extensive numerical analysis and evaluation to the proposed pinball attacks.

2.5 Experimental Results

In this section, we study the performance of the proposed pinball strategies and provide comparisons to other attack policies via extensive numerical experiments. Several network topologies are considered in our numerical evaluation including building-block topologies as shown in Fig. 2.6, 2.7, 2.8, 2.9. Moreover, we tested pinball attacks in a real commercial building topology as shown in Fig. 2.14. During policy iteration, we tested a range of attack costs for a channel separation constant $\Delta = 2$. To learn the approximate pinball attack via policy iteration, 35 representative states have been considered for each of the tested topologies.

To evaluate the performance of the proposed pinball policy, we compare its performance to other attack policies in terms of the average path reward accrued over 30 or more time steps. We also quantify the number of conflicts that each attack policy is able to create over time. The attack policies are (1) No-attack; (2) MCN (Most complex node); (3) Rand (Random) ; (4) DoS (Denial of service); (5) Myopic; (6) Pinball.

- In a no-attack policy, the attacker decides not to attack any node at every step. This is the lowest cost attack policy, and hence serves as a baseline policy against which all other attacking policies are compared. Since a network may start with conflicts and resolve itself over time, the attacker may see a non-zero path reward without expending any effort.

- In an MCN attack, the attacker always chooses to attack the node with the maximum degree. The attacker takes a greedy approach in selecting the most complex node within the interference radius of the largest number of APs. By selecting the highest degree node, the attacker

can potentially disrupt the maximum number of nodes with a single attack. If there is more than one AP with highest degree, the attacker picks the one with the shortest hop distance from its current location. Once at the victim AP, the attacker ignores any costs incurred and constantly attacks at every step with a guaranteed conflicting channel.

- In a random attack, the attacker naively selects any reachable AP at random and also chooses at random between broadcasting a channel or doing nothing. Since the random policy allows the attacker to withhold attacks at random, the attacker may miss critical attack opportunities and allow the system to resolve itself.

- In a DoS attack, the attacker selects any random reachable AP and broadcasts an interfering channel at every step. In a DoS policy, the attacker never chooses to refrain from attacking, but attacks at every step without considering the attack cost. This type of attack ignores the elevated cost stemming from constantly attacking in hope of inflicting substantial damage. However, since the victim nodes are picked at random, the attacker may be expending energy needlessly when there are more rewarding victims to attack.

- A myopic attack takes only the immediate reward into account and not the future value of possible future states, thus it selects victim nodes without considering the features of the potential states. By contrast, the pinball attack uses the features of the state and their weights computed during policy iteration to judiciously select an attack that would lead to more rewarding states.

For the 6-node chain network shown in Fig. 2.6, we tested a range of values for the scaling constant h in (2.7) to observe the attacker's behavior with high and low attack costs shown in Fig. 2.10. As expected, the path reward for baseline no-attack remained constant across all attack costs. When the attack cost is very low, the behavior of the pinball attacker approaches that of the DoS, while at a very high attack cost, the behavior approaches that of no-attack. As the attack cost increases, the

path reward of the DoS and Rand policies decreases, dropping well below that of no-attack for very high attack costs. For all values – even very low attack costs – the path reward of the pinball policy is higher than that of DoS. While DoS and Pinball are both constantly attacking at low attack costs (they launch the same number of attacks), Pinball could outperform DoS since it maximizes the inflicted damage by attacking intelligently balancing the potential of immediate reward and future states. In all the tested topologies, the attacker adopting the pinball policy is able to achieve the highest path reward.

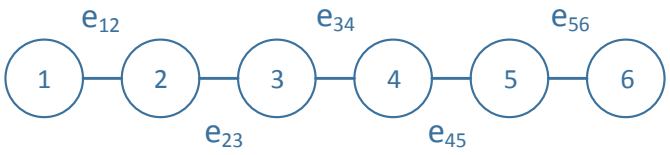

Figure 2.6: A 6-node chain network topology.

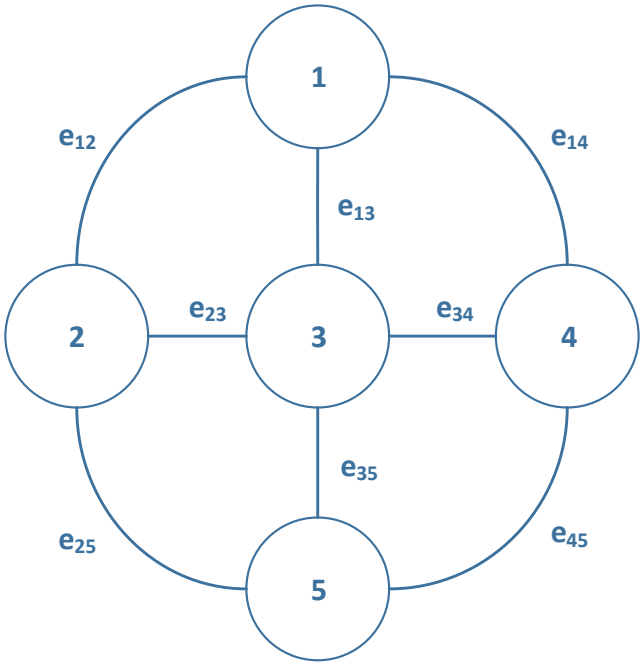

Figure 2.7: 5-node interconnected network topology.

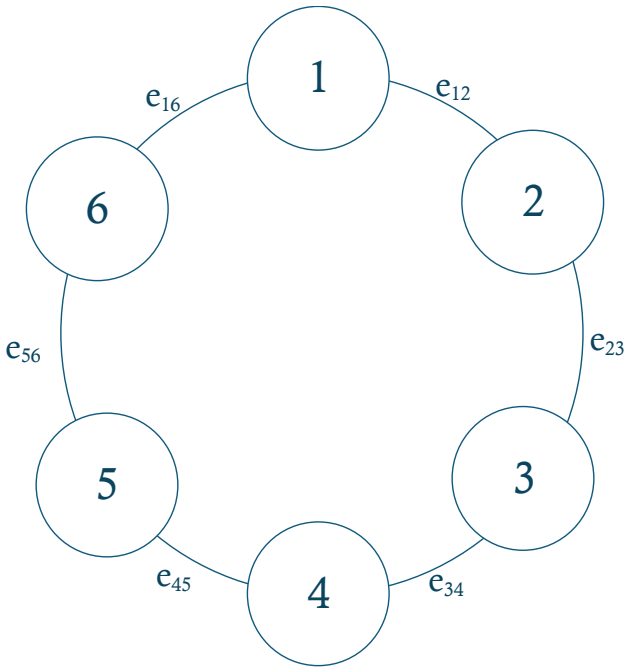

Figure 2.8: A 6-node ring network topology.

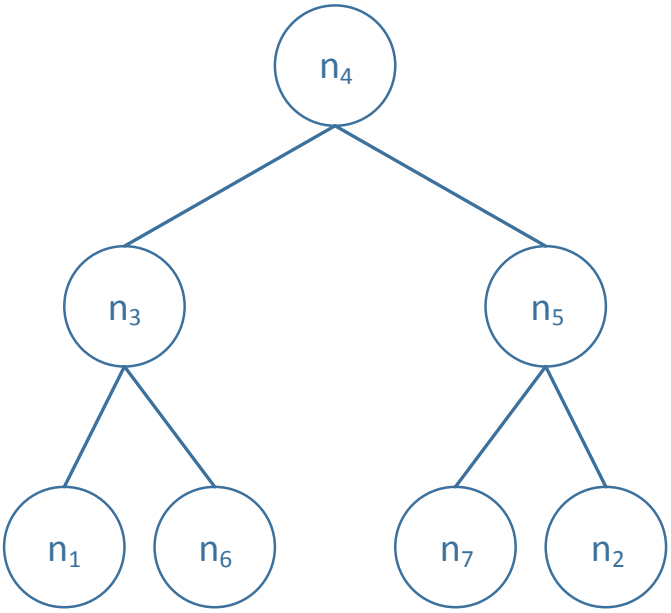

Figure 2.9: A 7-node tree network topology.

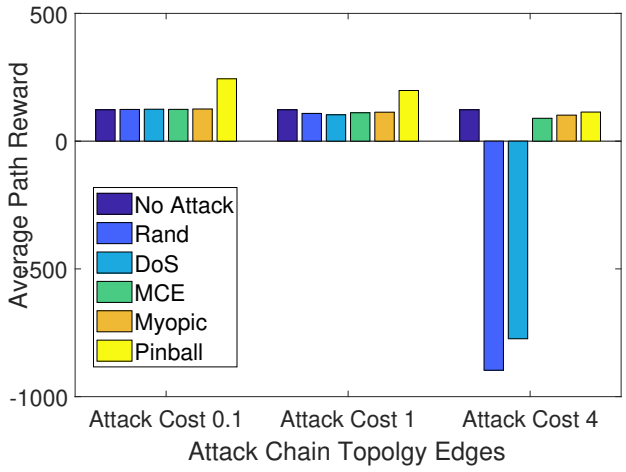

Figure 2.10: Comparing different attack policies on the 5-node chain topology.

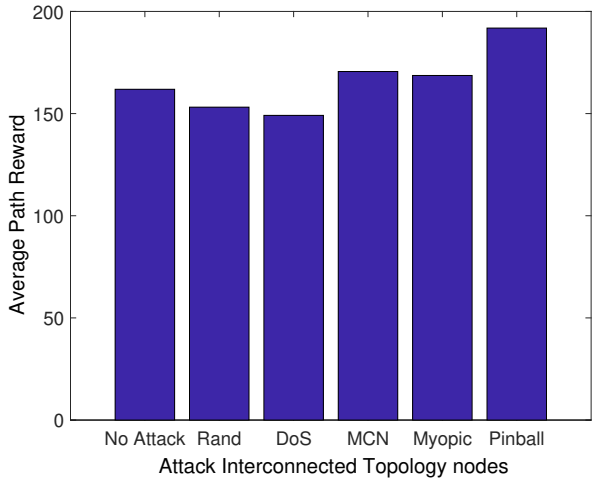

Figure 2.11: Comparing different attack policies on the 5-node interconnected network topology.

2.5.1 Average Path Reward

In our numerical evaluations of the proposed attack policy, we first quantify the main objective of the optimization problem, i.e., the average path reward for the attacker. Second, we quantify and

present the accumulated conflicts created by the attacker over 30 time steps for different policies to capture the potency of the attack. We then analyze and present the frequency of node attacks to understand the behavior of the proposed pinball policy. We also present the time evolution of the conflicts, as well as the conflict size distribution.

In Fig. 2.10, the average path reward is plotted for the 6-node chain topology of Fig. 2.6. The pinball policy always outperforms all other attack policies and yields the highest average path reward. The average path reward is plotted for different attack costs. As the attack cost increases from 0.1 to 4, DoS and Rand become very inefficient. At low attack costs, attacking any of the feasible nodes has low cost, hence the pinball policy chooses to attack at every step. The pinball policy yields higher path reward by directing attacks to the most rewarding nodes. Even though the attack cost is low, an aggressive policy like DoS is unable to achieve a higher average reward, as it occasionally misses rewarding actions depending on the network state at various times. At higher attack costs, the pinball attacker avoids high-cost actions. The attacker is better off withholding attacks due to the high risk of exposure, hence the average path reward converges to that of the no-attack policy. Since the Myopic policy also considers the immediate costs and rewards, it avoids the negative path reward by choosing not to attack as well.

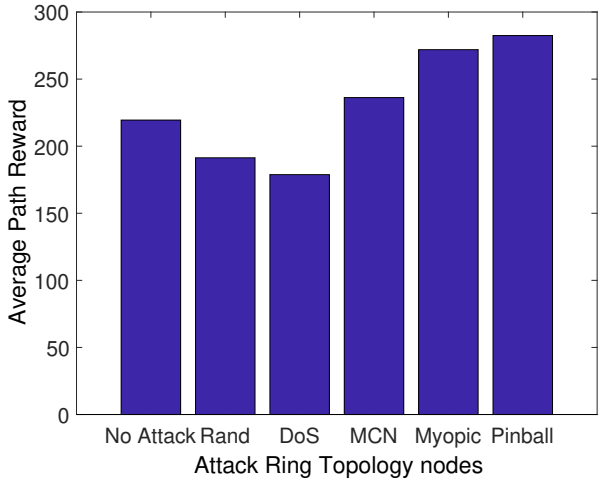

Figure 2.12: Comparing different attack policies on the 5-node ring topology.

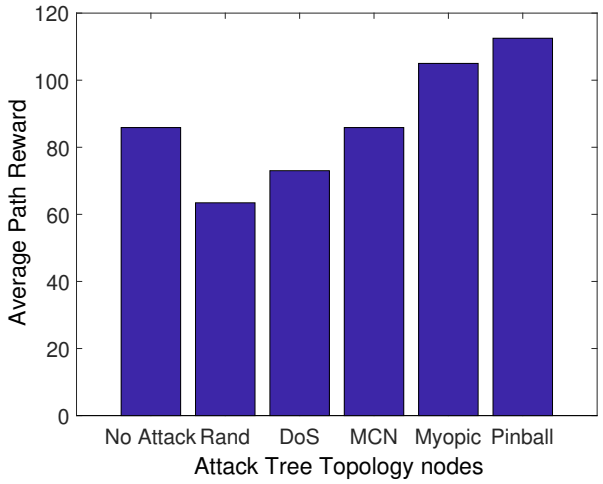

Figure 2.13: Comparing different attack policies on the 7-node tree topology.

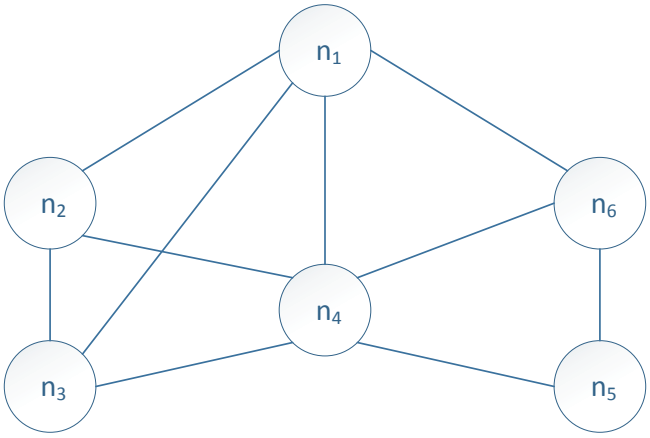

Figure 2.14: A 6-node topology representing the network of the tested building.

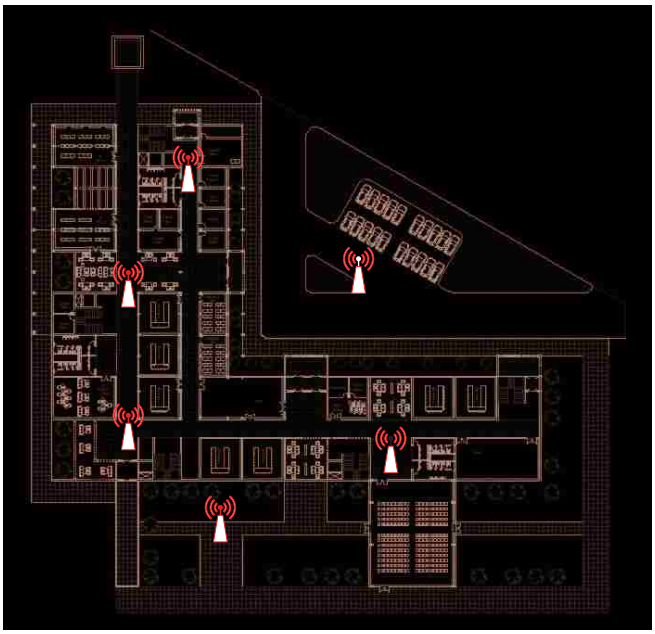

Figure 2.15: Plan view of the tested building.

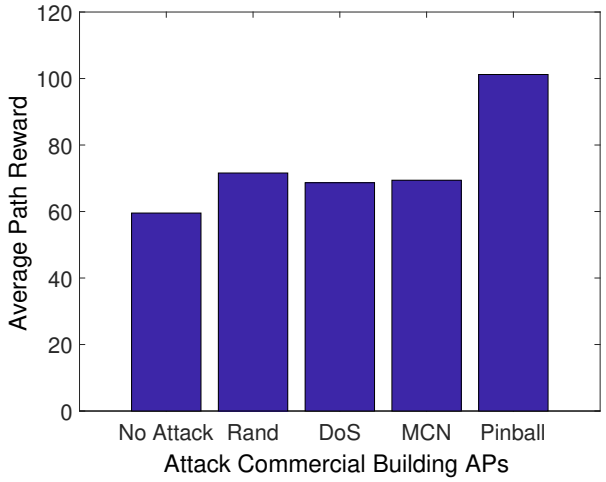

Figure 2.16: Comparing different attack policies on commercial building APs.

We evaluate the average path reward for the topologies shown in Figs. 2.7, 2.8, 2.9, and the topology in Fig. 2.14 representing the interference graph of the APs in the building shown in Fig. 2.15. As for the previous topology, the pinball policy outperforms all other policies in terms of average path reward as shown in Figs. 2.11, 2.12, 2.13 and 2.16, respectively.

2.5.2 Conflict Analysis

We also plot the number of conflicts accumulated over time for different topologies. In Figs. 2.17, 2.18 and 2.19 we show the conflicts accumulated in the network over 30 time steps for the 7-node tree topology, the 5-node interconnected topology and the 8-node topology of Fig. 2.20, respectively. For all these topologies, the pinball policy clearly creates more conflicts than the other policies. Adopting Pinball, the attacker is able to create a larger number of conflicts at the beginning of the attack as evident from the larger slope of the shown curve designating a higher rate of conflict generation. The attacker is taking advantage of the fact that the system is initially

experiencing a medium to large conflict size by creating more compounded conflicts. As time elapses, the system reassigns channels to the nodes reducing the number of conflicts while the attacker continues to induce new ones. The ability of the system to resolve conflicts with time is manifested by the reduced slope. For policies other than the pinball attack, the system is more able to resolve conflicts over time.

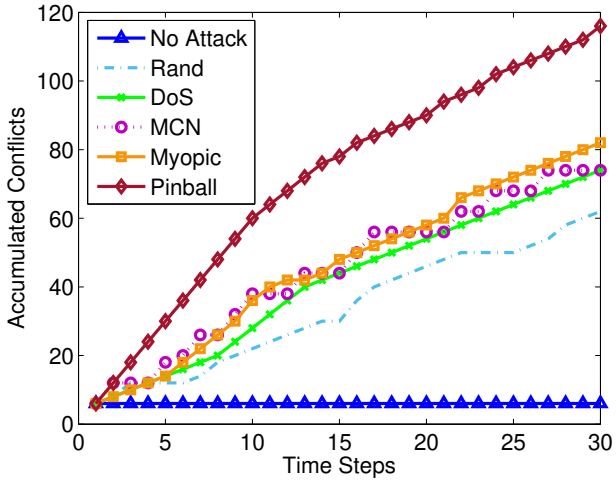

Figure 2.17: Accumulation of conflicts in the 7-node tree network for different attack policies.

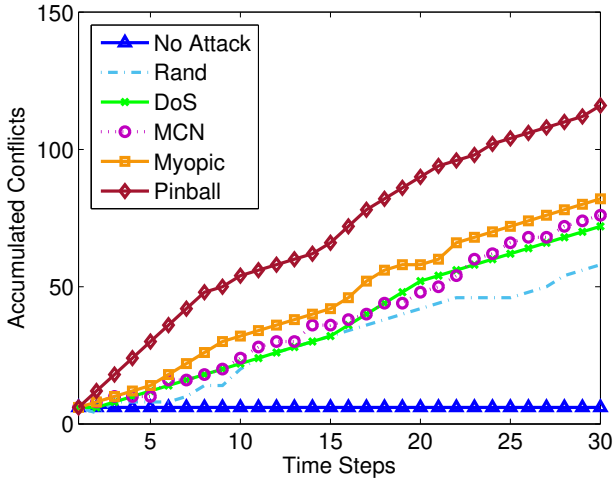

Figure 2.18: Accumulation of conflicts in the 5-node interconnected network for different attack policies.

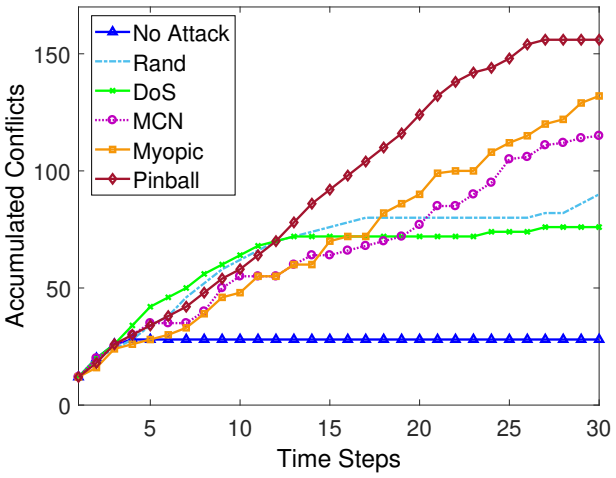

Figure 2.19: Accumulation of conflicts in the 8-node mixed network topology.

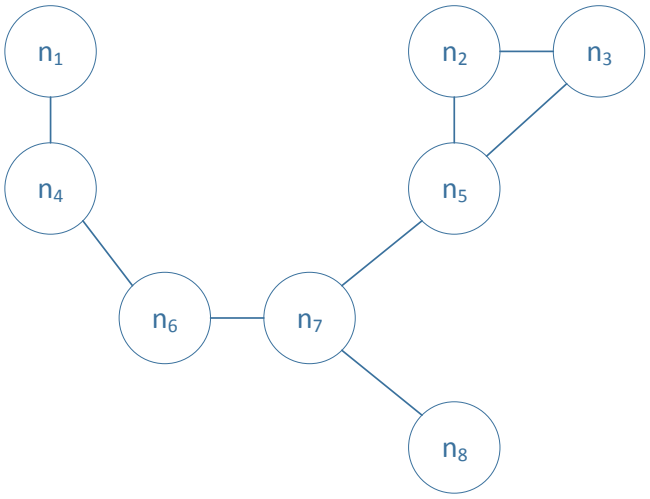

Figure 2.20: An 8-node mixed network topology.

2.5.3 Node Attack Frequency Analysis

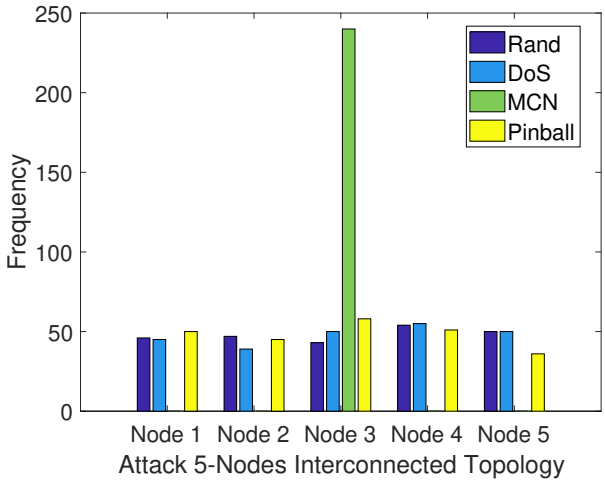

Figure 2.21: Frequency of attack actions for the interconnected network topology.

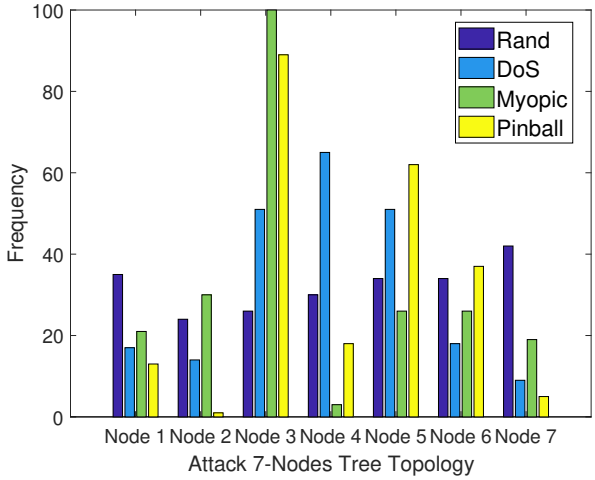

Figure 2.22: Frequency of attack actions for the tree topology.

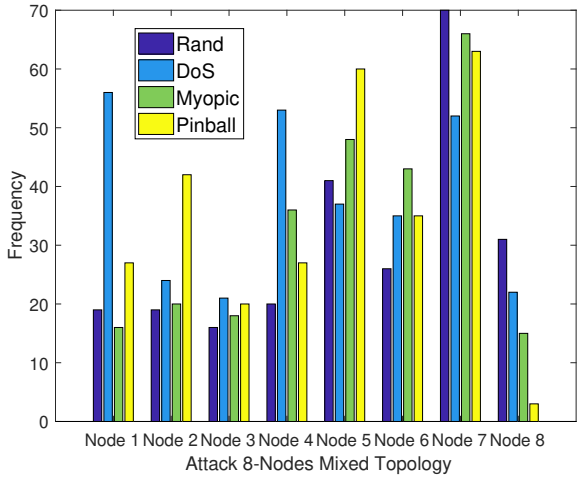

Figure 2.23: Frequency of attack actions for the 8-node mixed network topology.

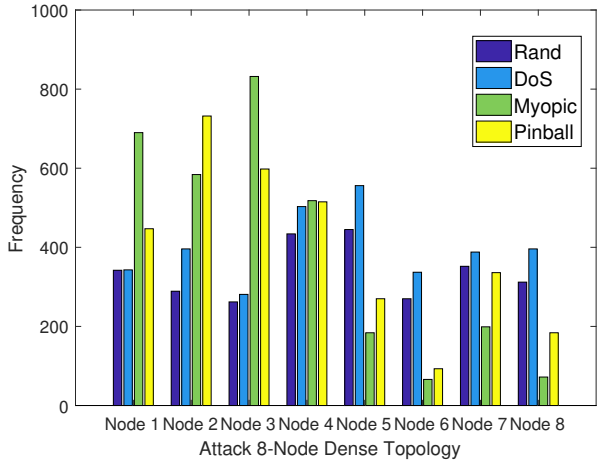

Figure 2.24: Frequency of attack actions for the 8-node dense network topology

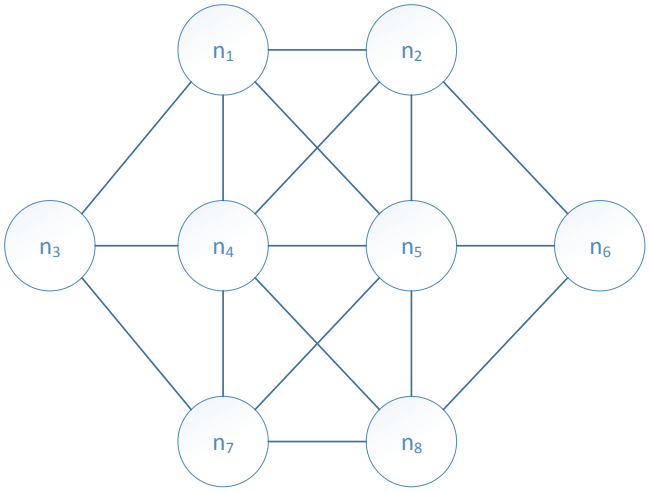

Figure 2.25: An 8-Node dense network topology.

Next, we plot the frequency of attacking different nodes in Figs. 2.21, 2.22, 2.23 and 2.24 for the topologies of Figs. 2.7, 2.9, 2.20 and 2.25, respectively. For the ring topology, all the nodes have equal importance on average (similar degree). In Fig. 2.21, although most of the nodes

were attacked at similar rates, the center node was attacked slightly more often. By comparison to the MCN policy (which only attacks the center node), it is evident that attacking the most complex node does not always yield the highest reward to the attacker. In addition, from Fig. 2.18 and Fig. 2.21, we see that while the MCN policy is always costly for the attacker, it fails to create as many conflicts as the pinball policy that balances the attack cost and rewards to achieve higher path reward. For the tree topology, the attacker utilizing the proposed pinball policy focuses most attacks on the two nodes in the second level with the higher degrees rather than attacking the leaf nodes. The same observation can be seen in Fig. 2.23, showing that the pinball policy attacks internal nodes more frequently than leaf nodes. But, it balances between attacking the most connected nodes and other internal nodes to avoid very costly actions. In Fig. 2.24, we plot the node attack frequency for the 8-node dense network illustrated in Fig. 2.25. Due to the structural similarity of most of the nodes, the pinball attack policy does not exhibit preferences for particular nodes or set of nodes. However, we have observed that the node attack frequency is more skewed towards the initial location of the attacker. Hence, for very dense topologies the attacker chooses to attack closer nodes more frequently rather than attacking farther away nodes with similar connectivities.

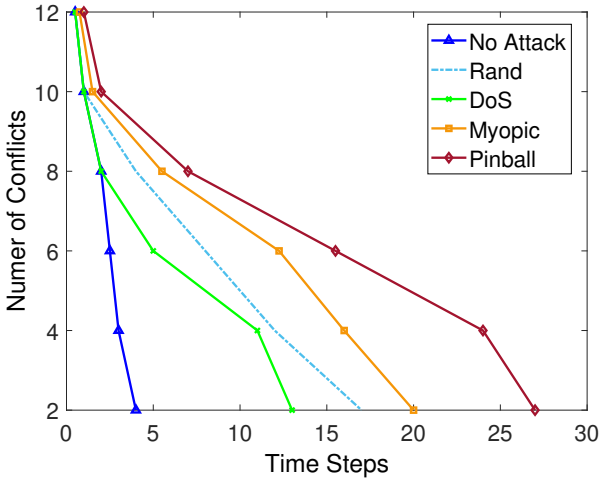

Figure 2.26: Conflicts resolution over time for the 8-node mixed network topology.

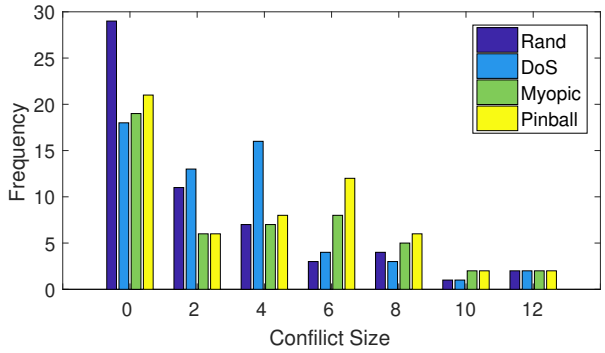

Figure 2.27: Frequency of different conflict sizes in the 8-node mixed network topology.

2.5.4 Conflict Decay Analysis

We select the 8-node Mixed topology of Fig. 2.20, which we purposely synthesized to have a variation of node connectivities, to study the pinball policy more closely. We show the conflict decay over time in Fig. 2.26 and make the following observations. First, the pinball policy is

the most capable at delaying the system resolution to a no-conflict state. Normally, the system takes 4 time steps to completely resolve its initial conflicts (under no-attack). It takes 13 and 17 time steps to resolve the conflicts under DoS and Rand attacks, respectively. In the pinball attack case, the system reaches a no-conflict state after 27 time steps. Although the attack policies will continue to build and induce new conflicts, these are not shown here as this experiment is focused on evaluating the ability of the network to resolve conflicts measured by the decay rate under various attacks. This experiment underscores that the proposed pinball attack is more damaging.

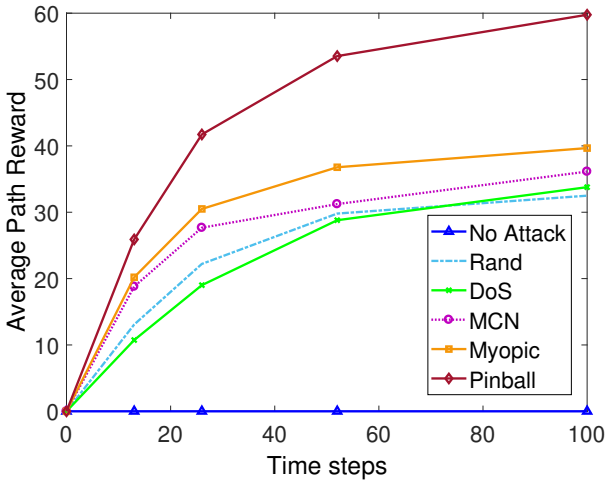

Figure 2.28: Path reward gained over time when 8-nodes mixed network topology starts from a zero-conflict state

2.5.5 Conflict Size Distribution Analysis

We study the conflict size distribution under different attack policies in Fig. 2.27. As shown, the pinball policy tends to create large-sized conflicts more frequently than smaller ones. Taking actions that cause larger conflicts explains the ability of Pinball to yield higher path rewards.

