GAME THEORETIC APPROACHES
FOR DESIGN OF INFORMATION CENTRIC NETWORKS (ICN) AND SPECTRUM SHARING

by

SEYEDMOHAMMAD HAJIMIRSADEGHI

A dissertation submitted to the

School of Graduates Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Narayan B. Mandayam

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2018

ABSTRACT OF THE DISSERTATION

Game Theoretic Approaches

For Design of Information Centric Networks (ICN) and Spectrum Sharing

By SeyedMohammad Hajimirsadeghi

Dissertation Director:

Prof. Narayan B. Mandayam

Future Internet designs call for increased security, performance reliability, social content distribution, mobility and distributed scalable resource allocation. The overarching goal of the research presented in this thesis is to use game theoretical approaches for the design of new networking paradigms for the future Internet in order to have better performance with respect to content distribution, security and resource management. The first part of the thesis studies information-centric networking (ICN) which is a new communication paradigm for future networks that replace the fixed-host/server model which has dominated today's Internet. ICN leverages in-network caching, multi-party communication through replication, and interaction models decoupling senders and receivers. We develop an analytical framework for distribution of popular content in an ICN that comprises of Access ICNs, Transit ICNs and Content Providers. Using a generalized Zipf distribution to model content popularity, we devise a game theoretic approach to jointly determine caching and pricing strategies in such an ICN.

Since the goal is to provide a network infrastructure that is better suited to content distribution and more resilient to disruptions and security attacks, the second part focuses on the security of content based networks and takes advantage of a classical problem in game theory called the Colonel Blotto game (CBG)–a multidimensional

strategic resource allocation game to study the defense strategies against Advanced Persistent Threat (APT) which applies multiple sophisticated methods to steal data from the target system. We model the interaction between an APT attacker and a cloud system defender in their allocation of the Central Processing Units (CPUs) over multiple storage devices using Colonel Blotto Game, which considers the competition of two players under given resource constraints over multiple battlefields. The Nash equilibria (NEs) of the CBG-based APT defense game are derived for the case of symmetric and asymmetric players with a different total number of CPUs to evaluate how the limited CPU resources, the size of storage devices and the number of storage devices impact the expected data protection level of the cloud storage system.

The increasing number of mobile users and services show the importance of edge wireless networks for connectivity and data transmission in future Internet. So, another major challenge for designing the future Internet are Radio Resource Management (RRM) and the task of allocating the scarce resources such as bandwidth in edge wireless networks. The static traditional approaches limit the usage and result in poor utilization and many spectrum holes. To overcome this problem and motivated by many real-world examples such as communication of mobile devices, localized Internet of Things (IoT) devices, or even autonomous vehicles, and aiming to capture the influence of spectral allocation in a competitive environment on the performance of communication devices, the third part of this thesis is devoted to study the problem of dynamic competitive spectrum allocation. We study the scenario of two network service providers (NSPs) which are trying to provide service for their regional users through spectrum bidding. We show that the dynamic process of competitive spectrum allocation can be described as a two-level game in which the upper level is modeled as an optimal control problem and the lower level is modeled using CBG. We adopt a dynamic non-cooperative repeated game as the decentralized approach for the NSPs to determine their optimal strategies for the next time slot. We also provide the optimal strategy and value function of the dynamic game using Dynamic Programming (DP).

Acknowledgements

First of all, I would like to express my deepest gratitude to my thesis adviser, Professor Narayan Mandayam, for his continuous encouragement, creative advice as well as the freedom he gave me during my graduate studies. I had this honor to complete my Ph.D. under his supervision. Working with him was always enjoyable for me because he would give me enough time to learn and explore different areas. I will consider myself always in his debt for believing in my abilities.

Furthermore, I would like to thank my thesis committee members: Professor Roy Yates, for kindly accepting to be among my committee members and for his constructive comments and all the recommendations letters he wrote for me. Professor Zoran Gajic, for his helpful comments and all of his support as the ECE graduate director during my Ph.D. at Rutgers. Many thanks to Professor Rasoul Etesami, not only as my committee member but also as a wonderful friend. Discussing with him was always beneficial for me, as I could learn so many new things from him.

I would also like to thank Prof. Waheed Bajwa, Prof. Predrag Spasojevic and Prof. Dario Pompili for being so generous to me with their time and friendship. I should mention late Prof. David Daut for supporting me during my hard time in my first year at Rutgers. May he rest in peace.

In addition, I would like to thank all WINLAB and ECE staff especially John Scafidi, Ivan Seskar, Elaine Conners, Rich Howard, Nory Martinez and for everything they did for me.

I would like to express my deepest thanks to my wonderful parents which I couldn't ask for better ones. They have sacrificed their rights by allowing me to study abroad to pursue my dreams. I would also like to thank my siblings: Naeemeh, Nayereh, and Mahdi.

I would also like to thank all my labmates and collaborators, especially Dr. Alex Reznik, Dr. Gokul Sridharan, Dr. Ahmed Alabdel Abass, Dr. Liang Xiao, Dr. Ratnesh Kumbhkar, Shreyasee Mukherjee, Dragoslav Stojadinovic, Parul Pandey and Minghui Min.

I am grateful for the financial support from the Department of Electrical and Computer Engineering, Rutgers University and from the National Science Foundation (NSF) which have provided me with the necessary resources to carry on research and finish this dissertation.

Last but not least, I would like to thank all my wonderful Iranian friends at Rutgers University, especially Parishad Karimi for always having lots of funny stories to tell, Mohsen Ghassemi for being a good and noisy roommate for 4 years, Parna Haghparast for being beside me in some of the hardest times in my life, Abolfazl Hajisami (Dash Kian) and Siamak Abbaslou (Dash Sia) which we started our journey together in Rutgers, Mahmoud Ramezani for being a loyal friend for 14 years, Masoud Janbaz, Zahra Shakeri, Alireza Zamani, Carolina Cocito, Mohsen Rajabpor and Arezoo Taghizadeh for being my second family here in Rutgers.

Dedication

To

My Parents, for their unconditional love and support

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Current Internet and future Internet challenges

Over the last few decades, the Internet has evolved from a small academic network to the most important commercial infrastructure. The success of the Internet has created high hopes and expectations for new applications and services [1]. Although the current Internet, as a ubiquitous and universal means for communication and computation, has been extraordinarily successful, due to diversity in types of users and applications, many technical and non-technical challenges have emerged, which the current Internet may not be able to support to a sufficient level [2]. Some of these problems and challenges could have not been foreseen when the first parts of the Internet were built since the Internet was a small scale network at the beginning but now it's a worldwide web service. According to [3], nowadays, almost 4 billion users are using the Internet. It is believed that the daily traffic of the Internet is more than 4 billion GB which is mostly related to the content generated by social media such as Youtube, Instagram, Facebook and etc as media files. To access this gigantic volume of content, the people use their mobile devices like tablets and smartphones which caused a shift from wired to mobile wireless devices for end-users.

However, the Internet today is very different from its original concept when the architecture and protocols were developed around the abstraction of communications between fixed end hosts [4]. Considering the fact that mobile wireless devices have outnumbered fixed end hosts, existing Internet protocols (e.g., TCP/IP) are not well-suited for mobile content services because they were designed under very different assumptions, both in terms of service requirements and technology constraints [5]. Moreover, the most predominant use of the Internet is centered on content creation, dissemination and delivery, and this trend will continue into the foreseeable future. So, while the basic client-server model has enabled a wide range of services and applications, it does not incorporate adequate mechanisms to support secure mobile content-oriented functionality. This high increase in demand for video content in the Internet and the need for new approaches to control this large volume of information have motivated the development of future Internet architectures based on named data objects (NDOs) instead of named hosts [6–15] for efficient delivery of media content to fixed and mobile users. Such architectural proposals are generally referred to as Information Centric Networking (ICN) which is a new communication paradigm to increase the efficiency of content delivery and also content availability [16–18]. In this new concept, the network infrastructure actively contributes to content caching and distribution and every ICN node can cache and serve the requested content. To fulfill that purpose, several architectures have been proposed for ICN to reflect current and future needs better than the existing Internet architecture [19–25]. To provide preferable services to the users in ICN, Internet service providers (ISPs) or access ICNs should be able to maintain the quality of service (QoS) by improving the response time for the file request. They need to cache the frequently requested or popular content locally and store them near the users in the network. To provide QoS, in-network caching is introduced to provide the network components with caching ability. Therefore every node actively contributes to content caching and operates as a potential source of content. This leads to the reduction in network congestion and user access latency and increases the throughput of the network by locally caching the more popular content [26–30].

Since each ICN requires cooperation in caching from other ICNs to provide a global high-performance network, it is necessary to have pricing policies to incentivize all the ICNs to contribute to the caching process [31]. Several works have been done to address the problem of the economics of service pricing in current Internet and interconnection networks [32–36]. Using contemporary pricing policies cannot incentivize the lower tier ISPs to cooperate in the future Internet architecture [37]; hence, it will be needed to have new models to provide them with monetary incentives to collaborate in caching and distributing content when content with different popularities are available in the network.

Another major concern is that the computer networks are vulnerable to cyber-attacks [38], in which an attacker (or multiple attackers) applies different types of methods such as distributed denial of service (DDoS) [39], hacking [40], malware [41] phishing [42] to disrupt the service or get access to information. In the future Internet, the situation will become even more complicated. Web-based applications will require access to the users data; therefore, security vulnerabilities in browsers could expose the users locally and remotely stored data to attacks [43]. As the networks, which store data and content (e.g. content centric networks and cloud system) are getting more popular, they get even more vulnerable since the attackers lunch more sophisticated attacks to steal the valuable information [44]. The advanced persistent threats (APTs) are an example of such attacks which aim to steal the data from storage systems by injecting of multiple malwares [45]. These attacks are difficult to detect and have caused privacy leakage and millions of dollars loss and need to be strategically defended [46].

The Internet has tremendously evolved in the last few years connecting billions of devices such as laptops, cell phones and any other things which have the ability of processing, communication and computing. Thus, the traditional Internet is turning into the smart future Internet, called the Internet of Things (IoT) [47]. The increasing number of mobile users and services and the demands for high-speed communications are in contrast to the scarce spectrum resources [48, 49]. So, another major challenge for designing the future Internet are Radio Resource Management (RRM) and the task of allocating the scarce resources such as bandwidth in edge wireless networks [50]. The traditional static approaches limit the usage and result in poor utilization and many spectrum holes [51]. To satisfy the growing demand of spectrum by better usage of the licensed and unlicensed band for real-world application such as communication of mobile devices, IoT devices, or even autonomous vehicles, the old policies of spectrum allocation should be getting replaced by dynamic [52] and opportunistic [53] spectrum allocation (DSA and OSA). DSA has primarily achieved using centralized coordination of spectrum based on the concept of "spectrum server", which is responsible for allocating radio system parameters and controlling access [54]. In some approaches [55], a regional "spectrum broker" provides centralized assignment of statistically multiplexed frequency resources across a given area which shows improved performance due to the globally coordinated allocation of radio parameters [56]. While the potential benefit of coordinated spectrum allocation can be significant, there are several intrinsic disadvantages to the centralized architecture for a large scale service of this nature. The central controller is difficult to scale considering the fact that managing spectrum for a large geographical area with the high volume of devices implies too many database entries. In addition, the central server can become a single point of failure for what is clearly a critical national resource. There are also somewhat fundamental economic arguments against centralizing spectrum allocation authority and associated algorithms with a single governmental or commercial entity [57], as this can lead to undue market power and insufficient local autonomy, along with less scope for innovation in technical and business models. In contrast, the Internet is an example of a fully distributed architecture which works very well without any central point of control. So, It is widely believed that a scalable solution for dynamic spectrum assignment can also be realized through decentralized architecture and competition among the network service providers (NSPs) [58]. Such an environment can be created by allowing users to choose their NSPs on an on-demand basis, without committing to any specified provider. NSPs are competing with one another to provide service to these users by competitively allocating the available radio resources. The term "users" has a broad meaning which can range from a single mobile user to a complete campus-wide WiFi network. Google's Project Fi [59] is an example of such an architecture, where a pool of mobile users with no dedicated cellular service provider opportunistically connect to the cellular service provider which offering them the best service.

Consequently, the increased reliability, availability and interoperability requirements of the new networked services on one hand, and the extremely high volumes of multimedia content on the other hand, challenge the today's Internet. Moreover, it has become extremely difficult to support the ever increasing demands for security, performance reliability, social content distribution, mobility, distributed scalable resource allocation and so on through traditional Internet, which call for potential new Internet paradigm and solutions to overcome these issues. Here is where the mathematics comes to help using tools like game theory which motivate our work in this thesis.

1.2 Game Theory approaches to design of future Internet

Game theoretic models began to be used in economic theory and political science in 50's and have been used widely in other social and behavioral sciences and engineering recently [60]. Game theory as an analytical tool aimed at modeling situations in which decision-makers have to make specific actions that have mutual, possibly conflicting, consequences [61–63]. Every defined game consists of some basic element:

- Players: decision makers in the game;
- Payoffs: expected rewards/punishment at the end of the game
- Actions: possible choices made by the player
- Strategies: specified plan of action for every player against other players

The basic assumption in game theory is that the players are rational, which means that they take into account their knowledge or expectations of other decision-makers' behavior and try to maximize their payoffs [64]. They also have common knowledge of rationality which means that everyone understands that everyone is rational. The players have the full knowledge of the game (payoffs and actions are observable and known by all) and only can communicate through the actions. These assumptions can be changed in special kinds of game. The players' rationality assumption has been challenged by altruistic behavior in some situation in nature, so the notion of bounded rationality introduced by Herbert A. Simon [65], which state that when players make decisions, their rationality is limited by the tractability of the decision problem, the cognitive limitations of their minds, and the time available to make the decision. But, many believe that in some cases (computer networks), most of the interactions can be captured using the concept of rationality, with the appropriate adjustment of the payoff function [61].

As the computer networks and mobile applications continuously evolve, the research area of networking is also changing [66]. With the optimization approaches, the strategy, allocation, or price choices can be defined independently of the reactions of other users or player, while, the game theory realizes the design of the large-scale system with lack of access to centralized information and subject to unexpected disturbances [67]. Since the future networks will rely on autonomous and distributed architectures to improve the efficiency and flexibility of mobile applications, the game theory provides the ideal framework to study the complex interactions among interdependent rational players for designing efficient and robust distributed algorithms [68].

During the past couple of years, game theory has been used to solve many problems in communication systems [66,68,69]. It has been used to propose new pricing strategies for Internet services [32, 33]. A lot of other issues in computer networks, especially wireless networks, has been modeled and analyzed using game theory such as, resource management [70–72], power control [73–76], flow and congestion control [77,78], network routing [79, 80], in-network caching [81], security [82–84] and etc.

As the communication devices getting more and more smart, the assumptions of game theoretic models become a still better match for future wireless networks. Game theory as a multi-agent decision theory models the rational players' behavior who are trying to maximize their utilities in the presence of other agents influentials. The cooperative and non-cooperative behaviors of the future networks' entities can be investigated using appropriate solution concepts in game theory [68, 69].

1.3 Organization of the Thesis

This rest of this dissertation is organized as follows.

Chapter 2 of the thesis introduces the reader to the fundamental issues at stake, discussing the architectural concept of Information Centric Networks and the power that lies behind it. Looking at the different aspect of this architecture presented over the years, we discuss how could this can meet the criteria for the design of the future Internet.