2.5.6 *Conflict Induction and Accumulation Analysis*

We also investigate the conflict induction and accumulation rates under different attack policies when the network starts from a state of zero initial conflicts. As shown in Fig. 2.28, not only is the pinball policy more capable at exploiting the large number of initial conflicts (as shown before), but also it outperforms the other policies in inducing conflicts and achieving higher net path rewards over time.

2.5.7 *Graph Connectivity Analysis*

We also tested the proposed pinball attack policy on six different network topologies, each of which is a 10-node graph, as we change the degree of connectivity between the 10 nodes. Examples for the 10-node graph with different connectivities are shown in Fig. 2.29 and 2.30. Fig. 2.31 shows the average path reward versus the average node degree δ_{avg} used as a connectivity metric. We chose the average node degree as it captures the average size of the neighboring set for each node. Increasing the degree of connectivity, the attacker is able to achieve higher rewards as the neighboring set for each node is getting larger allowing the attacker to create more conflicts under any of the attack policies. However, the pinball policy outperforms all the other strategies for low and moderate degrees of connectivity. For highly connected networks, we observe that the average path reward of Rand and DoS are close to that of Pinball as we approach a fully connected graph, since all actions inflict maximum conflict size when all nodes are more or less connected. Real networks are not typically fully connected, in which case the pinball strategy has clear advantages over the other strategies.

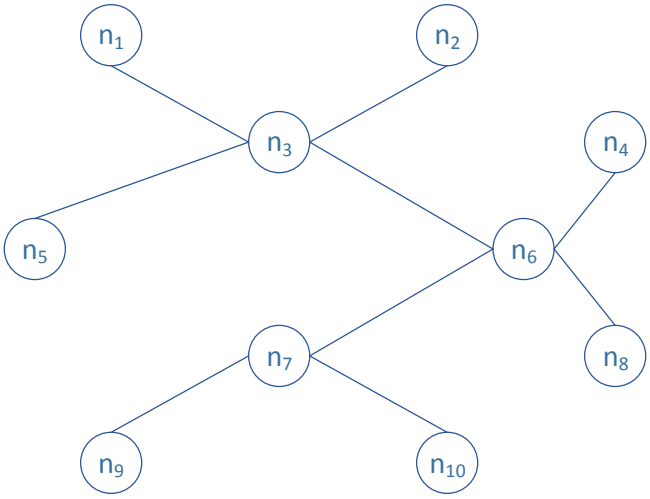

Figure 2.29: A 10-node graph with $\delta_{\text{avg}} = 1.8$.

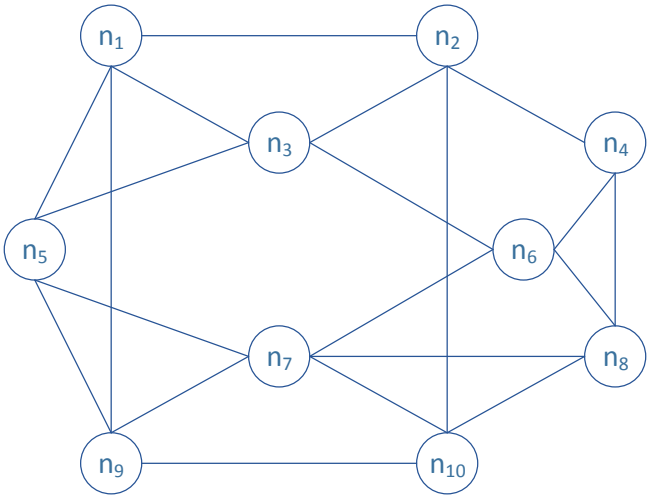

Figure 2.30: A 10-node graph with $\delta_{\text{avg}} = 4$.

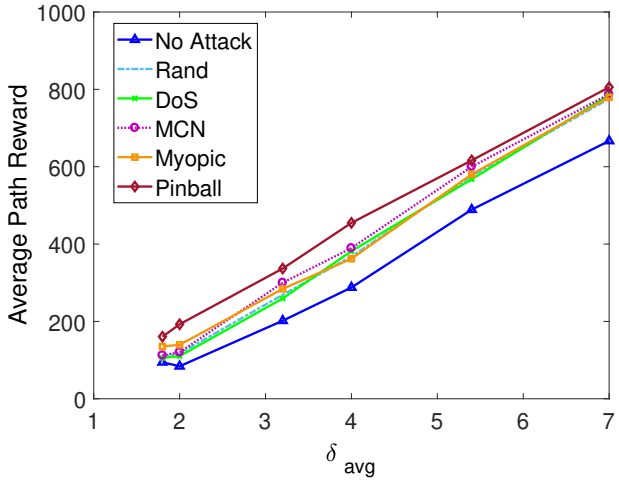

Figure 2.31: Comparing the performance of different attack policies as we increase the connectivity of a 10-node graph.

2.5.8 Attacking Networks of Repeated Structures

We generated three large graphs by repeating the ring, the 5-node interconnected graph and the tree topologies considered earlier as depicted in Figs. 2.32, 2.33 and 2.34, respectively. For instance, Fig. 2.32 illustrates a repeated graph structure with a basic unit graph of a 6-node ring topology. We adopt a pinball policy derived from the solution of the MDP corresponding to the small ring network, thereby alleviating the prohibitive computational complexity of computing a policy for the full scale graph.

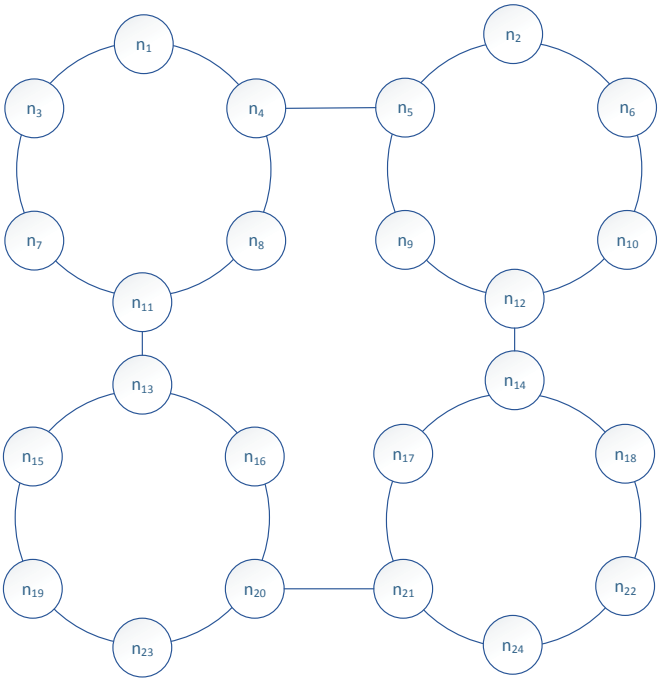

Figure 2.32: Graph with repeated ring structure.

Our results show that the pinball policy obtained from the small subgraph applied to the large graph yields higher rewards than the other policies. This is shown in Figs. 2.35, 2.36, and 2.37 for the three cases.

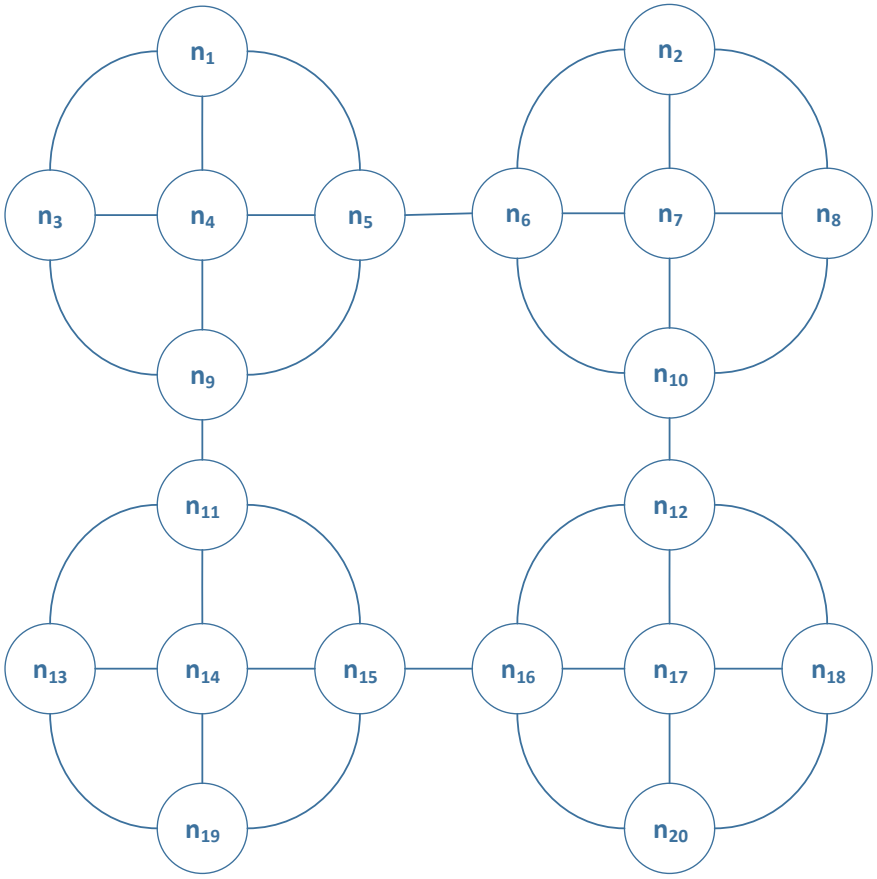

Figure 2.33: Graph with repeated interconnected subgraph structure.

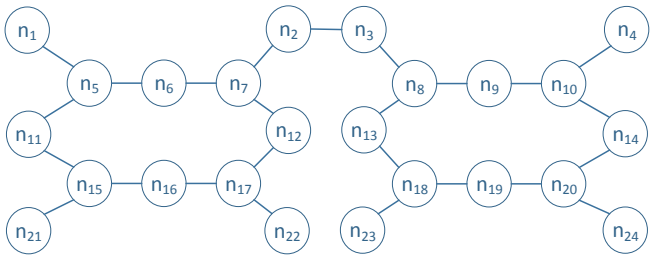

Figure 2.34: Graph with repeated tree structure.

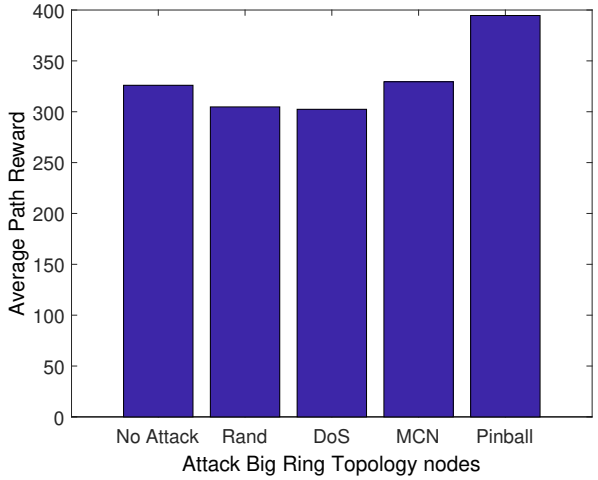

Figure 2.35: Comparing attack policies for graph with repeated ring structure.

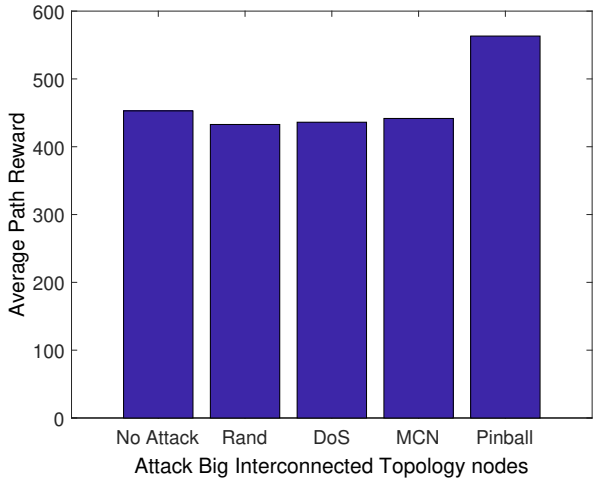

Figure 2.36: Comparing attack policies for graph with repeated interconnected subgraph structure.

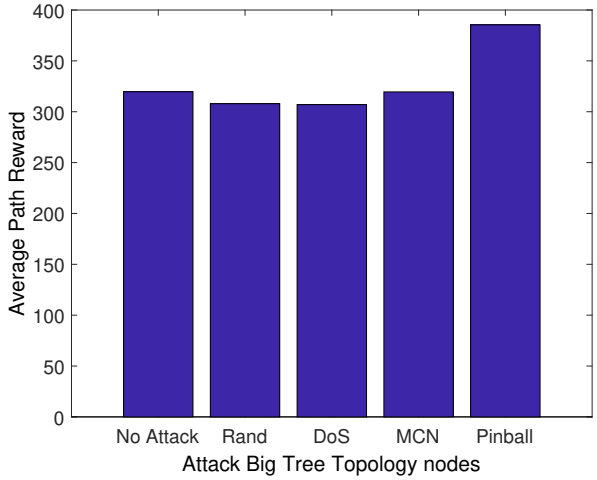

Figure 2.37: Comparing attack policies for graph with repeated tree structure.

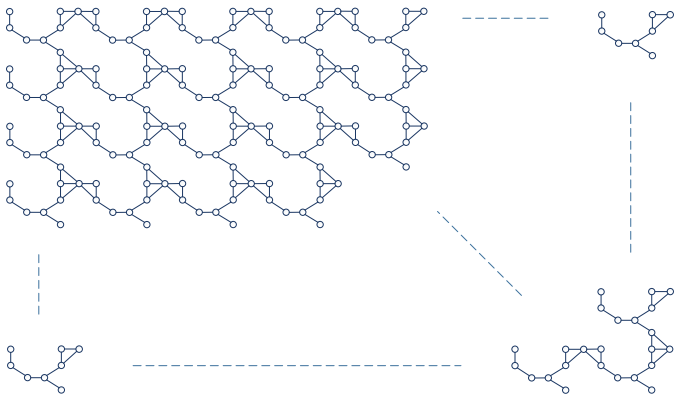

Figure 2.38: Partial view of an 800-node network with a repeated structure.

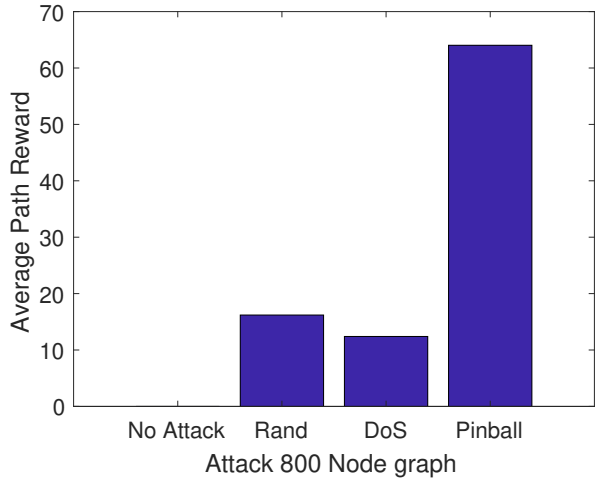

Figure 2.39: Comparing attack policies for the 800-node network.

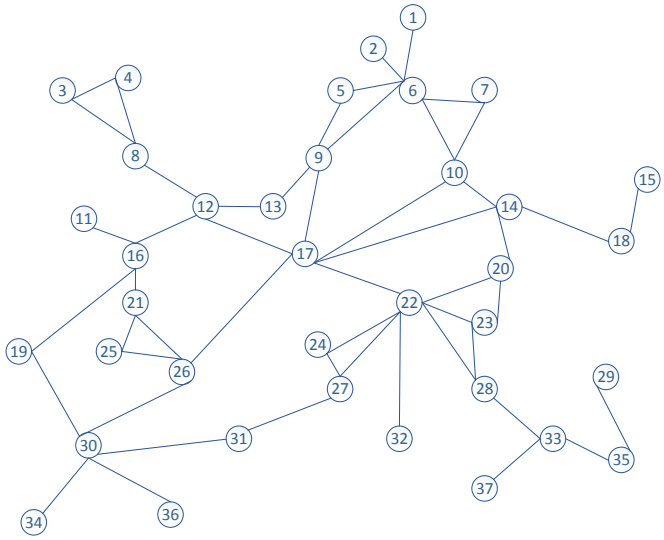

Figure 2.40: A 37-node network with node degree distribution.

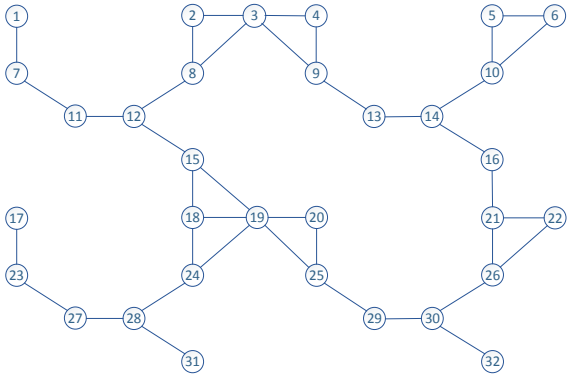

Figure 2.41: A 32-node network of a repeated structure

It is worth mentioning that, although the pinball policy has shown to yield the highest performance, the large scale graph starts from a state with a large number of conflicts, hence the number of additional conflicts due to the attack policy is rather small compared to the initial number of conflicts. However, the advantage of the pinball policy from the attacker's standpoint is that it hinders the network from reaching a state of zero-conflicts, i.e, it slows down the rate at which the network resolves its initial conflicts.

To even test our proposed policy on considerably larger graphs, we synthesized an 800-node graph (Fig. 2.38) as a repeated structure of the network topology of Fig. 2.20. In this case, we start from a no-conflict state and run the system for 10 steps. Fig. 2.39 shows the average path reward. The pinball policy yields a significantly higher reward. The no attack strategy gives zero reward since the network is started from a no-conflict state.

2.5.9 Attacking Networks of Arbitrary Structures

Finally, we extend the use of the pinball policy to arbitrary topologies. As an example, we consider the network shown in Fig. 2.40 which consists of 37 nodes. We approximate this graph to obtain

the same graph in Fig. 2.41, which has the repeated structure of Fig. 2.20 that only consists of 8 nodes. The pinball policy derived from the 8-node subgraph MDP problem is used for the attack.

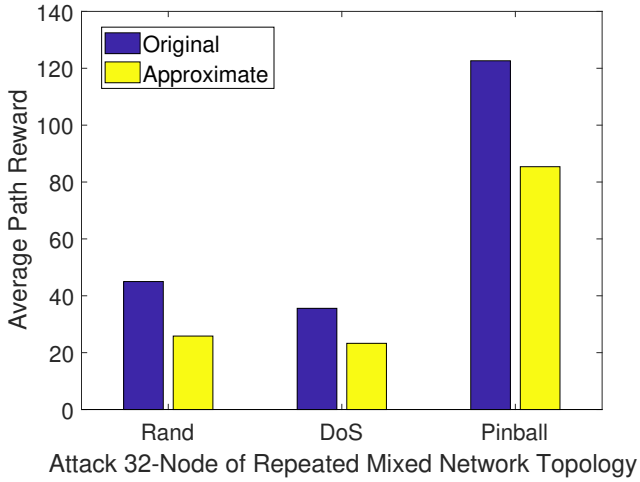

Figure 2.42: The gap due to approximating a graph with a repeated structure graph

Fig. 2.42 shows that the attacker incurs some loss in the average reward by using the approximate policy instead of the original one. However, the approximate pinball policy still outperforms DoS and Rand. In addition, the approximate approach to attacking arbitrary networks provides a substantial complexity reduction since it only needs to obtain the pinball policy for a small MDP problem (corresponding to the 8-node subgraph in this example) in lieu of the larger one (corresponding to the 37-node graph). We should mention that there is some additional overhead in identifying a graph with a repeated structure that well approximates an arbitrary graph.

CHAPTER 3: ADAPTIVE TOPOLOGIES AGAINST JAMMING ATTACKS IN WIRELESS NETWORKS

Shortage and tight control of the wireless spectrum coupled with the exponential demands for wireless bandwidth have put a lot of strain on our wireless resources. The deployment of wireless-enabled devices (e.g., Internet of Things, wearables, etc.) has made our networks larger and more dense. To accommodate such growth under spectrum constraints, significant research efforts focused on developing spectrum management techniques through the use of Software Defined Networks (SDN), Network Function Virtualization (NFV), and Cognitive Radios (CR) to improve the spectrum utilization [15, 51]. These techniques rely on sensing the current conditions (e.g., channels used, transmission power, interference levels, etc.) and dynamically making the appropriate allocation decisions. Current Access Points (APs) can dynamically select their channels and their transmission ranges to minimize interference with surrounding APs in wireless mesh networks (e.g., Cisco's Transmit Power Control [52]). The shared nature of wireless communication, however, has made many of the above techniques susceptible to attacks.

Various types of attacks have emerged that target such adaptation employed in SDN and CR at different layers [53, 42]. Jamming attacks, in particular, were shown to cause significant degradation in network performance through exploiting dynamic channel allocation feature and reactive power transmission techniques used in NFV and SDN (e.g., [37, 54, 34, 55]). In these attacks, the adversary creates interference on a given channel causing surrounding nodes to switch to other channels. Depending on the topology of the network, the effect of the attack can ripple through the network causing further nodes to switch. Once (and if) the network converges, the adversary can repeat the attack. Attacks on the control channel can also cause a similar outcome. Defense against jamming attacks has focused on the physical layer through frequency/channel hopping/selection

techniques – including the control channels – (e.g., [44, 56, 39, 38]), as well as power allocation methods (e.g., [40, 57]).

To study the interactions between the adversary and the defender, game theory has provided a rigorous framework to model the strategies of each player and has been instrumental in advancing the state-of-the-art in jamming games (e.g., [58, 8, 57, 44]) – which we explore in detail in Section 2.2. In many games, however, obtaining optimal policies is computationally prohibitive due to the exponential nature of the action space of the players. Some studies have investigated different techniques – that rely on either iteratively adding pure strategies to the problem or considering a marginal strategy through a minimax problem formulation – to make the problem more tractable [59, 60, 61]. We consider the defender's strategy in adapting the footprint of the wireless nodes to minimize the impact of a rational jammer, which presents itself as a combinatorial problem. We develop effective decomposition approaches that are scalable for larger and more practical problems.

In this chapter, we consider a wireless network of N nodes (APs) in which the defender adapts the transmission power level of each node to minimize interference due to jamming attacks. The setup, for example, corresponds to a wireless mesh network that exists in large buildings (e.g., academic, residential) configured to minimize overlap between nodes while ensuring proper coverage to service clients. We consider an adversary that aims to increase interference through selecting node(s) to jam. The defender adjusts the wireless footprint of the nodes to effectively change the topology for the adversary. The defender can be centralized as in wireless mesh networks in buildings that are managed by a single entity, or distributed as with the use of SDN-enabled APs in residential setups that are managed by multiple entities. In adapting the underlying topology via power control, we seek to optimize an inherent tradeoff between immunity to jamming and network coverage. If the footprints of the nodes recede, immunity to jamming could be enhanced at the expense of potential reduction in network coverage and vice-versa. In our model, we capture this tradeoff in the

cost function of various formulations of security games, classified according to the structure of the players' action spaces. To deal with the curse of dimensionality, we develop scalable approaches to handle the complexity of the various classes of the games considered.

3.1 Related Work

This work is related to two research areas: security in wireless channel allocation techniques and formulations of security games.

3.1.1 Security in Wireless Channel Allocation Techniques

There is a large body of research on the security of channel assignment algorithms in regards to exposing attacks and developing defense mechanisms (e.g., [37, 34, 43, 55, 62, 38, 39]). The work in [37] exposes three types of attacks against channel assignment algorithms that capitalize on attacking the most loaded channels. One of these attacks is the Low-Cost Ripple Effect Attack (LORA) that aims to force the network in quasi-stable state by continuously inducing channel conflicts. Moreover, the authors in [63] presented a game theoretical analysis to study the interaction between mobile ad-hoc networks (MANET) nodes where a subset of the nodes are malicious. In [43] the authors show a number of vulnerabilities in MAC protocols due to selfish cognitive radio users that seek to gain more than their fair share of resources. The authors in [34] expose three types of attacks against channel assignment mechanisms through creating utilization-based conflicts, link breakage and Denial of Service. The attacks above, however, were not studied from an optimization/game-theoretic standpoint and thus the attack strategies were rather arbitrary. The work in [55, 62] exposed attacks on dynamic channel allocation methods through Markov Decision Process (MDP) problems in which the attacker seeks to maximize the damage inflicted (captured

by the interference level in the network) subject to an attack cost (e.g., risk of being detected). These attacks, however, did not consider the defense strategies and in this dissertation we consider the impact of the defender through a game-theoretic formulation. On the defense front, the authors in [38, 39, 40] address the control-channel jamming attacks and propose a randomized distributed scheme in which nodes can reestablish the control channel using frequency hopping techniques. Furthermore, in [39] the authors propose two methods for identifying the jammers whether acting independently or colluding. The authors in [64] study and compare proactive versus reactive frequency hopping techniques against jamming attacks through a min-max formulation. The defense strategies, however, are limited to two defense actions: proactive vs. reactive. The author in [40] considers power control to enhance successful transmission. Our work, however, considers power control through a game-theoretic formulation with an attacker deciding which node(s) to jam. Game theory has been also used to study the power allocation problem among wireless nodes while managing the interference level between them [65, 66, 67]. In [65], the authors investigated the joint uplink subchannel and power allocation problem in cognitive small cells in a cooperative game-theoretic framework. In [66], the authors solved the subchannel assignment and power allocation joint optimization problem in heterogeneous small cell networks in a non-cooperative game setting. The power allocation problem for non-orthogonal multiple access amplify-and-forward two-way relay wireless networks in the presence of eavesdroppers was investigated in [67].

3.1.2 Game-Theoretic Formulations in Security Games

There have been some studies that focused on the game-theoretic aspect of jamming attacks as in [58, 9, 8, 57, 44, 68, 69, 70, 64, 71, 72, 73]. In [58] the authors consider a game in which the attacker aims to maximize the number of corrupted communication links through jamming while the defender aims to detect the presence of the attacker based on the percentage of collisions observed by a set of monitor nodes. In [8], the authors consider a game-theoretic approach for a finite

energy jamming game in which both players, the defender and the adversary, start with a power budget and decide in each round what power to use. A reward is obtained based on their power choices. The game is solved by computing the possible cases through dynamic programming, then identifying the strategies through a linear program. The authors, however, consider the defender as a single pair of nodes. The authors in [57] consider a jamming attack against a secondary user in cognitive radio setups in which a secondary user allocates different transmission powers to the fallow bands available. The attacker aims to induce interference by allocating different power levels in the same bands. The game is cast as a Colonel Blotto zero-sum game and a NE is obtained as randomized strategies over the power allocations of the players under certain conditions. The dynamics of this game are studied later in [71, 74] with focus on the single node case with an SINR objective function. Although the scope of the work is related to ours, our models and formulations are different. In particular, we focus on the topological adaptation of N nodes under coverage constraints whereby the power allocated to any node is coupled with the power allocated to its neighbors. Such a problem increases the complexity of the underlying games and leads to identifying scalable and computationally tractable defense mechanisms. In [44] the authors consider a stochastic game between a jamming adversary and a secondary user. In this game, multiple channels are reserved for control messages and are dynamically switched with data channels by the secondary user based on the strategy of the adversary. The work in [68] studies a problem of resource allocation in the presence of a jammer subject to total resource constraints. The authors in [70] studied the power allocation problem in dense small cell networks in a game-theoretic framework but at normal condition. i.e., when no threat is directed at the network. The authors establish the uniqueness of the NE for special utility functions (such as α- fairness) and show that it admits a closed-form parametrized solution that can be obtained by solving a non-linear system in few parameters, thereby evading the combinatorial complexity for that particular setting. This work was further extended in [69] to non-zero sum games where both selfish and cooperative strategies are characterized and shown to coincide under certain conditions. Authors in [75] considered a similar

problem in a different setup as a Stackelberg game [76] formulation, however they considered a decoupled SNR reward function that does not account for any interactions between the network neighbors. In the considered model, we obtain a marginal strategy for the footprint assignment of the nodes which satisfies a global coverage constraint through marginal relaxation of a zero-sum formulation.

Prior work focused on securing separate transmitter-receiver pairs against jammers. In sharp contrast, this work investigates the security of the whole network by considering the simultaneous adaptation of N wireless nodes through a game-theoretic framework. The proposed model uniquely considers a coupled objective function in which the adaptation of a node is not done in isolation from its neighbors, but rather depends on the power and frequency allocated to its neighbors. Furthermore, due to the complexity of solving the full game NE, we develop decomposition approaches to approximate the full games that have shown to yield efficient and scalable defense strategies for securing wireless networks.

3.2 System Model

We consider a wireless network of N Access Points (APs), represented by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of APs (nodes) such that $|\mathcal{V}| = N$, and $\mathcal{E}$ is the set of edges. Two nodes are considered adjacent if the distance between them is below a certain threshold so that they may interfere given the overlap of their footprints. Every node $v \in \mathcal{V}$ uses a channel frequency c_v from a set of usable frequencies $\mathcal{C}$. This set depends on the network operating spectral band.

3.2.1 Dynamic Channel Assignment

Channels should be assigned so as to avert Co-Channel Interference (CCI) and/or Adjacent Channel Interference (ACI) between neighboring nodes. Such an assignment is tantamount to a graph coloring problem since a fixed number of channels are assigned to N nodes in such a way that no two adjacent nodes share the same color, i.e, use interfering channels.

Let $\mathcal{N}_v := \{u : r_v + r_u > d_{u,v}\}, \forall v, u \in \mathcal{V}$ denote the neighboring set of node v predefined by the network administrator prior to playing the game and adapting the topology, where r_v is the power radius of node v and $d_{u,v}$ is the distance between the two nodes. We also define an interference set, $\mathcal{I}_v$, for every node $v \in \mathcal{V}$ as defined in equation (2.2). Thus, the set of non-interfering channels $\mathcal{L}_v$ available to node v is the difference set of $\mathcal{C}$ and the union of the interference sets of the neighbors of v as defined earlier in equation (2.3). We also define the conflict set $\mathcal{D}_v$ as the set of nodes in the neighborhood of v that use channels in its interference set, equation (2.4).

In practice, if the interference level exceeds a certain admissible threshold, the system should re-assign the channel frequencies. To further reduce interference with neighboring nodes, the footprint of the APs can be adapted by adjusting their transmission radii through power control. For instance, in the Cisco Radio Resource Management (RRM) system, the Transmit Power Control (TPC) protocol is automatically initiated if any AP is found to interfere with three or more neighbors with an RSSI exceeding a certain threshold (-70 dBm), thereupon its transmission power level is reduced to go below that threshold [52].

3.2.2 Jamming Attacks

Through jamming the APs, an adversary can trigger the channel assignment and adaptation protocols as mentioned earlier. Multiple jammers may further collude to inflict more damage by si-

multaneously attacking several APs. Initiating channel re-assignment to resolve conflicts could, in turn, result in increased network delay due to perpetual channel switching, and inadvertent changes in the footprint of the APs could lead to a loss in network coverage through the formation of coverage holes. The damage can be more drastic if these adaptations are automated on the software level. For instance, dynamic channel switching in the application layer as in SDNs and NFV – primarily intended to improve the network performance and stability [16] – can leave the network more susceptible to such exploits.

Next, we present a general game-theoretic formulation to identify effective and scalable defense strategies against jamming attacks based on adaptive network topologies.

3.2.3 General Framework

A game is defined as a tuple $\Gamma(\mathcal{K}, \mathcal{A}, \mathcal{R})$, where

- $\mathcal{K}$ is the set of players. $\mathcal{K} = \{1, 2\}$, denotes the defender (player 1) and the adversary (player 2).
- $\mathcal{A} = \mathcal{A}_d \times \mathcal{A}_a$ is the action space for the defender and adversary.
- $\mathcal{R} = \{R_d, R_a\}$ is the reward function, $\mathcal{R} : \mathcal{A} \rightarrow \mathbb{R}^2$ mapping actions to rewards, where R_d and R_a are the defender and adversary rewards, respectively. Since we consider zero-sum games, $R_d = -R_a = R$.

The defender sets the power profile of the nodes. Let $\mathbf{a}_d \in \mathcal{A}_d \subset \mathbb{R}^N$ be a power assignment vector designating the action of the defender. Hence, the defender action can be expressed as, $\mathbf{a}_d = [P_1, \cdots, P_N]$, where P_v is the power level assigned to node v. The adversary chooses one or more nodes to jam with interfering signals. Therefore, $\mathbf{a}_a = [J_1, \cdots, J_N]$, where J_v is the power

used by the adversary to jam node v. In the first instantiation, we consider a discrete game where $\mathcal{A}_d$ and $\mathcal{A}_a$ are both discrete and finite sets. As explained later, computing a NE in such games is computationally prohibitive for large network sizes, therefore we identify suboptimal scalable strategies. Also, in the second instantiation we further consider a defender with a continuous control space and propose a marginal-based approach to identify favorable strategies. At last, we consider a continuous game where both players have continuous action spaces.

The defender seeks to reduce the level of interference by adapting the power level of each node while maintaining an acceptable network coverage. On the other hand, the adversary seeks to inflict the maximum disruption to the network while reducing the cost of the launched attack. The cost of the attack captures its risk of exposure. We develop defense strategies where the defender chooses the power assignments for the nodes from both discrete and continuous sets.

3.3 Game Formulation

In this section, we use the general game-theoretic framework introduced above to present specific game formulations based on the control sets available to the players (e.g., discrete and continuous sets) as well as the corresponding utility functions (that capture the fundamental tradeoff between mitigating the jamming damage and maintaining minimum network coverage). We devise scalable solutions for the corresponding games to reduce the game complexity.

We focus on two-player zero-sum games. This class of games suits our system model as it does not allow for any type of cooperation between the players – a reward for one player is a cost for the other.

3.3.1 Discrete Game

In the first instantiation, we start off with simple defense and offense strategies whereby the defender selects one of two power levels, high or low, to assign for each node and the adversary is only capable of jamming a single node at a time. We investigate a minimax formulation to obtain a NE for the game. We remark that the use of two power levels is by no means restrictive and only used here to simplify the exposition. We elaborate more on that later in the section as we discuss the game complexity, as well as in Section 3.3.2 where we allow for more power levels along with explicit coverage constraints. The essential assumption for the formulation we consider in this section is the discrete nature of the action space for both players, hence the appellation 'Discrete Game'. This assumption is relaxed in Section 3.3.3 where we study a continuous game formulation.

With slight abuse of notation, we use P_v as a binary variable to designate the two power levels, i.e., $P_v \in \{0, 1\}\,,\ \forall v \in \mathcal{V}$, with $P_v = 0$ corresponding to low power for node v and $P_v = 1$ for high power. Hence, the pure action $\mathbf{a}_d \in \mathcal{A}_d = \{0, 1\}^N$ is a binary vector of length N. There exist 2^N pure strategies for a network of N nodes, thus the dimensionality of the action space grows exponentially with the graph size. The defender may choose to play any of these actions (a pure strategy), or a combination of these strategies through a mixed strategy as explained later. On the other hand, the jammer can select one node to jam at a time such that $J_v \in \{0, 1\},\ \forall v \in \mathcal{V}$, using the same convention used above for the power levels assigned by the defender. Hence, the jammer's pure action $\mathbf{a}_a \in \mathcal{A}_a \subset \{0, 1\}^N$ is a binary vector of length N with at most one entry of value 1. For example, if the adversary jams node $k \in \mathcal{V}$, $\mathbf{a}_a = \mathbf{e}_k$, where $\mathbf{e}_k$ is a vector of all zeros except for a one at the k-th position. Similarly, $\mathbf{a}_a = \mathbf{0}$ corresponds to a no-attack action.

A pure strategy is one that selects one of the actions from the action spaces for each player. Alternatively, a player may choose to use a randomized (mixed) strategy defined through a probability

distribution over the pure strategies. Given the set of actions of player 1, $\mathcal{A}_d$, let $\mathcal{X}(\mathcal{A}_d)$ denote the set of all probability distributions over $\mathcal{A}_d$, i.e., the set of mixed strategies for player 1. In a mixed strategy $\mathbf{x} \in \mathcal{X}$, action $\mathbf{a}_d$ is played with probability $x(\mathbf{a}_d)$. Similarly, the mixed strategy of the adversary is a probability distribution $\mathbf{y} \in \mathcal{Y}$ over the pure strategies.

Let $1_{\mathcal{S}}$ be the indicator function associated with set $\mathcal{S}$. The defender's reward R is defined as,

$$R(\mathbf{a}_d, \mathbf{a}_a) = \sum_{v \in \mathcal{V}} \Big(h \cdot 1_{\{J_v=1\}} \, \frac{\delta_v^- + \delta_v^+}{2} - 1_{\{P_v=0\}} \\ - \beta_v \left(1_{\{P_v=1\}} + 1_{\{P_v=0\}} 1_{\{J_v=1\}} \right) \Big), \tag{3.1}$$

where $\beta_v = \sum_{\{u \in \mathcal{N}_v : c_u \in \mathcal{I}_v\}} \left(1_{\{P_u=1\}} + 1_{\{P_u=0\}} 1_{\{J_u=1\}} \right)$. The first term in the summand on the RHS of (3.1) is the cost uncured by the adversary (reward for the defender) if he jams node v. Recall that the topology varies from its original state depending on the selected power profile of the nodes. The degree of node $v \in \mathcal{V}$ is δ_v^- and δ_v^+ prior and after playing the game, respectively, and h denotes the attack cost. Hence, considering any node $v \in \mathcal{V}$, the adversary incurs a cost $h \cdot \delta_v^+/2$ capturing the node degree after the game is played, plus another cost $h \cdot \delta_v^-/2$, which is a constant penalty reflecting the original importance of the attacked node and the power used by the adversary (proportional to node degree). The second term is a cost incurred by the defender for the coverage holes captured by the number of nodes transmitting at '*low*' power level. The third term counts the conflicts in the network, calculated by summing over the nodes in conflict, each weighted by the number of neighbors it conflicts with. Thus, b_v in (3.1) counts the number of interfering nodes in the neighborhood of v due to transmitting at an interfering frequency or because it is jammed by the adversary.

In order to obtain the NE of the game, the defender solves the following linear program (LP),

$$\underset{\mathbf{x}\in\mathcal{X}}{\text{maximize}} \qquad U \tag{3.2}$$

$$\text{subject to} \qquad \sum_{\mathbf{a}_d\in\mathcal{A}_d} R(\mathbf{a}_d,\mathbf{a}_a)x(\mathbf{a}_d) \geq U, \qquad \forall \mathbf{a}_a \in \mathcal{A}_a, \tag{3.3}$$

$$\mathbf{x}^T\mathbf{1} = 1, \quad \mathbf{x} \geq \mathbf{0}, \tag{3.4}$$

where U in (3.2) denotes the value of the game, which is the expected reward $\mathbf{x}^T R\mathbf{y}$. Thus, the defender seeks to maximize the value of the game with the first set of constraints (3.3) ensuring that the defender plays best response to every pure strategy by the adversary. The remaining two constraints in (3.4) ensure that $\mathbf{x}$ is a valid probability distribution vector.

Game complexity: While (3.2) can be readily solved to obtain a NE for the game, its complexity grows exponentially with the network size and the power of the attack. Indeed, recalling that $\mathcal{A}_d = \{0,1\}^N$, each constraint in (3.3) involves a summation over 2^N pure strategies. Note that the restriction to two power levels in this section is without loss of generality and is only used to simplify the exposition. Surely, for a fixed graph size N the complexity of the LP in (3.2) is polynomial in the number of admissible power levels, so generalizing to more power levels does not greatly impact complexity. Also, the number of constraints in (3.3) is linear in $M := |\mathcal{A}_a|$, the number of pure strategies for the jammer, albeit M itself is $O(N^K)$, i.e., exponential in the power of the attack measured by the number of nodes K the adversary can attack simultaneously, or equivalently the number of jammers colluding to attack different nodes at the same time. Typically $K \ll N$, yet the complexity of the game is still exponential in N. This motivates our next section, where we develop scalable and tractable defense strategies based on a newly proposed decomposition.

3.3.1.1 Decomposition-based approach

In this approach, we define a sub-game per node. Each sub-game is associated with a subgraph consisting of the corresponding node and its neighbors in a star topology. The defender solves N simple sub-games – in lieu of one complex game – whose solutions are then combined to decide on the power profiles for each node.

Similar to the game formulation of the full game in Sections 3.2.3 and 3.3.1, let $\Gamma_v(\mathcal{P}, \mathcal{A}_v, \mathcal{R}_v)$ denote the sub-game corresponding to node v. The action space $\mathcal{A}_v$ is the set of all possible actions by both players, i.e.,

$$\mathcal{A}_v = \{(P_v, J_v)\}, P_v, J_v \in \{0, 1\},$$

and the set of sub-game reward for both players $\mathcal{R}_v = \{R_v, -R_v\}$.

In solving the sub-game corresponding to node v to determine its power level, the defender has to make some assumptions about the unknown power profiles of the neighbors of v. This uncertainty is the result of our search for a decoupling scalable strategy versus a joint solution of all the power profiles as in the full LP (3.2). To tackle this difficulty, we consider an *all-high assumption*, that is, we assume that all the neighbors of v will be transmitting at '*high*'. This is a worst case assumption in terms of interference between neighbors in case conflicts exist between node v and any of its neighbors. Hence, the reward of the defender for this sub-game can be expressed as,

$$R_v(P_v, J_v) = h \cdot 1_{\{J_v=1\}} \frac{\delta_v^- + \delta_v^+}{2} - 1_{\{P_v=0\}} - \beta_v(P_v, J_v) \left(1_{\{P_v=1\}} + 1_{\{P_v=0\}} 1_{\{J_v=1\}}\right), \quad (3.5)$$

where all symbols are defined as before with respect to the subgraph of node v. As such, we only account for the cost and reward for the defender and jammer with respect to node v. The value of

the sub-game played at node v can be expressed as,

$$U_v = \begin{bmatrix} 1-x_v & x_v \end{bmatrix} \begin{bmatrix} R_v(0,0) & R_v(0,1) \\ R_v(1,0) & R_v(1,1) \end{bmatrix} \begin{bmatrix} 1-y_v \\ y_v \end{bmatrix}, \tag{3.6}$$

where x_v is the probability that the defender plays action $a_d = 1$, i.e, assigns '*high*' power level to node v, and y_v the probability of attacking node v. We can readily write the optimization problem solved by the defender to obtain a NE mixed strategy for sub-game Γ_v as,

$$\begin{aligned} \underset{x_v}{\text{maximize}} \quad & U_v \\ \text{subject to} \quad & R_v(1,0)x_v + R_v(0,0)(1-x_v) \geq U_v, \\ & R_v(1,1)x_v + R_v(0,1)(1-x_v) \geq U_v, \\ & x_v \geq 0. \end{aligned} \tag{3.7}$$

Again, this naturally generalizes to any finite number of power levels. The probabilities $x_v, v \in \mathcal{V}$ fully define the overall strategy $\mathbf{x}$ of the defender through a simple product measure, where each node is assigned its level based on its own distribution ($[1-x_v \;\; x_v]^T$ for the two-level case). The first two constraints in (3.7) guarantee that any mixed strategy x_v is an element of the best response set of the defender [7].

After solving the set of decomposed games using the same procedure for solving the full-size game, we obtain the jammer's NE strategy $y_v, v \in \mathcal{V}$ for each of the N sub-games. Then, the strategy of the jammer is obtained through proper normalization since the mixed strategy can only be supported on the jammer's M admissible pure strategies. For example, consider the scenario where the adversary can only jam one node at a time, i.e., $\mathcal{A}_a = \{\mathbf{e}_k\}_{k=0}^N$, recalling that $\mathbf{e}_k$ is the vector of all zeros except for a value one at the k-th position for $k = 1, \ldots, N$, and $\mathbf{e}_0 = \mathbf{0}$ is the vector of all zeros corresponding to no-attack. Hence, the probability that the adversary plays

action $\mathbf{e}_k$, denoted $y(\mathbf{e}_k)$, is obtained from the mixed strategies $y_v, v \in \mathcal{V}$, as

$$y(\mathbf{e}_k) = \frac{y_k \prod_{\substack{v=1 \\ v \neq k}}^{N} (1 - y_v)}{\sum_{k=0}^{N} y_k \prod_{\substack{v=1 \\ v \neq k}}^{N} (1 - y_v)}, k = 0, \ldots, N \tag{3.8}$$

where $y_0 = 1$.

Through this decomposition we have reduced the complexity of the original LP (3.2) – which was exponential in N (and polynomial in the number of power levels) – to linear complexity since the complexity of each sub-game is linear in the number of power levels (here we simply considered 2 levels) and we have to solve N such sub-games.

3.3.1.2 Progressive decomposition approach

Here, we develop a second alternative for decomposing the original LP other than the one based on the all-high assumption. In particular, we select an arbitrary ordering of the nodes. Then, we proceed in a greedy manner by solving decomposed sub-games progressively with respect to the predefined order of nodes, but this time sequentially feeding in the power assignment of the solved sub-games to subsequent ones rather than making an all-high assumption for all the neighbors. Hence, as we solve the sub-game corresponding to node v, the probability that *some of its neighbors* are transmitting at the '*high*' level is now known from the previous steps. For those neighbors whose assignment is not yet specified, we set their powers to '*high*'. Once we go over all the nodes in the specified order, we obtain the final strategy of the defender. We report on the performance of both decomposition-based approaches in the results section.

Remark 2. *When both players resort to decomposition owing to the combinatorial complexity of*

the full game, the decomposed strategies may yield a higher reward than the full game NE. This does not violate optimality since a NE is only optimal in the sense of unilateral deviations by one of the players [7].

3.3.2 Discrete Game with Explicit Constraints

In this section, we allow the action space of the defender to accommodate more than two power levels and account for explicit coverage constraints in the formulation. We also devise a different approach to address the aforementioned combinatorial complexity of the discrete game by allowing the defender to directly optimize over the marginal footprints of the nodes rather than enumerating all possible pure strategies. More specifically, let $\bar{\mathbf{P}} = \left[\bar{P}_1, \ldots, \bar{P}_v, \ldots, \bar{P}_N\right]$ denote a marginalized power strategy, where $\bar{P}_v$ is the marginal power of node v satisfying

$$P_L \leq \bar{P}_v \leq P_U, \tag{3.9}$$

where P_L and P_U are the admissible lower and upper power levels, respectively.

In this formulation, we consider scenarios where we have an explicit constraint on coverage (as opposed to absorbing it in the reward function as in the first formulation), for example to ensure a minimum level of quality of service. Specifically, the defender seeks to maintain a minimum network coverage measured by the sum of powers to avoid undue coverage loss, i.e., $\bar{\mathbf{P}}^T\mathbf{1} \geq C_{\min}$. Thus, we modify the optimization space for the defender's action as

$$\mathcal{A}_d = \{\bar{\mathbf{P}} \in \mathcal{P} \subset \mathbb{R}^N : \bar{\mathbf{P}}^T\mathbf{1} \geq C_{\min}, \bar{\mathbf{P}} \text{ satisfies } (3.9)\}, \tag{3.10}$$

where $\mathcal{P}$ is a discrete and finite subset of $\mathbb{R}^N$. For simplicity, we consider the action space of the adversary $\mathcal{A}_a = \{\mathbf{e_k}\}_{k=1}^N$, in which one node is jammed at a time. Instead of using the node degree

considered earlier, we adopt a measure of the interference overlap between the nodes. Therefore, the reward function R now captures the impact inflicted by the adversary on the network that is proportional to the degree of overlap between the jammed node and the rest of the nodes. The overlap L_{uv} between nodes v and u corresponding to assignment $\bar{\mathbf{P}}$ is defined as

$$L_{uv}(\bar{\mathbf{P}}) = \bar{P}_u + \bar{P}_v - \kappa d_{uv}^2, \tag{3.11}$$

where d_{uv} is the distance between the pair of nodes and κ is a weighting constant of proper units. This choice is motivated by the fact that (3.11) provides a tractable approximation of the overlap area of two circles, which consists of the sum of two terms proportional to the area of each circle (captured here by the power which is proportional to the coverage area) minus a term that scales with the square of the distance, [77]. We readily define the reward function as

$$R(\bar{\mathbf{P}}, \mathbf{e}_k) = -\sum_{v=1}^{N} (L_{kv}, 0)^+, \; k = 1, \ldots, N \tag{3.12}$$

where the function $(., 0)^+ := \max(., 0)$ captures the positive overlap and node k is the node being jammed by the adversary. To account for colluding adversaries, (3.12) can be simply modified by summing over the set of jammed nodes.

We consider a zero-sum game formulation in which the defender aims to maximize the worst case utility, solving for the optimal tradeoff in the footprint assignment that reduces the severity of the conflicts inflicted by the jammer subject to the minimum coverage requirement. Hence, the optimal mixed strategy for the zero-sum game can be obtained by solving the LP given in (3.2), (3.3), and (3.4), with the new definitions of the reward function and action spaces.

Solving this problem exactly is computationally prohibitive for large graphs since the number of pure strategies grows exponentially in N. To handle the large pure strategy space, here we adopt

a marginal-based approach [61], in which we directly optimize over the marginal assignment of footprints instead of enumerating all the pure strategies in $\mathcal{A}_d$.

Marginal-based approach: Here we require the marginal assignment $\bar{\mathbf{P}}$ to satisfy the coverage constraint. Bypassing the modeling of the mixed strategy and optimizing over the marginal variables, we solve the following optimization

$$\begin{aligned} &\underset{\bar{\mathbf{P}} \in \mathcal{A}_d}{\text{maximize}} \quad U \\ &\text{subject to} \quad U \leq -\sum_{v=1}^{N} (L_{kv}(\bar{\mathbf{P}}), 0)^+, \ \forall k = 1, \ldots, N, \end{aligned} \tag{3.13}$$

with $\mathcal{A}_d$ defined in (3.10), which dispenses with the constraints in (3.4) and enforces the coverage constraint in the marginal strategy. In the numerical evaluation, we used the *CPlex Optimizer* [78] to solve the LP in (3.13).

Implementation of the marginal strategy: Since an AP only supports a discrete set of power levels, we need to ensure that the obtained NE of the marginalized strategy is admissible, i.e., implementable using this set of power levels. Recall that a pure action, $\mathbf{a}_d$, is a vector of length N whose v-th entry, P_v, designates the power level assigned to node v. This amounts to a resource allocation problem with one resource (power) to be assigned to N agents (nodes) subject to minimum coverage constraints. A marginalized strategy is implementable if there exist positive numbers $\{\lambda_n\}_{n=1}^{T}$ such that, $\bar{\mathbf{P}} = \sum_{n=1}^{T} \lambda_n \mathbf{P}^n$, where $\mathbf{P}^n$ denotes the pure strategies.

Budish et al. established that a sufficient condition for marginalized strategies to be implementable using pure strategies is when the resource assignment constraints set forms the so-called 'hierarchy' [79]. The concept of a hierarchy pertains to constraint structures of assignment problems in which different objects/resources are assigned to a number of agents subject to a number of assignment constraints.

Constraint structures and hierarchies: To clarify, we briefly review the concept of hierarchies from [79], then show that our constraint structure naturally forms a hierarchy, wherefore the marginalized strategy is implementable.

Given the set of nodes $\mathcal{V}$ and a set of resources $\mathcal{O}$, a pure assignment is defined through a matrix $\mathbf{T} = [T_{vo}]$ indexed by all node-resource pairs (v, o) defining the amount of resource $o \in \mathcal{O}$ assigned to $v \in \mathcal{V}$. The assignment has to satisfy constraints of the form

$$\underline{q}_{\mathcal{S}} \leq \sum_{(v,o)\in\mathcal{S}} T_{vo} \leq \bar{q}_{\mathcal{S}}, \tag{3.14}$$

for sets $\mathcal{S}$ of node-resource pairs, i.e. $\mathcal{S} \subseteq \mathcal{V} \times \mathcal{O}$, called constraint sets. The bounds $\underline{q}_{\mathcal{S}}$ and $\bar{q}_{\mathcal{S}}$ are two assignment quotas, for example lower and upper bounds on coverage in our context. The full collection of constraint sets along with the quotas in an assignment problem form a constraint structure $\mathcal{H}$.

Definition 2. *$\mathcal{H}$ is a hierarchy if for every pair of constraints $\mathcal{S}$ and $\mathcal{S}'$ in $\mathcal{H}$, we have $\mathcal{S} \subset \mathcal{S}'$, or $\mathcal{S}' \subset \mathcal{S}$, or $\mathcal{S} \bigcap \mathcal{S}' = \emptyset$.*

Theorem 3. *The constraint sets $\mathcal{A}_d$ defined in (3.10) for the power assignment problem form a hierarchy, hence the marginalized strategy is implementable using pure strategies.*

Proof. Per [79], it suffices to show that our constraint sets form a hierarchy. To this end, we first note that we only have one resource (the power in a single band) to be assigned to N APs. Hence, the set of resources $\mathcal{O}$ consists of one resource, equivalently $\mathcal{O} = \{1\}$. The pure assignment vector $\mathbf{a}_d$ consists of power levels for each node, with two types of constraints, namely, individual power constraints (3.9) for every node, and a total coverage constraint which must be satisfied for any admissible pure assignment power vector $\mathbf{a}_d$. Hence, the constraint sets are singletons $\{(v, 1)\}$ for every node $v \in \mathcal{V}$, as well as the set $\mathcal{V} \times \{1\}$ of all nodes and resource pairs, with corresponding

constraints $P_L \leq \mathbf{a}_d(v) = P_v \leq P_U$, and $\sum_{v\in\mathcal{V}} \mathbf{a}_d(v) \geq C_{\min}$, respectively. Clearly, the constraint sets satisfy the requirements of a hierarchy given in Definition 2 since we have disjoint singletons, which are also subsets of $\mathcal{V} \times \{1\}$. Therefore, any marginalized assignment strategy $\bar{\mathbf{P}}$ satisfying the individual and total coverage constraints is implementable as a convex combination of pure strategies.

□

The formulation in (3.13) only includes a (global) total coverage constraint, but this is by no means restrictive. Specifically, we can also incorporate local coverage constraints – for example to avoid local holes in coverage around certain areas. In general, adding such linear constraints to the optimization problem will not affect its order complexity and only restricts the action space for the defender, $\mathcal{A}_d$. However, it may affect the implementability of the equilibrium marginal strategy.

To clarify, let ϱ^m denote a set of APs covering a local area indexed by m, where $m = 1, \ldots, M$, and let $C^m_{\min}$ be the corresponding minimum coverage. The local coverage constraints can thus be expressed as,

$$\sum_{v\in\varrho^m} \bar{P}_v \geq C^m_{\min}, \quad \forall m. \tag{3.15}$$

Hence, the action space for the defender can be modified as,

$$\mathcal{A}_d = \{\bar{\mathbf{P}} \in \mathcal{P} \subset \mathbb{R}^N : \bar{\mathbf{P}}^T\mathbf{1} \geq C_{\min}, \bar{\mathbf{P}} \text{ satisfies } (3.9), (3.15)\}, \tag{3.16}$$

Theorem 4. *The constraint sets in (3.16) for the power assignment problem in (3.13) form a hierarchy if and only if $\varrho^m \bigcap \varrho^{m'} = \emptyset$ for $m \neq m'$. If the problem forms a hierarchy, the marginalized strategy is implementable using pure strategies.*

Proof. If the sets $\varrho^m, m = 1, \ldots, M$, for which the local coverage constraints (3.15) are defined

are disjoint, then the corresponding constraint sets (sets of node-resource pairs) $\mathcal{S}^m := \{(v, 1) : v \in \varrho^m\}, m = 1, \ldots, M$, are disjoint. In addition, the constraint sets $\mathcal{S}^m$ are subsets of the set $\mathcal{V} \times 1$ corresponding to the total coverage constraint. Hence, the constraint structure is still a valid hierarchy by Definition 2. Thus, the marginalized strategy can be implemented as a convex combination of pure strategies. □

3.3.3 Continuous Game

In the first instantiation, we considered a game with a discrete control space for each player. We showed that computing a NE is computationally prohibitive, wherefore we devised decomposition-based approaches to derive suboptimal strategies. In the discrete game we leveraged marginalization to avert enumerating all possible pure actions for the defender. In this section, we consider a continuous control space for both players. The defender assigns a power level from a compact set to each node, and the jammer distributes his jamming power among the nodes to inflict damage subject to a power constraint. Specifically, the defender chooses a power assignment $\mathbf{P} = [P_1, P_2, ..., P_N]^T$. The power of each node is constrained as $P_L \leq P_v \leq P_U$, where P_L and P_U denote the minimum and maximum power levels for AP v, respectively. The strategy of the jammer is $\mathbf{J} = [J_1, J_2, ..., J_N]^T$, subject to a jamming power constraint, i.e., $\sum_v J_v \leq \bar{\mathbf{J}}$.

As considered in the earlier game instantiations, we define a utility function for the zero-sum game capturing the inherent tradeoff of increasing the power level for each node in order to enhance network coverage, and decreasing power to decrease the overlap between neighboring nodes thereby reducing the impact of jamming. In addition, the reward function in this game explicitly accounts for the impact of jamming on the users of the network. Meanwhile, the jammer seeks to reduce the utility of the defender by selecting the jamming powers $\mathbf{J}$ subject to his jamming power budget, while reducing the risk of exposure. The risk of exposure is a cost for the jammer that increases

proportionally to the importance of the attacked node captured by the number of users it can potentially serve. Hence, we account for the attacker cost by the term

$$T_1(v) = J_v \cdot \Delta_v, \tag{3.17}$$

where Δ_v denotes the density of users served by node v. Assuming that the active users are uniformly distributed over the network, the number of users served by AP v is proportional to its coverage area, hence proportional to the power transmitted by AP v, i.e., $\Delta_v = K_v P_v$, for some proportionality constant K_v. We also define the second term

$$T_2(v) = \sum_{u \in \mathcal{N}_v : u \in \mathcal{I}_v} P_u, \tag{3.18}$$

as a cost for the defender as he seeks to reduce the overlap between neighboring nodes[1] by reducing the power of the neighboring nodes while satisfying a minimum coverage constraint such that, $\sum_v P_v \geq C_{\min}$.

To capture the impact inflicted by the jammer, we consider its effect on the signal to noise ratio of a user served by AP v,

$$T_3(v) = \frac{\alpha_v P_v}{N_0 + h_v J_v}, \tag{3.19}$$

where α_v represents an *average* channel gain between AP v and a receiver within its coverage area. The average gain h_v is that of a channel between the jammer and the receiving device and N_0 denotes the additive white Gaussian noise level.

We can readily define the reward function of the defender $R(\mathbf{P}, \mathbf{J}) = \sum_v \theta_v T_1(v) - \gamma_v T_2(v) + T_3(v)$

[1]Even if neighbors are transmitting at non-interfering frequencies, undue overlap amounts to waste of resources since multiple APs cover the same area.

yielding the following reward,

$$R(\mathbf{P},\mathbf{J}) = \sum_v \theta_v K_v P_v J_v - \eta_v \left(\sum_{u\in\mathcal{N}_v, u\in\mathcal{I}_v} P_u \right) + \frac{\alpha_v P_v}{N_0 + h_v J_v}, \tag{3.20}$$

where θ_v and η_v, are weights with appropriate units.

To obtain the players best responses for the described zero-sum continuous game, the defender solves

$$\begin{aligned} \underset{\mathbf{P}}{\text{maximize}} \quad & R \\ \text{subject to} \quad & P_L \leq P_v \leq P_U, \qquad \forall v \\ & \sum_v P_v \geq C_{\min} \end{aligned} \tag{3.21}$$

while the jammer solves

$$\begin{aligned} \underset{\mathbf{J}}{\text{minimize}} \quad & R \\ \text{subject to} \quad & \sum_v J_v \leq \bar{\mathbf{J}}. \end{aligned} \tag{3.22}$$

The reward function of the defender (maximizer), R, is linear and continuously differentiable in $\mathbf{P} \in \mathbb{R}^N$, and convex and differentiable in $\mathbf{J} \in \mathbb{R}^N$ for the jammer (minimizer). Therefore, each player maximizes his own utility function for a given strategy of his opponent, yielding a best response function in terms of the opponent strategy. The NE strategy point, $\mathbf{z} = [\mathbf{P},\mathbf{J}]^T \in \mathbb{R}^{2N}$, lies at the intersection of the produced best-response functions. Since each player is solving an optimization problem that also depends on a set of variables shared with his opponent, at equilibrium the best strategy for both optimization problems can be simultaneously satisfied for both

players [80]. Therefore, an optimal strategy for the defender, denoted $\mathbf{P}^*$, must satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions for the optimization problem characterized in (3.21). Similarly, the optimal jamming strategy that minimizes the reward function must satisfy the KKT conditions of the optimization problem defined in (3.22), [81]. Since the constraints for both players are not coupled, the Lagrange multipliers of the two optimization problems are independent. Hence, a NE can be obtained by finding the optimal solution $\mathbf{z}^*$ that simultaneously satisfies the KKT conditions of both optimization problems, [80]. To characterize the optimal solution, we write the Lagrangian for (3.21),

$$L_d = R - \sum_{v=1}^{N} \left(\lambda_v (P_v - P_U) - \mu_v (P_L - P_v)\right) - \rho (C_{\min} - \sum_{v} P_v) \,. \tag{3.23}$$

The optimal power assignment strategy $\mathbf{P}^*$ for the defender must satisfy the following KKT conditions,

- Equality conditions:

$$P_v \frac{dL_d}{dP_v} = 0 \qquad\qquad \forall v = 1, \ldots, N \tag{3.24}$$

$$\lambda_v \frac{dL_d}{d\lambda_v} = 0, \quad \mu_v \frac{dL_d}{d\mu_v} = 0 \quad \forall v = 1, \ldots, N \tag{3.25}$$

$$\rho \frac{dL_d}{d\rho} = 0, \tag{3.26}$$

- Inequality conditions:

$$\frac{dL_d}{dP_v} \leq 0 \quad P_v \geq 0 \quad \forall v = 1, \ldots, N \tag{3.27}$$

$$\lambda_v \geq 0, \quad \mu_v \geq 0 \quad \forall v = 1, \ldots, N, \quad \rho \geq 0. \tag{3.28}$$

Similarly, the Lagrangian for the optimization problem (3.22) of the jammer is

$$L_a = R + \nu \, (\sum_{v=1}^{N} J_v - \bar{\mathbf{J}}). \tag{3.29}$$

An optimal jamming strategy $\mathbf{J}^*$ must satisfy the following KKT conditions

- Equality conditions:

$$J_v \frac{dL_a}{dJ_v} = 0 \quad \forall v = 1, \ldots, N \tag{3.30}$$

$$\nu \frac{dL_a}{d\nu} = 0, \tag{3.31}$$

- Inequality conditions:

$$\frac{dL_a}{dJ_v} \geq 0, \quad J_v \geq 0 \quad \forall v = 1, \ldots, N \tag{3.32}$$

$$\nu \geq 0, \tag{3.33}$$

Since the equations generated by the KKT conditions are nonlinear, we used the Newton method [82] to obtain the NE solution of this game. In addition, this NE $(\mathbf{P}^*, \mathbf{J}^*)$ is unique as stated in the next theorem. We establish uniqueness by leveraging a result in [83], which provided sufficient condition for NE uniqueness.

Theorem 5. *Given the utility function in (3.20), the pure strategies* $\mathbf{P}^*$ *and* $\mathbf{J}^*$ *that satisfy the conditions in (3.24)-(3.33) are the pure strategy NE and this equilibrium is unique.*

Proof. The fact that $\mathbf{P}^*$ and $\mathbf{J}^*$ which satisfy the KKT conditions form a pure NE follows directly from [81] as argued earlier in (3.3.3). It remains to establish uniqueness.

The utility function R in (3.20) is clearly linear in $\mathbf{P}$. To show that the jammer's reward function is concave in $\mathbf{J}$, we find the $N \times N$ Hessian matrix $\mathbf{Z} = diag\{Z_{v,v}\}$

, where $Z_{v,v} = \frac{-2\alpha_v h_v^2 P_v}{(N_0+h_v J_v)^3},\ v = 1, ..., N$. $\mathbf{Z}$ is diagonal with all negative elements, hence it is negative definite. Hence, the jammer's reward function, $-R$, is concave in $\mathbf{J}$.

We now return to the proof of uniqueness in Theorem 5. Consider a strategic form game as defined in Section 3.2.3. Assume that the strategy sets S_d (defender) and S_j (jammer) are given as defined in the constraints of (3.21) and (3.22), respectively, where these constraint sets are linear functions, and there exist some $\tilde{\mathbf{P}} \in \mathbb{R}^N$ and $\tilde{\mathbf{J}} \in \mathbb{R}^N$ for which the constraints in (3.21) and (3.22) are satisfied. To show that the game has a unique pure strategy NE, it suffices to show that the payoff functions $(R, -R)$ are diagonally strictly concave for $\mathbf{P} \in S_d$ and $\mathbf{J} \in S_j$ [83].