In **Chapter 3**, we develop an analytical framework for distribution of popular content in an Information Centric Network (ICN) that comprises of Access ICNs, a Transit ICN and a Content Provider. Using a generalized Zipf distribution to model content popularity, we devise a game theoretic approach to jointly determine caching and pricing strategies in such an ICN. Under the assumption that the caching cost of the access and transit ICNs is inversely proportional to popularity, we show that the Nash caching strategies in the ICN are 0-1 (all or nothing) strategies. Further, for the case of symmetric Access ICNs, we show that the Nash equilibrium is unique and the caching policy (0 or 1) is determined by a threshold on the popularity of the content (reflected by the Zipf probability metric), i.e., all content more popular than the threshold value is cached [85, 86].

Chapter 4 formulates the interactions between an APT attacker- which applies multiple sophisticated methods to stealthily attack targeted cyber systems- and a cloud system defender in their allocation of the Central Processing Units (CPUs) over multiple devices as a Colonel Blotto game (CBG). The Nash equilibria (NEs) of the CBG-based APT defense game are derived for the case with symmetric players and the case with asymmetric players each with a different total number of CPUs. The expected data protection level and the utility of the defender are provided for each game at the NE. An APT defense strategy based on the policy hill-climbing (PHC) algorithm is proposed for the defender to achieve the optimal CPU allocation distribution over the devices in the dynamic defense game without being aware of the APT attack model [87, 88].

Chapter 5 investigates a scenario where multiple network service providers (NSPs) compete to provide wireless connectivity to a set of users. The users could either be a single mobile device, a set of localized Internet-of-Things (IoT) devices, or even a campus-wide network requiring wireless backhaul. The NSPs compete with one another to provide wireless service to the users by strategically allocating the available bandwidth so as to maximize their total payoff. The NSPs present each user with an offer to provide wireless connectivity using a certain amount of bandwidth. Users then decide to connect to that NSP whose offered bandwidth maximizes their utility function. Under such an architecture, this chapter focuses on the optimal bandwidth allocation strategies for the NSPs [89].

Chapter 6 introduce a dynamic noncooperative repeated game as the decentralized approach for the NSPs to determine optimal strategies for NSPs over a finite time horizon. The problem of a dynamic bandwidth allocation game can be cast as a zero-sum dynamic game (ZSDG). Obtaining the optimal equilibrium strategies for the NSPs reduces to finding the saddle point strategies of such a ZSDG. We use a dynamic programming (DP) approach to find the optimal strategies over the horizon [90].

Chapter 7 summarizes our contributions as well as our observations and provides suggestions for future research directions that will push the state of the art in design of future Internet.

1.4 Summary of Specific Contributions of the Dissertation

This dissertation contains results from the following list of publications, which have addressed different parts of my research.

- M. Hajimirsadeghi, N. B. Mandayam, and A. Reznik, Joint caching and pricing strategies for popular content in information centric networks, IEEE Journal on Selected Areas in Communications, vol. 35, no. 3, pp. 654667, 2017. [86]

- M. Hajimirsadeghi, N. B. Mandayam, and A. Reznik, Joint caching and pricing strategies for information centric networks, in Global Communications Conference (GLOBECOM), 2015 IEEE. IEEE, 2015, pp. 16. [85]

- M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, Defense against advanced persistent threats in dynamic cloud storage: A colonel blotto game approach, IEEE Internet of Things Journal, 2018. [88]

- M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, Defense against advanced persistent threats: A colonel blotto game approach, in Communications (ICC), 2017 IEEE International Conference on. IEEE, 2017, pp.

16. [87]

- M. Hajimirsadeghi, G. Sridharan, W. Saad, and N. B. Mandayam, Inter-network dynamic spectrum allocation via a Colonel Blotto game, in Proc. IEEE Annu. Conf. Inf. Sci. Syst. (CISS), Princeton, NJ, Mar. 2016, pp. 252257. [89]

- M. Hajimirsaadeghi and N. B. Mandayam, A dynamic colonel blotto game model for spectrum sharing in wireless networks, in Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on. IEEE, 2017, pp. 287294. [90]

1.4.1 Other Contributions

Analysis and results from following list of publications have not been included in the dissertation, as they do not fit the scope of the topic.

- Pandey, Parul, Mohammad Hajimirsadeghi, and Dario Pompili. ”Region of feasibility of interference alignment in underwater sensor networks.” IEEE Journal of Oceanic Engineering 39.1 (2014): 189-202. [91]

- Ahmed A. Alabdel Abass, Mohammad Hajimirsadeghi, Narayan B. Mandayam, and Zoran Gajic. ”Evolutionary game theoretic analysis of distributed denial of service attacks in a wireless network.” In 2016 Annual Conference on Information Science and Systems (CISS), pp. 36-41. IEEE, 2016. [92]

- Mohammad Yousefvand, Mohammad Hajimirsadeghi, Narayan B. Mandayam. ”Impact of End-User Behavior on User/Network Association in HetNets” Accepted to IEEE ICC 2018, Kansas City, USA. [93]

Chapter 2

Information Centric Networks(ICN)

2.1 Overview and Motivation

The current Internet architecture was founded upon a host-centric communication model, which was developed around the abstraction of communication between fixed end host. Nowadays, however, the vast majority of Internet traffic requested by users are related to content access from the sources such as YouTube, Netflix, Amazon, Bit Torrent, Hulu, etc, estimated to be 80% of the Internet traffic by 2018. This high increase in demand for video content in the Internet and the need for new approaches to control this large volume of information, along with the pressing needs for better security and mobility support and more efficiently utilized infrastructure and simpler application [94] have motivated the development of future Internet architectures based on named data objects (NDOs)- where the users are just interested in information rather than its location or perhaps, even how it is delivered- instead of named hosts [7]. Such architectural proposals are generally referred to as information centric networking (ICN) which is a new promising communication paradigm for the architecture of the Future Internet to increase the efficiency of content delivery and also content availability [16]. ICN deploys in-network caching by naming the content at the network layer. It also can support multicast in order to facilitate the efficient and timely delivery of content to the users. However, ICN is not just the content distribution paradigm. It also addresses a series of other limitation in the current Internet such as mobility management and security enforcement to fulfill all the future Internet requirements [17].

2.2 Information Naming in ICN

Users are more interested in receiving information wherever it may be located, rather than accessing a specified server. However, the current Internet works as a host-centric network which requires the user to specify in each request both information and the location which it can be retrieved from. So, fetching the content from the optimal location would be challenging. The ICN approach fundamentally decouples content from its sources, by naming and addressing the information independently of its location, thus the content can be located anywhere in the network. This is the main abstraction of ICN called Named Data Objects (NDO) [95]. The NDO can be anything, basically, all types of objects that we store in and access via computers. The NDO is independent of location, storage method, an application program, and transportation method. So, every NDO has its own identity and every copy of NDO will be treated the same way, regardless of how it is copied or stored. By this definition, any node holding a copy of the NDO can provide the requester with that object. Some ICN designs use metadata associated with NDOs. For example for naming a piece of music, they use author, creation date or any other attribute. In ICN, when a NDO is being requested by a user, the network is locating the best source that can provide that piece of content. This type of design based on NDO, can provide security primitives, multi-path forwarding and in-network caching.

Since names are used for identifying objects independent of its location, for having a reliable network, the ICN requires having a unique name for each individual NDOs. This unique binding between the object and its name assure the network elements and receiver about the content originality, otherwise, no one can trust the object authenticity, which would endanger the network reliability by enabling denial of service attack using injecting spoofed content into the network.

Three naming schemes in ICN [96] have been proposed in order to ensure above functionality:

- **Hierarchical naming:** This is similar to current DNS and correlates to underlying network topologies. It enables aggregation of routing information and improving the scalability of the routing system.

- **Flat naming:** This is a self-certified scheme, meaning that the objects name-data integrity can be verified without needing a public key infrastructure (PKI) or another third party to first establish trust in the key and usually is done by hashing [16].

- **Attribute-based naming:** This uses the object attributes for naming and is more expressive and richer in semantic structures. It can be combined with previous two naming schemes.

2.3 Mobility in ICN

As the traffic from wireless terminals exceeds the traffic from the fixed host, the IP address won't be fit for the future Internet. It can not support continuous connectivity properly while on the move, which is becoming an increasingly important requirement. Mobile IP [97] which was proposed to alleviate this problem is also inefficient since traffic has to travel along a path longer than the optimal. However, since in ICN architecture, there is no host-based connection, the mobility is easier to be addressed. The mobile users just request for the NDO on every new access and that access point are responsible for finding the optimal source which has that NDO, thus there is no need to maintain the old connection with the previous source. The proposed improvement in mobility support opens up many potential benefits such as Host Multihoming, Network Address Consistency, Removal of Connection-Oriented Sessions, Scoping of Content and Location and Resilience through Replication [98]. These properties allow for the efficient support of mobility and mobile nodes can simply reissue subscriptions for the content after handoffs and the nearest caches, who stored that content, rather than the original publisher will provide the service.

2.4 Name resolution and Routing in ICN

In ICN, when a request is made for a specific NDO, the first thing is to route the request to the closest node which has stored a copy of that NDO and deliver the request for NDO to that node. The second function is to route the content to the requester by finding a path for delivering the NDO. To do this, there are two ways of name resolution and name-based routing. In name resolution, a resolution service is queried and an information name is getting matched to a provider or source that can supply that information. After that one or more low-layer locators are returned and using them the NDO will be retrieved using HTTP or direct IP protocol. In name-based routing, the request will directly be forwarded to a copy of that NDO in the network based on its name, without first resolving the object name into some lower-layer locators.

2.5 In-network caching in ICN

The massive requests for popular content at a specific time and space make a huge traffic on the network which needs to be managed properly. Primary Internet which has a data-agnostic architecture fail to address this issue to enable efficient content delivery. Content Delivery Networks (CDNs) which is a critical component of any modern web application is been used to improve the delivery of content by replicating commonly requested files across a globally distributed set of caching servers. The CDNs typically employ network-unaware mechanisms and lack of unique identification of identical objects makes it hard to take advantage of caching, which lead to inefficient utilization of the underlying network resources.

In ICN architecture, in-network caching is a non-separable part of ICN service [26, 99]. The ICN provide transparent, ubiquitous in-network caching to alleviate the rapid traffic growth and speed up content distribution and improve network resource utilization. To do this, all of the network components including routers, intermediate nodes and mobile terminals have an inner cache which is able to store the most requested content by the geographically nearby users. Thus, not only the original source can provide the content, but also all the caches in the network which hold a copy of the content are able to provide users with their desired information. The caching in ICN is application-independent and applies to all providers.

ICN aims to provide the cache transparency to provide a shared cache infrastructure so that each application can manage the cache space independently and making routing and caching decisions on unified content name. It also provides cache ubiquity and fine granularity which support arbitrary topology and different options of granularity such as file-level and chunk level, respectively.

2.6 Security in ICN

In today Internet which is based on the TCP/IP protocol, the client/server communication channel is protected using Transport Layer Security (TLS) or a similar technique. This security model requires the client to trust the server to deliver correct information over the channel. However, the design of Internet let any traffic to be injected into the network. This characteristic results in many different types of attacks such as denial of service attacks against the Internet infrastructure or against Internet hosts and services. ICN architecture is, in contrast, interest-driven, it means that no unwanted traffic will be injected into the network. This helps to reduce the amount of unwanted and harmful traffic. Using this, forensic mechanisms can be deployed on the network points that handle availability and interest signaling. Moreover, ICN architectures provide self-certifying name-data integrity and origin verification of NDOs, independent of the immediate source, which helps in filtering malicious data. The DoS attack can also be somehow prevented by decoupling the communication between the two parties which one of them own the content and the other one requesting it. It can also help in making a private environment for the users, as a publisher does not need to know the identity of the content requesters. In ICN, securing the content itself is much more important than securing the infrastructure or the endpoints. Having an understanding about the ICN attacks can be the doorway into the protection of the network and users information. [44] has provided a survey of attacks unique to ICN architectures and other generic attacks that have an impact on ICN.

2.7 ICN well-known designs

Various existing proposed ICN designs are future Internet architecture candidates. Although they are still under active development, they are trying to address some key functionalities such as naming, caching, mobility, security and name resolution and routing. This section briefly reviews some of the well-known designs for ICN.

2.7.1 Named Data Networking (NDN)

The NDN [7,100] project advances the CCN approach. It provides a topology-independent naming scheme and is exploring greedy routing for better router routing scalability. Whenever a user needs a piece of content, it broadcasts the request packets containing the desired content name, and the routing protocols are employed to distribute information about the location of the content based on the name. The routing is based on a hierarchical naming scheme. The content provider, or any other network node with a copy of the requested content, route the required content, along with additional authentication and data-integrity information, along the interests reverse route. In NDN, the security is provided by encrypting each data packet using publisher's signature. Every NDN node is provided by a cache which is able to store the content for the future request. Consumer mobility in NDN is intrinsic due to its consumer-driven nature. When a consumer relocates, it can re-issue any previously sent interest packets that have not been satisfied yet.

2.7.2 Data Oriented Network Architecture (DONA)

DONAs architecture [6] redesign the current Internet naming. DNS names are replaced with flat, self-certifying names, and DNS name resolution is replaced with any cast name resolution process. The architecture provides improved data retrieval as well as improved service by providing persistence, authentication, and availability. Routing in DONA produces overhead for each request, which makes the network unable to use packet-size objects, unlike NDN. DONA supports on-path caching via the resolution handler (HR) infrastructure. Mobile users can simply issue new FIND messages from their current location, relying on the RH infrastructure to provide them with the closest copy of the information. Names in DONA are self-certifying which removes the necessity for public key infrastructures (PKIs).

2.7.3 Publish-Subscribe Internet Technology (PURSUIT)

PURSUIT [101] have produced an architecture that completely replaces the IP protocol stack with a publish-subscribe protocol stack. The PURSUIT architecture consists of three separate functions: rendezvous, topology management and forwarding. A rendezvous system helps in locating the scope and publications in the network. It directs the topology management function to create a route between the publisher and the subscriber. This route is finally used by the forwarding function to perform the actual transfer of data. PURSUIT uses a flat namespace with two types of names, called, the rendezvous identifier (RI) and the scope identifier (SI). These identifiers together establish the name of the content. PURSUIT uses self-certifying names, which alleviate the need for a public key infrastructure (PKI); therefore, nodes can easily check the name-data integrity based on the received datas name. It supports both on-path and off-path caching. Mobility is also been supported by this architecture. When a consumer changes its location, it re-subscribes to the content using another source by computing a new forwarding indicator (FI) for the hosts new location. Clearly, the efficiency of consumer mobility is therefore dependent on the speed at which new FIs can be generated.

2.7.4 MobilityFirst

The MobilityFirst project [8, 11], proposes a clean-slate Future Internet architecture with an emphasis on treating mobile devices as first-class citizens. It provides detailed mechanisms to handle mobility, multicast, multi-homing, in-network caching and security. It separates the names of all entities attached to the network from their network address. Thus, every entity will have global name which can be used throughout the network and can be translated into network addresses at various points in the network. This specification allows the messages to be dynamically redirected. On-path caching is being supported in this architecture by caching the passing messages at intermediate local content routers in an opportunistic manner.

Chapter 3

Joint caching and pricing Strategies for Popular Content in ICN

3.1 Overview and motivation

The vast majority of Internet traffic relates to content access from the sources such as YouTube, Netflix, Bit Torrent, Hulu, etc. This rapid increase of content delivery on the Internet has revealed the need for a different networking paradigm. Further, as Fig. 3.1 describes [85], the emerging trend is that the users are just interested in the information (content), and not where it is located or perhaps, even how it is delivered. This high increase in demand for video content on the Internet and the need for new approaches to control this large volume of information have motivated the development of future Internet architectures based on named data objects (NDOs) instead of named hosts [7]. Such architectural proposals are generally referred to as Information Centric Networking (ICN) which is a new communication paradigm to increase the efficiency of content delivery and also content availability [16–18] of future fifth generation (5G) networks.

In this new concept, the network infrastructure actively contributes to content caching and distribution and every ICN node can cache and serve the requested content. To fulfill that purpose, several architectures have been proposed for ICN to reflect current and future needs better than the existing Internet architecture [19–25]. To provide preferable services to the users in ICN, Internet service providers (ISPs) or access ICNs should be able to maintain quality of service (QoS) by improving the response time for file request. They need to cache the frequently requested or popular content locally and store them near the users in the network. To provide QoS, in-network caching is introduced to provide the network components with caching ability. Therefore every node

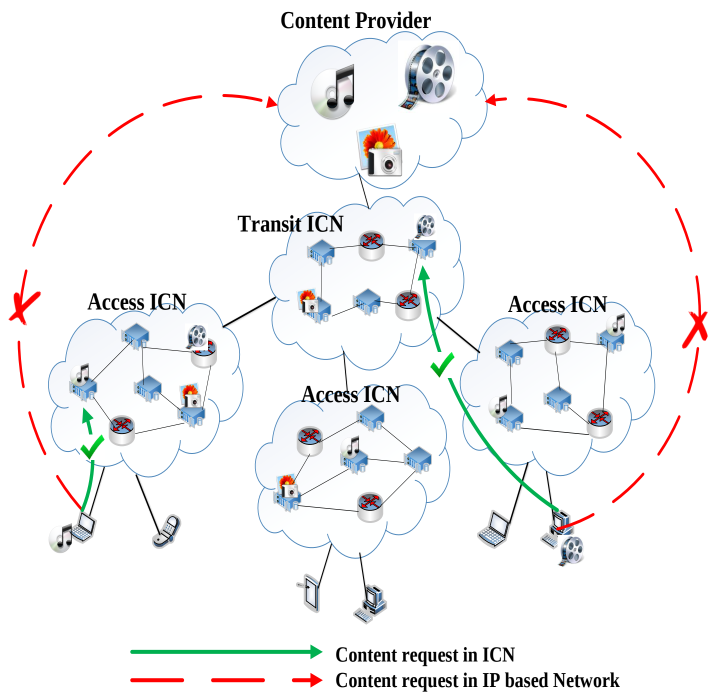


Figure 3.1: ICN communication model: unlike the current Internet which the users are interested in the location of the files, the ICN users just look for the content regardless of file location.

actively contributes on content caching and operates as a potential source of content.

This leads to the reduction in network congestion and user access latency and increases the throughput of the network by locally caching the more popular content [26–30]. Several works have claimed that web (file) requests in Internet are distributed according to Zipfs law [102–107]. Zipfs law states that the relative probability of a request for the ith most popular content is proportional to $\frac{1}{i}$. However, several other studies have found out that the request distribution generally follows generalized Zipf distribution where the request rate for the ith most popular content is proportional to $\frac{1}{i^\gamma}$ and γ is a positive value less than unity [108–110].

Since each ICN requires cooperation in caching from other ICNs to provide a global high-performance network, it is necessary to have pricing policies to incentivize all the ICNs to contribute to the caching process [110]. Several works have been done to address the problem of the economics of service pricing in current Internet and interconnection networks [32–36]. Using contemporary pricing policies cannot incentivize the lower tier ISPs to cooperate in the future Internet architecture [37]; hence, we need to have new models to provide them with monetary incentives to collaborate in caching and distributing content when content with different popularities are available in the network.

3.2 Related Work

The benefits of in-network caching have been investigated before in the setting of distributed file systems in several recent works. In [29], the problem of caching is studied from an information-theoretic viewpoint. They propose a coded caching approach that in addition to the local caching gain is able to achieve a global caching gain. A novel Cooperative Hierarchical Caching (CHC) framework is proposed in [111–113] in the context of Cloud Radio Access Networks (C-RAN). In [114], a collaborative joint caching and processing strategy for on-demand video streaming in mobile-edge computing network is envisioned.Content caching and delivery in device-to-device (D2D) networks have been studied in [115]. The aim of this work is to improve the performance of content distribution by the use of caching and content reuse. Several approaches such as base station assisted D2D network and other schemes based on caching at the user device are compared to show the improvement of the network throughput in the presence of in-network caching. Another recent article [116] studies the limitation of current reactive networks and proposes a novel proactive networking paradigm where caching plays a crucial role. It shows that peak data traffic demands can be substantially reduced by proactively serving users' demands via caching. [117] has used a mean field game model to study distributed caching in ultra dense small cell networks. Zhang et al have proposed an optimal cache placement strategy based on content popularity in content centric networks (CCN) in [118]. The authors in [119] have proposed a collaborative caching and forwarding design for CCN. The problem of joint caching and pricing for data service for a single Internet service provider (ISP) is studied in [120]. Similar problem but for multiple ISPs in the setting of small cell networks is investigated in [121] using a Stackelberg game. [122] proposes an incentive proactive cache mechanism in cache-enabled small cell networks (SCNs) in order to motivate the content providers to participate in the caching procedure.

One of the earliest studies of economic incentives in ICNs has been conducted by Rajahalme et al. [37] and has demonstrated that top level providers are not willing to cooperate in the caching process since they cannot get enough revenue. Another recent study by Agyapong et al. [123] has addressed the economic incentive problem in ICN by building a simple economic model to evaluate the incentive of different types of network players in a hierarchical network infrastructure. They qualitatively showed that without explicit monetary compensation from publishers, the network will fail to deploy the socially optimal number of caches. Few prior works have used game theoretic approaches to solve the problem of caching and pricing in ICN. In [124] the authors have presented a game theoretic approach using matrix payoff to analyze the process of economic incentives sharing among the major network components. A pricing model was proposed in [125] to study the economic incentives for caching and sharing content in ICNs which consists of access ICNs, a transit ICN and a content provider. This work has shown that if each player's caching (pricing) strategy remains fixed, the utility of each player becomes a concave function of its own pricing (caching) strategy. Therefore a unique Nash Equilibrium exists in a non-cooperative pricing (caching) game among different players.

3.3 System Model

In this section, we investigate joint caching and pricing strategies of the access ICNs, the transit ICN and the content provider based on content popularity. We study Nash strategies for a non-cooperative game among the above entities using a probabilistic model by assuming that access requests generally follow the generalized Zipf distribution. We then use the insights gained to simplify the problem by replacing two caching threshold indices instead of caching parameters for the symmetric case; where all access ICNs have the same parameters. In our model, the ICNs caching costs vary in respect to content popularity while the content provider cost per unit data is fixed for all content types.

3.3.1 Content Popularity

There are M different types of content in the network that each user is trying to access. Each type of content has a different measure of popularity reflected by the probability of requests for it. We consider a model where the popularity of content is uniformly similar in all parts of the network, i.e., all users in the network have the same file popularity distribution. Analyzing the impact of different per-user file popularities is an open problem. As in previous works (e.g. [28–30, 102, 107–109, 115–119]), in this chapter the distribution of user requests for content is described by a generalized Zipf distribution function as follows:

$$q_M(m) = \frac{\Omega}{m^\gamma}, m = 1, ..., M, \tag{3.1}$$

where $\Omega = \left(\sum_{i=1}^{M} \frac{1}{i^\gamma}\right)^{-1}$ and $0 \leq \gamma \leq 1$ is the exponent characterizing the Zipf distribution in which $\gamma = 0$ makes the distribution uniform and all the content will be identical in popularity, whereas the case of $\gamma = 1$ corresponds to one where the content popularity distribution is following the classic Zipf's law and more popular content is dominant in the network. The content is ranked in order of their popularity where content m is the mth most popular content, i.e., $m = 1$ is the most popular content and $m = M$ is the least popular content.

In most of the aforementioned works that have studied in-network caching, the popularity profile of content was assumed to be identical and perfectly known by all the network components. In reality, the demand and popularity are not predictable and certain [126, 127]. The problem of caching has been studied in [128] wherein the users have access to demand history but no knowledge about popularity. Several other papers have used learning-based approaches to estimate the popularity profile at the user side [129–134]. While content learning is more accurate for modeling content popularity, the reason we have used the Zipf model is due to (1) experimental results showing reasonable fit to the Zipf model and (2) analytical tractability provided by the Zipf model. Our framework can be extended by changing the demand model and considering a repeated game with a parametric Zipf distribution. In each time slot of the game, this parameter can be estimated in an optimal way using a learning process.

3.3.2 Cost Model

Although the prices are fixed for all types of content, since the ICNs want to earn more profit by caching the content, they are more willing to locally store the content which is more popular. Thus, the access ICNs' and the transit ICN's caches treat the content differently in regard to their popularity. As the content gets more popular, the ICNs incur less caching costs to locally store the content.

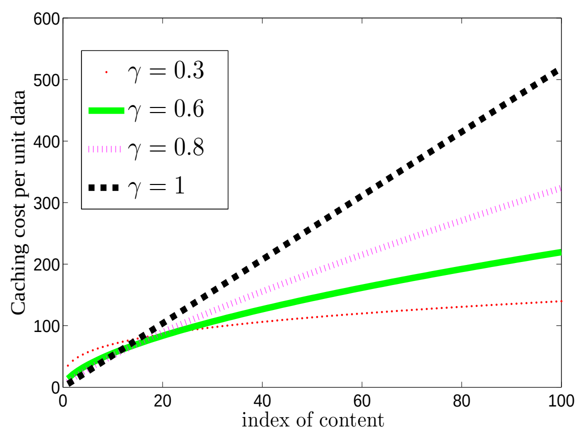


Figure 3.2: Access ICN caching costs vs index of content for different Zipfs factor γ.

Definition 3.1. *For a finite cache, the caching costs of access and transit ICNs is defined to be inversely proportional to the content popularity as follows*

$$c_{x_i} = \frac{c_{x0}}{q_M(i)}, \tag{3.2}$$

where, without loss of generality, c_{x_0} is a fixed initial caching cost at ICN x.

Using equations (3.1) and (3.2), we see in Fig. 3.2, for fixed values of i, c_{x_i} is a decreasing function of γ when i is small (i.e., more popular content). On the other hand, c_{x_i} is an increasing function of γ when i is large (i.e., less popular content). Unlike the

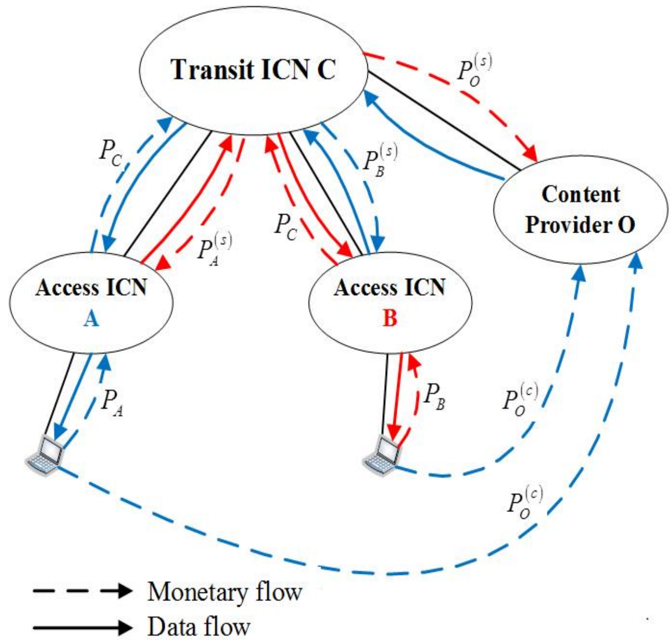


Figure 3.3: Interaction between different entities in a simplified model of an ICN.

access and transit ICNs, the content provider has no priority for caching the content and caches all types of content. The content provider incurs the constant cost c_O for every unit of data that it serves for the transit ICN.

3.3.3 Access and Demand Model

For simplicity in illustration, as shown in Fig. 3.3 [85], we begin with a hierarchical network model [125, 135] with two access ICNs (A and B) one transit ICN (C), one content provider (O) and an arbitrary number of users who can switch from one access ICN to another. The access ICNs connect the end-users to the content network and the transit ICN provides wide-area transport for the access ICNs while the content provider provides the content for the users. Fig. 3.3 also shows the monetary and data flows among different entities with the various prices described in Table 3.1. The network economy depends on two effective factors: caching and pricing. Under the assumption that each ICN can have access to all content, it can decide to either cache the entire or portion of the requested content, or get it from somewhere else. The caching strategy adopted by each entity is denoted by the parameter α that takes values in the interval $[0, 1]$. Every ICN decides to cache different types of content independently, therefore, we have a specific caching variable for each type of content. We have denoted α_{I,S_i}, where I can either be access or transit ICN and S is any cache in the network (possibly another ICN or content provider), as the fraction of ICN I's demand for content type i that comes from cache S. Each ICN also has different pricing strategies. These strategies are the prices that each player charges others for the provided service. The pricing is based on the usage, i.e., price per unit data. Each access ICN sets two different prices: (1) the network price per unit data for transporting the content; and (2) the storage price per unit data for providing content from its cache for other ICNs. For example, the network and storage prices for access ICN A are denoted as $P_A^{(n)}$ and $P_A^{(s)}$, respectively. The total price per unit data is the sum of these two prices and is denoted by $P_A = P_A^{(n)} + P_A^{(s)}$. We will find it useful to utilize an alternative form of the above. Following the charging policies of several storage services for the Internet such as Amazon S3, it would be practical to assume that the storage price is typically less than the network price. The linear relationship between network price and storage price while empirical, has been used earlier in [125] and we follow this assumption. It can be represented by $P_A^{(n)} = \beta_A P_A^{(s)}$ where $\beta_A > 1$. Thus, the relationship between P_A and $P_A^{(s)}$ would be $P_A^{(s)} = \left(\frac{1}{1+\beta_A}\right) P_A$. As a result, each access ICN or transit ICN will have a set of strategies for pricing in the interval $[0, \infty)$. The content provider pricing strategies set also consists of the content price $P_O^{(c)}$ that the users should pay for content and the storage price $P_O^{(s)}$ which is the price for providing the content from the content provider cache. The access ICN A and B charge prices P_A and P_B to their users and $P_A^{(s)}$ and $P_A^{(s)}$ to the transit network if they store or forward the content that the transit ICN had asked for. The transit ICN C charges access ICNs A and B with price P_C, if it stores or forwards their requested content. The content provider charges users with content price $P_O^{(c)}$ and transit ICN C with storage price $P_O^{(s)}$ if it stores transit ICN requested content.

To model the behavior of users, we have considered content demand at each access ICN to be a linear function of the prices.

Definition 3.2. *The users' demands are affected by both content price and access price and defined to be a linear function as follows:*

$$
\begin{aligned}
\sigma_A &= 1 - \rho_A P_A + \rho_B P_B - \rho_0 P_o^{(c)} \\
\sigma_B &= 1 + \rho_A P_A - \rho_B P_B - \rho_0 P_o^{(c)}
\end{aligned}
\tag{3.3}
$$

where ρ_A, ρ_B and ρ_0 are the reflective coefficients of the prices' influence on users' demands and are positive. The demands are normalized per unit data. We observe that the users' demands directly depend on the access ICNs' prices and the content price. For example, as the content price $P_O^{(c)}$ increases, the users demands decrease. If one of the access ICNs increases its price, the users will switch to the other ICN. Table 3.1 summarizes our notation.