Definition 3. *(Diagonal strict concavity [83]): Let $\mathbf{x} = [\mathbf{P}, \mathbf{J}]^T$ denote the combined strategy played by the two players. Let (u_1, u_2) denote the payoff functions such that, $u_1 = R$ and $u_2 = -R$. The payoff functions (u_1, u_2) are said to be diagonally strictly concave for $\mathbf{x} \in S_d \times S_j$, if for every $\mathbf{x}^*$, $\bar{\mathbf{x}} \in S_d \times S_j$, we have that*

$$(\bar{\mathbf{x}} - \mathbf{x}^*)^T \nabla u(\mathbf{x}^*) + (\mathbf{x}^* - \bar{\mathbf{x}})^T \nabla u(\bar{\mathbf{x}}) > 0, \tag{3.34}$$

where $\nabla u(\mathbf{x}) = [\nabla_1 u_1(\mathbf{x}), \nabla_2 u_2(\mathbf{x})]^T$, and where

$$\nabla_1 u_1(\mathbf{x}) = \left[\frac{\partial u_1(\mathbf{x})}{\partial P_1} \quad \frac{\partial u_1(\mathbf{x})}{\partial P_2} \quad \dots \quad \frac{\partial u_1(\mathbf{x})}{\partial P_N} \right]^T \tag{3.35}$$

$$\nabla_2 u_2(\mathbf{x}) = \left[\frac{\partial u_2(\mathbf{x})}{\partial J_1} \quad \frac{\partial u_2(\mathbf{x})}{\partial J_2} \quad \dots \quad \frac{\partial u_2(\mathbf{x})}{\partial J_N} \right]^T. \tag{3.36}$$

A sufficient condition for Diagonal Strict Concavity was further given in [83], which we state next for completeness. Let the strategy sets S_d and S_j be as defined earlier. Assume that the symmetric matrix $(U(\mathbf{x})+U^T(\mathbf{x}))$ is negative definite for all $\mathbf{x} \in S_d \times S_j$, i.e., we have $y^T(U(\mathbf{x})+U^T(\mathbf{x}))y < 0, \forall y \in \mathbb{R}^{2N} \neq 0$, where $U(\mathbf{x})$ denotes the Jacobian of $\nabla u(\mathbf{x})$. Then, the payoff functions (u_1, u_2) are diagonally strictly concave for $\mathbf{x} \in S_d \times S_j$.

For our defined two-player game, $U(\mathbf{x})$ is a $2N \times 2N$ Jacobian matrix.

We have that

$$y^T(U(\mathbf{x}) + U^T(\mathbf{x}))y = \sum_{k=1}^{N} \sum_{l=1}^{N} (U_{k,l} + U_{l,k}) y_k y_l, \tag{3.37}$$

where,

$$U_{k,l} = \begin{cases} -U_{l,k} & ; \quad |l-k| = N, \\ \frac{-2\alpha_v h_v^2 P_v}{(N_0 + h_v J_v)^3} & ; \quad k = l = N+1 : 2N, \;\; v = K - N \\ 0 & ; \quad \text{otherwise} \end{cases} \tag{3.38}$$

Therefore,

$$y^T(U(\mathbf{x}) + U^T(\mathbf{x}))y = \sum_{k=N+1}^{2N} 2U_{k,l} y_k^2 < 0. \tag{3.39}$$

Therefore, the symmetric matrix $(U(\mathbf{x}) + U^T(\mathbf{x}))$ is negative definite, which completes the proof of Theorem 5. □

3.3.4 Extension to Multiple Bands

Our game formulations allow us to consider the new generation of wireless APs that support multiple wireless bands whereby the defender adjusts the power radius for each supported band. Let ω denote a specific wireless band in a set Ω of all supported bands (e.g., $\Omega =\{2.4$ GHz, 5 GHz$\}$). Since each band $\omega \in \Omega$ is generally treated independently from the other bands, this truly amounts to solving separate optimization problems, one for each band. As an example, we consider the second formulation which considers a marginalized strategy. In this case, the marginalized action space of the defender changes to

$$\mathcal{A}_d = \{\bar{\mathbf{P}} \in \mathcal{P} \subset \mathbb{R}^{N\times|\Omega|} : \bar{\mathbf{P}}_\omega^T \mathbf{1} \geq C_\omega \, , \forall \omega \in \Omega\}, \tag{3.40}$$

where $\bar{\mathbf{P}}_\omega$ is the power assignment vector corresponding to a wireless band $\omega \in \Omega$. The lower band C_ω generally depends on ω to reflect different coverage constraints for different bands. Thus, the defender seeks to solve the following game,

$$\begin{aligned} \underset{\bar{\mathbf{P}}\in\mathcal{A}_d}{\text{maximize}} \quad & \sum_{\omega} U(\omega) \\ \text{subject to} \quad & U(\omega) \leq -\sum_{v=1}^{N} (L_{kv}^{\omega}(\bar{\mathbf{P}}), 0)^+, \forall \omega \in \Omega, k = 1, \ldots, N \end{aligned} \tag{3.41}$$

This approach extends naturally to the other formulations, namely, the discrete and continuous games of Sections 3.3.1 and 3.3.3. In Section 2.5, we show results on a real topology in a 3-floor building for dual-band case where $\Omega = \{2.4$ GHz$, 5$ GHz$\}$. Each of the bands has different coverage constraints.

Remark 3. *For the multiple-band marginalized strategy with constraints (3.40), we have a pure assignment matrix, one column per frequency band. The corresponding constraint structure is also a hierarchy since the constraint sets per band are disjoint.*

3.4 Experimental Results

In this section we present numerical and simulation results obtained from extensive experimental studies for the proposed defense approaches for a variety of topologies and under different conditions and attack scenarios. Our assessment is performed on 3 types of topologies: (i) simple topologies that are shown in Fig. 3.1 that describe "building block" structures for mesh networks, (ii) a real deployment of a 28-node topology in a 3-story academic building (as shown later in Figures 3.7 and 3.10), and (iii) a larger 64-node random topology.

In general, we are interested in assessing the performance of our decomposition approaches in comparison to the full-game (which we can only solve for small networks due to the exponential complexity) and assessing the gap between them. We present theoretical results (obtained as solutions to the optimization problems presented in (3.2), (3.7) and (3.13)), as well as simulations obtained by playing the game using actions sampled from the computed randomized strategies and averaging over 1000 independent runs. Figs. 3.2 through 3.9 are for the first instantiation of the reward (Section 3.3.1), and Fig. 3.10 through 3.12 concern the second instantiation (Section 3.3.2). Performance evaluation for the continuous game (Section 3.3.3) is presented in Figs. 3.14, 3.15, 3.16 and 3.17.

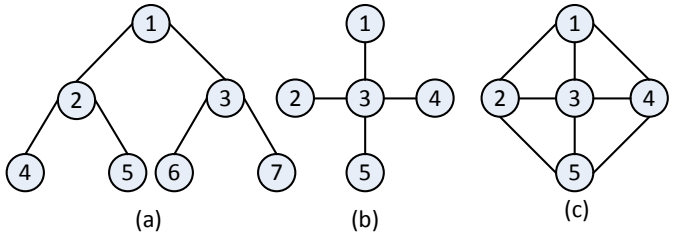

Figure 3.1: Building block topologies, (a) 7-node tree, (b) 5-node star, (c) 5-node interconnected network.

Defense and attack strategies: Fig. 3.2,3.3 and 3.4 show the defender's reward under different attack costs (h) for the building block topologies (a), (b) and (c) of Fig. 3.1. In this set of experiments we consider an adversary jamming at most one node. Our results show that the defender's reward obtained by the decomposition-based approach (the resulting strategy is termed decomposed NE) performs closely to the NE obtained without decomposition, i.e., when solving jointly for the power profiles (termed full game NE). Unless otherwise specified, the decomposed NE is when both players apply a decomposition.

The decomposed NE is shown to yield a slightly lower reward for the defender (higher reward for jammer) particularly in the low attack cost regime due to the all-high assumption for the neighbors in solving the sub-games (cf. Sec. 3.3.1.1), which in turn motivates the defender to play more conservatively when attacks are emboldened by the lower attacking cost. When the attack cost is sufficiently high, both strategies converge to the same value of the game as the jammer chooses to back-off with higher probability. Fig. 3.2,3.3, and 3.4 also show the performance as the defender continues to play his best response against a random attack strategy in which the jammer picks one node at random to attack regardless of the attack cost value. Since the jammer is unilaterally changing its strategy, the reward of the defender increases by the definition of the NE.

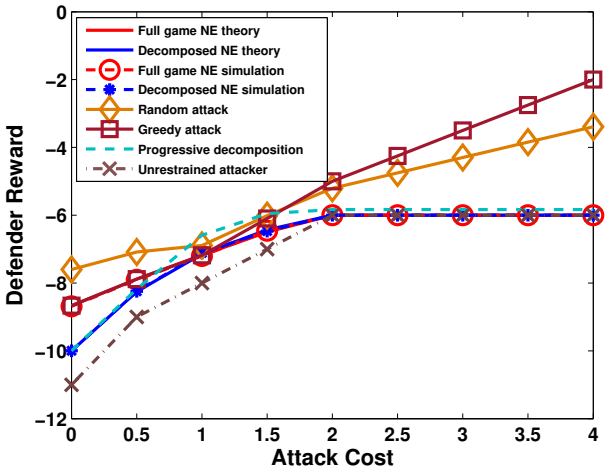

Figure 3.2: Defender reward for the 7-node tree network topology versus attack cost.

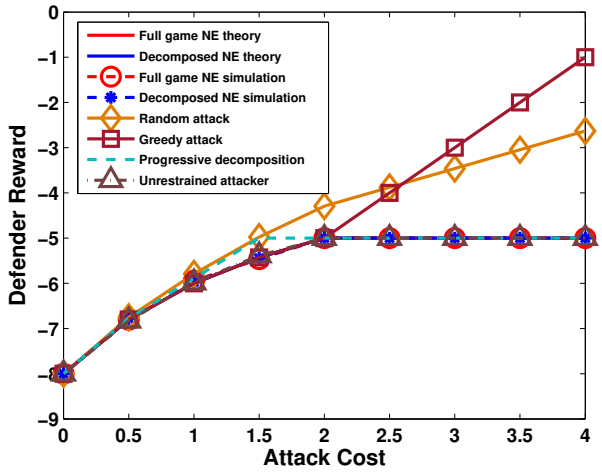

Figure 3.3: Defender reward for different attack costs for the 5-node star network.

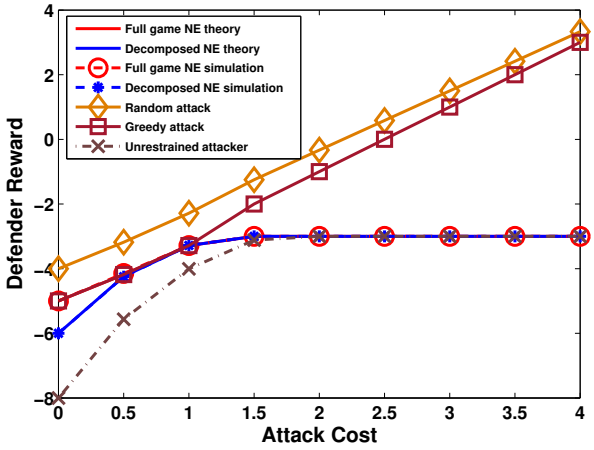

Figure 3.4: Defender reward for the 5-node interconnected network topology vs attack cost.

In addition, we compare the performance of the defender's strategy against a jammer that targets the highest degree node (greedy attack). It is shown that the defender performs fairly close to the NE for low attack cost (e.g., values less than 1 for topologies (a) and (c), and 2 for topology (c)) as attacking high-degree nodes can induce a large number of conflicts while being fairly uncostly. As the cost of the attack increases, however, the greedy attack shows to be very expensive for the jammer and enables the defender to secure larger rewards.

Unrestrained adversary: Our investigations have shown that the defense strategy based on decomposition yields comparable performance to that of the (non-decomposed) NE (when its computation is non-prohibitive). However, obtaining an analytical characterization of the performance gap is rather challenging due to the aforementioned coupling inherent to the minimax optimization problem. In order to better understand the performance gap, we also study the defender's reward when only the defender (unilaterally) uses the decomposition to calculate his defense strategy. The attacker, on the other hand, has the computational capability to obtain the optimal (non-decomposed) attack strategy, hence the designation 'unrestrained adversary'.

In Figs. 3.2,3.3 and 3.4, the defender plays a decomposed NE strategy against a computationally unrestrained adversary that plays its NE strategy (i.e. the adversary does not decompose the game). Since this corresponds to a unilateral change of strategy by the defender, the reward of the defender has to degrade by definition of the NE. However, the defender is still able to perform fairly well even when the adversary has unlimited computational capability. More importantly, the small gap from the full game NE indicates that the defender's decomposed NE strategy only incurs minimal loss in reward.

Fig. 3.2,3.3 and 3.4, also shows the reward with the progressive decomposition described at the end of Section 3.3.1.2. The reward is fairly close, and sometimes higher (c.f. Remark 2), than that of the full game NE. Since this is a successive approach in which the solution to sub-games are successively fed into subsequent ones, the outcome will depend on the order of the nodes. For fair comparison, the order of the nodes was selected randomly.

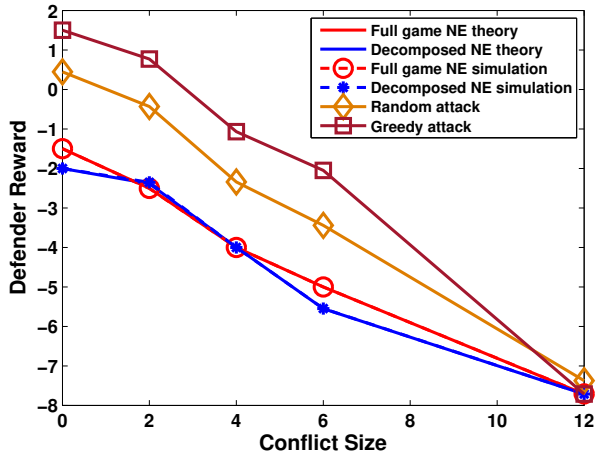

Figure 3.5: Defender reward for topology (a) vs number of conflicts.

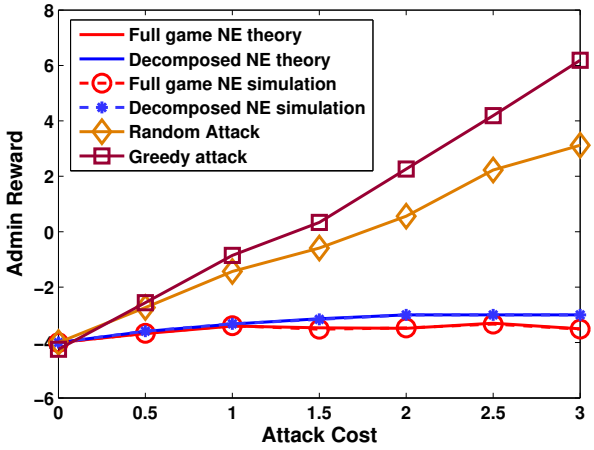

Figure 3.6: Defender reward for topology (a) vs attack cost against two colluding jammers.

Number of conflicts: Fig. 3.5 shows the defender's reward with and without decomposition at attack cost $h = 1$ for topology (a) as a function of the number of conflicts that exist prior to playing the game. In a network with a small number of initial conflicts, the jammer's task is harder, hence the larger reward for the defender. In this regime, the defender is able to play less conservatively and to safely assign '*high*' power levels to a larger number of nodes. The defender's reward increases if the adversary chooses to use a random or greedy attack, and the gap between the decomposed and full game NE strategies is negligible.

Colluding jammers: Fig. 3.6 shows the defender's reward for topology (a) as a function of the attack cost with two colluding jammers. We also compare the performance of the defender's NE strategy when the jammers deviate from the NE and choose to launch a random or greedy attack leading to higher reward for the defender.

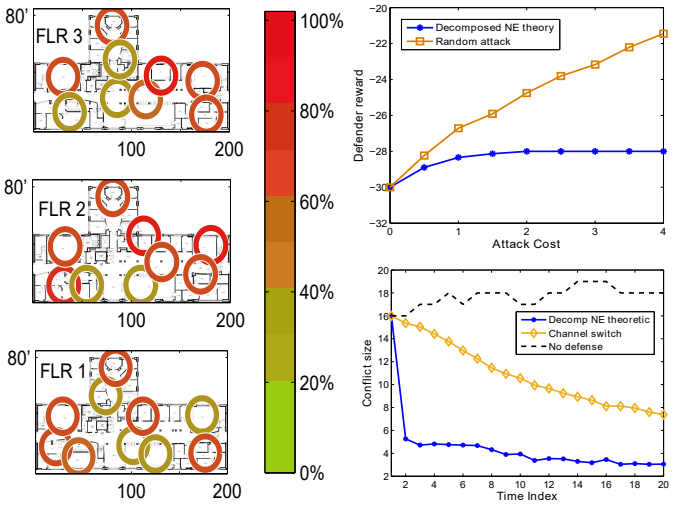

Figure 3.7: (Left) Mixed strategy for power assignment of 28 nodes in 3-floor building, (Right) The top plot shows the defender reward vs the attack cost, and the bottom plot displays the time evolution of the conflicts with and without defense.

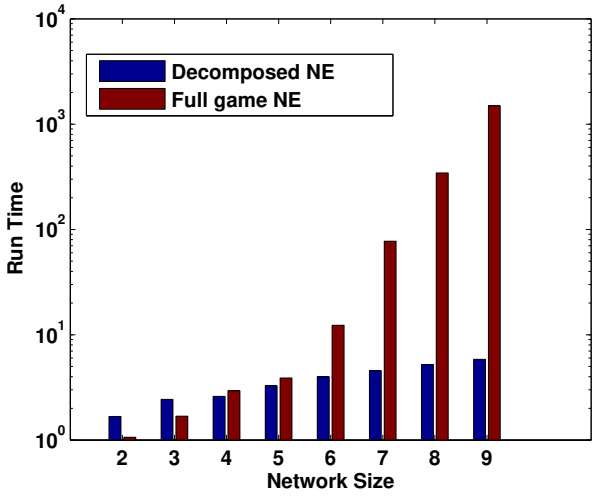

Figure 3.8: Run time in seconds.

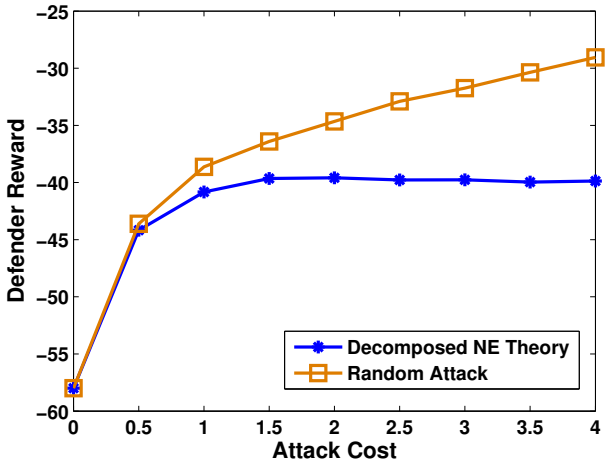

Figure 3.9: Defender's reward for a randomly generated 64-node topology.

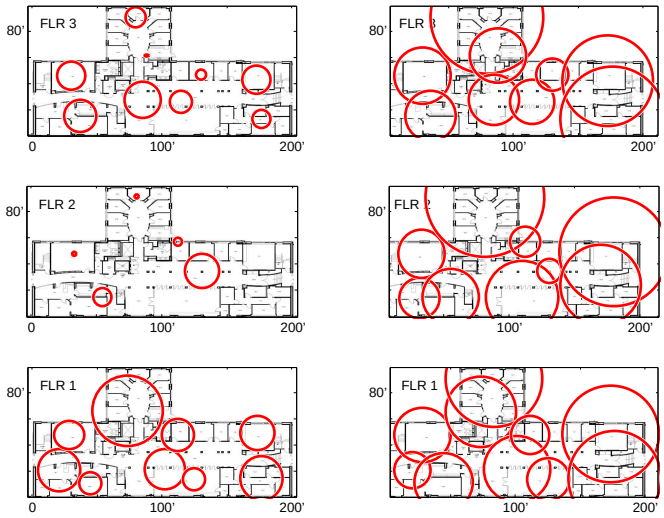

Figure 3.10: Defender's strategies in a 3-floor building. 2D projection of the 3D footprint of each access point, without coverage constraints (left) and with coverage constraints (right).

Real topology and value of defense: Our next experiment considers a real topology in one of our academic buildings. The building consists of three floors with 10, 9 and 9 CISCO APs deployed

on the first, second and third floor, respectively. In Fig. 3.7 (Left) we show the mixed strategy for power assignments for the 28 nodes using a color map on each floor. We used fixed radii to indicate node locations. Colors corresponding to higher percentages designate a higher probability of assigning 'high' power level to a particular node. Note that for this network size computing a full game NE is computationally prohibitive, so this mixed strategy is obtained through decomposition. For the same topology, the top plot of Fig. 3.7 (Right) shows that the utility of the defender increases significantly if the adversary deviates unilaterally from a decomposed NE and uses a random attack. To demonstrate the gains emerging from using the defense strategy, we study the evolution of the number of conflicts over time with and without defense as the game is played repeatedly. Each game is initialized with the residual conflicts from the previous game. The lower curve in the bottom plot of Fig. 3.7 (Right) shows the evolution when the defender adapts the footprints using the decomposed NE. The middle curve is when the defender resolves the conflicts by re-assigning the channels – each time switching the channel causing the largest number of conflicts – but otherwise does not adapt the transmission footprints. The upper curve corresponds to a no-defense scenario. In each scenario, the adversary plays its decomposed NE strategy. The proposed defense strategy quickly resolves conflicts, thereby successfully protects the network, and outperforms channel re-assignment. This experiment also underscores the value of using such a defense mechanism versus no defense.

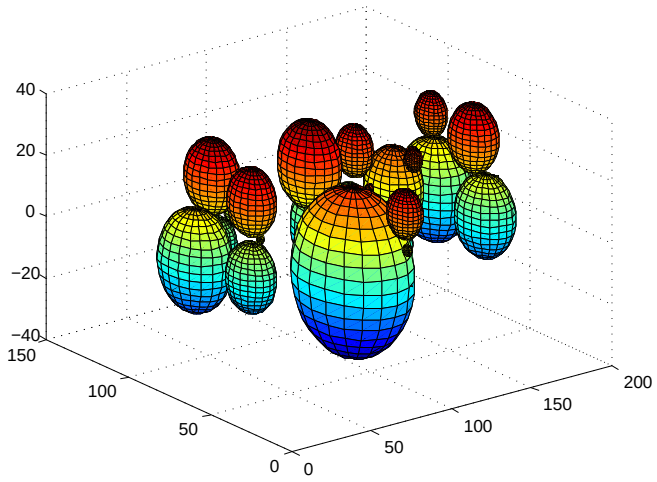

Figure 3.11: 3D wireless footprints based on marginal strategy.

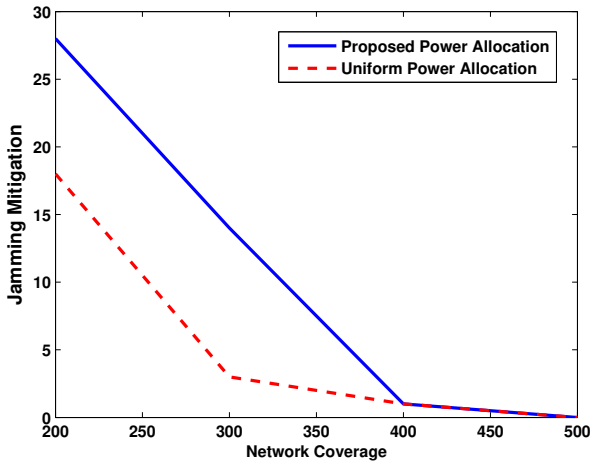

Figure 3.12: Tradeoff between the number of non-conflicting APs and the minimum network coverage.

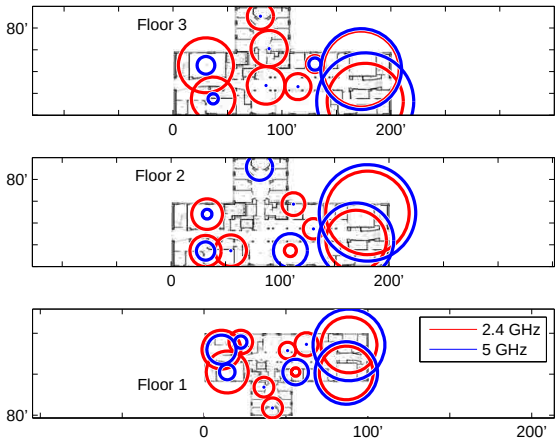

Figure 3.13: Power assignment in 3-floor building over two different wireless bands to enhance coverage

Complexity reduction: The proposed decomposition brings about significant reduction in complexity (c.f. Section 3.3.1.1 and 3.3.1.2). In Fig. 3.8 we compare the run time in computing a full game and decomposed NE as we increase the network size. The complexity for the full game NE increases exponentially in the size of the network – hence is computationally intractable beyond a certain point – versus a linear increase for the decomposed NE. As such, in Fig. 3.9 we obtain the reward based on a decomposed NE (and compare to random attack) for a randomly generated topology of 64 nodes. The decomposed strategy is perfectly scalable.

Marginal strategy: Our next experiment considers the same 28-node topology inside the building. We identify the marginal strategy of the two player zero-sum game developed in Section 3.3.2 obtained as a solution to (3.13) subject to the coverage constraint. Fig. 3.10 shows the defender's marginal strategy where we plot the 2D projection of the computed 3D wireless footprints for each AP on each floor. A sample of computed 3D wireless footprints of the first and second floors is shown in Fig. 3.11. Fig. 3.10 (left) shows the footprints with no coverage constraints – in this case the marginal strategy amounts to no overlap between the nodes achieving zero value for the game.

Fig. 3.10 (right) shows the footprints subject to a coverage constraint. Herein, we used a minimum coverage that is twice the actual coverage that resulted from the marginal strategy of Fig. 3.10 (left).

In this case, the overlap maximizes the worst case utility of the defender. While the deployed CISCO nodes do not implement or optimize an actual defense mechanism against adversaries, this case study partially explains an actual poor wireless coverage phenomenon on the second floor of the building. In particular, as the occupancy on and near the first floor tends to be high in this busy area on campus, the APs on the first floor expand their footprints to serve more coverage based on the TPC protocol of their RRM system. This in turn would trigger the APs on the second floor to reduce their footprint to avert significant overlap with the ones on the first floor akin to the middle plot of Fig. 3.10 (left) which uses no coverage constraints, leading to poor coverage. Note that while the footprints in Fig. 3.10 are chosen to limit the worst-case effect of jamming attacks, they share a similar objective with radio resource management – that is to minimize overlap between the nodes. This issue has been brought up to IT who are currently investigating some solutions. At a high level, this issue may be resolved by enforcing a minimal coverage constraint as in the right figure. To further demonstrate the effectiveness of our proposed power adaptation approach, in Fig. 3.12 we plot the tradeoff between the network immunity to jamming – measured in terms of the number of APs with no conflicts (overlap) – and the network coverage, i.e., different values of $C_{\min}$. We compare our proposed scheme to a power assignment scheme that uniformly allocates the power levels across all the nodes. For the same network coverage, the proposed NE strategy is shown to achieve about 36% improvement in jamming mitigation over uniform power allocation when $C_{\min} \leq 300$ power units. The gains are pronounced in the low $0 \leq C_{\min} \leq 300$ and medium $300 \leq C_{\min} \leq 400$ coverage regimes as the proposed scheme takes the node degrees into account when assigning the power levels. For more demanding coverage requirements, the tradeoffs are shown to concide since the maximum power is used up to meet the coverage constraint and there

is no freedom to avoid overlap between many nodes. In Fig. 3.13, we show the power radii when each AP is supporting two wireless bands as in the next generation APs to enhance the network coverage and the QoS. We consider two different coverage constraints for each band. In practice, the 5 GHz band has fewer applications in specific locations, hence is associated with less stringent coverage constraints.

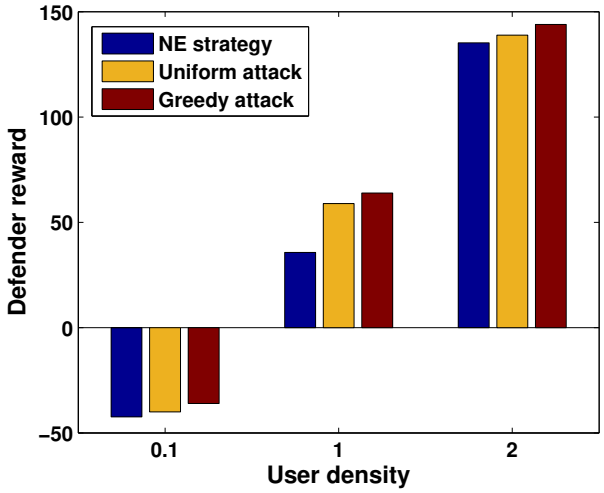

Figure 3.14: Defender reward for the 7-node tree network topology versus attack cost.

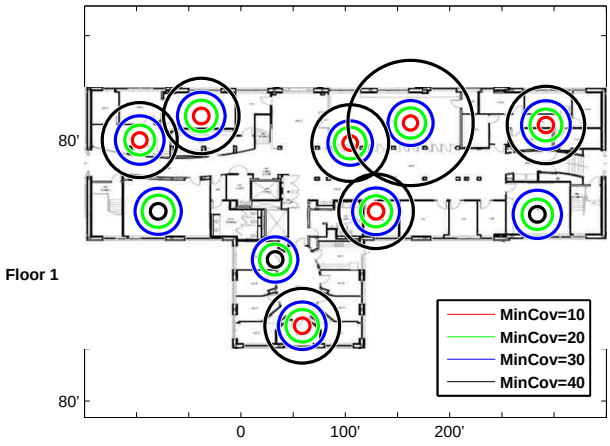

Figure 3.17: Power radii for different minimum coverage constraints.

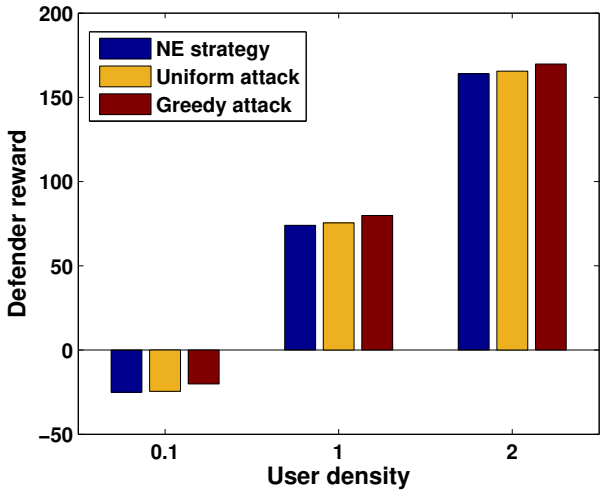

Figure 3.15: Defender reward for different attack costs for the 5-node star network.

In order to identify the unique NE for the continuous game in (3.21), the KKT equalities and inequalities should be satisfied simultaneously for both players. We used Newton's method to solve the system of nonlinear equations numerically [81]. In Figs. 3.14, 3.15 and 3.16 we plot the defender's reward for the building block topologies.For this experiment, we set $P_L = 1$, $P_U = 5$ and $C_{\min} = 14$, as defined in 3.3.3. We compare the performance of the NE strategy to other strategies where the adversary deviates unilaterally from the NE. In one instance, the jammer chooses to attack only middle nodes (Greedy attack), and in another distributes the jamming power uniformly across all 5 nodes regardless of their relative importance (Uniform attack). In both cases, this indeed leads to higher reward for the network defender, equivalently loss in the jammer's reward. Fig. 3.17 demonstrates how the overlap increases by enforcing a more strict coverage constraint. These results are obtained for the topology of the first floor of the building.

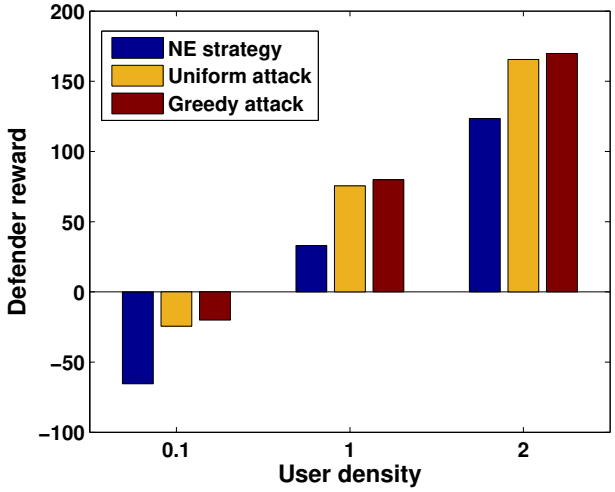

Figure 3.16: Defender reward for the 5-node interconnected network topology vs attack cost.