As illustrated in Fig. 3.3, the interaction between different entities in the ICN model results in a conflict among the ICNs (players) when they unilaterally try to maximize their revenue. In the following section, we use a game theoretic approach to solve the joint caching and pricing strategies for each entity in the ICN network.

3.4 Joint Caching and Pricing Strategy

3.4.1 Utility Function

Each ICN can cache the content or just forward the requests to other ICNs or content provider based on the utility that it gains. The utility function for each player is defined as the utility received by providing the services for others. Each player incurs a caching cost with respect to the popularity of the content when it stores a unit of data. Therefore, as shown in Fig. 3.3, the utility functions for the access ICNs A and B, the transit ICN C and the content provider O can be formulated as an opportunistic function in terms of the prices, caching costs, demands and fraction of content stored and popularity.

The access ICN A incurs a caching cost of c_{A_i} for content type i if it decides to store a unit of data. Therefore, the utility function of the access ICN A is given as the

Table 3.1: Summary of notation for joint caching and pricing problem

Notation	Description
K	Number of access ICNs
$P_k^{(n)}$	Network price of ICN k per unit data
$P_k^{(s)}$	Storage price of access ICN or content provider k per unit data
P_C	Transit price charge by transit ICN C
P_k	Access ICN k's price to its users per unit data
$P_O^{(c)}$	Content price of content provider O per unit data
ρ_k	Reflective coefficient of prices influence on access ICN k's user demand
σ_k	Normalized total user demands for ICN k per unit data
σ_{k_i}	Normalized total user demands for ICN k for content type i per unit data
α_{I,S_i}	Fraction of ICN I's demand for content type i that comes from cache S
c_{k_i}	Caching cost of ICN k for content type i
c_{k_0}	Initial caching cost of ICN k
c_O	Content provider O cost
β_k	Scaling parameter between network price and storage price (greater than unity)
γ	Zipf popularity law exponent
Th	Caching threshold index for access ICNs
Th_C	Caching threshold index for transit ICN C
$q_M\left(m\right)$	Popularity (request rate) for content type m over a set of M content

average utility for all different types of content as follows:

$$U_A = \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \sigma_A \alpha_{A,A_i} \left(P_A - c_{A_i}\right) + \\ \sigma_A \alpha_{A,Out_i} \left(P_A - P_C\right) + \\ \sigma_B \alpha_{B,A_i} \left(\left(\frac{1}{1+\beta_A}\right) P_A - c_{A_i}\right) \end{array} \right\}, \tag{3.4}$$

where the first term in (3.4) is the utility that results when ICN A stores a portion of its users demands in its own cache. $q_M(i)$ denotes the popularity (request rate) of content type i over a set of M different types of content, σ_A is the total demand that users have requested to the access ICN A and α_{A,A_i} is the fraction of ICN A's demand for content type i that is going to be served by ICN A's cache. Therefore, σ_A multiplied by $q_M(i)$ and α_{A,A_i} is the demand of access ICN A for content type i that is served by transit ICN A's cache and $(P_A - c_{A_i})$ is the revenue of access ICN A by serving this portion of the requested demand. The second term is the utility that results when ICN A forwards a portion of its users demand to the transit ICN C. The third term is the utility that results when the transit ICN C forwards a portion of ICN B's users' demand to ICN A. The access ICN A can control only the caching and pricing parameters α_{A,A_i}, P_A and $P_A^{(s)}$. Note that $\alpha_{A,Out_i} = 1 - \alpha_{A,A_i}$ or $\alpha_{A,Out_i} = \alpha_{A,B_i} + \alpha_{A,C_i} + \alpha_{A,O_i}$. The utility function of the access ICN B can be defined in a similar way as follows:

$$U_B = \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \sigma_B \alpha_{B,B_i} \left(P_B - c_{B_i}\right) + \\ \sigma_B \alpha_{B,Out_i} \left(P_B - P_C\right) + \\ \sigma_A \alpha_{A,B_i} \left(\left(\frac{1}{1+\beta_B} \right) P_B - c_{B_i} \right) \end{array} \right\}. \tag{3.5}$$

The access ICN B can control only the caching and pricing parameters α_{B,B_i}, P_B and $P_B^{(s)}$. Note that $\alpha_{B,Out_i} = 1 - \alpha_{B,B_i}$ or $\alpha_{B,Out_i} = \alpha_{B,A_i} + \alpha_{B,C_i} + \alpha_{B,O_i}$. The transit ICN C gains a profit if the access ICNs A and B request a content through it. Equation (3.6) consists of two terms. The first term is the utility that results when transit ICN C stores ICN A and B's content in its own cache and the second term is the utility that results when it forwards ICN A and B's content to other ICN's or content provider. We model the ICN C's utility in the following way:

$$\begin{array}{c} U_C = \sum\limits_{X \in \{A,B\}} \sum\limits_{i=1}^{M} q_M(i) \sigma_X \alpha_{X,C_i} (P_C - c_{C_i}) + \\ \sum\limits_{X \in \{A,B\}} \sum\limits_{L \in \{A,B,O\}, L \neq X} \sum\limits_{i=1}^{L} q_M(i) \sigma_X \alpha_{X,L_i} (P_C - P_L^{(s)}) \end{array}, \tag{3.6}$$

where the transit ICN C has control over caching variables α_{A,B_i}, α_{A,C_i}, α_{A,O_i}, α_{B,A_i}, α_{B,C_i}, α_{B,O_i} and P_C. Finally, if the requests are not served by any access ICNs or transit ICN, they will be forwarded to the content provider that has all the content in its cache and the costs for all types of content are identical and equal to c_O. The content provider's utility function can be expressed formally as

$$U_O = \sum_{i=1}^{M} q_M(i) \left[\sigma_A \alpha_{A,O_i} + \sigma_B \alpha_{B,O_i}\right] \left(P_O^{(s)} - c_O\right) + \left(\sigma_A + \sigma_B\right) P_O^{(c)}. \tag{3.7}$$

The first term in (3.7) results when the content provider O charges transit ICN C with storage price to deliver the requested content to it. The second term comes from the content price that content provider charges the users. The content provider can control pricing parameters $P_O^{(c)}$ and $P_O^{(s)}$.

Because of the competitive nature of this problem, we can present a solution in the analytical setting of a game theoretic framework. Let $G = [N, \{S_j\}, \{U_j(.)\}]$ denote the non-cooperative game among players from the set $N = \{A, B, C, O\}$, where $S_j = (P_j, \alpha_j)$ is the set of joint caching (α_j) and pricing (P_j) strategies and $U_j(.)$ is the utility function of player j. The strategy space of all the entities excluding the jth player is denoted by S_{-j}. In the joint caching and pricing game, each player tries to maximize its own utility by solving the following optimization problem for all $j \in N$,

$$\underset{s_j \in S_j}{max}\, U_j\left(s_j, S_{-j}\right). \tag{3.8}$$

It is necessary to characterize a set of caching and pricing strategies where all the players are satisfied with the utility they receive, given the strategy selection of other players. Such an operating point, if it exists, is called an equilibrium. The notion that is most widely used for game theoretic problems is the *Nash Equilibrium (NE)* [136]. A set of pricing and caching strategies $S_j^* = \left(P_j^*, \alpha_j^*\right)$ constitutes a *NE* if for every $j \in N$, $U_j\left(s_j^*, S_{-j}\right) \geq U_j\left(s_j, S_{-j}\right)$ for all $s_j \in S_j$. The *NE* of the game is one where no player benefits by deviating from her strategy unilaterally.

3.4.2 Characterization of Nash Strategies

To find the *NE* using the best response functions we need to solve the four following optimization problems for ICN A, ICN B, ICN C and the content provider O, respectively.

The maximization problem for ICN A is:

$$\max_{\alpha_{A,A_i},P_A} U_A = \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \sigma_A \alpha_{A,A_i} \left(P_C - c_{A_i}\right) + \\ \sigma_A \left(P_A - P_C\right) + \\ \sigma_B \alpha_{B,A_i} \left(\left(\frac{1}{1+\beta_A}\right) P_A - c_{A_i}\right) \end{array} \right\}, \tag{3.9}$$
$$s.t. \quad 0 \leq \alpha_{A,A_i} \leq 1 \ , P_A > 0$$

where the access ICN A tries to maximize her utility by changing its caching (α_{A,A_i}) and pricing (P_A) strategies while other players' strategies are unknown to her. Similarly, the maximization problem for ICN B is:

$$\max_{\alpha_{B,B_i},P_B} U_B = \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \sigma_B \alpha_{B,B_i} \left(P_C - c_{B_i}\right) + \\ \sigma_B \left(P_B - P_C\right) + \\ \sigma_A \alpha_{A,B_i} \left(\left(\frac{1}{1+\beta_B}\right) P_B - c_{B_i}\right) \end{array} \right\}. \tag{3.10}$$
$$s.t. \quad 0 \leq \alpha_{B,B_i} \leq 1 \ , P_B > 0$$

The content provider O maximizes its utility using equation (3.11)

$$\begin{aligned} \max_{P_O^{(c)},P_O^{(s)}} U_O &= \left(\sigma_A + \sigma_B\right) P_O^{(c)} \\ &+ \sum_{i=1}^{M} q_M(i) \left[\sigma_A \alpha_{A,O_i} + \sigma_B \alpha_{B,O_i}\right] \left(P_O^{(s)} - c_O\right), \\ s.t. \quad & P_O^{(C)} > 0, P_O^{(s)} > c_O \end{aligned} \tag{3.11}$$

and the transit ICN C maximization problem is given as the following equation:

$$\begin{aligned} &\max_{\alpha_{A,C_i},\alpha_{B,C_i},\alpha_{A,B_i},\alpha_{A,O_i},\alpha_{B,A_i},\alpha_{B,O_i},P_C} U_C = \sum_{i=1}^{M} q_M(i)\sigma_A \\ &\times \left\{ \begin{array}{l} \alpha_{A,C_i} \left(P_C - c_{C_i}\right) + \\ \alpha_{A,B_i} \left(P_C - P_B^{(s)}\right) + \alpha_{A,O_i} \left(P_C - P_O^{(s)}\right) \end{array} \right\} + \\ &\sum_{i=1}^{M} q_M(i)\sigma_B \left\{ \begin{array}{l} \alpha_{B,C_i} \left(P_C - c_{C_i}\right) + \\ \alpha_{B,A_i} \left(P_C - P_A^{(s)}\right) + \alpha_{B,O_i} \left(P_C - P_O^{(s)}\right) \end{array} \right\}. \end{aligned} \tag{3.12}$$

$$s.t. \quad \begin{array}{l} \alpha_{A,A_i} + \alpha_{A,B_i} + \alpha_{A,C_i} + \alpha_{A,O_i} = 1 \\ \alpha_{B,B_i} + \alpha_{B,A_i} + \alpha_{B,C_i} + \alpha_{B,O_i} = 1 \\ 0 \leq \alpha_{A,B_i} \leq 1, 0 \leq \alpha_{A,C_i} \leq 1, 0 \leq \alpha_{A,O_i} \leq 1 \\ 0 \leq \alpha_{B,A_i} \leq 1, 0 \leq \alpha_{B,C_i} \leq 1, 0 \leq \alpha_{B,O_i} \leq 1 \\ P_C \geq 0 \end{array}$$

Theorem 3.1. *The caching variables α_{I,S_i} take on values of either 0 or 1 at the equilibrium of the caching and pricing strategies game.*

Proof. The solution of a maximization (minimization) problem with an objective function that has a linear relationship with the variable is the boundary point of the feasible interval. Therefore, since the relationship between utility function and caching parameters are linear to maximize the utility functions, they just take on the boundary values. Since $\alpha_{I,S_i} \in [0, 1]$, therefore, they can be either 0 or 1. $\square$

Given Theorem 3.1, it follows that all caching variables adopt binary values. For example, according to equation (3.9), if $P_C > c_{A_i}$, the caching variable α_{A,A_i} should be 1 to maximize the access ICN A's utility function. Whenever $P_C > c_{A_i}$ it means that the transit price for delivering the requested content type i to users under the access ICN A is smaller than the caching cost of access ICN A for locally storing the requested file itself. Therefore the access ICN decides to store the content locally in its cache rather than transferring the request to transit ICN C. That is the reason that caching variable α_{A,A_i} gets the value 1. The caching strategies at equilibrium for different conditions are summarized in Table 3.2.

Unlike the caching parameters which only get binary values; the pricing parameters can be continuous. Therefore, this optimization problem is a mixed integer program with multiple objective functions, and in general, the uniqueness of the NE in terms of pricing and caching strategies cannot be characterized.

3.5 Generalization to K Access ICNs

We can extend the case of two access ICNs, one transit ICN and one content provider to a generalized scenario of K access ICNs, one transit ICN and one content provider. We consider $\tilde{K} = \{A_1, A_2, ..., A_K\}$ as the set of access ICNs which are connected to transit ICN C and α is the set of all caching variables. The demand function of each access ICN is defined in Definition 3.3.

Definition 3.3. *The received demands at access ICN A_j is defined as*

Table 3.2: *Caching table for each content type i

	Condition	α_{A,A_i}	α_{A,B_i}	α_{A,C_i}	α_{A,O_i}	α_{B,B_i}	α_{B,A_i}	α_{B,C_i}	α_{B,O_i}
1	$P_C > (c_{A_i} \& c_{B_i})$	1	0	0	0	1	0	0	0
	$P_C > c_{A_i} \& P_C < c_{B_i}$								
2	$c_{C_i} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	1	0	0	0	0	0	1	0
3	$P_O^{(s)} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	1	0	0	0	0	0	0	1
4	$P_A^{(s)} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	1	0	0	0	0	1	0	0
	$P_C < c_{A_i} \& P_C > c_{B_i}$								
5	$c_{C_i} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	0	0	1	0	1	0	0	0
6	$P_O^{(s)} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	0	0	0	1	1	0	0	0
7	$P_B^{(s)} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	0	1	0	0	1	0	0	0
	$P_C < (c_{A_i} \& c_{B_i})$								
8	$c_{C_i} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	0	0	1	0	0	0	1	0
9	$P_O^{(s)} = min\left\{c_{C_i}, P_O^{(s)}, P_A^{(s)}, P_B^{(s)}\right\}$	0	0	0	1	0	0	0	1

$*$ This table shows the possible caching strategies at the equilibrium under different conditions. In each case one of the components (Access ICN, Transit ICN or Content Provider) serve the request which the corresponding caching variable takes on value 1.

$$\sigma_{A_j} = 1 - \rho_{A_j} P_{A_j} + \tfrac{1}{K-1}\left[\sum_{k=1,k\neq j}^{K} \rho_{A_k} P_{A_k}\right] - \rho_0 P_o^{(c)}, \forall j = 1, ..., K. \tag{3.13}$$

The received demand of content type i by access ICN A_j can be shown as $\sigma_{A_{j_i}} = \sigma_{A_j} q_M(i)$. Following the previous section, the maximization problem of each access ICN $A_j \in K$ can be defined as follows:

$$\begin{aligned} &\max_{\alpha_{A_j,A_{j_i}}, P_{A_j}} U_{A_j} = \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \sigma_{A_j} \alpha_{\left(A_j,A_j\right)_i} \left(P_C - c_{A_{j_i}}\right) + \sigma_{A_j}\left(P_{A_j} - P_C\right) + \\ \left(\left(\frac{1}{1+\beta_{A_j}}\right) P_{A_j} - c_{A_{j_i}}\right) \times \sum\limits_{k \in \tilde{K}, k \neq A_j} \sigma_k \alpha_{(k,A_j)_i} \end{array} \right\} \\ &s.t. \qquad 0 \leq \alpha_{A_j,A_{j_i}} \leq 1 \ , P_{A_j} > 0 \end{aligned} \tag{3.14}$$

The transit ICN C maximization problem is given by the following equation:

$$\begin{aligned}\max_{\alpha,P_C} U_C = &\sum_{k\in\tilde{K}}\sum_{i=1}^{M} q_M(i)\sigma_k\alpha_{k,C_i}(P_C - c_{C_i})\\ &+\sum_{k\in\tilde{K}}\sum_{L\in\tilde{K}\cup\{O\},L\neq k}\sum_{i=1}^{M} q_M(i)\sigma_k\alpha_{k,L_i}(P_C - P_L^{(s)})\end{aligned}$$

$$s.t. \quad \begin{aligned} &\alpha_{(A_j,A_j)_i} + \sum_{\substack{L\in\tilde{K}\cup\{C,O\}\\ L\neq A_j}} \alpha_{(A_j,L)_i} = 1\\ &0 \leq \alpha \leq 1\\ &P_C \geq 0\end{aligned} \tag{3.15}$$

and the content provider O maximizes its utility using equation (3.16).

$$\begin{aligned}\max_{P_O^{(c)},P_O^{(s)}} U_O = \sum_{i=1}^{M} q_M(i)\left[\sum_{k\in\tilde{K}}\sigma_k\alpha_{k,O_i}\right]\left(P_O^{(s)} - c_O\right) \quad + \sum_{k\in\tilde{K}}\sigma_k P_O^{(c)}\\ s.t. \qquad P_O^{(C)} > 0, P_O^{(s)} > c_O\end{aligned} \tag{3.16}$$

Theorem 3.1 can be extended for the generalized K Access ICN case with the same reasoning and all the caching variables take binary values (i.e., all or nothing 0-1 strategies). So, the joint caching and pricing strategy game in the general form is also a mixed integer program. In the next section we will simplify the problem with the assumption of symmetric access ICNs with similar characteristics and try to give some analytical and intuitive results.

3.6 Symmetric Access ICNs Scenario Analysis

In the previous section, the general form of the caching and pricing strategies for ICNs was formulated through equations (3.14)-(3.16) as a set of mixed integer programs. In this section, in order to analytically study the equilibrium of our proposed model, we consider the symmetric scenario, where all access ICNs have the same specifications. For the symmetric scenario, where all the access ICNs are exactly the same, we consider $\rho_k = \rho, \beta_k = \beta, c_{k_0} = c_0 \forall k \in \tilde{K}$.

Theorem 3.2. *In the symmetric case, for each content type i, $\alpha_{A_k,A_{j_i}} = 0$, $\forall k \neq j$.*

Proof. According to equations (3.14), when access ICN A_j receives a request for content type i and transit price is greater than its caching cost for that particular type of content ($P_C \geq c_{A_{j_i}}$), access ICN A_j decides to serve the requested content itself by adopting value 1 for caching parameter $\alpha_{A_j,A_{j_i}}$. Since $\alpha_{A_j,A_{j_i}} = 1$, then all the other caching parameters for content type i would be equal to zero. On the other hand, if the transit price is less than the access ICN's caching cost for content type i ($P_C < c_{A_{j_i}}$), the access ICN A_j will forward the request to the transit ICN C to be served by choosing $\alpha_{A_j,A_{j_i}} = 0$. When the transit ICN C receives the request, it should decide to either cache the content or forward it to the content provider or other access ICNs based on the payoff that it gains according to equations (3.15). Considering the Theorem 3.1 and the constraint $\alpha_{(A_j,A_j)_i} + \sum\limits_{L \in \tilde{K} \cup \{C,O\}, L \neq A_j} \alpha_{(A_j,L)_i} = 1$, one of the caching parameters should be 1 and others should be 0. If c_{C_i} or $P_O^{(s)}$ are the minimums among $\left\{c_{C_i}, P_O^{(s)}, P_k^{(s)} \forall k \in \tilde{K}, k \neq A_j\right\}$, the transit ICN C ($\alpha_{A_j,C_i} = 1$) or the content provider O ($\alpha_{A_j,O_i} = 1$) will serve the request, respectively. Now assume that one of the access ICN's storage price $P_k^{(s)}$ is the minimum. If $P_k^{(s)}$ is the minimum, it means that $P_k^{(s)} < \left(c_{C_i} \& P_O^{(s)}\right)$. On the other hand, $P_k^{(s)} > c_{K_i}$ in order to the access ICN K accepts the request; otherwise it does not accept the content to prevent from losing payoff. Therefore, $c_{K_i} < P_k^{(s)} < c_{C_i}$. Note that $c_{K_i} = c_{A_{j_i}}$ in symmetric scenario. Thus $P_k^{(s)}$ should adopt a value greater than access ICN A_j's caching cost and less than transit ICN C's caching cost for content type i ($c_{A_{j_i}} < P_k^{(s)} < c_{C_i}$). On the other hand, we know that $P_C < c_{A_{j_i}}$, therefore $P_C < c_{A_{j_i}} < P_k^{(s)} < \left(c_{C_i} \& P_O^{(s)}\right)$. It means that $(P_C - c_{C_i})$, $\left(P_C - P_k^{(s)}\right)$ and $\left(P_C - P_O^{(s)}\right)$ are negative and that is a contradiction since the transit ICN C is trying to choose P_C in a way to get at least zero payoff. Thus $P_k^{(s)}$ can never be the minimum value among the others and accordingly $\alpha_{A_k,A_{j_i}} = 0$, $\forall k \neq j, \forall i$, in the symmetric scenario case. □

To better understand the implication of Theorem 3.2, we can refer to Table 3.2 that shows the caching strategies at the equilibrium for asymmetric case with two access ICNs which are not identical. By assuming two identical access ICNs that have the same characteristics, the cases 2-7 can be removed and the table will reduce to just three cases and in all of these cases $\alpha_{A,B_i} = \alpha_{B,A_i} = 0$. This table can also be extended for the generalized scenario with K access ICNs.

What the above theorem reveals is that in the symmetric case, access ICNs have no motivation to serve each other's users. This is not against the philosophy of the content centric network paradigm, since in this setup the access ICNs and also the transit ICN are capable of caching the requested content. Besides, this theorem is just for the *Symmetric Scenario* and in the asymmetric setup the access ICNs are able to cache requests for users of other access ICNs.

Moreover, according to Theorem 3.2, when the system is symmetric, we can add the following facts to our models.

$$\begin{aligned} \alpha_{A_j,C_i} &= \alpha_{A_k,C_i}, \qquad \forall i \\ \alpha_{A_j,O_i} &= \alpha_{A_k,O_i}, \qquad \forall i \end{aligned} \tag{3.17}$$

So using Theorem 3.2 and (3.17), we can reformulate our maximization problem described in (3.14)-(3.16) for symmetric case. The maximization problems for access ICN A_j can be expressed in equation (3.18) as follows:

$$\begin{aligned} &\max_{P_{A_j},\alpha_{A_j,A_{j_i}}} \quad U_{A_j} = \sigma_{A_j} \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \alpha_{A_j,A_{j_i}} \left(P_C - c_{A_{j_i}}\right) \\ + \left(P_{A_j} - P_C\right) \end{array} \right\} \\ &s.t. \qquad \alpha_{A_j,A_{j_i}} \in \{0,1\}, P_{A_j} > 0 \end{aligned} \tag{3.18}$$

The maximization problem for transit ICN C can be defined in the following equation:

$$\begin{aligned} &\max_{\substack{\alpha_{A_j,C_i} \\ \alpha_{A_j,O_i} \\ P_C}} U_C = \sum_{k=1}^{K} \sigma_{A_k} . \sum_{i=1}^{M} q_M(i) \left\{ \begin{array}{l} \alpha_{A_j,C_i} \left(P_C - c_{C_i}\right) + \\ \alpha_{A_j,O_i} \left(P_C - P_O^{(s)}\right) \end{array} \right\} \\ &s.t. \qquad \begin{array}{c} \alpha_{A_j,A_{j_i}} + \alpha_{A_j,C_i} + \alpha_{A_j,O_i} = 1, \\ \alpha_{A_j,C_i} \in \{0,1\}, \alpha_{A_j,O_i} \in \{0,1\}, P_C > 0, \end{array} \end{aligned} \tag{3.19}$$

and finally, the content provider maximization problem can be reformulated as the following

$$\begin{aligned} &\max_{\substack{P_O^{(c)} \\ P_O^{(s)}}} U_O = \sum_{k=1}^{K} \sigma_{A_k} \left[P_O^{(c)} + \sum_{i=1}^{M} q_M(i) \alpha_{A_j,O_i} \left(P_O^{(s)} - c_O\right) \right] \\ &s.t. \qquad P_O^{(C)} > 0, P_O^{(s)} > c_O \end{aligned} \tag{3.20}$$

As mentioned in Theorem 3.1, the caching parameters still take on binary values. Moreover, since the content popularity probability function $q_M(i)$ is monotonically decreasing; according to (3.1), the access and transit ICNs' caching costs are monotonically increasing depending on the content type. For access ICN A_j, the caching parameter $\alpha_{A_j,A_{j_i}}$ adopts value 1 when the transit price P_C is greater than the access ICN caching cost $c_{A_{j_i}}$. Hence, if $\alpha_{A_j,A_{j_i}}$ is 1 for content type i, it would also be 1 for content type $i-1$. It means that if access ICN decides to cache the content type i, it will cache all the other content that are more popular than it. So, there would be a caching threshold index for the number of content type that access ICN is willing to locally store. We denote the optimum caching threshold index by Th_{Aj} for access ICN A_j. In the symmetric scenario, the caching threshold indices Th_{Aj} are identical for all access ICNs, so we consider Th as the caching threshold index for all the access ICNs. Since caching content types $Th+1$ to M is not beneficial for access ICNs, they will forward these to the transit ICN C to be served. The transit ICN C should decide to either serve the content itself or forward it to somewhere else. In the symmetric case, as mentioned in Theorem 3.2, in case it decides not to serve the content itself, it can forward it to the content provider O. According to (3.19), if the content provider storage price $P_O^{(s)}$ is greater than the transit ICN C caching cost c_{C_i}, the caching parameter α_{A_j,C_i} adopts value 1 and the transit ICN C caches the content type i. On the other hand, if $P_O^{(s)}$ is less than c_{C_i}, the caching parameter α_{A_j,C_i} will be 0 and α_{A_j,O_i} adopts value 1. In this case, the content provider O will take care of the request for content type i. As discussed, for access ICNs, the transit ICN also can have a caching threshold index. It means that if it caches content type i, it would also be able to cache the content type $i-1$. So there would be a caching threshold index for the number of content type that the transit ICN C is willing to locally store. We denote this caching threshold index by Th_C. So the transit ICN C will serve the content with popularity index $Th+1$ to Th_C and content with popularity index greater than Th_C will be served by the content provider O. We can summarize these new parameters in (3.21) and (3.22).

$$\alpha_{A_j,A_{j_i}} = \begin{cases} 1 & i \le Th \\ 0 & i > Th \end{cases}, \forall j \in \{1,...,K\} \tag{3.21}$$

$$\begin{aligned} \alpha_{A_j,C_{j_i}} &= \begin{cases} 1 & Th+1 \le i \le Th_C \\ 0 & i \le Th \end{cases}, \forall j \in \{1,...,K\} \\ \alpha_{A_j,O_{j_i}} &= \begin{cases} 1 & Th_C+1 \le i \le M \\ 0 & i \le Th_C \end{cases}, \forall j \in \{1,...,K\} \end{aligned} \tag{3.22}$$

Thus, all the caching variables α will be replaced by two caching threshold indices Th and Th_C. By (3.21) and (3.22), the problem set of (3.18)-(3.20) can be rearranged using new parameters Th and Th_C. The new maximization problem for access ICN A_j is given as the following:

$$\begin{aligned} &\underset{P_{A_j},Th}{max}\ U_{A_j} = \sigma_{A_j}\left[P_{A_j} - Th.c_0 - P_C \sum_{i=Th+1}^{M+1} q_M(i)\right] \\ &s.t. \quad 0 \le Th \le M,\ P_{A_j} > 0 \end{aligned} \tag{3.23}$$

The transit ICN C maximizes its utility function as the following:

$$\begin{aligned} &\underset{P_C,Th_C}{max}\ U_C = \left(K - K\rho_0 P_O^{(c)}\right)\left\{\begin{array}{l} P_C \sum\limits_{i=Th+1}^{M+1} q_M(i) \\ -\left(Th_C - Th\right) c_{C_0} \\ -P_O^{(s)} \sum\limits_{i=Th_C+1}^{M+1} q_M(i) \end{array}\right\} \\ &s.t. \quad Th \le Th_C \le M,\ P_C > 0 \end{aligned} \tag{3.24}$$

And the content provider O maximization problem is formulated using new parameters in (3.25)

$$\begin{aligned} &\underset{P_O^{(c)},P_O^{(s)}}{max}\ U_O = \left(K - K\rho_0 P_O^{(c)}\right)\left\{\begin{array}{l} P_O^{(c)} + \left(P_O^{(s)} - c_O\right). \\ \sum\limits_{i=Th_C+1}^{M+1} q_M(i) \end{array}\right\} \\ &s.t. \quad P_O^{(C)} > 0, P_O^{(s)} > c_O \end{aligned} \tag{3.25}$$

Note that in the above equations $q_M(0) = q_M(M+1) = 0$.

The problem of joint caching and pricing strategies for the case of symmetric ICNs can be decomposed into two independent caching and pricing optimization problems. The caching problem is dealing with the parameters that affect the caching process and is stated as follows:

- **Caching Problem:**

$$\begin{aligned} &\min_{Th} Th.c_0 + P_C \sum_{i=Th+1}^{M+1} q_M(i) \\ &s.t. \quad 0 \le Th \le M \end{aligned} \tag{3.26}$$

$$\begin{aligned} \max_{P_C, Th_C} \; & P_C \sum_{i=Th+1}^{M+1} q_M(i) + (Th - Th_C)\, c_{C_0} \\ & - P_O^{(s)} \sum_{i=Th_C+1}^{M+1} q_M(i) \\ s.t. \quad & Th \le Th_C \le M, P_C > 0 \end{aligned} \tag{3.27}$$

$$\begin{aligned} &\max_{P_O^{(s)}} \left(P_O^{(s)} - c_O\right). \sum_{i=Th_C+1}^{M+1} q_M(i) \\ &s.t. \quad P_O^{(s)} > c_O \end{aligned} \tag{3.28}$$

The outcome of this problem is a 4-tuples $\left(Th^*, {Th_C}^*, {P_C}^*, {P_O^{(s)}}^*\right)$. The pricing problem is defined in (3.29) and (3.30) by substituting the 4-tuple resulting from the caching problem.