3.5 Stochastic Games

A stochastic game Γ is defined as a tuple $(\mathcal{S}, \mathcal{N}, \mathcal{A}, P, \mathbf{R})$, where

- $\mathcal{S}$ is a finite set of games, where each game represents a state. Hence, $\mathcal{S}$ is the state-space of the game Γ.
- $\mathcal{K}$ is a finite set of k players.
- $\mathcal{A} = \mathcal{A}^1 \times ... \times \mathcal{A}^k$, is a game action space, where the action space $\mathcal{A}^i$ is a finite set of all possible actions for player i.
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is a transition probability function, where $P(s, a, s')$ is the probability of transitioning from state s to another state s' after action profile $a = (a^{(1)}, ..., a^{(k)})$ is played.
- $\mathbf{R} = (R^1, \ldots, R^k)$, where $R^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is a real-valued payoff function for player i.

Stochastic games provide a fairly broad framework as they generalize both MDPs and repeated games. An MDP is a stochastic game with only one player, while a repeated game is a stochastic game with a single stage. In the next section, we formulate a dynamic stochastic game-theoretic framework to identify optimal defense policies against jamming attacks based on adaptive network topologies. In this section, we present our stochastic Markov game formulation. A Markov game can be seen as an MDP in which multiple agents influence the system dynamics. Such games have several important properties. Every discounted reward Markov game has a non-empty set of optimal policies, at least one of which is stationary. An optimal stationary policy is independent of the time at which the game is played and only depends on the current state of the game [7],[84].

Following the definition of stochastic games introduced in the previous section, we define our stochastic game as follows:

• A state $s \in \mathcal{S}$ is a normal form game played by two players, where $\mathcal{S} = \{\mathsf{s}^1, \ldots, \mathsf{s}^{|\mathcal{S}|}\}$. We generally use s to denote an arbitrary state. A state at time t is denoted $s_t = (\mathbf{H}_t, \mathbf{c}_t)$, where $\mathbf{H}_t$ is the network adjacency matrix and $\mathbf{c}_t$ is the channel assignment vector. Note the dependence of the network topology and the channel assignment on t, the matrix $\mathbf{H}_t$ depends on the power assignment $a^{(d)}$ set by the defender (defined below). The channel assignment vector $\mathbf{c} = [c_1, \ldots, c_N]^T$, where c_v is the channel assigned to node v. The channel assignment is controlled by a graph coloring algorithm as discussed earlier. We refer to this algorithm as the system action which represents a source of randomness as an exogenous variable. In case of conflicts, the system takes an action to resolve these conflicts sequentially through channel re-allocation.

• $\mathcal{K} = \{\text{defender}, \text{attacker}\}$ is a set of 2 players.

• The action space $\mathcal{A} = \mathcal{A}^d \times \mathcal{A}^a$, where $\mathcal{A}^d = \{\mathsf{d}^1, \ldots, \mathsf{d}^{|\mathcal{A}^d|}\}$ and $\mathcal{A}^a = \{\mathsf{a}^1, \ldots, \mathsf{a}^{|\mathcal{A}^a|}\}$ are the action spaces for the defender and the attacker, respectively. A pure action $a^{(d)} \in \mathcal{A}^d$, is a vector of length N, where each entry $a^{(d)}(v)$ represents the power level assigned to node v. For simplicity,

we assume that $a^{(d)}(v) \in \{0, 1\}$, where $a^{(d)}(v) = 0$ means the defender reduces the power level of node v such that node v is not heard by any of its neighbors, and $a^{(d)}(v) = 1$ means that node v operates at its maximum possible coverage with respect to all of its neighbors.

A pure action for the attacker $a^{(a)} \in \mathcal{A}^a$ is a binary vector of length N. The choice $a^{(a)}(v) = 1$ means the attacker is attacking node v, otherwise $a^{(a)}(v) = 0$. For simplicity and without loss of generality, we assume the attacker has a limited power budget so that it can only attack one node at a time or choose to completely back off.

• $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function, i.e., $P(s, a, s')$ is the probability of transitioning to a future state s' from the current state s after action profile $a = (a^{(d)}, a^{(a)})$.

• $\mathbf{R} = \left(R^d, R^a\right)$, is the real-valued payoff function. We consider a zero-sum game, hence $R^d = -R^a = R$. We consider a discounted long-term reward objective in which rewards are accumulated over time. The immediate reward of the defender due to a transition from the current state $s_t = s$ to state $s_{t+1} = s'$ is defined a

$$R(s, a, s') = \sum_{v \in \mathcal{V}} \left(h \cdot \mathbf{1}_{\{a^{(a)}(v)=1\}} \frac{\delta_v^o + \delta_v(s')}{2} - \mathbf{1}_{\left\{a^{(d)}(v)=0\right\}} - \beta_v(s') \right) \tag{3.42}$$

The first term of the summand on the RHS of (3.42) accounts for the cost paid by the attacker for attacking node v, where h is the attack cost parameter and $\mathbf{1}_{\{.\}}$ is the indicator function. Recall that the topology varies from its original state according to the power transmission profile specified by the defender's actions. Hence, at terminal state s', the node degree $\delta_v(s')$ varies accordingly from its maximum value δ_v^o, where δ_v^o is the node degree when all the nodes are transmitting at their maximum power levels. Thus, considering any node $v \in \mathcal{V}$, the attacker's cost is the average of two terms. The first cost term is a cost $h\delta_v(s')$ that captures the node degree after the game is played reflecting the risk of getting exposed as the attacker is avoiding attacking very dense areas.

The other cost term is $h\delta_v^o$, which is a constant penalty reflecting the original importance of the attacked node since hub nodes that have higher degrees are more expensive to attack than leaf nodes. The attack cost is a reward for the defender.

The second term in (3.42) is the cost incurred by the defender for the coverage holes captured by the number of nodes transmitting at '*low*' power level.

The third term accounts for the conflicts in the network, where $\beta_v(s')$ is the conflict size for node v, defined in (3.43) which counts the number of nodes in the neighborhood of v with channels in its interference set that are either assigned high power or attacked by the adversary.

$$\beta_v(s') = \sum_{u \in \mathcal{D}_v} \left(\mathbf{1}_{\left\{a^{(d)}(u)=1\right\}} + \mathbf{1}_{\left\{a^{(d)}(u)=0\right\}} \mathbf{1}_{\{a^{(a)}(u)=1\}} \right) \tag{3.43}$$

where we have made the dependence of the conflict set $\mathcal{D}_v$ defined in (2.4) on the future state s' explicit since $\mathcal{D}_v$ depends on the neighborhood set and the channel assignment, which are both part of the state.

3.5.1 Mixed strategy, state evolution and objective function

Unlike MDPs, the optimal stationary policy for stochastic games need not be deterministic. Since, we will be searching for an optimal mixed (randomized) stationary policy we start with a definition of stationary mixed strategies for both players.

Let $x^i(s), i = 1, \ldots, |\mathcal{A}^d|$, and $y^j(s), j = 1, \ldots, |\mathcal{A}^a|$, denote the probability that the defender and attacker play the pure action $\mathsf{d}^i \in \mathcal{A}^d$ and $\mathsf{a}^j \in \mathcal{A}^a$ while at state s, respectively. A mixed strategy is then defined as a probability distribution over the whole action space. Specifically, given a state

s, $\mathbf{x}(s) = [x^1(s), \ldots, x^{|\mathcal{A}_d|}(s)]^T$ and $\mathbf{y}(s) = [y^1(s), \ldots, y^{|\mathcal{A}_a|}(s)]^T$ are the mixed strategies for the defender and the attacker at state s, respectively. A stochastic stationary policy is readily defined as $\pi = \{\mathbf{x}(s^1), \ldots, \mathbf{x}(s^n)\}$ for the defender, and $\theta = \{\mathbf{y}(s^1), \ldots, \mathbf{y}(s^n)\}$ for the attacker, where n is the total number of states.

The network evolves from state s at time t to state s' at time $t+1$ according to the evolution equation

$$s' = f(s, a^{(d)}, a^{(a)}, \zeta_t) \tag{3.44}$$

for some deterministic function f, where ζ_t is a random variable representing the system action based on a graph coloring algorithm for channel assignment that picks one node in conflict at random and re-assigns it to a non-interfering channel if such a channel exists. Eq. (3.44) defines the transition probability $P(s, a, s')$.

The defender goal is to maximize the expected sum of discounted rewards. Under randomized stationary defense and attack policies π and θ, the expected sum of discounted rewards starting from state $s \in \mathcal{S}$ at time $t = 0$ is,

$$V(s, \pi, \theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t^{(d)}, a_t^{(a)}, s_{t+1}\right) \middle| s_0 = s, \pi, \theta\right], \tag{3.45}$$

where the expectation is over the randomness of the players' actions (recalling that π and θ are randomized policies) and the state evolution (the exogenous variables). Subscript t denotes the t-th stage and $0 < \gamma < 1$ is a discount factor. The goal is to solve for the optimal stationary policies for both players, i.e., obtain the solution to

$$V^*(s) := V(s, \pi^*, \theta^*) = \max_{\pi} \min_{\theta} V(s, \pi, \theta). \tag{3.46}$$

The optimal randomized stationary policies $\mathbf{x}^*(s)$ and $\mathbf{y}^*(s)$ for state s are the solutions to

$$V^*(s) = \max_{\mathbf{x}(s)} \min_{\mathbf{y}(s)} \mathbb{E}\left[R(s, a^{(d)}, a^{(a)}, s') + \gamma V^*(s') \middle| \mathbf{x}(s), \mathbf{y}(s)\right] \tag{3.47}$$

where the immediate reward term is given by

$$\mathbb{E}\left[R(s, a^{(d)}, a^{(a)}, s') \middle| \mathbf{x}(s), \mathbf{y}(s)\right] = \sum_{s' \in \mathcal{S}} \sum_{i=1}^{|\mathcal{A}^d|} \sum_{j=1}^{|\mathcal{A}^a|} R(s, \mathsf{d}^i, \mathsf{a}^j, s') x^i(s) y^j(s) p(s' | s, \mathsf{d}^i, \mathsf{a}^j). \tag{3.48}$$

Thus, the optimal stationary policies are $\pi^* = \{\mathbf{x}^*(\mathsf{s}^1), \ldots, \mathbf{x}^*(\mathsf{s}^n)\},\ \theta^* = \{\mathbf{y}^*(\mathsf{s}^1), \ldots, \mathbf{y}^*(\mathsf{s}^n)\}$.

3.5.2 Q-minmax Algorithm

MDPs (single-player Markov game) can be solved using value iteration [45]. There, the total expected discounted reward $V^*(s)$ is termed the value of state s since larger rewards are collected from states with larger values of V^*. The value of a state $V^*(s)$ satisfies

$$V(s) = \max_{a' \in \mathcal{A}} Q(s, a'), \tag{3.49}$$

where $Q(s, a)$ is the quality of the state-action pair (s, a) defined as the total expected discounted reward attained by the non-stationary policy that takes action a then follows with the optimal policy from this point on. Given two players, this notion can be extended so that

$$Q(s, a^{(d)}, a^{(a)}) = \mathbb{E}\left[R(s, a^{(d)}, a^{(a)}, s') + \gamma V^*(s') \middle| a^{(d)}, a^{(a)}\right]$$

To solve (3.46) for the optimal policy, we adopt the Q-minmax value-iteration based algorithm of

Littman [85] extended to zero-sum stochastic games, which replaces the maximization in (3.49) with a maxmin operator to account for the actions of the second player. Hence,

$$V^*(s) = \max_{\mathbf{x}(s)\in\chi(\mathcal{A}^d)} \min_{a^{(a)}\in\mathcal{A}^a} \sum_{i=1}^{|A^d|} Q(s, \mathsf{d}^i, a^{(a)}) x^i(s), \tag{3.50}$$

where $\chi(\mathcal{A}^d)$ is the space of probability distributions defined over $\mathcal{A}^d$. In the next section, we discuss the complexity of the Markov game and propose an approach to handle complexity.

3.6 Approximate Policy

The complexity of the formulated game grows exponentially in the network size N and in the number of usable channels $|\mathcal{C}|$. Even with limiting the number of power levels to 2 levels, the number of possible states is $2^N \times |\mathcal{C}|^N$. In order to implement the Q-minimax algorithm, one needs to iterate over all possible states, which is computationally prohibitive even for medium-size networks. In addition to enumerating all the states, in each iteration the algorithm solves for a NE of a matrix game of size $2^N \times (N+1)$. We focus on addressing the state-space complexity problem. In our previous work [86], we developed a decomposition-based approach for static games that can substantially speed up the search for a NE in large scale networks. Next, we propose an approach to reduce the state-space size using features to represent each state.

Approximate value iteration: The idea underlying this approach is to use representative states and features instead of enumerating all possible states. A set of representative states is selected for 'training' for policy improvement. These states should be selected so as to capture most of the possible system evolutions. Here, we select representative states containing a spectrum of possible conflict states, ranging from minimal conflicts or conflict-free assignment to maximum conflicts where all APs are assigned the same channel. The assignment to APs may start from any default

channel, so an experimental run may begin from any number of conflicts.

From these representative states, a set of features are extracted to capture the characteristics of the state and its conflict profile. A weighted sum of those features is then used to approximate the value for a given state as

$$\tilde{V}(s) = \sum_{i=1}^{M} r_i \phi_i(s). \tag{3.51}$$

Where r_i is a weight for the i-th feature, $\phi_i(s)$ the value of the i-th feature for state s, and M the number of used features. Hence, the algorithm can find an approximate value for any state and plug it in equation (3.50) using this feature representation.

Monte Carlo simulations are used to evaluate the feature weights over a number of independent trajectories. In our simulations, we used the following 5 features to capture the degree of conflict in a given state. Feature ϕ_1 is the number of APs in conflict with one or more neighbors, ϕ_2 the ratio of maximum number of APs involved in the same conflict to the degree of the network in the original graph, ϕ_3 the average number of APs involved in the same conflict, ϕ_4 the average number of channels unavailable to an AP, and ϕ_5 the average conflict size of the highest degree AP(s).

3.7 Experimental Results

In this section, we evaluate our proposed defense approach using numerical simulations. We test the defense approach on different network topologies considered as building blocks for many real world networks. The numerical evaluation consists of two main steps. First, we implement the Q-minimax learning algorithm in order to learn the optimal policy for each state – this is the NE strategy upon convergence. Second, we evaluate the learned NE strategies through comparisons to other strategies.

3.7.1 Learning step

The Q-minimax algorithm proposed by Littman in [85] is a reinforcement learning algorithm. It follows the same procedure of the standard Q-value algorithm which starts by assigning initial values for each state [87]. Then, to solve the two-player zero-sum game the 'max' operator in the update step of the standard Q-learning algorithm is replaced by a 'minimax' operator, which can be evaluated by solving a linear program.

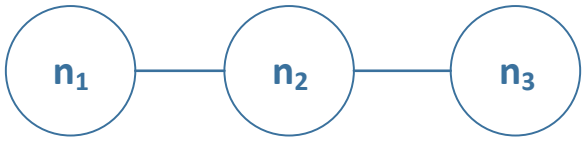

Figure 3.18: A 3-node chain network topology

For illustration, we start off with a very simple network that consists of three nodes forming a chain topology as shown in Fig. 3.18. Hence, the defender has 8 different pure actions and the attacker has 4 possible pure actions (including the no-attack action). Thus, when only 2 possible channels are usable, the state space consists of 32 states. Fig. 3.19 shows the convergence of the value function for a particular state using the Q-minimax algorithm. The value function converges in approximately 130 steps. The corresponding defense and attack strategies are plotted in Figs. 3.20 and 3.21. The defender's strategy converges to the NE (mixed) policy at this particular state as shown in Fig. 3.20, which is a probability 0.6 for pure action $a_d = [1\ 1\ 0]$ and probability 0.4 for action $a_d = [1\ 1\ 1]$, and zero probability for the remaining 6 pure actions. The attacker's strategy converges to the randomized strategy in which he attacks node 3 with probability 0.333 and backs off (no attack) with probability 0.6667.

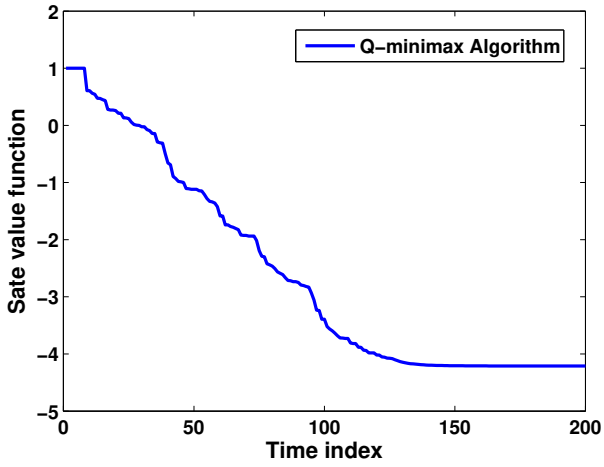

Figure 3.19: Convergence of the state value function for a 3-node chain network.

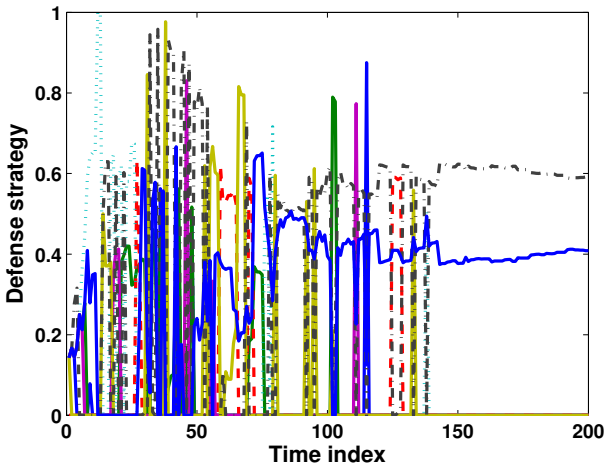

Figure 3.20: The defender strategy at a sample state for a 3-node chain network converges to a mixed NE strategy.

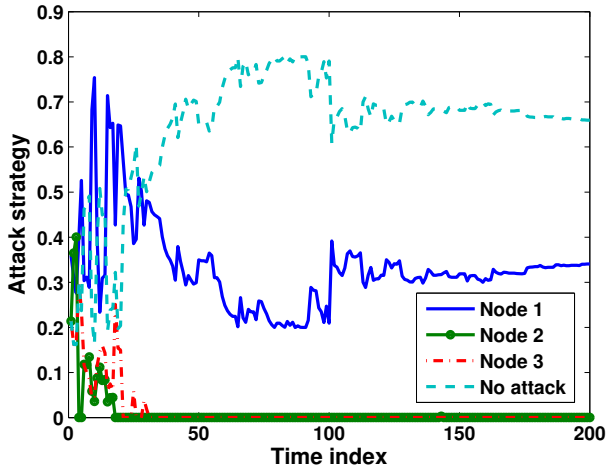

Figure 3.21: The attack strategy at a sample state for a 3-node chain network converges to a mixed NE strategy.

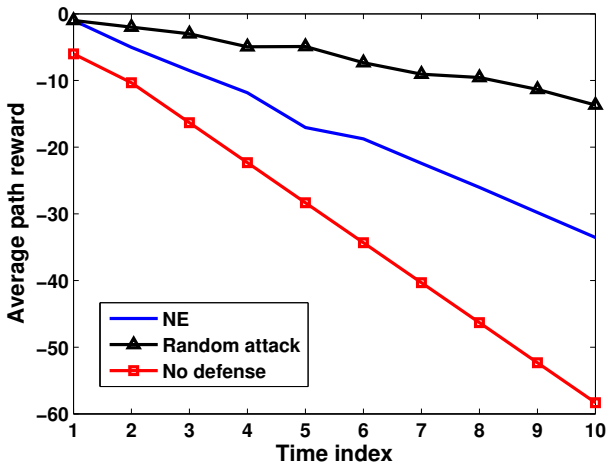

Figure 3.22: The defender's NE average path reward for a 3-node chain network against an attacker that plays best response and one that uses a random attack. Comparison to no-defense is also shown.

3.7.2 Testing step

After obtaining the stationary defense and attack policies for both players at NE for the Markov game using the Q-minimax algorithm, we evaluate the performance of our proposed defense approach against jamming attacks. In Fig. 3.22, we plot the objective function over 10 time steps. For each step we obtain the average (defender) reward of 1000 runs. Clearly, the NE policy yields substantial performance gains for the defender versus not using any defense mechanism. Furthermore, if the attacker deviates from the NE by launching a random attack (where attacked nodes are chosen uniformly at random), the defender will be able to secure higher reward.

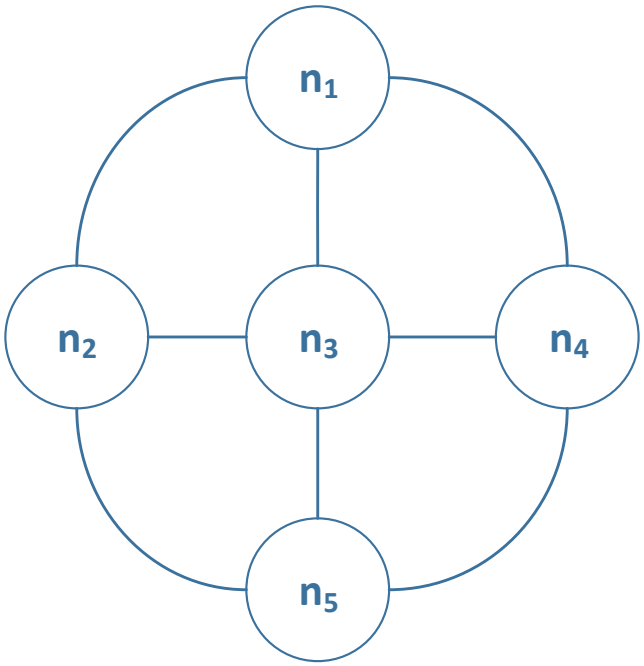

Figure 3.23: A 5-node interconnected network topology.

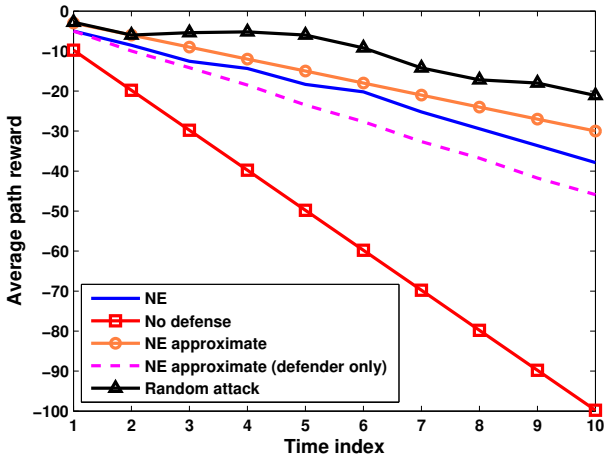

Figure 3.24: The defender's average path reward for a 5-node interconnected network compared with other attack and defense policies.

We also tested our defense approach on the 5-node interconnected topology of Fig. 3.23. In Fig. 3.24 we compare the defender's reward for equilibrium strategies obtained by both the exact Q-minimax algorithm and the approximate approach described in Section 3.6. The scalable approximate policy is shown to yield a comparable reward to that of the exact NE. Note that the approximate NE curve (shown with the circular markers) is obtained when both players use an approximate strategy, i.e, when both players deviate from the exact NE. Thus, the fact that the defender achieves a higher reward using the approximate policy than with the exact NE policy does not violate the definition of the NE which is only optimal in the sense of unilateral deviation by one of the players. We also plot the average path reward when only the defender uses the approximate policy while the attacker is assumed to have superior computational resources to compute his exact NE policy (dashed line). As shown, the reward achieved is only slightly smaller than that of the NE. This demonstrates the effectiveness of the approximate policy since it only costs the defender little loss when deviating from his exact NE.

CHAPTER 4: VIRTUAL MACHINES MIGRATION TIMING PROBLEM

4.1 Introduction

One of the main characteristics of the cloud that allows scalable and cost-effective operation is multi-tenancy. Multi-tenancy is achieved through virtualization to enable cloud providers to host multiple virtual machines (VMs) on the same physical machine while providing isolation between them. Recent attacks, however, have been shown to bypass such isolation [88]. A malicious VM collocating on the same physical machine with a victim VM can seek unauthorized access to sensitive and private data and/or intellectual property, or can render some of its computational functionality unusable.

This has prompted cloud providers to develop various strategies for VM placement, migration and reconfiguration to mitigate some of these attacks. Moving target defense (MTD) strategies aim to dynamically shift the attack surface, making it more difficult for attackers to launch effective attacks [89]. When developing an MTD strategy, two main questions generally arise: *which* targets should be moved? and *when* should they be moved? The answers to these questions largely depend on the context of the problem and the nature of the attack. For example, if an attacker contemplates to infer the underlying topology of the cloud, then the target is the machine connectivity that should then be adapted over time. However, if the attacker seeks to crack system credentials that protect the users' databases, then the target are the keys that should be constantly reconfigured (i.e., moved). In the proposed model, we consider collocation attacks whereby an attacker can access sensitive data from a targeted victim by running a VM on the same physical node (e.g., through launching a side-channel attack). Thus, for securing such systems, VMs should be periodically migrated, i.e., moved to different physical machines. While much work focused on the scheduling and placement aspect of VM migration, the timing problem is largely understudied. This motivates

the work in this chapter, which is primarily focused on the second question, that is, *when* to move the identified targets.

In the MTD literature, this question is usually referred to as the timing problem of the MTD strategy. In this chapter, we study this question in a game-theoretic framework seeking an understanding of the interplay of the strategies of the cloud provider (i.e., the defender) and the adversary. In our formulation, the adversary seeks to prolong the collocation time with the victim VMs to maximize the amount of information she can access. Since the adversary has no guarantees for being successfully collocated on the same node with the victim (different cloud providers implement different placement algorithms according to different criteria that the attacker has no control over), her best-effort is to increase the number of VMs to launch (which is a cost metric we capture). After the adversary is placed on given physical machines, she can check whether she had a successful collocation or not [90]. The cloud provider, on the other hand, migrates VMs between physical machines to minimize the collocation times between VMs. VM live migration, while efficient at not significantly disrupting the tasks running on a VM in the event of migration, is not free [91]. In practice, the number of cache pages read by an adversary from shared memory pages is proportional to the duration of a side-channel attack. It also depends on the technique used to access the last level cache (LLC) (e.g, PRIME+PROBE and FLUSH+RELOAD attacks) as shown in [92]. To read the cache, an attacker would need to adjust the time of the PROBE phase, which in turn affects the error rate of the attack covert channel. Thus, the question as to when to migrate is crucial. At the same time, the defender controls the migration time in order to mitigate the collocation attack threats while not burdening the system with significant overhead, e.g., due to VM downtime and undue memory usage.

Contributions: While VM migration strategies have been proposed as defense mechanisms against collocation attacks in various studies, such work focused on the VM assignment problem (mapping VMs to physical nodes) as a single player scheduling problem. In proposed model, however, we

consider the *timing* problem of the MTD as a game between the attacker and the cloud provider. Our work contributes to the theory of timing games [93, 94], which is largely unexplored in cloud computing settings. We leverage the results of the leakage model in the FlipIt game considered previously in [95, 96, 97, 98, 99, 100] to develop a novel formulation to study the VM collocation problem in an extended FlipIt game-theoretic framework. To the best of our knowledge, this is the first work to investigate the following aspects of timing games.

4.2 Related work

This work is at the intersection of two areas focused on securing cloud computing: Cross-VM side-channel attacks and mitigation, and game-theoretic modeling and techniques in cloud security. In this section, we put our work in context within these two areas.

4.2.1 Cross-VM side channel attacks and mitigation strategies

Cloud security has received considerable attention recently [88, 101]. Various studies have investigated the impact of cross-VM side-channel attacks [102, 103, 104, 90, 105, 92, 106]. Users' cryptographic keys have been shown to be vulnerable to exfiltration attacks when adversaries perform Prime+Probe attacks on the square-and-multiply implementation of GnuPG [105]. The authors in [90, 106, 92] have shown that some side-channel attacks can extract cryptographic keys by exploiting the last-level shared caches of the memory. Other attacks have identified pages that a VM shares with its collocated neighboring VMs revealing information about the victim's applications [103] and OS [104].

To combat cross-VM side-channel attacks, various approaches have been proposed at the hypervisor [107, 108, 105, 109, 110]), the guest OS [111], the hardware level [112, 113], and the ap-

plication layer [114]. These techniques, however, suffer from two fundamental limitations. First, they cannot be generalized to different types of side-channel attacks [115]. Second, they require major changes to the hypervisor, OS, hardware, and applications [116]. VM live migration, on the other hand, has been proposed as an effective mechanism to combat side-channel attacks [91, 117]. The authors in [118] provided a detection mechanism known as *CloudRadar* that works as a real-time side-channel attack detector based on monitoring hardware performance counters. The authors in [119] proposed another detection system that can differentiate between friendly and other malicious activities of neighboring tenants. The authors in [120] showed that by controlling the placement process, a defense mechanism can mitigate the effect of cross-VM attacks through reducing the co-run probability between users. The approach, however, is only effective in the case of time-sensitive attacks and when the number of assigned virtual CPUs is large. Motivated by the MTD concept, the authors in [121] presented a migration engine in which VMs are migrated to balance the load between different nodes in the cloud. Although MTD is a well-known defense methodology, the authors in [122] demonstrated that in certain scenarios the migrated VMs can be tracked by adversaries. Hence, they proposed a stealthy approach to migrate VMs that can hide them on the network. In [123], the authors study an MTD migration strategy against an attacker that seeks to collocate with VMs of high rewards by solving a multi-armed bandit problem.

4.2.2 Cloud security using game-theoretic techniques

The use of game theory has largely focused on the VM allocation problem in the presence of adversaries [124, 125, 126, 127, 128]. A common assumption in such formulations is that the adversary is known, which may not hold in practice. Additionally, existing formulations do not consider the timing question for the VM migration problem, which is a critical one for the cloud provider wishing to migrate VMs for security. A more practical leakage model was considered in [129, 130], based on the FlipIt game model. FlipIt is a two-player game in which a defender and

an attacker compete over the control of a given resource, which can only be held by one player at a time. A flip is an action performed by a player to gain control of the resource. The goal is to hold the resource for the longest duration possible with the least number of flips (i.e., flips are costly). Over time, the resource generates rewards for the player holding the resource. The state of the resource is obscured from each player until they "flip". Several variants of the FlipIt game model were considered to study different security situations [95, 96, 131, 132, 97, 98, 99, 100]. In [96], the authors studied different strategies for each player and calculated dominant strategies and Nash equilibria. In [131], the game model was studied under the assumption that the players know the state of the resource before taking actions. In [132, 97] the game was extended to the case of a system where insiders can work in favor of external adversaries. The authors in [98] considered the game with both players having limited budgets. Pawlick *et al.* investigated the game model with characteristics of signaling games [99]. In [100], Farhang *et al.* studied a variant of the FlipIt game with an associated data leakage model in which the defender can partially eliminate the foothold of the attacker. The attacker exploits the system vulnerabilities that appear based on a periodic process. The authors assume that the attacker's strategy is fixed since she always starts to attack right after the defender takes his action. This, however, requires the attacker to fully observe the defender's strategy which we do not assume here.

In this work, we consider a significantly different and a realistic threat model that captures data leakage due to cross-VM side-channel attacks and develop defense strategies for identifying the best time(s) to migrate VMs. We do this through a game-theoretic framework in which the attacker only controls the attack rate and does not fully observe the defender's strategy. In addition, we assume that the attacker controls the probability of a successful attack by choosing the attack rate as opposed to the time to launch the attack.

4.3 System Model

4.3.1 The cloud

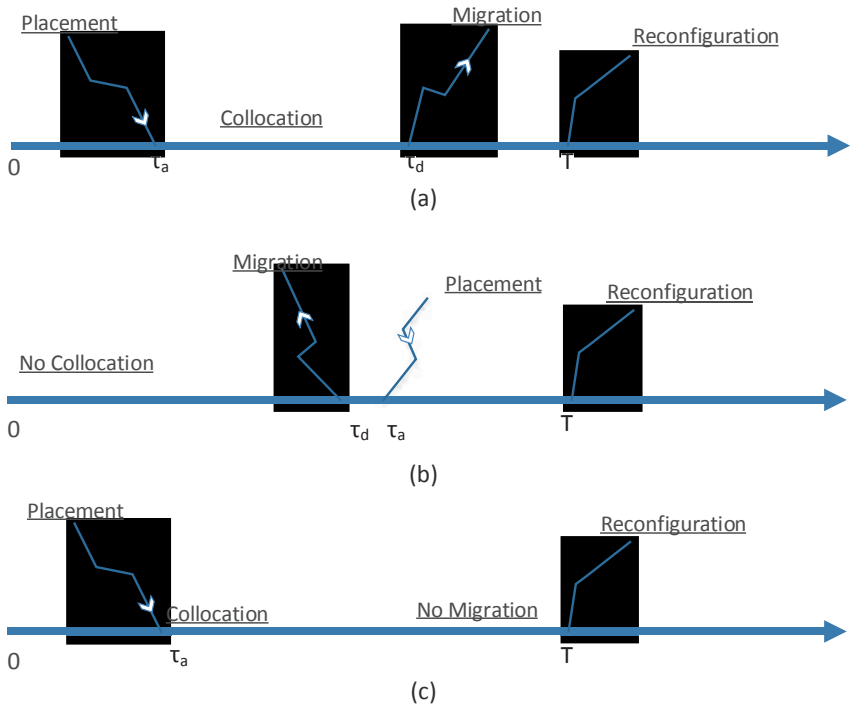

Figure 4.1: System model illustration for different placement events.