- **Pricing Problem:**

$$\begin{aligned} &\max_{P_{Aj}} U_{Aj} = \sigma_{Aj} \left[P_{Aj} - Th^*.c_0 - {P_C}^* \sum_{i=Th^*+1}^{M+1} q_M(i) \right] \\ &s.t. \quad P_{Aj} > 0 \end{aligned} \tag{3.29}$$

$$\begin{aligned} \max_{P_O^{(c)}} U_O &= \left(K - K\rho_0 P_O^{(c)}\right) \left\{ \begin{array}{l} P_O^{(c)} + \sum\limits_{i=Th_C{}^*+1}^{M+1} q_M(i) \\ \times \left(P_O^{(s)^*} - c_O\right) \end{array} \right\} \\ s.t. \quad & P_O^{(c)} > 0 \end{aligned} \tag{3.30}$$

The K+1-tuple $\left(P_{A1}^*, ..., P_{AK}^*, P_O^{(c)^*}\right)$ is the outcome of the pricing problem. The *NE* of the joint caching and pricing problem is $\left(Th^*, Th_C{}^*, P_C{}^*, P_O^{(s)^*}, P_{A1}^*, ..., P_{AK}^*, P_O^{(c)^*}\right)$.

Theorem 3.3. *The caching problem introduced above is a two player matrix game between transit ICN C and content provider O.*

Proof. According to (3.14), when the transit price for ICN C is greater than access ICN's caching cost for content type i $\left(P_C \geq c_{A_{j_i}}\right)$, the access ICN caches all the content which are more popular than content type i. Therefore, when $c_{A_{j_i}} \leq P_C < c_{A_{j_{i+1}}}$, the optimum caching threshold index chosen by access ICN will be $Th^* = i$. On the other hand, since the utility function of the transit ICN C has a linear relationship with the transit price P_C , the transit ICN will choose the maximum value possible that is $c_{A_{j_{i+1}}} - \varepsilon$ (ε is a very small value). Thus, the transit ICN C also can adopt its actions from a discrete set. There is a caching threshold index Th corresponding with each transit price chosen by ICN C. It shows that the transit ICN is the leader in its relationship with the access ICNs and its action is (P_C, Th) from a set of $M+1$ feasible choices. The relationship between the transit ICN C and the content provider is also a leader follower game. Depending on the storage price $P_O^{(s)}$, the transit ICN C might forward some part of demands to the content provider O to be served. If the access ICN decides to cache the content more popular than content type Th, the rest of the content should be forwarded to transit ICN. Therefore, the content type with index $Th+1$ to M is going to be served in either the transit ICN or the content provider. When $c_{C_j} \leq P_O^{(s)} < c_{C_{j+1}}$, the optimum caching threshold index chosen by the transit ICN C will be $Th_C{}^* = j$. Since, the utility of the content provider O has a linear relationship with the storage price $P_O^{(s)}$; the content provider will pick the maximum value possible for the storage price that is $c_{C_{j+1}} - \varepsilon$. Thus, for every content provider storage price, there is a corresponding caching threshold index chosen by the transit ICN C. Since both the transit ICN C and the content provider O have a limited set of discrete actions, the problem introduced in (3.26)-(3.28) is a matrix game between the transit ICN C and the content provider O when the transit ICN action is the transit price P_C and content provider action is the storage price $P_O^{(s)}$. The access ICNs cannot change the results and they just follow the transit ICN and their actions. $\square$

By Theorem 3.3, we can discard (3.26) and solve equations (3.27) and (3.28) jointly to find the integer thresholds Th and Th_C. Note that P_C and $P_O^{(s)}$ are functions of Th and Th_C, respectively as follows:

$$P_C\left(Th\right) = \begin{cases} c_{A_{j_{Th+1}}} - \varepsilon & 0 \le Th \le M-1 \\ c_{A_{j_{Th}}} + \varepsilon & Th = M \end{cases} \tag{3.31}$$

$$P_O^{(s)}\left(Th_C\right) = \begin{cases} c_{C_{Th+1}} - \varepsilon & 0 \le Th_C \le M-1 \\ c_{C_{Th}} + \varepsilon & Th_C = M \end{cases} \tag{3.32}$$

Proposition 3.1. *$f\left(Th\right) = P_C\left(Th\right)\sum\limits_{i=Th+1}^{M+1} q_M(i) + Thc_{C_0}$ and $g\left(Th_C\right) = \left(P_O^{(s)}\left(Th_C\right) - c_O\right) \cdot \sum\limits_{i=Th_C+1}^{M+1} q_M(i)$ are concave sequences and have a unique maximum.*

Proof. A sequence S is strictly concave if the below inequality holds for every n.

$$S\left(n+1\right) + S\left(n-1\right) - 2S\left(n\right) < 0$$

As stated before, $q_M\left(i\right) = \frac{1}{\left(\sum\limits_{j=1}^{M} \frac{1}{j^\gamma}\right) i^\gamma}$ for $= 1, ..., M$ and $q_M\left(i\right) = 0$ for $i > M$. Using (3.31) for $Th = 1, ..., M-1$, we have

$$P_C\left(Th\right) = c_0(Th+1)^\gamma \sum_{i=1}^{M} \left(\frac{1}{i}\right)^\gamma - \varepsilon$$

Since ε is so small and tends to zero, it can be considered as zero. Therefore, for $Th = 1, ..., M-1$, the sequence f is defined as follows

$$f\left(Th\right)=\left[c_0(Th+1)^{\gamma}\sum_{i=1}^{M}\left(\tfrac{1}{i}\right)^{\gamma}\right]\sum_{i=Th+1}^{M+1}q_M(i)+Thc_{C_0}$$
$$\Rightarrow f\left(Th\right)=\left[c_0(Th+1)^{\gamma}\right]\sum_{i=Th+1}^{M+1}\left(\tfrac{1}{i}\right)^{\gamma}+Thc_{C_0}$$

To prove concavity, we need to have

$$f\left(Th+1\right)+f\left(Th-1\right)-2f\left(Th\right)<0\Rightarrow$$
$$\left[(Th+2)^{\gamma}-2(Th+1)^{\gamma}+Th^{\gamma}\right]\sum_{i=Th+2}^{M+1}\left(\tfrac{1}{i}\right)^{\gamma}$$
$$<1-\left(\tfrac{Th}{Th+1}\right)^{\gamma}.$$

If this inequality holds for all M and $Th=1,...,M-1$, then the sequence f is concave. We introduce $\varphi_{\gamma}\left(Th\right)=\left[(Th+2)^{\gamma}-2(Th+1)^{\gamma}-Th^{\gamma}\right]$. Two functions $\varphi_0\left(Th\right)$ and $\varphi_1\left(Th\right)$ are zero for all Th that satisfy the above inequality. It means that for $\gamma=0$ and $\gamma=1$, f is a concave sequence. Moreover, $\varphi_{\gamma}\left(Th\right)$ can be written as

$$\varphi_{\gamma}\left(Th\right)=\left[(Th+2)^{\gamma}-(Th+1)^{\gamma}\right]-\left[(Th+1)^{\gamma}+Th^{\gamma}\right]$$
$$\Rightarrow\varphi_{\gamma}\left(Th\right)=\left(\tfrac{Th}{Th+1}\right)^{\gamma}+\left(\tfrac{Th+2}{Th+1}\right)^{\gamma}-2.$$

Since $\left(\tfrac{Th}{Th+1}\right)^{\gamma}+\left(\tfrac{Th+2}{Th+1}\right)^{\gamma}<2$ for $0<\gamma<1$, $\varphi_{\gamma}\left(Th\right)$ is negative for all $Th=1,...,M-1$ and M which cause that above inequality be satisfied. Therefore, f is a concave sequence for all M and all $0\leq\gamma\leq1$.

Using (3.32) for $Th_C=1,...,M-1$, we have

$$P_O^{(s)}\left(Th_C\right)=c_{C_0}(Th_C+1)^{\gamma}\sum_{i=1}^{M}\left(\tfrac{1}{i}\right)^{\gamma}-\varepsilon$$

Since ε is so small and tends to zero, it can be considered as zero. Therefore, for $Th_C=1,...,M-1$, the sequence g can be defined as

$$g=c_{C_0}(Th_C+1)^{\gamma}\sum_{i=Th_C+1}^{M}\left(\tfrac{1}{i}\right)^{\gamma}-c_O\sum_{i=Th_C+1}^{M+1}q_M(i)$$

On the other hand, we can show that if two sequences are concave the sum of them is also concave. Assume that $g=\Psi+\Delta$. If Ψ and Δ are concave, we have

$$\Psi\left(n+1\right)+\Psi\left(n-1\right)-2\Psi\left(n\right)<0$$

and

$$\Delta\left(n+1\right)+\Delta\left(n-1\right)-2\Delta\left(n\right)<0.$$

Therefore,

$$\begin{aligned}&\Psi(n+1)+\Delta(n+1)+\Psi(n-1)+\Delta(n-1)\\&-2\left[\Psi(n)+\Delta(n)\right]<0\end{aligned}$$

That means

$$g(n+1)+g(n-1)-2g(n)<0$$

We already showed that the first term of sequence g is concave, so we just need to show that the second term $\Delta(Th_C)=-c_O\sum_{i=Th_C+1}^{M+1}q_M(i)$ is also concave. Then, we have to show that the following inequality holds for every $Th_C=1,...,M-1$

$$\Delta(Th_C+1)+\Delta(Th_C-1)-2\Delta(Th_C)<0$$

By doing some algebra, we will get $q_M(Th_C)>q_M(Th_C+1)$ that holds for $Th_C=1,...,M-1$, since $q_M(i)$ is a strictly decreasing function. That completes the proof. $\square$

Theorem 3.4. *The symmetric joint caching and pricing game has a unique NE.*

Proof. The solution of the caching and pricing problems in (3.26)-(3.30) is the *NE*. On the one hand, the caching problem set is like a leader-follower game and the transit ICN C (leader) maximizes (3.27) and the content provider O (follower) maximizes (3.28). Since both of them are concave sequences based on Proposition 3.1, they have only one maximum in their feasible sets. By (3.27), Th^* can be defined as the unique optimum value for the access ICN caching threshold index and by (3.28), ${Th_C}^*$ can be defined as the optimum caching threshold index of the transit ICN C that should always be greater or equal than Th^*. Assume that $Th_{C_{\max}}$ is the value that maximizes (3.28). If ${Th_C}^*<Th_{C_{\max}}$ then ${Th_C}^*=Th_{C_{\max}}$ and if ${Th_C}^*\geq Th_{C_{\max}}$ then ${Th_C}^*=Th^*$. By finding the first set of parameters and substituting them in (3.29) and (3.30), we can find the second set of parameters. Since these are concave quadratic functions the

problem has the following unique solution

$$\begin{aligned} P_O^{(c)^*} &= max\left\{0, \frac{1-\rho_0\left(P_O^{(s)^*}-c_O\right)\sum\limits_{i=Th_C^*+1}^{M+1} q_M(i)}{2\rho_0}\right\}, \\ P_A{}^* = P_B{}^* &= \frac{1}{\rho}\left[\begin{array}{l} \rho\left(Th^*.c_0 + P_C{}^* \sum\limits_{i=Th^*+1}^{M+1} q_M(i)\right) \\ +1-\rho_0 P_O^{(c)^*} \end{array}\right]. \end{aligned} \tag{3.33}$$

Note that $P_{A_j}^*$'s are always greater than zero. Hence, as we have unique set of results for Th^* and $Th_C{}^*$, so $P_{A_j}^*$'s and $P_O^{(c)^*}$ are also unique. Therefore the NE exists and is unique. □

As the analytical results show, the NE for the symmetric case is independent of number of access ICNs. So, for the numerical results section, we consider the scenario with only two access ICNs.

3.7 Numerical Results

We consider the interaction among two symmetric access ICNs, one transit ICN and a content provider who are competing to maximize their utilities. In this scenario, the reflective coefficients of price's influence on users' demands ρ_A, ρ_B and ρ_0 are set identically to 0.1. These parameters model the sensitivity of the demands to an increase in the prices. The scaling parameter between network price and storage price is set to $\beta_A = \beta_B = 10$, i.e., the storage price is an order of magnitude less than the network price. There are $M = 100$ different types of content in this network which are randomly requested by users according to a generalized Zipf distribution. Fig. 3.2 shows how the access ICN caching cost varies for different types of content and different values of Zipfs factor γ when initial caching cost $c_0 = 1$. As Zipf's factor γ is increasing, the distribution of content requested is getting skewed and according to (3.2), the caching costs of more popular content decrease while the caching costs of less popular content increase. Note that the transit ICN C possibly has (on average) access to cheaper caches. We denote the ratio of transit ICN caching cost to access ICN caching cost by R. We assume that the content provider cost (c_O) for all types of content is identical.

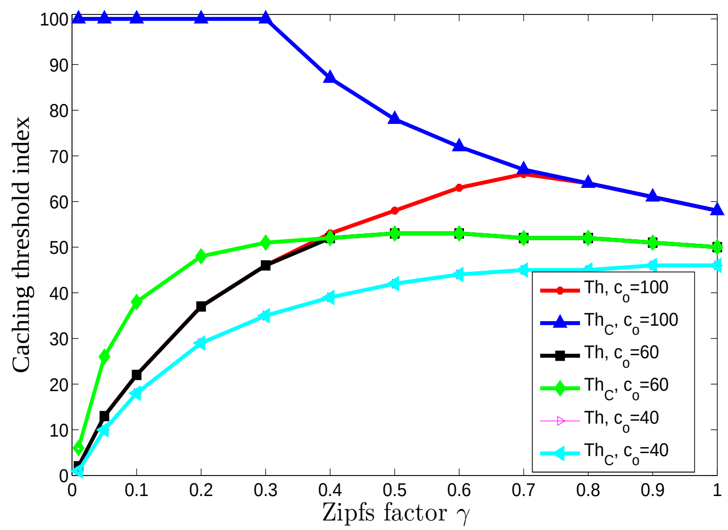


Figure 3.4: Access ICN/transit ICN Caching threshold vs Zipfs factor γ for various content provider cost c_O when $R = 0.7$.

The caching strategies Th^* and $Th_C{}^*$ at the equilibrium are shown in Fig. 3.4, where Zipfs factor γ is varying in the interval of 0.01 to 1. In this scenario, the caching cost of transit ICN C is set to 70% of the caching cost of access ICNs for every type of content i, i.e., $R = 0.7$. The results are compared for $c_O = 40, 60, 100$. As seen, for small amount of γ the content are less skewed and the caching cost for them is very similar. Since the caching cost of transit ICN is less than access ICNs' and content provider's, it decides to cache most of the content. But as γ increases, some of the content is getting relatively more popular. In this case, the access ICNs prefer to cache the more popular content locally and smaller amount of content will be cached by transit ICN and content provider. For the higher γ, the caching cost of access ICNs and transit ICN is getting higher and they do not have an incentive to locally store them. Therefore, at this point, the content provider starts to serve more content than before and the transit ICN does not cache content and just forwards requests to the content provider. For higher content provider cost $c_O = 100$, since the cost is so high the content provider just serves the less popular requested content but as the cost c_O increases, it starts to serve more content. For the case which c_O is relatively small ($c_O = 40$), the transit ICN does not have the incentive to spend its resources for caching the requested content

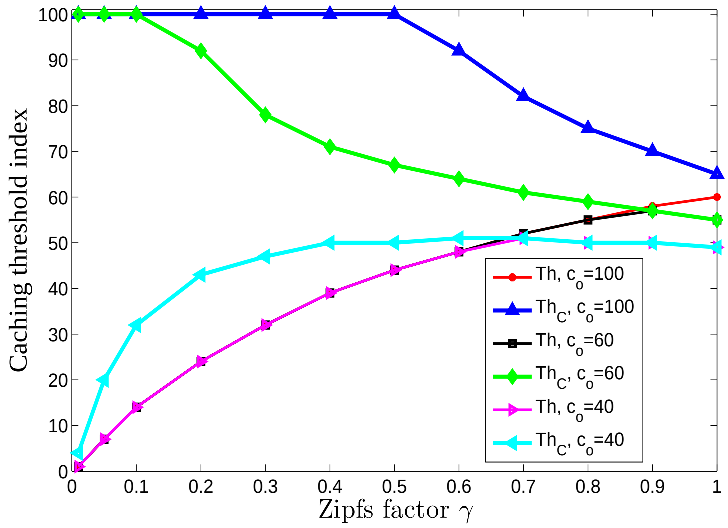


Figure 3.5: Access ICN/transit ICN Caching threshold vs Zipfs factor γ for various content provider cost c_O when $R = 0.5$.

and just forwards all the requests for content from access ICNs to the content provider. The same scenario with $R = 0.5$ is examined in Fig. 3.5. In this case, the transit ICN caching cost is half of the access ICN caching cost for each type of content. So, the transit ICN caches more requested content in its cache.