We model the cloud as a set of physical machines and each machine can host a number of VMs from different users. The cloud provider uses a placement strategy to initially assign VMs to physical machines. The details of the placement strategy do not affect our analysis and we assume that the adversary (or any user) has no control over it. We assume the adversary is interested in targeting a set of victim VMs by collocating with them on the same physical machines. We study the interaction between the cloud provider (defender) and the adversary through a game-theoretic framework in which the rewards are time-dependent. In particular, the defender's strategy is to

choose the time to re-assign VMs to different machines to defend against collocation attacks. The adversary, on the other hand, chooses an attack rate to launch more VMs to increase her chances for prolonged collocation with her victims. Fig. 4.1 illustrates three possible placement scenarios for the game. In plot (a), the attacker's VM is successfully collocated at time τ_a with her target VM on the same hypervisor before the target VM is migrated to another node at time τ_d. This scenario represents a successful collocation event, which results in information leakage. In plot (b), the target VM is migrated before the malicious VM is placed on the hypervisor, hence the collocation event does not occur. Finally, the plot in (c) illustrates a no-migration policy, where the collocation duration is maximized. We define the game next.

4.3.2 The game

A game is defined as a tuple $\Gamma(\mathcal{K}, \mathcal{A}, \mathcal{U})$, where

- $\mathcal{K}$ is the set of players. Here, $\mathcal{K} = \{1, 2\}$, denoting the defender (player 1) and the adversary (player 2).
- $\mathcal{A} = \mathcal{A}_d \times \mathcal{A}_a$ is the action space for the defender and adversary.
- $\mathcal{U} = \{u_d, u_a\}$ is the utility function, $\mathcal{U} : \mathcal{A} \rightarrow \mathbb{R}^2$. In this chapter we use $\mathcal{U}$ to denote a utility function instead of a reward function since the defender reward is always negative for the considered system model as explained next.

4.3.2.1 Defender's action space

Since we are investigating the timing factor, the cloud provider (referred to as the system defender) is assumed to control the re-allocation period. Let $\tau_d \in \mathcal{A}_d$ denote the time instant at which the

defender migrates a running VM to a new physical node, such that $\mathcal{A}_d = [\tau_{\min}, T]$, where T is a system parameter at which the credentials are reset and $\tau_{\min}$ is the smallest reconfiguration time. Since we assume a leakage model, at time T when the system credentials are reset, the attacker can no longer benefit from the side-channel attack. Therefore, the whole game will be reset every T. The defender seeks to optimize the value of τ_d to minimize chances for information leakage and avoid loading the system with unnecessary migrations. Thus, the defender's goal is to optimize the tradeoff between security and stability. In particular, a smaller τ_d ensures the system is more secure since the co-residency times between any two VMs will be small. However, the system's overhead increases due to frequent migration of the VMs between the physical nodes. The overhead of VM live migration has been investigated in [91, 133], and in general depends on the VM workload. The work in [133] has shown that the main factors affecting the VM migration overhead are the VM memory size and the network speed. On the other hand, a larger τ_d leads to a more stable system. However, the co-residency times between VMs on the same node will be large making the system more susceptible to a data breach through collocation attacks.

4.3.2.2 Attacker's action space

Here, we assume that the attacker does not know the system placement algorithms, hence only tries to increase her co-residency chances via increasing the number of requests submitted to the cloud provider. Let $\lambda_a \in \mathcal{A}_a$ denote the rate of requests (rate of attack) submitted to the cloud, where $\mathcal{A}_a = [\lambda_{\min}, \lambda_{\max}]$ is an interval of non-negative attack rates. The game is assumed to start at time $t = 0$, and let τ_a denote the actual time at which the attacker successfully collocates with her targeted victim. Hence, $\tau_a > 0$ is a non-negative random variable with a probability density function (pdf) $f_a(.; \lambda_a)$ parametrized by λ_a. Since the attacker pays a cost for each submitted job, she needs to optimize over the attack rate λ_a. Hence, the attacker's tradeoff can be summarized as follows. When λ_a is very small, it is less probable for the attacker to successfully co-reside

with her victim and in turn steal any information before VMs are migrated. When λ_a is very large, the attacker increases her chances of successful collocation at the expense of a higher attack cost. Therefore, the pdf f_a should be such that $f_a(\tau_a; \lambda_{a_1})$ yields a higher probability of early collocation than $f_a(\tau_a; \lambda_{a_2})$, when $\lambda_{a_1} > \lambda_{a_2}$. Mathematically, this requirement is expressed through the following assumption.

Assumption 1. $F_a(t; \lambda_{a_1}) \geq F_a(t; \lambda_{a_2})$ *for* $\lambda_{a_1} \geq \lambda_{a_2}$*, where* $F_a(t; \lambda_a) := \Pr(\tau_a \leq t)$ *denotes the cumulative distribution function (CDF) of the collocation time.*

If $\lambda_{\min} = 0$, then the attacker can choose to back off (i.e., not attack). In such case, $f_a(\tau_a; 0)$ is a degenerate deterministic distribution such that $F_a(T; 0) = 0$ since the probability of collocation is 0.

We focus only on the timing factor of the problem, and the mapping of VMs to physical nodes is carried out through the placement engine. The separation of the placement and timing strategies allows for layered functionality highly desirable in practice. In particular, the developed timing policies can be implemented on any existing platform without modifying the existing placement engine. This is especially true since allocation decisions are typically developed around widely differing load balancing and power reduction objectives, and other operational constraints. Next, we define the players' utility functions in a nonzero-sum two-person game.

4.3.2.3 Attacker's utility

Once the attacker's VM is successfully placed on the same node where the victim VM resides, she immediately starts accumulating rewards by reading out data from the target VM. The amount of information leakage depends in practice on the duration of collocation as shown in [92]. Let $G(\tau_d, \tau_a)$ denote the reward accumulated by the attacker capturing the relation between the collo-

cation duration and the amount of the data leaked.

Assumption 2. *$G(\tau_d, \tau_a)$ is a stationary function and monotonically non-decreasing in the collocation duration $t = \tau_d - \tau_a$. Therefore, $G(\tau_d, \tau_a) = G(\tau_d - \tau_a, 0) = G(t)$, where $G(t)$ is an abbreviated notation indexed by one variable.*

Stationarity signifies that the attacker's accumulated reward depends on the collocation and migration times only through their difference, i.e., the duration of collocation. The accumulated reward is assumed to be zero if $\tau_a \geq \tau_d$. The attacker incurs a cost C_a for launching an attack. Hence, the total cost is scaled by the rate of attack λ_a. Therefore, the attacker's expected utility is given by

$$u_a(\tau_d, \lambda_a) = \int_0^{\tau_d} G(\tau_d, \tau_a) f_a(\tau_a, \lambda_a)\, d\tau_a - C_a \lambda_a. \tag{4.1}$$

4.3.2.4 Defender's utility

The defender, on the other hand, incurs a loss due to the collocation of a victim VM with the attacker equal in magnitude to the gain of the attacker. In addition, the defender pays a cost per migration denoted by C_d. This cost captures the migration overhead, which stems from the VM downtime, performance degradation of the running applications (e.g., due to successive iterations of memory pre-copying [134]), and the amount of memory and cache usage. Accordingly, the defender's expected utility can be written as

$$u_d(\tau_d, \lambda_a) = -\int_0^{\tau_d} G(\tau_d, \tau_a) f_a(\tau_a, \lambda_a)\, d\tau_a - \frac{C_d}{\tau_d}\,. \tag{4.2}$$

4.4 Theoretical Analysis

A NE characterizes a solution for non-cooperative games in which no player can gain by deviating from his own equilibrium strategy while the other players' strategies are fixed [135, 7]. In this section, we establish sufficient conditions for the existence of a NE for the formulated game model in Theorems 8 and 13. We characterize the players' best responses (c.f. Definition 1) in Theorem 10 and Lemma 14. By definition, a NE secures a minimum utility for the cloud admin since the defender's utility cannot decrease if only the adversary deviates from her best response. Since the utilities for both players depend on the cost parameters C_a and C_d (c.f. Section 2.3), the role of these parameters is also analyzed in Lemmas 11, 12 and Theorems 15 and 16.

Existence of NE depends on the properties of the payoff functions. First, we derive existence conditions for a general accumulated reward function $G(\tau_d, \tau_a)$ and pdf $f_a(\tau_a; \lambda_a)$ of collocation time, then we provide analysis for a special instantiation of the payoff functions. We also characterize the best response curves for both players and derive existence conditions for the corresponding NE strategies. First, we restate a general theorem from [7] that provides sufficient conditions for N-person nonzero-sum games to admit a pure strategy NE.

Theorem 6. *[6, 7] For each player i in the set $\mathcal{N}$ of N players, let the action space $\mathcal{A}_i$ of player i be a closed, bounded and convex subset of a finite-dimensional Euclidean space, and the cost functional $J_i : \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \mathbb{R}$ be jointly continuous in all its arguments and strictly convex in $a_i \in \mathcal{A}_i$, for every $a_j \in \mathcal{A}_j, j \in \mathcal{N}, j \neq i$. Then, the associated N-person nonzero-sum game admits a Nash equilibrium in pure strategy.*

4.4.1 General reward functions

For the general payoff formulation described in equations (4.1) and (4.2), the following lemma establishes sufficient conditions for the concavity of the payoff functions.

Lemma 7. *For the 2-person nonzero-sum game defined in Section 4.3.2 with payoff functions defined in equations (4.1) and (4.2) under Assumptions 1 and 2, if* $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ *is strictly concave in* $\lambda_a \in \mathcal{A}_a$ *for any* τ_d*, then* $u_a(\tau_d, \lambda_a)$ *is strictly concave in* λ_a *for any* $\tau_d \in \mathcal{A}_d$*, and if* $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ *is convex in* $\tau_d \in \mathcal{A}_d$*, then* $u_d(\tau_d, \lambda_a)$ *is strictly concave in* τ_d *for any* $\lambda_a \in \mathcal{A}_a$*, where* $\mathbf{1}_{\{.\}}$ *is an indicator function.*

Proof. If $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ is strictly concave in λ_a, then $\int_0^{\tau_d} G(\tau_d, \tau_a) f_a(\tau_a; \lambda_a)\, d\tau_a$ is strictly concave in λ_a. Since the expected payoff u_a in (4.1) is a linear combination of a strictly concave function and a linear function $\lambda_a C_a$, it follows that u_a is strictly concave. Similarly, if $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ is convex in τ_d, then $\int_0^{\tau_d} G(\tau_d, \tau_a) f_a(\tau_a; \lambda_a)\, d\tau_a$ is strictly convex. From (4.2), it follows that u_d is strictly concave in τ_d since $\frac{C_d}{\tau_d}$ is strictly convex in τ_d. □

Therefore, we can readily state sufficient conditions for our game to admit a pure strategy NE.

Theorem 8. *The 2-person nonzero-sum game defined in Section 4.3.2 under Assumptions 1 and 2 with the payoff functions in (4.1) and (4.2) admits a NE in pure strategy if* $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ *is continuous and strictly concave in* $\lambda_a \in \mathcal{A}_a$*, and* $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ *is convex and* G *is continuous in* $\tau_d \in \mathcal{A}_d$*.*

The proof of Theorem 8 follows directly from Lemma 7, which establishes strict concavity of the payoff functions under the conditions in the statement of the theorem, and Theorem 6 from [7].

Proposition 9. *For the game defined in Section 4.3.2 with* $\lambda_{\min} = 0$*, there exists an equilibrium in which the attacker backs off (i.e., does not attack) and the defender does not migrate if the reward*

function $G(t)$ satisfies

$$\mathbb{E}_{\lambda_a}\left[G(T-\tau_a)\right] \leq \lambda_a C_a, \tag{4.3}$$

for every $\lambda_a \in \mathcal{A}_a$, where $\mathbb{E}_{\lambda_a}[.]$ denotes the expectation w.r.t. the measure induced by $f_a(.;\lambda_a)$.

Proof. If the attacker backs off, i.e., chooses $\lambda_a = \lambda_{\min} = 0$, then the defender's payoff in (4.2) becomes

$$u_d(\tau_d, 0) = \frac{-C_d}{\tau_d},$$

which attains its maximum at $\tau_d = T$ for any $C_d > 0$. Hence, the defender's best response is to not migrate over the game interval. Also, if condition (4.3) in the statement of Proposition 9 is satisfied, then the attacker's best response to the defender's action $\tau_d = T$ is $\lambda_a = 0$. To see that note that if

$$\mathbb{E}_{\lambda_a}\left[G(T,\tau_a)\right] = \int_0^\infty G(T,\tau_a) f_a(\tau_a;\lambda_a) d\tau_a \leq \lambda_a C_a \,,$$

then

$$\int_0^{\tau_d} G(\tau_d,\tau_a) f_a(\tau_a;\lambda_a) d\tau_a \leq \lambda_a C_a$$

since $G(t)$ is monotonically non-decreasing in t per Assumption 2. Recalling the attacker's payoff function in (4.1), the attacker's decision to back off is at least as good as launching an attack at an alternative non-vanishing rate since the cost of the attack upper bounds the leakage reward for any $\lambda_a \neq 0$.

□

In the following theorem, we characterize the best response for both players.

Theorem 10. *For the 2-person nonzero-sum game defined in Section 4.3.2, if the attacker's payoff function in (4.1) is strictly concave in λ_a, then the attacker's best response λ_a^* to any defense strategy can be described as*

- $\lambda_a^* = \lambda_{\max}$, *if* $\frac{\partial u_a}{\partial \lambda_a} > 0,\ \forall\ \lambda_a \in \mathcal{A}_a$
- $\lambda_a^* = \lambda_{\min}$, *if* $\frac{\partial u_a}{\partial \lambda_a} < 0,\ \forall\ \lambda_a \in \mathcal{A}_a$
- $\lambda_a^* \in \left\{ \lambda_a \mid \frac{\partial}{\partial \lambda_a} \mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a < \tau_d\}}] = C_a \right\}$, *if* $\frac{\partial u_a}{\partial \lambda_a} = 0$, *for any* $\lambda_a \in \mathcal{A}_a$.

Also, if the defender's payoff function in (4.2) is strictly concave in τ_d*, then the best response* τ_d^* *can be described as*

- $\tau_d^* = T$, *if* $\frac{\partial u_d}{\partial \tau_d} > 0,\ \forall\ \tau_d \in \mathcal{A}_d$
- $\tau_d^* = \tau_{\min}$, *if* $\frac{\partial u_d}{\partial \tau_d} < 0,\ \forall\ \tau_d \in \mathcal{A}_d$
- $\tau_d^* \in \left\{ \tau_d \mid \tau_d^2 \frac{\partial}{\partial \tau_d} \mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a < \tau_d\}}] = C_d \right\}$, *if* $\frac{\partial u_d}{\partial \tau_d} = 0$, *for any* $\tau_d \in \mathcal{A}_d$.

Proof. Given the concavity of the payoff function u_a in $\lambda_a \in \mathcal{A}_a$, the derivative $\frac{\partial u_a}{\partial \lambda_a}$ is monotone. Hence, there exist three possibilities for the behavior of u_a. If $\frac{\partial u_a}{\partial \lambda_a} > 0$, then u_a is strictly increasing in λ_a for all $\lambda_a \in \mathcal{A}_a$, thus the payoff is maximized by $\lambda_a^* = \lambda_{\max}$. If $\frac{\partial u_a}{\partial \lambda_a} < 0,\ \forall \lambda_a\ \in \mathcal{A}_a$, then u_a is strictly decreasing in λ_a for all $\lambda_a \in \mathcal{A}_a$, thus the payoff is maximum at $\lambda_a^* = \lambda_{\min}$. Otherwise, u_a attains its maximum when $\frac{\partial u_a}{\partial \lambda_a} = 0$, hence the best response λ_a^* belongs to the set $\Lambda_a = \left\{ \lambda_a \mid \int_0^{\tau_d} \frac{\partial f_a}{\partial \lambda_a} G(\tau_d, \tau_a) d\tau_a = C_a \right\}$ at which $\frac{\partial u_a}{\partial \lambda_a} = 0$. The second part of Theorem 10 which characterizes the defender's best response can be proven similarly. □

Next, we study the effect of the attack cost C_a and the moving cost C_d and state bounds on the costs beyond which no player is interested in the game. When the cost C_a exceeds a certain threshold, the cost of the attack dominates the attacker's tradeoff, i.e., the attacker is better off backing off over attempting to access the victim's information. Similarly, if C_d is too high, the defender incurs a cost for migration that exceeds any benefit he would get at any migration rate.

In the following lemma, we derive a lower bound on the attack cost C_a beyond which the attacker is always better off attacking with the minimum rate $\lambda_{\min}$. If $\lambda_{\min} = 0$, then the attacker will back off.

Lemma 11. *For the two person nonzero-sum game Γ defined in Section 4.3.2, if $\mathbb{E}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ is strictly concave in $\lambda_a \in \mathcal{A}_a$, and $C_a > \frac{\partial}{\partial\lambda_a}\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}] \mid_{\lambda_a=\lambda_{\min}}$, then the attacker's best response to any defense strategy τ_d is to attack at the minimum permissible rate $\lambda_{\min}$.*

Proof. We argue that under the condition stated in the lemma, the attacker's payoff is monotonically decreasing in λ_a. Hence, $\lambda_a^* = \lambda_{\min}$ is the attacker's best response to any τ_d. To show that $\lambda_{\min}$ is the unique best response, assume for contradiction there exists $\lambda^* = \lambda_1 \neq \lambda_{\min}$. If $C_a > \int_0^{\tau_d} G f_a'(\lambda_a)\, d\tau_a \mid_{\lambda_a=\lambda_{\min}}$, where $f_a'(\lambda_a) = \frac{\partial f_a}{\partial\lambda_a}$, then u_a is monotonically decreasing, therefore $u_a(\lambda_{\min}) > u_a(\lambda_1)$ since $\lambda_1 > \lambda_{\min}$. Hence, λ_1 is not in the best response set. □

Similarly, the following lemma establishes a lower bound on the migration cost C_d of the defender, beyond which it is more advantageous not to migrate before the system reconfiguration cycle T.

Lemma 12. *For the two person nonzero-sum game Γ defined in Section 4.3.2, if $\mathbb{E}_{\lambda_a}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right]$ is strictly convex in λ_a and G is continuous in $\tau_d \in \mathcal{A}_d$, and $C_d > T^2\frac{d}{d\tau_d}\mathbb{E}_{\lambda_a}\left[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}\right] \mid_{\tau_d=T}$, then the action of not migrating any VM before T is the defender's unique best response regardless of the attacker's strategy λ_a, where $\mathbb{E}_{\lambda_a}[.]$ is the expectation with respect to $f_a(\tau_a; \lambda_a)$.*

Proof. By an argument similar to the proof of Lemma 11, under the condition in the statement of the lemma, the defender's payoff is monotonically increasing in τ_d. Hence, $T \in R_1(\lambda_a)$ for any λ_a. Establishing the uniqueness of T as a best response action follows the same argument used in the proof of Lemma 11. □

4.4.2 Special instantiation analysis

In Section 4.4.1, we provided conditions for the existence of an equilibrium for generic reward functions. The conditions imposed were the strict concavity of f_a in addition to the non-negativity, monotonicity and stationarity of G (stationarity in that the accumulated reward depends on the collocation and migration times only through their difference, i.e., the duration of collocation). In this section, we study existence conditions for equilibrium and characterize the best response sets of both players for specific choices of the reward function G and the collocation pdf $f_a(\tau_a; \lambda_a)$. Since the amount of information leakage depends in practice on the duration of collocation, here we provide an analysis for the case where $G(t)$ increases linearly in the collocation duration t. Hence, we analyze the formulated timing game for the following choice of G,

$$G(\tau_d, \tau_a) = \begin{cases} \alpha\left(\tau_d - \tau_a\right), & \tau_a \leq \tau_d \leq T \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

In Section 4.6.4, we provide numerical results on the best response for other (non-linear) functions, including when G scales sublinearly and quadratically in t. Without loss of generality, we always consider $\alpha = 1$. The case $\alpha \neq 1$ corresponds to the case $\alpha = 1$ with the migration cost C_d replaced by $\frac{C_d}{\alpha}$.

In our numerical evaluation we consider an exponential pdf f_a for the collocation time, i.e.,

$$f_a(\tau_a; \lambda_a) = \lambda_a e^{-\lambda_a \tau_a}, \quad \tau_a \geq 0. \tag{4.5}$$

This choice of $f_a(.; \lambda_a)$ is motivated by the interpretation of λ_a as the rate of attacks launched by the adversary. In other words, the attacker controls the rate of the submitted requests to the cloud

server. Her requests are served within the queue of the placement engine and hence assigned to different physical machines according to a Poisson arrival process justifying the exponential arrival time.

Next, we derive sufficient conditions for the existence of a NE for the choice of functions in (4.4) and (4.5).

Theorem 13. *The 2-person nonzero-sum game defined in Section 4.3.2 with $G(t)$ and $f_a(\tau_a; \lambda_a)$ defined in (4.4) and (4.5) admits a pure strategy NE.*

The proof of Theorem 13 rests upon proving the strict concavity of u_a and u_d, which translates into existence of a NE in pure strategy from [7, Theorem 1].

Proof. From Theorem 6, it suffices to show that for every pair of $(\tau_d, \lambda_a) \in \mathcal{A}_d \times \mathcal{A}_d$, u_a is strictly concave in λ_a for every $\tau_d \in \mathcal{A}_d$ and that u_d is strictly concave in τ_d for every $\lambda_a \in \mathcal{A}_a$. Per Lemma 2, it suffices to show that $\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}]$ is strictly concave in $\lambda_a \in \mathcal{A}_a$ and strictly convex in $\tau_d \in \mathcal{A}_d$. For the functions G and f_a defined in (4.4) and (4.5),

$$\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}] = \frac{\lambda_a\tau_d + e^{-\lambda_a\tau_d} - 1}{\lambda_a}. \tag{4.6}$$

Taking the derivative with respect to λ_a twice,

$$\frac{\partial^2}{\partial\lambda_a^2}\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}] = \frac{(\lambda_a^2\tau_d^2 + 2\lambda_a\tau_d + 2)e^{-\lambda_a\tau_d} - 2}{\lambda_a^3}. \tag{4.7}$$

By Taylor's expansion of e^x for $x > 0$, $e^x > \frac{x^2}{2} + x + 1$. Hence, $(\lambda_a^2\tau_d^2 + 2\lambda_a\tau_d + 2)e^{-\lambda_a\tau_d} < 2$. Therefore, $\frac{\partial^2}{\partial\lambda_a^2}\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}] < 0, \forall\lambda_a \in \mathcal{A}_a$. Thus, $\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}]$ is strictly concave in λ_a for every τ_d.

Next, we prove that $\mathbb{E}_{\lambda_a}[G(\tau_d, \tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}]$ is convex in τ_d. To this end, we obtain its second

derivative with respect to τ_d,

$$\frac{\partial^2}{\partial \tau_d^2}\mathbb{E}_{\lambda_a}[G(\tau_d,\tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}] = \lambda_a^2 e^{-\lambda_a \tau_d}\,. \tag{4.8}$$

Since, $\frac{\partial^2}{\partial \tau_d^2}\mathbb{E}_{\lambda_a}[G(\tau_d,\tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}] > 0, \forall \tau_d \in \mathcal{A}_d$, then $\mathbb{E}_{\lambda_a}[G(\tau_d,\tau_a)\mathbf{1}_{\{\tau_a<\tau_d\}}]$ is convex in τ_d for every λ_a. Accordingly, existence of a NE follows from Theorem 8.

□

To characterize NE for both players, we start off by characterizing the best response set for each player in the following lemma whose proof follows the same argument used in the proof of Theorem 10.

Lemma 14. *For the 2-person game defined in Section 4.3.2 with the reward function $G(t)$ and the probability density function $f_a(\tau_a;\lambda_a)$ defined in (4.4) and (4.5), the attacker's best response pure strategy is characterized as*

- $\lambda_a^* = \lambda_{\max}$, *if* $1-\lambda_a\tau_d e^{-\lambda_a\tau_d} - e^{-\lambda_a\tau_d} - C_a\lambda_a^2 > 0$
- $\lambda_a^* = \lambda_{\min}$, *if* $1-\lambda_a\tau_d e^{-\lambda_a\tau_d} - e^{-\lambda_a\tau_d} - C_a\lambda_a^2 < 0$
- $\lambda_a^* = \left\{\lambda_a \mid 1-\lambda_a\tau_d e^{-\lambda_a\tau_d} - e^{-\lambda_a\tau_d} = C_a\lambda_a^2\right\}$, *otherwise,*

for any action τ_d by the defender.

The best response strategy for the defender is characterized as

- $\tau_d^* = T$, *if* $C_d - \tau_d^2\left(1-e^{-\lambda_a\tau_d}\right) > 0$
- $\tau_d^* = \tau_{\min}$, *if* $C_d - \tau_d^2\left(1-e^{-\lambda_a\tau_d}\right) < 0$

- $\tau_d^* = \left\{\tau_d \mid \tau_d^2\left(1 - e^{-\lambda_a \tau_d}\right) = C_d\right\}$, *otherwise,*

for any action λ_a by the attacker.

Remark 4. *The optimal NE strategies can be obtained analytically using the players' best response curves derived in Lemma 14 (c.f. Section 4.6). In situations where an analytical solution is intractable, they can be obtained numerically as the equilibrium solution of convex optimization problems associated with each player. Specifically, for $\lambda_a \in \mathcal{A}_a$, $\lambda_a^* = \arg\max u_a$ and for $\tau_d \in \mathcal{A}_d$, $\tau_d^* = \arg\max u_d$. Since both objective functions were shown to be convex and differentiable, each player is guaranteed to converge to his optimal action for every action by the opponent. The equilibrium (τ_d^*, λ_a^*) satisfies the Lagrangian equations corresponding to both problems [136] and can be obtained by solving both problems simultaneously using standard techniques such as Newton's method with convergence guarantees [80, 11].*

The following two theorems establish bounds on both the attack cost C_a and the migration cost C_d beyond which the players' best response strategies are on the boundaries of their action intervals.

Theorem 15. *For the two person nonzero-sum game defined in Section 4.3.2 with the reward function in (4.4) and the exponentially distributed collocation time τ_a in (4.5), if*

$$C_a > \frac{1 - (1 + \lambda_{\max}\tau_d)e^{-\lambda_{\max}\tau_d}}{\lambda_{\min}^2},$$

then the attacker's best response to the action τ_d of the defender is $\lambda_a^*(\tau_d) = \lambda_{\min}$.

Proof. Given the reward function $G(\tau_d, \tau_a)$ in (4.4) and the pdf of τ_a in (4.5), the attacker's expected payoff function can be expressed as in (4.11). If the lower bound on C_a in the statement of the theorem is satisfied, i.e.,

$$C_a > \frac{1 - (1 + \lambda_{\max}\tau_d)e^{-\lambda_{\max}\tau_d}}{\lambda_{\min}^2},$$

then,

$$C_a > \frac{1-(1+\lambda_a\tau_d)e^{-\lambda_a\tau_d}}{\lambda_a^2},\ \ \forall\lambda_a \in \mathcal{A}_a$$

since the function in the numerator of the RHS of the inequality is monotonically increasing in λ_a for $\lambda_a, \tau_d \geq 0$. Hence,

$$1 - C_a\lambda_a^2 - (\lambda_a\tau_d + 1)e^{-\lambda_a\tau_d} < 0\,.$$

Dividing both sides by λ_a^2,

$$\frac{1 - C_a\lambda_a^2 - (\lambda_a\tau_d + 1)e^{-\lambda_a\tau_d}}{\lambda_a^2} < 0.$$

The left hand side of the above inequality is $\frac{\partial u_a}{\partial \lambda_a}$. Thus, u_a is monotonically decreasing in λ_a, therefore, $\lambda_a^* = \lambda_{\min}$, which completes the proof. □

Theorem 16. *For the two person nonzero-sum game defined in Section 4.3.2 with the reward function in (4.4) and the exponentially distributed collocation time τ_a in (4.5), if*

$$C_d > T^2(1 - e^{-\lambda_a T})\,,$$

then the defender's best response to the action λ_a of the attacker is to stop migrations, i.e, $\tau_d^*(\lambda_a) = T$.

Proof. Similar to the argument used in the proof of Theorem 15 above, if $C_d > T^2(1 - e^{-\lambda_a T})$, then

$$C_d > \tau_d^2(1 - e^{-\lambda_a\tau_d}),\ \forall\tau_d < T$$

since $\tau_d^2(1 - e^{-\lambda_a\tau_d})$ is monotonically increasing in $\tau_d \geq 0$. Dividing by τ_d^2,

$$0 < \frac{C_d}{\tau_d^2} - (1 - e^{-\lambda_a\tau_d}) = \frac{\partial u_d}{\partial \tau_d}\,.$$

Hence, u_d is monotonically increasing in τ_d, therefore, the best response $\tau_d^*(\lambda_a) = T$. □

4.5 Generalization: Game Model with IDS

In the aforementioned model, the attacker's goal is to be collocated with her victim as soon as possible before the victim is migrated. Evidently, upon collocation with her victim, the attacker will choose to reside there until τ_d since no detection mechanism is in place to urge her to evade. In this section, we extend the existing system model and consider the case in which the cloud data center is equipped with an IDS. The IDS monitors suspicious activities and captures malicious behavior of any user after a sufficient period of time δ, which is a random variable with distribution $y(\delta)$, $\delta \in [0, T]$. For useful detection, $\delta < \tau_d$. Hence, the attacker may need to stop her collocation attacks before being detected. This introduces another control variable s to be optimized by the attacker, namely how long she should continue to carry on the attack after successful collocation. The distribution $y(\delta)$ accounts for the entire range of priors between the extreme of an uninformative prior (a uniform distribution) in which the players do not have useful information about the time-to-detection δ, and the extreme of a fully degenerate distribution in which the players know δ exactly. For the latter case, the attacker will surely choose to stop after a duration δ from the onset of successful collocation, i.e., right before $\tau_a + \delta$.

Next, we modify the attacker's payoff function u_a in order to account for the probability of detection. In the event of detection, the attacker incurs a cost D (since this user will be black-listed), but her gain is in the amount of data read out until detection. Therefore, we redefine the attacker's

expected utility by averaging over both the time-to-detection δ and the collocation time τ_a,

$$
\begin{aligned}
u_a(\tau_d, \lambda_a, s) = & \int_{\delta=s}^{T} G(s) y(\delta) d\delta \int_{\tau_a=0}^{\tau_d - s} f_a(\tau_a; \lambda_a) d\tau_a \\
& + \int_{\delta=0}^{s} (G(\delta) - D) \left(\int_{0}^{\tau_d - \delta} f_a(\tau_a; \lambda_a) d\tau_a \right) y(\delta) d\delta \\
& + \int_{\delta=0}^{s} \left(\int_{\tau_d - \delta}^{\tau_d} G(\tau_d - \tau_a) f_a(\tau_a; \lambda_a) d\tau_a \right) y(\delta) d\delta \\
& + \int_{\delta=s}^{T} \left(\int_{\tau_d - s}^{\tau_d} G(\tau_d - \tau_a) f_a(\tau_a; \lambda_a) d\tau_a \right) y(\delta) d\delta - C_a \lambda_a.
\end{aligned}
\tag{4.9}
$$

The first term in (4.9) accounts for the attacker's expected payoff in the event of no detection as the attacker stopped malicious activities before the IDS alarm, i.e., $s < \delta$, as illustrated in Fig. 4.2. The second term represents the event of detection, hence collocation ends at $\tau_a + \delta$, i.e., after a collocation duration δ as $\delta < s$, as shown in Fig. 4.3. Therefore, the attacker incurs a detection loss D. The third and fourth terms account for the event of no detection but due to the migration mechanism. In other words, the attacker is not identified because $\tau_d - \tau_a < \min(\delta, s)$. The last term accounts for the cost of launching the attack.

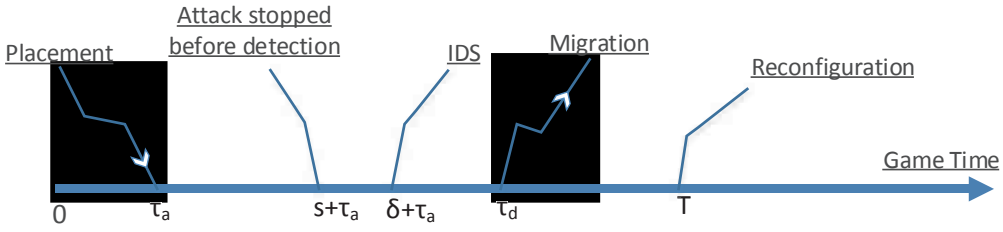

Figure 4.2: Attacker evades IDS by early stopping of malicious activity.