Fig. 3.6 shows the access ICN price P_A as function of γ for different content provider costs c_O. P_A decreases as the γ gets higher when the content provider cost c_O is relatively low. The reason is that as γ increases, the caching costs of more popular content at access ICNs caches are getting lower. Therefore, the access ICNs need to decrease their price P_A in order to compete with other access ICNs. But as the relative content provider cost c_O increases, the access price is getting higher since both the access ICN and the transit ICN have a greater incentive to locally cache the content. However, the price for greater value of c_O displays a bimodal behavior as a function of γ. According to Fig. 3.2, for relatively small value of γ, the different types of content have similar popularity and the access and transit ICN should incur more or less similar caching costs for each type of content. But as γ increases, some of the content is getting more popular and the cost of caching them at the access ICN is decreasing. Therefore, the access ICN decreases its access price in order to induce increased demand from

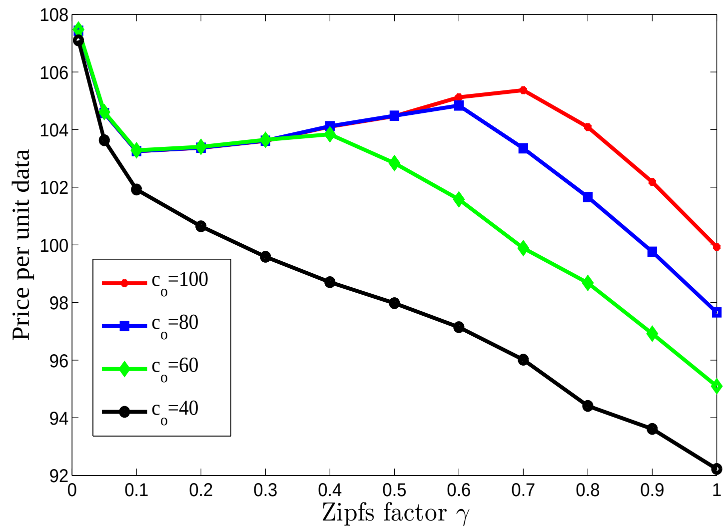


Figure 3.6: Access price P_A vs Zipfs factor γ for various content provider cost c_O and $R = 0.7$.

the users (see relationship between P_A and demand in equation (3.3)). On the other hand, when γ keeps increasing, the transit price and the content provider storage price increase. Thus, the access ICN needs to slightly increase its price to compensate the increase in the transit price. After the slight increase, as γ increases further, the access price P_A decreases again. This is because, as seen from Fig. 4, the caching strategies Th^* and ${Th_C}^*$ are the same for large γ. At this point, the access ICNs decide to cache the content that are more popular to get higher payoff. As a result, the access ICN decreases the price to further incentivize greater user demand for popular content.

The transit price P_C and the content provider storage price $P_O^{(s)}$ are shown in Fig. 3.7. As γ increases, the caching costs of less popular content that are going to be cached in transit ICN or content provider caches are increasing. Therefore, both the transit ICN and content provider should increase their prices as shown in the figure.

3.8 Discussion

In this chapter, we developed an analytical framework for distribution of popular content in an Information Centric Network (ICN) that comprises of Access ICNs, a Transit

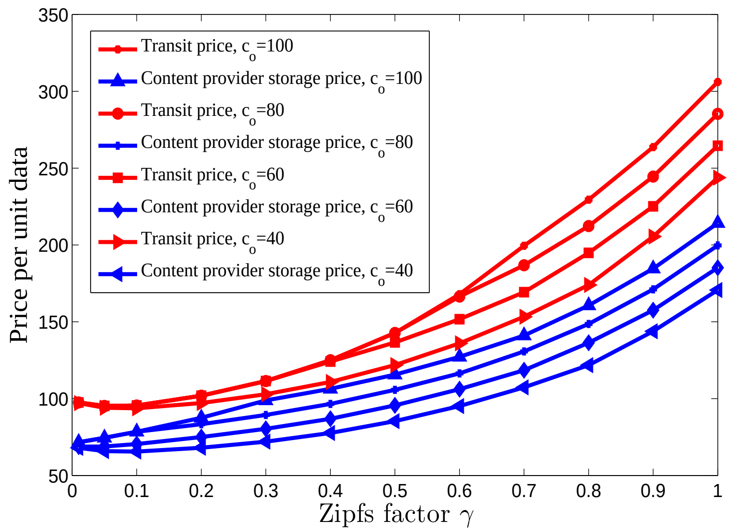


Figure 3.7: Transit price / content provider storage price vs Zipfs factor γ for various content provider cost c_O and $R = 0.7$.

ICN and a Content Provider. By modeling the interaction of the above entities using game theory and under the assumption that the caching cost of the access ICNs and transit ICNs is inversely proportional to popularity, which follows a generalized Zipf distribution, we first showed that at the NE, the caching strategies turn out to be 0-1 (all or nothing). Further, for the case of symmetric Access ICNs, we showed that a unique NE exists and the caching policy (0 or 1) is determined by a threshold on the popularity of the content (reflected by the Zipf probability metric), i.e., all content more popular than the threshold value is cached. Moreover, we showed that the resulting threshold indices and prices can be obtained by a decomposition of the joint caching and pricing problem into two independent caching only and pricing only problems. Finally, using numerical results we showed that as the Zipf's factor γ increases and the relative popularity of the content gets skewed, the access ICN just caches the more popular content and the content provider serves only requests for less popular content while the transit ICN just forwards the demands to the content provider without locally caching any content itself. The insights obtained from the analysis here warrants further investigation into the case of asymmetric access ICNs. In this chapter, we discussed a hierarchical scenario with K access ICNs, one transit ICN and one content provider.

In the special case that the transit ICNs are not ”interconnected’ to each other, the model in this chapter is readily extendable for multiple transit ICNs.

Chapter 4

Defense Against Advanced Persistent Threats

4.1 Overview and Motivation

Cloud storage and cyber systems are vulnerable to Advanced Persistent Threats (APTs), in which an attacker applies multiple sophisticated methods such as the injection of multiple malwares to continuously and stealthily steal data from the targeted cloud storage system [137–140]. APT attacks are difficult to detect and have caused privacy leakage and millions of dollars' loss [46, 141]. According to [142], more than 65% of the organizations in a survey in 2014 have experienced more APT attacks in their IT networks than last year.

The FlipIt game proposed in the seminal work [143] formulates the stealthy and continuous APT attacks and designs the scan interval to detect APTs on a given cyber system. The game theoretic study in [144] has provided insights to design the optimal scan intervals of a cyber system against APTs. Prospect theory has been applied in [145] to investigate the probability distortion of an APT attacker against cloud storage and cumulative prospect theory has been used in [146] to model the frame effect of an APT attacker to choose the attack interval. Most existing APT games ignore the strict resource constraints in the APT defense, such as the limited number of Central Processing Units (CPUs) of a storage defender and an APT attacker [143, 147]. However, a cloud storage system with limited number of CPUs cannot scan all the data stored on the storage devices in a given time slot. To this end, encryptions and authentication techniques are applied to protect data privacy of cloud storage systems. On the other hand, an APT attacker with limited CPU resources cannot install malware to steal all the data on the cloud storage system in a single time slot either [148, 149].

It is challenging for a cloud storage system to optimize the CPU allocation to scan the storage devices under a large number of CPUs and storage devices without being aware of the APT attack strategy. Therefore, we use the Colonel Blotto game (CBG), a two-player zero-sum game with multiple battlefields to model the competition between an APT attacker and a storage defender, each with a limited total number of CPUs over a given number of storage devices. The player who applies more resources on a battlefield in a Colonel Blotto game wins it, and the overall payoff of a player in the game is proportional to the number of the winning battlefields [150]. The Colonel Blotto game has been recently applied to design the spectrum allocation of network service providers [89], the jamming resistance methods for the Internet of Things [151, 152].

In our earlier work [153], we assumed that each storage device has the same amount of data and addresses APT attackers that do not change their attack policies. However, the storage devices usually have different amount of the data with different priority levels, and the data size and their priority level also change over time. By allocating more CPUs to scan the storage devices with more data, a storage defender can achieve a higher data protection level. Therefore, here, we extend the scenario to a dynamic cloud storage system whose data size changes over time and addresses smart APTs, in which an attacker first infers the security mechanism applied by the cloud storage system, and then deliberately induces the storage defender to use a specific defense strategy, if the defender applies a learning-based APT detection scheme. For example, an attacker keeps sending malware to a specific storage device for a long period of time to exhaust most CPU resources and then suddenly attacks another device in the cloud storage system.

By applying time sharing (or division), a defender can use a single CPU to scan multiple storage devices as battlefields to detect APTs in a time slot, and an attacker can use a single CPU to attack multiple devices with a single CPU yielding a roughly continuous CBG. According to [150], a pure-strategy Colonel Blotto game rarely achieves Nash equilibria (NEs). Therefore, we focus on the CBG with mixed strategies, in which both players choose their CPU allocation distribution and introduce randomness in their action selection to fool their opponent. The conditions under which the NEs exist in the CPU allocation game are provided to disclose how the number of storage devices, the size of the data stored in each storage device and the total number of CPUs in which the defender observes the impact on the data protection level and the utility of the cloud storage system against APTs.

The CBG-based CPU allocation game provides a framework to understand the strategic behavior of both sides, and the NE strategy relies on the detailed prior knowledge about the APT attack model. In particular, the cloud defender has to know the total number of the attack CPUs and the attack policy over the storage devices, which is challenging to accurately being estimated in a dynamic storage system. On the other hand, the repeated interactions between the APT attacker and the defender over multiple time slots can be formulated as a dynamic CPU allocation game, and the defender can choose the security strategy according to the attack history. The APT defense decisions in the dynamic CPU allocation game can be approximately formulated as a Markov decision process (MDP) with finite states, in which the defender observes the state that consists of the previous attack CPU allocation and the current data storage distribution. Therefore, a defender can apply reinforcement learning (RL) techniques such as Q-learning to achieve the optimal CPU allocation over the storage devices to detect APTs in a dynamic game.

The policy hill-climbing (PHC) algorithm as an extension of Q-learning in the mixed-strategy game [154] enables an agent to achieve the optimal strategy without being aware of the underlying system model. For instance, the PHC-based CPU allocation scheme in [153] enables the defender to protect the storage devices with a limited number of CPUs without being aware of the APT attack model. In this work, a "hotbooting" technique as a type of transfer learning [155] exploits the APT defense experiences in similar scenarios to save the random explorations at the initial stage of the dynamic APT defense game, and thus accelerate the learning speed. We propose a hotbooting PHC-based CPU allocation scheme that chooses the number of the CPUs on each storage device based on the current state and the quality or Q-function that is initialized according to the APT detection experiences to reduce the exploration time at the initial learning stage.

We apply deep Q-network (DQN)[1], a deep reinforcement learning technique recently developed by Google DeepMind in [156, 157] to accelerate the learning speed of the defender for the case with a large number of storage devices and defense CPUs. More specifically, the DQN-based CPU allocation exploits the deep convolutional neural network (CNN) to determine the Q-value for each CPU allocation and thus suppress the state space observed by the cloud storage defender. Simulation results demonstrate that this scheme can improve the data protection level, increase the APT attack cost, and enhance the utility of the cloud storage system against APTs.

4.2 Related work

The seminal work in [143] formulates a stealthy takeover game between an APT attacker and a defender, who compete to control a targeted cloud storage system. The APT scan interval on a single device has been optimized in [158] based on the FlipIt model without considering the constraint of scanning CPUs. The game between an overt defender and a stealthy attacker as investigated in [159] provides the best response to the periodic detection strategy against a non-adaptive attacker. The online learning algorithm as developed in [160] achieves the optimal timing of the security updates in the FlipIt game and reduces the regret of the upper confidence bound compared with the periodic defense strategy. The APT defense game formulated in [148] extends the FlipIt game in [143] to multi-node systems with limited resources. The game among an APT attacker, a cloud defender and a mobile device as formulated in [161] combines the APT defense game in [143] with the signaling game between the cloud and the mobile device. The evolutionary game can capture the long-term continuous behavior of APTs on cloud storage [162]. The information-trading and APT defense game formulated in [163] analyzes the joint APT and insider attacks. The subjective view of APT attackers under uncertainty scanning duration was analyzed in [145] based on prospect theory.

[1] The section 4.6 is part of the reference [88] which I am co-author. I don't feel getting credit for this section and most of the credit will go to Minghui Min, Liang Xiao, Caixia Xie ,and Narayan B. Mandayam. This part is included to this thesis to keep the section 4.7 consistent.

Colonel Blotto game models the competition between two players each with resource constraints. For example, the Colonel Blotto game with mixed-strategy as formulated in [89] studies the spectrum allocation of network service providers, yielding a fictitious play based allocation approach to compute the equilibrium of the game with discrete spectrum resources. The anti-jamming communication game as developed in [164] optimizes the transmit power over multiple channels in cognitive radio networks based on the NE of the CBG. The CBG-based jamming game as formulated in [151] shows that neither the defender nor the attacker can dominate with limited computational resources. The CBG-based jamming game as formulated in [152] shows how the number of subcarriers impacts the anti-jamming performance of Internet of Things with continuous and asymmetric radio power resources. The CBG-based phishing game is formulated in [165] and investigates the dynamics of the detect-and-takedown defense against phishing attacks.

There are some other APT detection methods rather than game theoretic approaches. The APT detection scheme in [166] applies a digital signature and anomaly analysis to detect APTs for cyber systems. The classification-based APT detection scheme as proposed in [167] uses the feature vector of TCP/IP session information to minimize the APT damage. The APT detection in [168] uses log-lines, such as search patterns, event classes and rules to detect APTs in networks.

Reinforcement learning techniques have been used to improve network security. For instance, the minimax-Q learning based spectrum allocation as developed in [169] increases the spectrum efficiency in cognitive radio networks. The DQN-based anti-jamming communication scheme as designed in [170] applies DQN to choose the transmit channel and node mobility and can increase the signal-to-interference-plus-noise ratio of the secondary users against cooperative jamming in cognitive radio networks. The PHC-based CPU allocation scheme as proposed in [153] applies PHC to improve the data protection level of the cloud storage system against APTs. Compared with [153], this work improves the game model by incorporating the time-variant data storage model. Both the hotbooting technique and DQN are applied to accelerate the learning speed and thus improve the security performance for the case with a large number of storage devices and CPUs against the smart APT attacks in the dynamic cloud storage system.