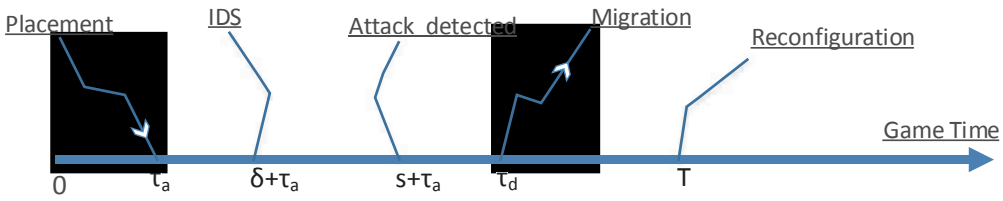

Figure 4.3: Attacker detected by the IDS.

Similarly, we redefine the defender's expected payoff function,

$$\begin{aligned} u_d(\tau_d, \lambda_a, s) = &- \int_{\delta=s}^{T} G(s) y(\delta) d\delta \int_{\tau_a=0}^{\tau_d - s} f_a(\tau_a; \lambda_a) d\tau_a \\ &- \int_{\delta=0}^{s} (G(\delta) - D) \left(\int_{0}^{\tau_d - \delta} f_a(\tau_a; \lambda_a) d\tau_a \right) y(\delta) d\delta \\ &- \int_{\delta=0}^{s} \left(\int_{\tau_d - \delta}^{\tau_d} G(\tau_d - \tau_a) f_a(\tau_a; \lambda_a) d\tau_a \right) y(\delta) d\delta \\ &- \int_{\delta=s}^{T} \left(\int_{\tau_d - s}^{\tau_d} G(\tau_d - \tau_a) f_a(\tau_a; \lambda_a) d\tau_a \right) y(\delta) d\delta - \frac{C_d}{\tau_d} . \end{aligned} \tag{4.10}$$

4.6 Numerical Analysis

In this section, we provide numerical analysis of the proposed game model. To characterize the payoff functions for both players, we need to specify $G(t)$ and $f_a(\tau_a; \lambda_a)$. For the linear reward function $G(t)$ and the exponential density function $f_a(\tau_a; \lambda_a)$ described in (4.4) and (4.5), the utility functions can be readily expressed as

$$u_a(\tau_d, \lambda_a) = \frac{\lambda_a \tau_d + e^{-\lambda_a \tau_d} - C_a \lambda_a^2 - 1}{\lambda_a}, \tag{4.11}$$

$$u_d(\tau_d, \lambda_a) = \frac{1 - \lambda_a \tau_d - e^{-\lambda_a \tau_d}}{\lambda_a} - \frac{C_d}{\tau_d}, \tag{4.12}$$

for $\tau_d \in \mathcal{A}_d, \lambda_a \in \mathcal{A}_a$. In the following analysis, we study the behavior of the payoff functions for both players. We illustrate the utility of the defender as a function of the migration time τ_d for a range of attack rates λ_a. For the attacker, we plot her utility as a function of λ_a for different τ_d. Afterwards, we investigate the effect of the migration cost C_d and the attack cost C_a on the utility functions and the players' best response curves. We also demonstrate existence of NE when the game satisfies the concavity conditions. Finally, we generalize our analysis to investigate different scaling regimes of the reward function, including sublinear and superlinear regimes.

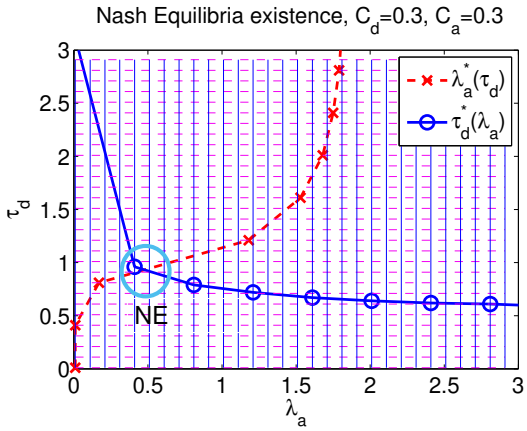

Figure 4.4: For the shown action space, $\mathcal{A} = \mathcal{A}_d \times \mathcal{A}_a$, the game admits a NE in pure strategies.

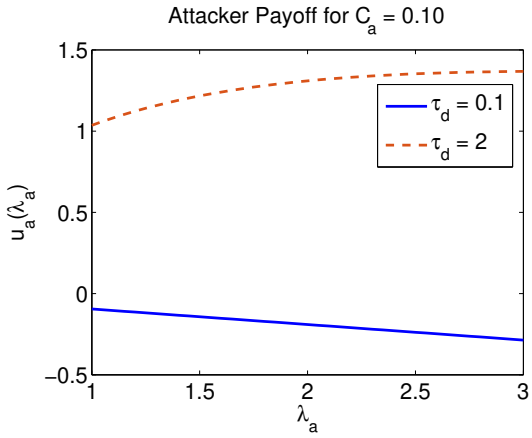

Figure 4.5: At $\tau_d = 0.1$, the attacker payoff is monotonically decreasing, but not for $\tau_d = 2$, in agreement with the bound on C_a in Theorem 15.

We start our numerical analysis by reflecting on the theoretical analysis in Section 4.4.2. In Fig. 4.4, we plot the NE existence region that satisfies strict concavity of both u_a and u_d. Per Theorem 13, for G and f_a as defined in (4.4) and (4.5), the game played over the illustrated action space admits a NE in pure strategies circled in Fig. 4.4. The figure illustrates the best response curves along with the game action space at $C_d = 0.3$ and $C_a = 0.3$. Fig. 4.4 verifies our analytical results of NE existence. An equilibrium point lies at the intersection of the two best response curves for both players. By definition, this point is a NE at which each player makes the best decision taking into account the opponent's best fixed strategy. In this setting, the NE is unique – the unique intersection point of the best response curves for both players at $\tau_d^* = 0.93$ and $\lambda_a^* = 0.48$. Theorems 15 and 16 established lower bounds on C_a and C_d beyond which u_a and u_d are monotone. Fig. 4.5 shows the attacker's utility function at different migration rates, verifying Theorem 15. We numerically verify the monotonicity of u_a for $\tau_d = 0.1$ and $\tau_d = 2$. Let $\lambda_{\min} = 1$ and $\lambda_{\max} = 3$, hence according to Theorem 15, u_a is monotonically decreasing when $C_a > 0.04$ when $\tau_a = 0.1$. However, at $\tau_d = 2$, the attack cost $C_a > 0.98$ ensures that u_a is monotonically decreasing in λ_a. In Fig. 4.5 where $C_a = 0.1$, it is shown that the corresponding u_a is monotonically decreasing for

all $\lambda_a \in [1, 3]$ for $\tau_d = 0.1$. At $\tau_d = 2$ when the condition on C_a is not satisfied, the payoff u_a is not monotonically decreasing. Next, we study and discuss the effect of different system parameters on the players' payoff and best response in comparison to other defense and attack policies.

4.6.1 Utility functions

Fig. 4.6 shows the payoff function of the defender u_d versus the migration time τ_d for $C_d = 0.3$, $\tau_{\min} = 0.1$, and $T = 3$. The figure highlights the tradeoff of the defender as he seeks to optimize τ_d to secure the system through VM migration while avoiding a large overhead. Evidently, the optimal migration time τ_d^* depends on the attacker's strategy λ_a. The tradeoff shown in Fig. 4.6 agrees with our intuition based on the game model. Specifically, a very small τ_d signifying a high VM migration rate is associated with a high migration cost that dominates the payoff function u_d. On the other hand, with a larger τ_d, the VMs dwell for a longer duration on the same physical node leaving more room for the attacker to collocate and steal data from her target VM. In Fig. 4.6, we compare the defender's utility at different attack rates λ_a. Clearly, when the attack is more aggressive, the defender is able to maximize his payoff by reducing the migration time τ_d at the expense of higher migration cost. Therefore, when λ_a increases from 1 to 2.5, the optimal τ_d reduces from 0.8 to 0.6.

In Fig. 4.7, we plot the attacker's expected payoff u_a versus the attack rate λ_a for different defense actions τ_d for an attack cost $C_a = 0.2$. As shown, the optimal attack rate depends on the defender's action. As the attack rate increases, the cost of attack increases and eventually becomes the dominant term in the payoff function. Moreover, as the defender reduces his time to migrate τ_d, the attacker's utility decreases. This is due to the fact that when τ_d is small (a higher migration rate), there is a shorter time window for the attacker to successfully collocate with her victim. Contrariwise, when the migration rate is not too high (i.e., τ_d is fairly large), the attacker can max-

imize her utility by increasing the attack rate λ_a. For example, when τ_d is reduced from $\tau_d = 3.5$ to $\tau_d = 1.5$, the optimal attack rate that maximizes the payoff u_a decreases from $\lambda_a^* = 2.21$ to $\lambda_a^* = 2$. However, if the defender is migrating the VMs at a very high rate, i.e., τ_d is very small, the attacker's best response is to attack at the minimum possible rate or completely back-off since the attack is useless. To better understand the effect of the migration (attack) cost on the optimal migration (attack) rate for the defender (attacker), in the following two subsections we study the behavior of the payoff functions at different values of the cost. We also study the behavior of the best response curves to gain more insight into the tradeoffs associated with this game.

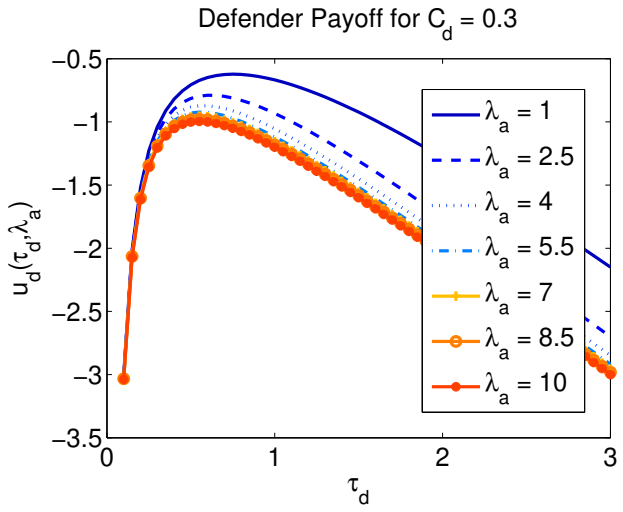

Figure 4.6: Defender's utility versus migration time τ_d

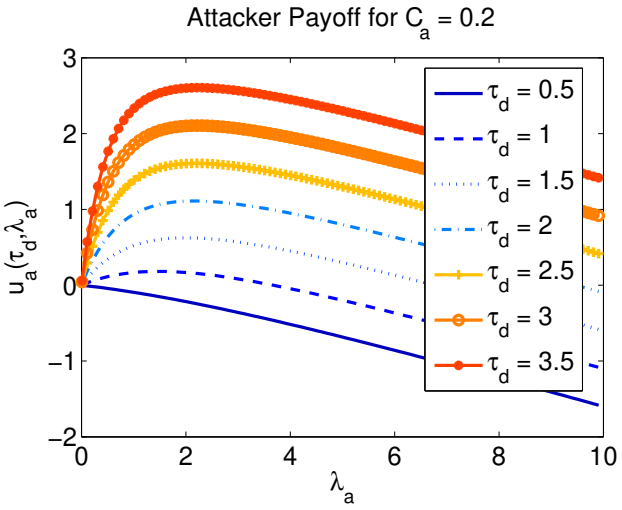

Figure 4.7: Attacker's utility versus attack rate λ_a

4.6.2 Cost effect and monotonicity

To show the effect of the migration and attack costs C_d and C_a, we plot the players' utility functions for different values of the cost. In Fig. 4.8, we plot the defender's payoff versus τ_d for different attack strategies for a fairly small migration cost $C_d = 0.03$. At this small migration cost, the defender's best response is to always migrate at the highest permissible rate, i.e., $\tau_d^* = \tau_{\min}$ regardless of the attack rate λ_a. Hence, the leakage loss term dominates the defender's payoff function u_d at this small migration cost. Indeed, referring to (4.12), u_d is monotonically decreasing in τ_d when $C_d \to 0$. On the other hand, when the migration cost is too high as shown in Fig. 4.9 where $C_d = 10$, the defender's best response is $\tau_d^* = T$ to reduce the associated migration cost. We remark that the utility function is monotonically increasing in τ_d for such high migration cost, a fact which was established analytically in Theorem 16.

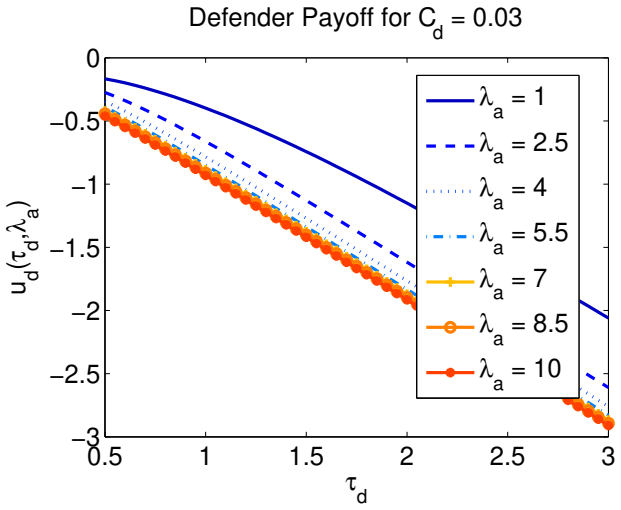

Figure 4.8: Defender's utility versus migration time τ_d for $C_d = 0.03$

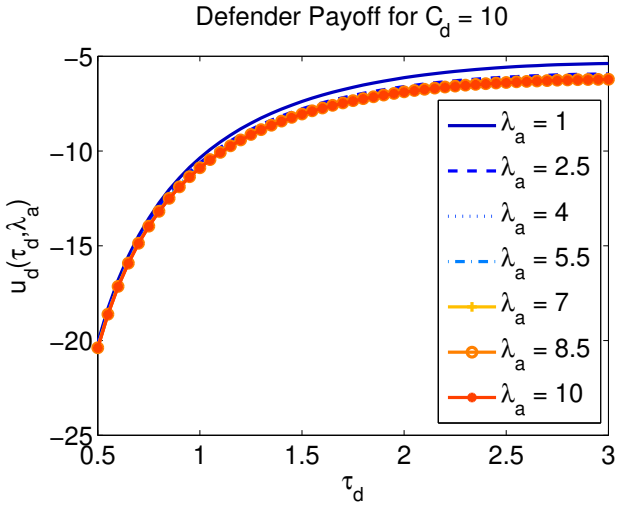

Figure 4.9: Defender's utility versus migration time τ_d for $C_d = 10$

Similarly, the effect of the attack cost C_a can be shown in Fig. 4.10 and 4.11. At a very small attack cost, $C_a = 0.01$, as shown in Fig. 4.11, the attacker's best attack strategy is to attack aggressively at $\lambda_{\max}$ to maximize the chances of successful collocation regardless of the defender's

action. Recalling the attacker's payoff function in (4.11), u_a is monotonically increasing in λ_a when $C_a \to 0$. In case of a high attack cost, the behavior of the payoff function is reversed as shown in Fig. 4.11 where $C_a = 6$. In this case, the cost of the attack term dominates the payoff function. Therefore, the best action for the attacker is $\lambda_{\min}$ regardless of the action of the defender. This behavior is confirmed by the analysis in Theorem 15.

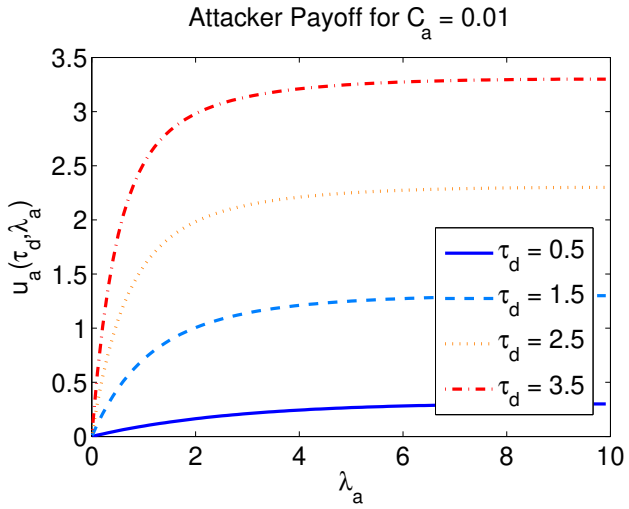

Figure 4.10: Attacker's utility versus attack rate λ_a for $C_a = 0.01$

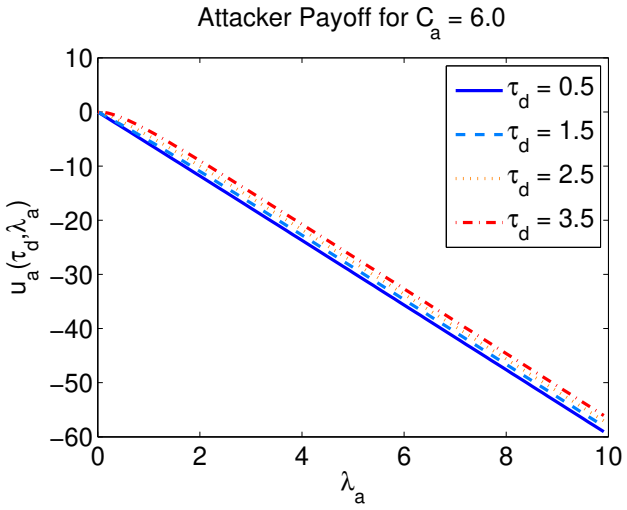

Figure 4.11: Attacker's utility versus attack rate λ_a for $C_a = 6$.

4.6.3 Best response curves

In this section, we study the best response curves for both players based on Definition 1 to provide more insight into the optimal action of a player as function of the action of the opponent. The solid blue line in Fig. 4.12 shows the defender's best response curve τ_d^* as function of λ_a. The attacker's best response curve λ_a^* as function of the defender's action τ_d is shown in dashed red line. In this scenario, we set $T = 3$, $\lambda_{\max} = 3$, $C_d = 0.3$, and $C_a = 0.1$. In Fig. 4.12, the intersection point of the two response curves is the unique NE. The point(s) of equilibria depend on the values of C_a and C_d as detailed next. The best response curves also underscore the tradeoff for each player. For example, at equilibrium the defender migrates with $\tau_d = 0.66$ while the attacker uses rate $\lambda_a = 1.8$ for the attack. Clearly, at low attack rate, VM migration at a very small migration rate, i.e, larger τ_d, is more favorable. As the attack rate increases, the defender is urged to migrate the VMs at faster rate, wherefore τ_d^* decreases as λ_a increases. On the attacker's side, a similar tradeoff is observed. The attacker attacks the system at the minimum rate $\lambda_{\min}$ as long as the VM stays on

the same physical node for a duration $\tau_d < 0.4$ since it is very hard to collocate when migration is taking place at such high rates. If the defender increases the time before migrating, i.e $\tau_d > 0.4$, the attacker is enticed to attack the system at higher rates to increase the amount of data leaked out to the attacker as long as the defender is reducing the migration rate. The best response curves also demonstrate the monotonicity of the payoff functions with respect to C_a and C_d as explained earlier in Section 4.6.2. To show this, Fig. 4.13, 4.14 and 4.15 illustrate the best response curves at extreme cost values. In particular, in Fig. 4.13, both C_d and C_a are set to zero. It is obvious that the defender is migrating with the highest permissible frequency such that, $\tau_d^* = \tau_{\min}$ for any attack rate. In response, the attacker's best action is $\lambda_a = \lambda_{\max}$ regardless of the defender's action. Hence, when the costs of migration and attack are zero, both players do not face any tradeoffs and the game is zero-sum. Fig. 4.14 shows another extreme scenario where only the defender faces a very high cost for migration. His best response is $\tau_d^* = T = 3$, which corresponds to the lowest migration rate possible. In Fig. 4.15, the attack cost $C_a = 6$ while the defender incurs zero cost for migration. Hence, the defender adopts the highest migration rate at $\tau_d^* = \tau_{\min}$ against any attack rate. In response, it is more rewarding for the attacker to attack at $\lambda_{\min}$ for any τ_d.

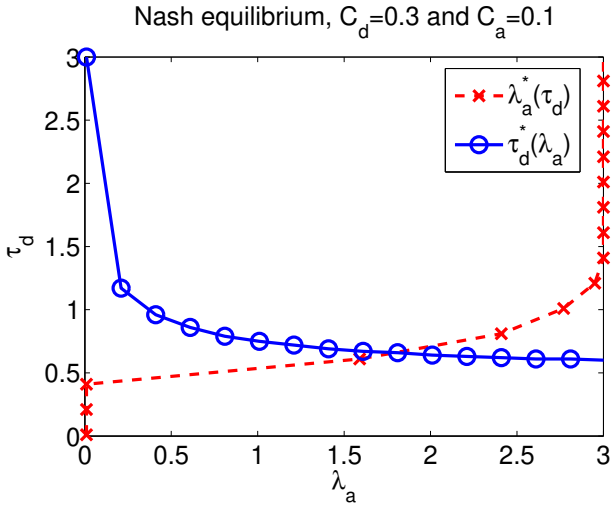

Figure 4.12: Players best response curves for $C_d = 2.5$ and $C_a = 1$

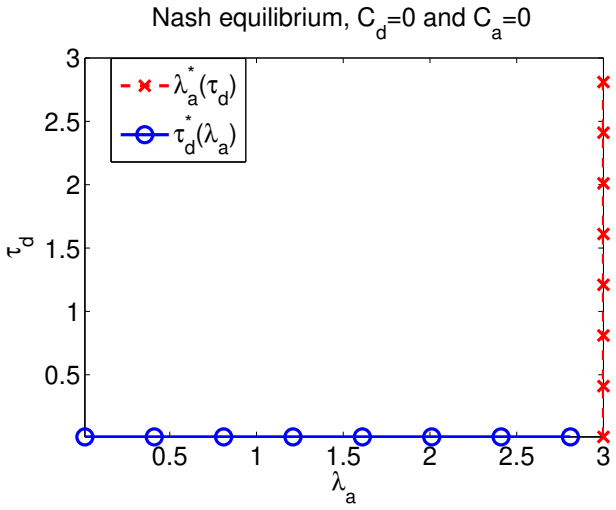

Figure 4.13: Players best response curves for $C_d = 0$ and $C_a = 0$

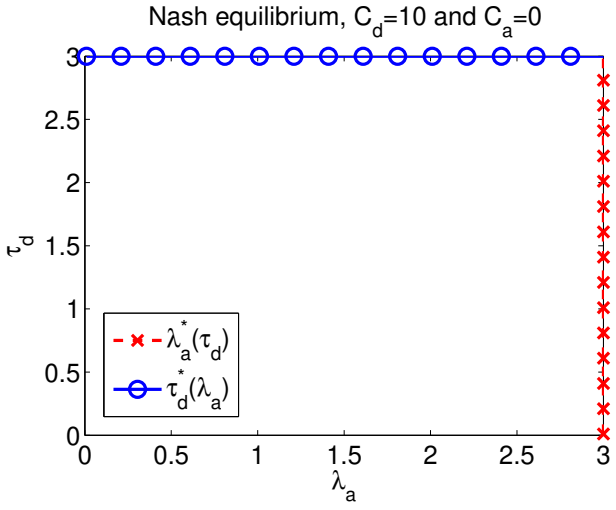

Figure 4.14: Players best response curves for $C_d = 10$ and $C_a = 0$

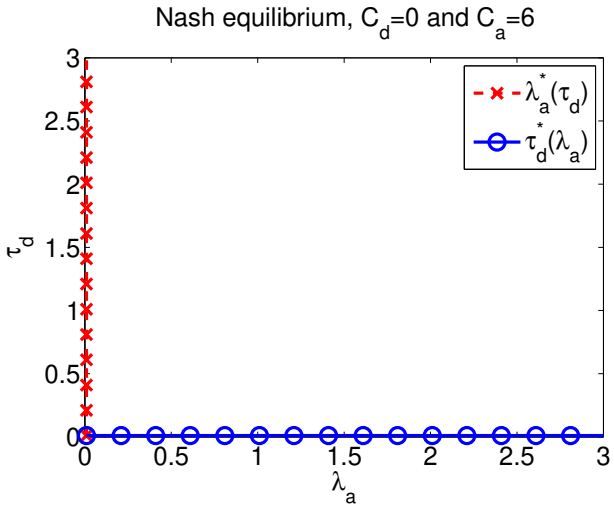

Figure 4.15: Players best response curves for $C_d = 0$ and $C_a = 6$

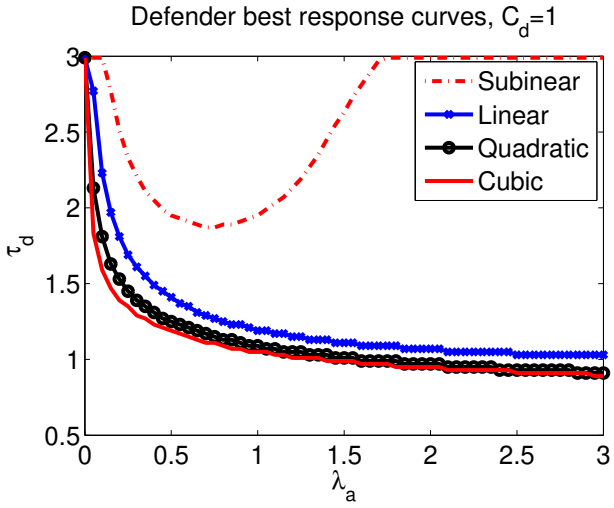

Figure 4.16: Defender's best response curves for different reward scaling regimes.

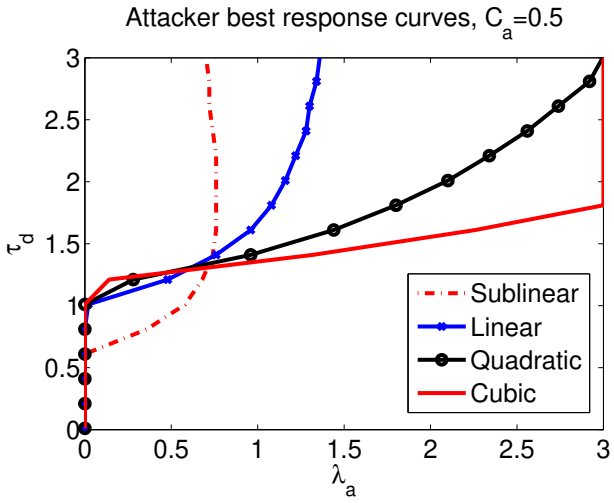

Figure 4.17: Attacker's best response curves for different reward scaling regimes.

4.6.4 Different reward scaling regimes

In the numerical analysis above, we considered the reward function G to be linearly increasing in the collocation duration. However, the reward function need not be linear. In this section, we study other scaling regimes. In particular, we consider the scenario where the reward $G(\tau_d, \tau_a)$ scales sub-linearly, quadratically or cubically with the collocation duration. In other words, $G(\tau_d, \tau_a) = \max\{0, (\tau_d - \tau_a)^n\}$, where $n = \{0.2, 1, 2, 3\}$. The scaling of the reward function relates to the power of the attack as well as the vulnerability of the system under attack. The more powerful the attacker, the more data she is able to read out for a given duration of collocation. The power of the attack also depends on the side-channel technique used. The authors in [92] have shown that the amount of data leaked practically depends on the technique used to access the last level cache (LLC) (e.g., PRIME+PROBE and FLUSH+RELOAD attacks [102, 105]). For example, to read the cache, an attacker would need to adjust the time of the PROBE phase, which in turn affects the error rate of the attack covert channel.

In Fig. 4.16, we plot the defender's best response curves for sub-linear, linear, quadratic, and cubic reward functions. Intuitively, higher order reward functions are more disposed to dominate the payoff functions than for the linear and sub-linear scaling. In Fig. 4.16, the migration cost is set to $C_d = 1$. For the sub-linear regime, the defender's payoff function is more dominated by the migration cost. As the attack rate λ_a increases, the defender is urged to migrate the VMs at a faster rate (wherefore τ_d^* decreases), but only until a certain point where faster migration becomes futile. Indeed, when the attack rate is overwhelming, it is more rewarding for the defender to use a large τ_d to alleviate high migration costs. For the linear regimes, the defender is facing exactly the same tradeoff discussed earlier in Section 4.6.3. Similarly, for the higher order reward regimes, the leakage term dominates the payoff over the entire range of attack rates. Therefore, the defender is consistently urged to increase the migration rate as the attacker increases her attack rates. With the quadratic and cubic reward functions the defender's best response is shown to exhibit a similar behavior, but conceivably the cubic reward incentivizes a faster increase in the rate of migration.

In Fig. 4.17, the attacker's best response curves are plotted for different reward functions. The higher the order of the reward regime, the more is the attacker enticed to attack. For the sub-linear regime, the attacker attacks at rates higher than $\lambda_{\min}$ when $\tau_d > 0.6$. However, in the linear regime, the attacker's best response rate is non-vanishing and increasing in τ_d for $\tau_d > 1$, reaches $\lambda_a = 1.4$ as soon as the cost of the attack starts to dominate the attacker's payoff. Evidently, the higher the order of the reward, the more is the attacker willing to attack at higher rates. As shown in Fig. 4.17, the cubic regime is extremely rewarding to the attacker, and as a result the attacker affords to attack at the maximum permissible rate as the reward term dominates her utility function.

4.6.5 Game simulation and implementation

In this section, we compare the payoff of both players playing NE strategies to the payoffs of other defense and attack strategies. As per our theoretical analysis in Section 4.4, the players' optimal (NE) policies depend on the values of the associated costs C_d and C_a. Table 4.1 presents the results of a simulation of the game for the linear reward regime in which $G(\tau_d, \tau_a) = (\tau_d - \tau_a)^+ = \max(\tau_d - \tau_a, 0)$ at different values of C_d and C_a. For the numerical results, the maximum collocation time is set to $T = 3$ and the maximum attack rate is $\lambda_{\max} = 3$. The results underscore that a rational attacker would adapt the attack rate to the attack cost to avoid incurring high cost and/or launching useless attacks. For example, when $C_a = 8$, the payoff corresponding to the NE converges to that of the No Attack strategy, which is substantially higher than the payoff of an aggressive attacker of -23.68 due to a substantial attack cost. Similarly, the defender should not resort to very high frequency migration (equivalently, small τ_d), unless the migration cost is fairly low. For example, the results in the table show that the payoff of the defender adopting the NE policy tends to that of the No defense policy as C_d increases. The last column designated as worst case (for the defender) corresponds to the scenario where the attacker is attacking at the highest rate while the defender does not adopt any migration policy. The loss of the defender for not migrating compared to the NE strategy is more pronounced when the NE point has $\tau_d^* < T$, i.e., when the defender is in a position to defend the system through VM migrations.

Real system implementation: To demonstrate the effectiveness of the proposed approach, we implemented the migration defense approach on a proof-of-concept cloud setup using the Xen hypervisor [137], which allows us to run many instances of an operating system on a single machine. The setup is composed of five physical nodes in addition to the orchestrating controller node. The specifications of the nodes are: Dell Inc. PowerEdge 1900 Intel(R) Xeon(R), 4-core CPU E5335 (2.00GHz) and 8GB Ram running Ubuntu 16.04.4 LTS with Xen 4.6. Each hypervisor initially

runs 20 VMs. The number of VMs residing on each hypervisor changes slightly over time due to live migration. Three target VMs are uniformly distributed over the five hypervisors. We validate our results by comparing the performance of the proposed defense approach to a no-defense approach and to a random migration defense policy with a uniform distribution over the interval $[\tau_{\min}, T]$. In Fig. 4.20, we plot the collocation duration per target VM (left y-axis) and the average number of collocation events (right y-axis). As shown, the no-migration approach results in collocation events of longer duration. The proposed defense approach can reduce both the durations and number of occurrences of collocation events. VM migration defense policies are shown to reduce the duration of collocation events by half. In Fig. 4.21, we evaluate the defender's utility at different migration cost values. The proposed defense is shown to outperform the random migration policy, which does not adapt to the migration cost C_d as it chooses a random τ_d.

4.6.6 Extended model with IDS

In Fig 4.18, we compare the attacker's payoff with and without an IDS in place based on the analysis in Section 4.5. In this experiment, we set $D = 0.2$, $C_a = 0.2$ for different stop times s. It is clear that the IDS drastically reduces the attacker's utility. In addition, while the IDS is in place the attacker can increase her expected payoff by shortening the attack duration. Hence, the defender's utility increases since the amount of data leaked out is reduced.

Fig. 4.19 illustrates the attacker's payoff as function of the attack stopping time s for different values of the detection cost D. Obviously, the best time to stop the attack depends on the detection cost D. As D increases, the attacker's payoff decreases and the optimal stopping time s (corresponding to the highest payoff) is shown to decrease. At a certain point, the attacker is forced to stop as soon as she collocates to evade a high penalty if detected.

Table 4.1: Players' Payoff For Several Attack and Defense Strategies.

Cost		NE		No Defense		No Attack		Aggressive Attack		Worst case	
C_d	C_a	u_d	u_a	u_d	u_a	u_d	u_a	u_d	u_a	u_d	u_a
0	0	-1.49E-04	1.49E-04	-2.6778	2.6778	-5.00E-07	5.00E-07	-1.47E-04	1.47E-04	-2.6722	2.6722
0.1	0	-0.4184	0.1745	-2.711	2.6778	-0.2447	8.39E-04	-0.4164	0.1725	-2.7054	2.6722
0.1	0.1	-0.292	0.0147	-1.9578	1.8396	-0.1937	3.50E-04	-0.447	-0.0414	-2.7054	2.3762
0	0.1	-5.00E-07	-1.00E-03	-0.0448	0.0438	-5.00E-07	-1.00E-03	-1.47E-04	-0.2959	-2.6722	2.3762
0.4	0.2	-0.7561	0.0603	-2.2234	1.8825	-0.4912	0.0014	-0.9998	-0.08	-2.8051	2.0802
0.4	0.4	-0.6141	0.0331	-1.5869	1.254	-0.3864	0.0015	-1.1082	-0.4567	-2.8051	1.4882
0.4	0.6	-0.5024	0.0158	-1.0553	0.7664	-0.3379	0.0013	-1.2121	-0.8944	-2.8051	0.8962
0.8	0.6	-0.9674	0.0732	-1.7374	1.1656	-0.5974	0.0032	-1.6164	-0.7478	-2.938	0.8962
0.8	1	-0.7914	0.0345	-1.1881	0.6624	-0.5098	0.0029	-1.7719	-1.685	-2.938	-0.2878
2	4	-0.9212	0.0145	-0.9206	0.0162	-0.7112	0.0046	-3.3289	-9.1778	-3.3367	-9.1678
10	0	-6	2.6667	-5.999	2.6767	-3.3779	0.0446	-5.9955	2.6622	-5.9945	2.6722
0	8	-5.00E-07	-0.08	-0.0448	-0.0352	-5.00E-07	-0.08	-1.47E-04	-23.6799	-2.6722	-21.0078

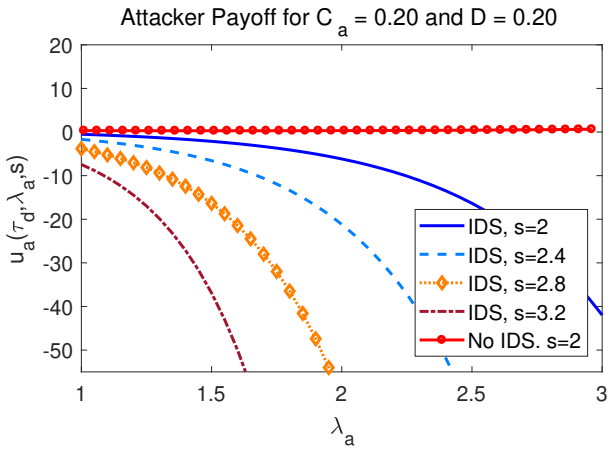

Figure 4.18: Attacker's payoff with and without IDS.

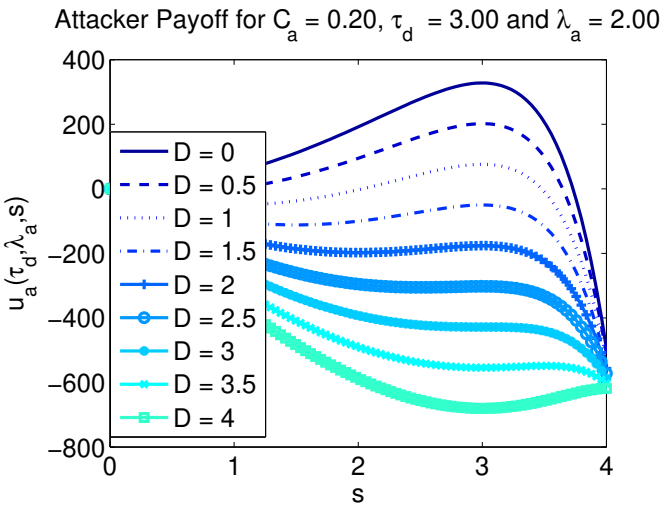

Figure 4.19: Attacker's utility vs stopping time s at different costs of detection.

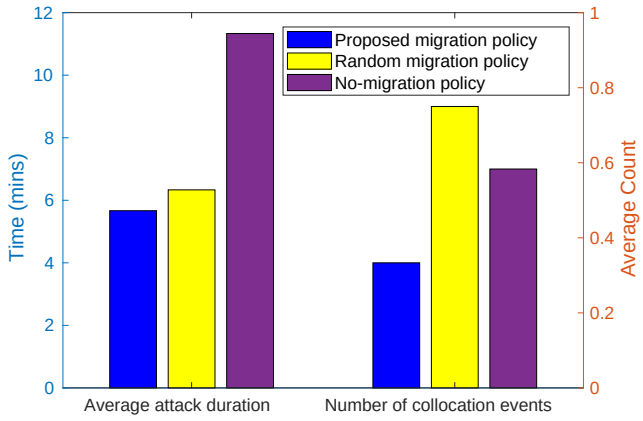

Figure 4.20: Comparison between the proposed migration approach and no-migration.

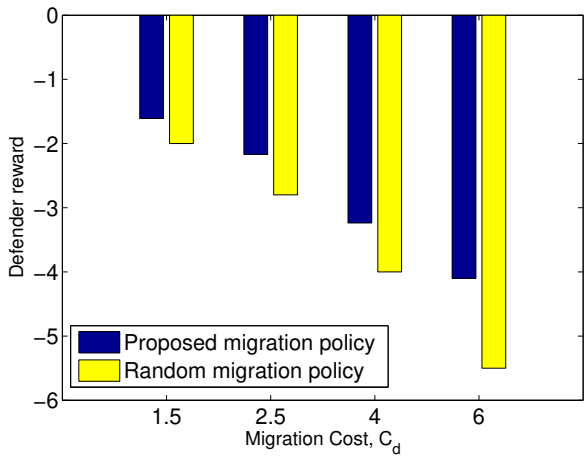

Figure 4.21: Comparing the defender's utility with the proposed and the random migration policies at several migration cost values.

CHAPTER 5: CONCLUSION

We investigated the security of DCA in wireless networks. We formulated the attack as an MDP problem solved by the attacker to yield pinball jamming attacks that optimize the tradeoff between the damage inflicted in creating conflicts and the attack cost. Given that the original MDP is not scalable, we proposed an approximate yet effective policy which was shown to outperform other policies such as DoS, MCN, myopic and random attacks. The proposed policy judiciously adapts the level of the attack from highly aggressive at low attack costs to less aggressive at higher costs. Pinball attacks are shown to cause cascading channel switching and prevent the network from resolving conflicts to reach a zero-conflict state.

The performance gaps of the different policies compared to pinball attacks are more pronounced in sparser graphs since pinball attacks are capable of exploiting the graph structure and the variability in node degrees to yield substantially higher rewards. Also, we formulated the dual problem of attacking the network communication links and established the equivalence of the edge and node attack problems. We also extended pinball strategies to large scale networks. We considered large graphs with repeated structures and derived policies obtained as solutions to MDPs corresponding to the small repeating subgraphs. Based on a thorough experimental evaluation with numerous topologies, we have shown that the proposed pinball policy outperforms the other attack policies in different scenarios.

On the defense side, we proposed a game-theoretic defense against jamming attacks via adapting the transmission profiles of the wireless access points, i.e. adapting the underlying network topology. Given the combinatorial complexity of the best response strategy, we developed scalable strategies based on newly proposed game decompositions in which the defender solves multiple, yet simple, sub-games instead of one large combinatorial game. The decoupled approach was

shown to yield mixed strategies with performance close to (and could outperform) that of the full-scale NE. They also come with marginal reward loss against computationally unrestrained adversaries. Also, we developed a scalable marginal strategy for selecting the footprints in terms of expected power levels which enforce a minimum coverage constraint. The proposed marginal strategy is shown to substantially enhance the network dependability against jamming attacks over uniform power allocation. Further, we extended the marginal-based approach and considered a continuous game in which the defender and the jammer assign powers from a continuous set. We characterized and proved uniqueness of a pure strategy NE for such game.

Finally we developed a moving target defense framework for the VM migration timing problem. Live migration of VMs between different physical nodes is studied in a game-theoretic framework to defend multi-tenant clouds against side channel attacks launched by malicious users co-residing on the same physical node. We characterized best strategies for the players and established NE existence conditions. We also considered an extended system model in which the cloud is equipped with an IDS. The IDS is a reactive defense approach which, combined with our proactive VM migration defense approach, enhances the cloud security against side channel attacks. We also verified our theoretical results numerically for different settings of the game. The theoretical and numerical analyses provided characterize the performance of the migration defense approach against collocation attacks. We also demonstrated the proposed migration defense on a cloud network implemented in a Xen-based cluster.

LIST OF REFERENCES

[1] Cisco. Radio resource management white paper: Chapter: Dynamic channel assignment (DCA). http://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-2/b_RRM_White_Paper/b_RRM_White_Paper_chapter_0100.html, 2016.

[2] Ed Tittel. Sdn vs. nfv: What's the difference? URL https://www.cisco.com/c/en/us/solutions/software-defined-networking/sdn-vs-nfv.html.

[3] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1):1–11, 2011.

[4] Robinson E Pino. *Network science and cybersecurity*. Springer, 2014.

[5] Roger B Myerson. *Game theory*. Harvard university press, 2013.

[6] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[7] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[8] Bruce DeBruhl, Christian Kroer, Anupam Datta, Tuomas Sandholm, and Patrick Tague. Power napping with loud neighbors: optimal energy-constrained jamming and anti-jamming. In *Proceedings of the ACM conference on Security and privacy in wireless & mobile networks*, pages 117–128, 2014.

[9] Giacomo Bacci, Luca Sanguinetti, and Marco Luise. Understanding game theory via wireless power control [lecture notes]. *IEEE Signal Processing Magazine*, 32(4):132–137, 2015.

[10] Ahmed H Anwar, Janiece Kelly, George Atia, and Mina Guirguis. Pinball attacks against dynamic channel assignment in wireless networks. *Computer Communications*, Available online, 16 April 2019.

[11] Ahmed H Anwar, George Atia, and Mina Guirguis. Adaptive topologies against jamming attacks in wireless networks: A game-theoretic approach. *Journal of Network and Computer Applications*, 121:44–58, 2018.

[12] Ahmed H Anwar, George Atia, and Mina Guirguis. Dynamic game-theoretic defense approach against stealthy jamming attacks in wireless networks. In *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*, pages 252–258. IEEE, 2017.

[13] Ahmed H Anwar, George Atia, and Mina Guirguis. A game-theoretic framework for the virtual machines migration timing problem. *IEEE Transactions on Cloud Computing*, 2019.

[14] Cisco. Cisco visual networking index: Forecast and methodology 2016–2021, 2017.

[15] Joseph Mitola III and Gerald Q Maguire Jr. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, 1999.

[16] Christopher Monsanto, Joshua Reich, Nate Foster, Jennifer Rexford, and David Walker. Composing software defined networks. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 1–13, 2013.

[17] David López-Pérez, Alvaro Valcarce, Guillaume De La Roche, and Jie Zhang. OFDMA femtocells: A roadmap on interference avoidance. *IEEE Communications Magazine*, 47(9): 41–48, 2009.

[18] Surachai Chieochan, Ekram Hossain, and Jeffrey Diamond. Channel assignment schemes

for infrastructure-based 802.11 WLANs: A survey. *IEEE Communications Surveys & Tutorials*, 12(1):124–136, 2010.

[19] Paul Goransson, Chuck Black, and Timothy Culver. *Software defined networks: a comprehensive approach*. Morgan Kaufmann, 2016.

[20] Mithun Acharya and David Thuente. Intelligent jamming attacks, counterattacks and $(\text{counter})^2$ attacks in 802.11b wireless networks. In *Proceedings of the OPNETWORK Conference*, Washington DC, USA, 2005.

[21] John Bellardo and Stefan Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *USENIX security*, pages 15–28, 2003.

[22] B Konings, Florian Schaub, Frank Kargl, and Stefan Dietzel. Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard. In *IEEE 34th Conference on Local Computer Networks (LCN)*, pages 14–21. IEEE, 2009.

[23] Adnan Akhunzada, Abdullah Gani, Nor Badrul Anuar, Ahmed Abdelaziz, Muhammad Khurram Khan, Amir Hayat, and Samee U Khan. Secure and dependable software defined networks. *Journal of Network and Computer Applications*, 61:199–221, 2016.

[24] David Murray, Michael Dixon, and Terry Koziniec. Scanning delays in 802.11 networks. In *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST'07)*, pages 255–260. IEEE, 2007.

[25] Eldad Perahia and Robert Stacey. *Next generation wireless LANs: 802.11n and 802.11ac*. Cambridge university press, 2013.

[26] Arunesh Mishra, Suman Banerjee, and William Arbaugh. Weighted coloring based channel assignment for WLANs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):19–31, 2005.

[27] Mohamad Haidar, Rabindra Ghimire, Hussain Al-Rizzo, Robert Akl, and Yupo Chan. Channel assignment in an ieee 802.11 wlan based on signal-to-interference ratio. In *Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 001169–001174. IEEE, 2008.

[28] Krishna N Ramachandran, Elizabeth M Belding-Royer, Kevin C Almeroth, and Milind M Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Infocom*, volume 6, pages 1–12, 2006.

[29] Anand Prabhu Subramanian, Himanshu Gupta, Samir R Das, and Jing Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE transactions on mobile computing*, 7(12):1459–1473, 2008.

[30] Mahesh K Marina, Samir R Das, and Anand Prabhu Subramanian. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Computer networks*, 54(2):241–256, 2010.

[31] Ejaz Ahmed, Abdullah Gani, Saeid Abolfazli, Liu Jie Yao, and Samee U Khan. Channel assignment algorithms in cognitive radio networks: Taxonomy, open issues, and challenges. *IEEE Communications Surveys & Tutorials*, 18(1):795–823, 2014.

[32] Elias Z Tragos, Sherali Zeadally, Alexandros G Fragkiadakis, and Vasilios A Siris. Spectrum assignment in cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 15(3):1108–1135.

[33] Yueshi Wu and Mihaela Cardei. Multi-channel and cognitive radio approaches for wireless sensor networks. *Computer Communications*, 94:30–45, 2016.

[34] Qijun Gu, Meng Yu, Wanyu Zang, and Peng Liu. Lightweight attacks against channel

assignment protocols in MIMC wireless networks. In *IEEE International Conference on Communications (ICC)*, pages 1–6, 2011.

[35] 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.

[36] David Thuente and Mithun Acharya. Intelligent jamming in wireless networks with applications to 802.11 b and other networks. In *Proc. 25th IEEE Communications Society Military Communications Conference (MILCOM06), Washington, DC*, pages 1–7, 2006.

[37] A. Naveed and S. Kanhere. Security vulnerabilities in channel assignment of multi-radio multi-channel wireless mesh networks. In *IEEE Global Telecommunications Conference*, pages 1–5. IEEE, 2006.

[38] Loukas Lazos, Sisi Liu, and Marwan Krunz. Mitigating Control-channel Jamming Attacks in Multi-channel Ad hoc Networks. In *WiSec*, 2009.

[39] Sisi Liu, Loukas Lazos, and Marwan Krunz. Thwarting Control-channel Jamming Attacks from Inside Jammers. *IEEE Transactions on Mobile Computing*, 11(9), 2012.

[40] Wenyuan Xu. On adjusting power to defend wireless networks from jamming. In *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pages 1–6, 2007.

[41] Alireza Attar, Helen Tang, Athanasios V Vasilakos, F Richard Yu, and Victor CM Leung. A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE*, 100(12):3172–3186, 2012.

[42] Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. SDN security: A survey. In *IEEE SDN for Future Networks and Services*, pages 1–7, 2013.

[43] Yan Zhang and Loukas Lazos. Vulnerabilities of Cognitive Radio MAC Protocols and Countermeasures. *IEEE Networks*, 27(3), 2013.

[44] Beibei Wang, Yongle Wu, KJ Liu, and T Charles Clancy. An anti-jamming stochastic game for cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 29(4): 877–889, 2011.

[45] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[46] Yang Xiao. Ieee 802.11n: enhancements for higher throughput in wireless lans. *Wireless Communications, IEEE*, 12(6):82–91, Dec 2005. ISSN 1536-1284. doi: 10.1109/MWC.2005.1561948.

[47] Z. Liu, H. Liu, W. Xu, and Y. Chen. Exploiting jamming-caused neighbor changes for jammer localization. *IEEE Transactions on Parallel and Distributed Systems*, 23(3):547–555, March 2012. ISSN 1045-9219.

[48] X. Wei, Q. Wang, T. Wang, and J. Fan. Jammer localization in multi-hop wireless network: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 19(2):765–799, 2017.

[49] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[50] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[51] Ian F Akyildiz, Won-Yeol Lee, Mehmet C Vuran, and Shantidev Mohanty. Next generation/-dynamic spectrum access/cognitive radio wireless networks: a survey. *Computer networks*, 50(13):2127–2159, 2006.

[52] Cisco. Radio resource management white paper: Chapter: Transmit power control (tpc) algorithm. `http://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-2/b_RRM_White_Paper/b_RRM_White_Paper_chapter_0101.html`, 2016.

[53] Gianmarco Baldini, Taj Sturman, Abdur Rahim Biswas, Ruediger Leschhorn, Győző Gódor, and Michael Street. Security aspects in software defined radio and cognitive radio networks: a survey and a way ahead. *Communications Surveys & Tutorials, IEEE*, 14(2):355–379, 2012.

[54] Bruce DeBruhl and Patrick Tague. How to jam without getting caught: Analysis and empirical study of stealthy periodic jamming. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 496–504. IEEE, 2013.

[55] Janiece Kelly, Mina Guirguis, and George Atia. Pinball attacks: Exploiting channel allocation in wireless networks. In *Proceedings of the IEEE ICC*, Kuala Lumpur, Malaysia, 2016.

[56] Matthias Ihmig and Peter Steenkiste. Distributed dynamic channel selection in chaotic wireless networks. In *13th European Wireless Conference, Paris, France*, 2007.

[57] Yongle Wu, Beibei Wang, and KJ Ray Liu. Optimal power allocation strategy against jamming attacks using the colonel blotto game. In *Global Telecommunications Conference (GLOBECOM)*, pages 1–5, 2009.

[58] Mingyan Li, Iordanis Koutsopoulos, and Radha Poovendran. Optimal jamming attacks and

network defense policies in wireless sensor networks. In *26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1307–1315, 2007.

[59] Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordónez, and Milind Tambe. Security games with arbitrary schedules: A branch and price approach. In *Proceedings of AAAI*, 2010.

[60] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 327–334, 2011.

[61] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 689–696, 2009.

[62] Ahmed H Anwar, Janiece Kelly, George Atia, and Mina Guirguis. Stealthy edge decoy attacks against dynamic channel assignment in wireless networks. In *IEEE Military Communications Conference, MILCOM*, pages 671–676, 2015.

[63] Feng Li, Yinying Yang, and Jie Wu. Attack and flee: game-theory-based analysis on interactions among nodes in manets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(3):612–622, 2010.

[64] Sherif Khattab, Daniel Mosse, and Rami Melhem. Jamming mitigation in multi-radio wireless networks: Reactive or proactive? In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, page 27. ACM, 2008.

[65] Haijun Zhang, Chunxiao Jiang, Norman C Beaulieu, Xiaoli Chu, Xianbin Wang, and

Tony QS Quek. Resource allocation for cognitive small cell networks: A cooperative bargaining game theoretic approach. *IEEE Transactions on Wireless Communications*, 14(6): 3481–3493, 2015.

[66] Haijun Zhang, Jiali Du, Julian Cheng, Keping Long, and Victor CM Leung. Incomplete CSI based resource optimization in SWIPT enabled heterogeneous networks: A non-cooperative game theoretic approach. *IEEE Transactions on Wireless Communications*, 17(3):1882–1892, 2018.

[67] H. Zhang, N. Yang, K. Long, M. Pan, G. K. Karagiannidis, and V. C. M. Leung. Secure communications in NOMA system: Subcarrier assignment and power allocation. *IEEE Journal on Selected Areas in Communications*, pages 1–1, 2018. ISSN 0733-8716. doi: 10.1109/JSAC.2018.2825559.

[68] Eitan Altman, Konstantin Avrachenkov, and Andrey Garnaev. Fair resource allocation in wireless networks in the presence of a jammer. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, pages 33:1–33:7, 2008.

[69] Eitan Altman, Konstantin Avrachenkov, and Andrey Garnaev. *Transmission Power Control Game with SINR as Objective Function*, pages 112–120. Springer Berlin Heidelberg, 2009.

[70] Jianchao Zheng, Yuan Wu, Ning Zhang, Haibo Zhou, Yueming Cai, and Xuemin Shen. Optimal power control in ultra-dense small cell networks: A game-theoretic approach. *IEEE Transactions on Wireless Communications*, 16(7):4139–4150, 2017.

[71] Yongle Wu, Beibei Wang, KJ Ray Liu, and T Charles Clancy. Anti-jamming games in multi-channel cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 30 (1):4–15, 2012.

[72] Mohammad Ashiqur Rahman, Mohammad Hossein Manshaei, and Ehab Al-Shaer. A game-theoretic approach for deceiving remote operating system fingerprinting. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 73–81. IEEE, 2013.

[73] Manjesh K Hanawal, Mohammad J Abdel-Rahman, and Marwan Krunz. Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems. *IEEE Transactions on Mobile Computing*, 15(9):2247–2259, 2016.

[74] Changlong Chen, Min Song, ChunSheng Xin, and Jonathan Backens. A game-theoretical anti-jamming scheme for cognitive radio networks. *IEEE Network*, 27(3):22–27, 2013.

[75] Dejun Yang, Guoliang Xue, Jin Zhang, Andrea Richa, and Xi Fang. Coping with a smart jammer in wireless networks: A stackelberg game approach. *IEEE Transactions on Wireless Communications*, 12(8):4038–4047, 2013.

[76] Marwaan Simaan and Jose B Cruz. On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555, 1973.

[77] Eric W Weisstein. Circle-circle intersection. 2003.

[78] IBM ILOG CPLEX optimization studio. URL `https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer`.

[79] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Designing random allocation mechanisms: Theory and applications. *The American Economic Review*, 103(2): 585–623, 2013.

[80] Eitan Altman, Konstantin Avrachenkov, and Andrey Garnaev. Fair resource allocation in wireless networks in the presence of a jammer. *Performance Evaluation*, 67(4):338–349, 2010.

[81] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[82] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[83] J Ben Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965.

[84] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[85] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157, pages 157–163, 1994.

[86] Ahmed H Anwar, George Atia, and Mina Guirguis. Game theoretic defense approach to wireless networks against stealthy decoy attacks. In *54th Annual Allerton Conference on Communication, Control, and Computing*, pages 816–821, 2016.

[87] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[88] Kui Ren, Cong Wang, and Qian Wang. Security challenges for the public cloud. *IEEE Internet Computing*, 16(1):69–73, 2012.

[89] Moving target defense. URL `https://www.dhs.gov/science-and-technology/csd-mtd`.

[90] Yuval Yarom and Katrina Falkner. Flush+ reload: A high resolution, low noise, l3 cache side-channel attack. In *USENIX Security Symposium*, pages 719–732, 2014.

[91] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Cost of virtual machine live migration in clouds: A performance evaluation. *CloudCom*, 9:254–265, 2009.

[92] Fangfei Liu, Yuval Yarom, Qian Ge, Gernot Heiser, and Ruby B Lee. Last-level cache side-channel attacks are practical. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 605–622. IEEE, 2015.

[93] David Blackwell. The noisy duel, one bullet each. Technical report, arbitrary accuracy. Technical report, The RAND Corporation, D-442, 1949.

[94] Tadeusz Radzik. Results and problems in games of timing. *Lecture Notes-Monograph Series*, pages 269–292, 1996.

[95] Kevin D Bowers, Marten Van Dijk, Robert Griffin, Ari Juels, Alina Oprea, Ronald L Rivest, and Nikos Triandopoulos. Defending against the unknown enemy: Applying flipit to system security. In *GameSec*, pages 248–263. Springer, 2012.

[96] Marten Van Dijk, Ari Juels, Alina Oprea, and Ronald L Rivest. Flipit: The game of "stealthy takeover". *Journal of Cryptology*, 26(4):655–713, 2013.

[97] Pengfei Hu, Hongxing Li, Hao Fu, Derya Cansever, and Prasant Mohapatra. Dynamic defense strategy against advanced persistent threat with insiders. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 747–755. IEEE, 2015.

[98] Ming Zhang, Zizhan Zheng, and Ness B Shroff. Stealthy attacks and observable defenses: A game theoretic model under strict resource constraints. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 813–817. IEEE, 2014.

[99] Jeffrey Pawlick, Sadegh Farhang, and Quanyan Zhu. Flip the cloud: cyber-physical signal-

ing games in the presence of advanced persistent threats. In *International Conference on Decision and Game Theory for Security*, pages 289–308. Springer, 2015.

[100] Sadegh Farhang and Jens Grossklags. Flipleakage: a game-theoretic approach to protect against stealthy attackers in the presence of information leakage. In *International Conference on Decision and Game Theory for Security*, pages 195–214. Springer, 2016.

[101] Sukhpal Singh and Inderveer Chana. A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of grid computing*, 14(2):217–264, 2016.

[102] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 199–212. ACM, 2009.

[103] Kuniyasu Suzaki, Kengo Iijima, Toshiki Yagi, and Cyrille Artho. Memory deduplication as a threat to the guest os. In *Proceedings of the Fourth European Workshop on System Security*, page 1. ACM, 2011.

[104] Rodney Owens and Weichao Wang. Non-interactive os fingerprinting through memory deduplication technique in virtual machines. In *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, pages 1–8. IEEE, 2011.

[105] Yinqian Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Cross-vm side channels and their use to extract private keys. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 305–316. ACM, 2012.

[106] Gorka Irazoqui, Mehmet Sinan Inci, Thomas Eisenbarth, and Berk Sunar. Wait a minute! a fast, cross-vm attack on aes. In *International Workshop on Recent Advances in Intrusion Detection*, pages 299–319. Springer, 2014.

[107] Bhanu C Vattikonda, Sambit Das, and Hovav Shacham. Eliminating fine grained timers in xen. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 41–46. ACM, 2011.

[108] Peng Li, Debin Gao, and Michael K Reiter. Stopwatch: a cloud architecture for timing channel mitigation. *ACM Transactions on Information and System Security (TISSEC)*, 17 (2):8, 2014.

[109] Himanshu Raj, Ripal Nathuji, Abhishek Singh, and Paul England. Resource management for isolation enhanced cloud services. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 77–84. ACM, 2009.

[110] Taesoo Kim, Marcus Peinado, and Gloria Mainar-Ruiz. Stealthmem: System-level protection against cache-based side channel attacks in the cloud. In *USENIX Security symposium*, pages 189–204, 2012.

[111] Yinqian Zhang and Michael K Reiter. Düppel: Retrofitting commodity operating systems to mitigate cache side channels in the cloud. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 827–838. ACM, 2013.

[112] Zhenghong Wang and Ruby B Lee. A novel cache architecture with enhanced performance and security. In *Microarchitecture, 2008. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, pages 83–93. IEEE, 2008.

[113] Fangfei Liu and Ruby B Lee. Random fill cache architecture. In *Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on*, pages 203–215. IEEE, 2014.

[114] Erman Pattuk, Murat Kantarcioglu, Zhiqiang Lin, and Huseyin Ulusoy. Preventing cryp-

tographic key leakage in cloud virtual machines. In *USENIX Security Symposium*, pages 703–718, 2014.

[115] Yinqian Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Cross-tenant side-channel attacks in paas clouds. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 990–1003. ACM, 2014.

[116] Soo-Jin Moon, Vyas Sekar, and Michael K Reiter. Nomad: Mitigating arbitrary cloud side channels via provider-assisted migration. In *Proceedings of the 22nd acm sigsac conference on computer and communications security*, pages 1595–1606. ACM, 2015.

[117] Vivek Shrivastava, Petros Zerfos, Kang-Won Lee, Hani Jamjoom, Yew-Huey Liu, and Suman Banerjee. Application-aware virtual machine migration in data centers. In *INFOCOM, 2011 Proceedings IEEE*, pages 66–70. IEEE, 2011.

[118] Tianwei Zhang, Yinqian Zhang, and Ruby B Lee. Cloudradar: A real-time side-channel attack detection system in clouds. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 118–140. Springer, 2016.

[119] Yinqian Zhang, Ari Juels, Alina Oprea, and Michael K Reiter. Homealone: Co-residency detection in the cloud via side-channel analysis. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 313–328. IEEE, 2011.

[120] Wen Qi, Jin Wang, Hermine Hovhannisyan, Kejie Lu, Jianping Wang, and Junda Zhu. A generic mitigation framework against cross-vm covert channels. In *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*, pages 1–10. IEEE, 2016.

[121] Qian Sun, Qingni Shen, Cong Li, and Zhonghai Wu. Selance: Secure load balancing of

virtual machines in cloud. In *Trustcom/BigDataSE/I SPA, 2016 IEEE*, pages 662–669. IEEE, 2016.

[122] Stefan Achleitner, Thomas La Porta, Patrick McDaniel, Srikanth V Krishnamurthy, Alexander Poylisher, and Constantin Serban. Stealth migration: Hiding virtual machines on the network. In *Infocom. IEEE*, 2017.

[123] Terry Penner and Mina Guirguis. Combating the bandits in the cloud: A moving target defense approach. In *Cluster, Cloud and Grid Computing (CCGRID), 2017 17th IEEE/ACM International Symposium on*, pages 411–420. IEEE, 2017.

[124] Yi Han, Tansu Alpcan, Jeffrey Chan, and Christopher Leckie. Security games for virtual machine allocation in cloud computing. In *International Conference on Decision and Game Theory for Security*, pages 99–118. Springer, 2013.

[125] Charles A Kamhoua, Luke Kwiat, Kevin A Kwiat, Joon S Park, Ming Zhao, and Manuel Rodriguez. Game theoretic modeling of security and interdependency in a public cloud. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, pages 514–521. IEEE, 2014.

[126] Luke Kwiat, Charles A Kamhoua, Kevin A Kwiat, Jian Tang, and Andrew Martin. Security-aware virtual machine allocation in the cloud: A game theoretic approach. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 556–563. IEEE, 2015.

[127] Charles Kamhoua, Andrew Martin, Deepak K Tosh, Kevin A Kwiat, Chad Heitzenrater, and Shamik Sengupta. Cyber-threats information sharing in cloud computing: A game theoretic approach. In *Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on*, pages 382–389. IEEE, 2015.

[128] Achintya Prakash and Michael P Wellman. Empirical game-theoretic analysis for moving

target defense. In *Proceedings of the Second ACM Workshop on Moving Target Defense*, pages 57–65. ACM, 2015.

[129] Aron Laszka, Benjamin Johnson, and Jens Grossklags. Mitigation of targeted and non-targeted covert attacks as a timing game. In *International Conference on Decision and Game Theory for Security*, pages 175–191. Springer, 2013.

[130] Benjamin Johnson, Aron Laszka, and Jens Grossklags. Games of timing for security in dynamic environments. In *International Conference on Decision and Game Theory for Security*, pages 57–73. Springer, 2015.

[131] Viet Pham and Carlos Cid. Are we compromised? modelling security assessment games. *Decision and Game Theory for Security*, pages 234–247, 2012.

[132] Xiaotao Feng, Zizhan Zheng, Pengfei Hu, Derya Cansever, and Prasant Mohapatra. Stealthy attacks meets insider threats: a three-player game model. In *Military Communications Conference, MILCOM 2015-2015 IEEE*, pages 25–30. IEEE, 2015.

[133] Haikun Liu, Cheng-Zhong Xu, Hai Jin, Jiayu Gong, and Xiaofei Liao. Performance and energy modeling for live migration of virtual machines. In *Proceedings of the 20th international symposium on High performance distributed computing*, pages 171–182. ACM, 2011.

[134] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 273–286. USENIX Association, 2005.

[135] Heinrich Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

[136] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[137] David Chisnall. *The definitive guide to the xen hypervisor*. Pearson Education, 2008.