4.3 System Model

As illustrated in Fig. 4.1, the cloud storage system consists of D storage devices, where device i stores data of size $B_i^{(k)}$ at time k, with $1 \leq i \leq D$. Let $\mathbf{B}^{(k)} = \left[B_i^{(k)}\right]_{1\leq i\leq D}$ be the data size vector of the cloud storage system, and $\widehat{B}^{(k)} = \sum_{i=1}^{D} B_i^{(k)}$ denote the total amount of the data stored in the cloud storage system at time k.

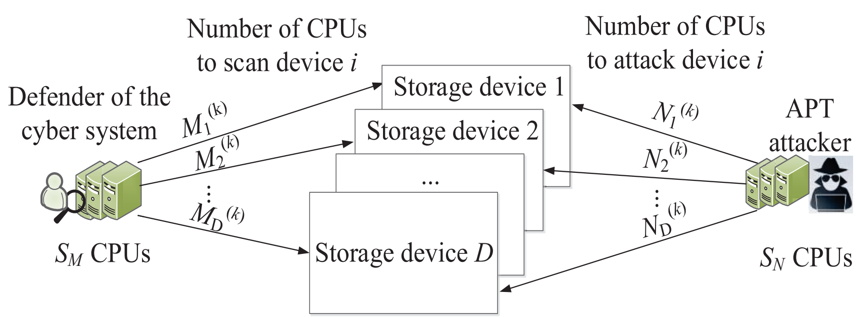


Figure 4.1: CPU allocation game, in which a defender with S_M CPUs chooses the CPU allocation strategy to scan the D storage devices in the cloud storage system against an APT attacker with S_N CPUs.

In this work, we consider an APT attacker who combines multiple attack methods, tools, and techniques such as one mentioned in [141] to steal data from the targeted cloud storage system over a long period of time. The attacker aims to steal as much as possible data from the D storage devices with S_N CPUs without being detected. At time k, $N_i^{(k)}$ out of the S_N CPUs are used to attack storage device i, with $\sum_{i=1}^{D} N_i^{(k)} \leq S_N$. The attack CPU allocation at time k is given by $\mathbf{N}^{(k)} = \left[N_i^{(k)}\right]_{1\leq i\leq D} \in \triangle_A$, where the attack action set $\triangle_A$ is given by

$$\triangle_A = \left\{ [N_i]_{1\leq i\leq D} \,\middle|\, 0 \leq N_i \leq S_N;\ \sum_{i=1}^{D} N_i \leq S_N \right\}. \tag{4.1}$$

The defender uses S_M CPUs to scan the D storage devices in the cloud storage system and aims to detect APTs as early as possible to minimize the total amount of the data stolen by the attacker. At time k, $M_i^{(k)}$ out of the S_M CPUs are allocated to scan the device i for APTs, with $\sum_{i=1}^{D} M_i^{(k)} \leq S_M$. As each time slot is quite short, the storage defender can not scan all the data stored in the D storage devices in a single time slot. The defense CPU allocation vector denoted by $\mathbf{M}^k$ is defined as $\mathbf{M}^{(k)} = \left[M_i^{(k)}\right]_{1\leq i\leq D} \in \triangle_D$, where the defense action set $\triangle_D$ is given by

$$\triangle_D = \left\{ [M_i]_{1\leq i\leq D} \,\middle|\, 0 \leq M_i \leq S_M;\ \sum_{i=1}^{D} M_i \leq S_M \right\}. \tag{4.2}$$

If the attacker uses more CPUs than the defender in the APT defense game, the data stored in the storage device are assumed to be at risk. More specifically, the data stored in the storage device i are assumed to be safe if the number of the defense CPUs is greater than the number of attack CPUs at that time, i.e., $M_i^{(k)} > N_i^{(k)}$, and the data are at risks if $M_i^{(k)} < N_i^{(k)}$. If $M_i^{(k)} = N_i^{(k)}$, both players have an equal opportunity to control the storage device. Let $\mathrm{sgn}(x)$ denote a sign function, with $\mathrm{sgn}(x) = 1$ if $x > 0$, $\mathrm{sgn}(x) = -1$ if $x < 0$, and 0 otherwise. Therefore, the data protection level of the cloud storage system at time k denoted by $R^{(k)}$ is defined as the normalized size of the "safe" data that are protected by the defender and is given by

$$R^{(k)} = \frac{1}{\widehat{B}^{(k)}} \sum_{i=1}^{D} B_i^{(k)} \mathrm{sgn}\left(M_i - N_i\right). \tag{4.3}$$

For ease of reference, our commonly used notations are summarized in Table 4.1. The time index k in the superscript is omitted if no confusion occurs.

4.4 CBG-Based CPU Allocation Game

Colonel Blotto game is a powerful tool to study the strategic resource allocation of two agents each with limited resources in a competitive environment. Therefore, the interactions between the APT attacker and the defender of the cloud storage system regarding their CPU allocations can be formulated as a Colonel Blotto game with D battlefields. By applying the time sharing (or division) technique, the defender (or attacker) can scan (or attack) multiple storage devices with a single CPU in a time slot, which can be approximately formulated as a continuous CBG. In this game, the

Table 4.1: Summary of symbols and notations

D	Number of storage devices
$S_{M/N}$	Total number of defense/attack CPUs
$\mathbf{M}^{(k)}/\mathbf{N}^{(k)}$	Defense/attack CPU allocation vector
$\triangle_{D/A}$	Action set of the defender/attacker
$u_{D/A}^{(k)}$	Utility of the defender/attacker at time k
$\widehat{B}^{(k)}$	Total size of the stored data at time k
$\mathbf{B}^{(k)}$	Data size vector of D devices at time k
$R^{(k)}$	Data protection level at time k

defender chooses the defense CPU allocation vector $\mathbf{M}^{(k)} \in \triangle_D$ to scan the D devices at time k, while the APT attacker chooses the attack CPU allocation $\mathbf{N}^{(k)} \in \triangle_A$.

The utility of the defender (or the attacker) at time k denoted by $u_D^{(k)}$ (or $u_A^{(k)}$) depends on the size of the data stored in the D devices, and the data protection level of each device at the time. In the zero-sum game, by (4.3) the utility of the defender is set as

$$\begin{aligned} u_D^{(k)}(\mathbf{M},\mathbf{N}) &= -u_A^{(k)}(\mathbf{M},\mathbf{N}) \\ &= \sum_{i=1}^{D} B_i^{(k)} \operatorname{sgn}(M_i - N_i). \end{aligned} \tag{4.4}$$

The CBG-based CPU allocation game rarely has a pure-strategy NE, because the attack CPU allocation $\mathbf{N}^{(k)}$ can be chosen according to the defense CPU allocation $\mathbf{M}^{(k)}$ and to defeat it for a higher utility $u_A^{(k)}$. Therefore, we study the CPU allocation game with mixed strategies, in which each player randomizes the CPU allocation strategies to fool the opponent.

In the mixed-strategy CPU allocation game, the defense strategy at time k denoted by $x_{i,j}^{(k)}$ is the probability that the defender allocates j CPUs to scan device i, i.e., $x_{i,j}^{(k)} = \Pr\left(M_i^{(k)} = j\right)$. Let $p_{m,j} \in [0,1]$ be the m-th highest feasible value of $x_{i,j}^{(k)}$. The mixed-strategy defense action set denoted by P_M is given by

$$\mathrm{P}_M = \left\{ [p_{m,j}]_{1\le m\le D,\ 0\le j\le S_M} \middle| p_{m,j} \ge 0,\ \forall j; \sum_{j=0}^{S_M} p_{m,j} = 1,\ \forall m \right\}. \tag{4.5}$$

The defense mixed strategy vector denoted by $\mathbf{x}^{(k)}$ is given by

$$\mathbf{x}^{(k)} = \left[x_{i,j}^{(k)}\right]_{1\le i\le D,\ 0\le j\le S_M} \in \mathrm{P}_M. \tag{4.6}$$

Similarly, let $y_{i,j}^{(k)}$ denote the probability that $N_i^{(k)}$ CPUs are used to attack device i, i.e., $y_{i,j}^{(k)} = \Pr\left(N_i^{(k)} = j\right)$, and $q_{m,j} \in [0,1]$ be the m-th highest feasible value of $y_{i,j}^{(k)}$. The action set of the attacker in the mixed-strategy game denoted by P_N is given by

$$\mathrm{P}_N = \left\{ [q_{m,j}]_{1\le m\le D, 0\le j\le S_N} \middle| q_{m,j} \ge 0,\ \forall j; \sum_{j=0}^{S_N} q_{m,j} = 1,\ \forall m \right\}. \tag{4.7}$$

The attacker chooses the CPU allocation strategy in this game denoted by $\mathbf{y}^{(k)}$ with

$$\mathbf{y}^{(k)} = \left[y_{i,j}^{(k)}\right]_{1\le i\le D,\ 0\le j\le S_N} \in \mathrm{P}_N. \tag{4.8}$$

The expected utility of the defender (or the attacker) averaged over all the feasible defense (or attack) strategies is denoted by $U_D^{(k)}$ (or $U_A^{(k)}$) and given by (4.4) as

$$\begin{aligned} U_D^{(k)}(\mathbf{x},\mathbf{y}) &= -U_A^{(k)}(\mathbf{x},\mathbf{y}) \\ &= E_{\substack{\mathbf{M}\sim\mathbf{x}\\ \mathbf{N}\sim\mathbf{y}}} \left(\sum_{i=1}^{D} B_i^{(k)} \mathrm{sgn}\left(M_i - N_i\right) \right). \end{aligned} \tag{4.9}$$

The NE of the CBG-based CPU allocation game with mixed strategies denoted by $(\mathbf{x}^*, \mathbf{y}^*)$ provides the best-response policy, i.e., no player can increase his or her utility by unilaterally changing from the NE strategy. For example, if the defender chooses the CPU allocation strategy $\mathbf{x}^*$, the APT attacker cannot do better than selecting $\mathbf{y}^*$ to attack the D storage devices. By definition, we have

$$U_D(\mathbf{x}^*, \mathbf{y}^*) \ge U_D(\mathbf{x}, \mathbf{y}^*), \quad \forall\ \mathbf{x} \in \mathrm{P}_M \tag{4.10}$$

$$U_A(\mathbf{x}^*, \mathbf{y}^*) \ge U_A(\mathbf{x}^*, \mathbf{y}), \quad \forall\ \mathbf{y} \in \mathrm{P}_N. \tag{4.11}$$

We first consider a CBG-based CPU allocation game $\mathbb{G}_1$ with symmetric CPU resources, $S_M = S_N$, i.e., the defender and the attacker have the same amount of computational resources. Let $\mathbf{1}_{m\times n}$ (or $\mathbf{0}_{m\times n}$) be an all-1 (or 0) $m \times n$ matrix, $\lfloor\ \rfloor$ be the lower floor function, and the normalized defense CPUs $\beta = 2S_M/\widehat{B}$.

Theorem 4.1. *If $S_M = S_N$ and $B_i < \sum_{1\le h\ne i\le D} B_h$, the CPU allocation game $\mathbb{G}_1$ has a NE $(\boldsymbol{x}^*, \boldsymbol{x}^*)$ given by*

$$\boldsymbol{x}^* = \begin{bmatrix} \frac{1}{\lfloor\beta B_1\rfloor+1}\mathbf{1}_{1\times(\lfloor\beta B_1\rfloor+1)} & \mathbf{0}_{1\times(S_M-\lfloor\beta B_1\rfloor)} \\ \frac{1}{\lfloor\beta B_2\rfloor+1}\mathbf{1}_{1\times(\lfloor\beta B_2\rfloor+1)} & \mathbf{0}_{1\times(S_M-\lfloor\beta B_2\rfloor)} \\ \vdots & \vdots \\ \frac{1}{\lfloor\beta B_D\rfloor+1}\mathbf{1}_{1\times(\lfloor\beta B_D\rfloor+1)} & \mathbf{0}_{1\times(S_M-\lfloor\beta B_D\rfloor)} \end{bmatrix}. \tag{4.12}$$

Proof. The CPU allocation game $\mathbb{G}_1$ can be formulated as a CBG with symmetric players on D battlefields. The resource budget of the defender is S_M, the value of the i-th battlefield is B_i, and the total value of D battlefields is $\widehat{B}$. Let $\mathcal{U}(m, n)$ denote the uniform distribution between m and n. By Proposition 1 in [171], the mixed-strategy CBG game has an NE given by $(\mathbf{x}^*, \mathbf{x}^*)$, where $\mathbf{x}^*$ is the probability distribution of $\mathbf{M}$, and each vector coordinate M_i is uniform distribution between 0 and $2S_M B_i/\widehat{B}$. Therefore, the CPU allocation of the i-th device M_i^* is uniformly distributed on $[0, 2S_M B_i/\widehat{B}]$, i.e.,

$$M_i^* \sim \mathcal{U}\left(\left\{0, 1, 2, ..., \left\lfloor \frac{2S_M B_i}{\widehat{B}} \right\rfloor\right\}\right). \tag{4.13}$$

Thus this game has an NE given by $(\mathbf{x}^*, \mathbf{x}^*)$, where $\forall\ 0 \le j \le \lfloor 2S_M B_i \widehat{B}\rfloor$, $0 \le i \le D$, each element of $\mathbf{x}$ is given by

$$x_{i,j}^* = \Pr\left(M_i^* = j\right) = \frac{1}{\lfloor 2S_M B_i/\widehat{B}\rfloor + 1}, \tag{4.14}$$

which results in (4.12). □

Corollary 4.2. *At the NE of the symmetric CPU allocation game $\mathbb{G}_1$, the expected data protection level is zero and the utility of the defender is zero.*

Proof. By (4.3) and (4.12), the data protection level over all the realizations of the mixed-strategy NE $(\mathbf{x}^*, \mathbf{x}^*)$ is given by

$$E_{\mathbf{x}^*}(R) = E_{\mathbf{x}^*}\left(\frac{1}{\widehat{B}}\sum_{i=1}^{D} B_i \mathrm{sgn}\left(M_i^* - N_i^*\right)\right) \tag{4.15}$$

$$= \frac{1}{\widehat{B}}\sum_{i=1}^{D} B_i \Big(\Pr\left(N_i^* < M_i^*\right) - \Pr\left(N_i^* > M_i^*\right)\Big) = 0. \tag{4.16}$$

Similarly, by (4.4) and (4.9) , we have $U_D = U_A = 0$. □

Remark: If the APT attacker and the defender have the same number of CPUs and no storage device dominates in the game (i.e., $B_i < \sum_{1 \le h \ne i \le D} B_h$, $\forall i$), both players choose a number from $\{0, 1, ..., \lfloor 2S_M B_i/\widehat{B} \rfloor\}$ to attack or scan the storage device i with probability $1/\Big(\lfloor 2S_M B_i/\widehat{B} \rfloor + 1\Big)$ by (4.14). The data protection level R by definition ranges between -1 and 1. Therefore, the game makes a tie, yielding zero expected data protection level and zero utility of the defender.

We next consider a CBG-based CPU allocation game with asymmetric players denoted by $\mathbb{G}_2$, in which the attacker and the defender have different number of CPUs and compete over D storage devices with an equal data size, i.e., $B_i = B$, $\forall i$.

Theorem 4.3. *If $2/D \le S_N/S_M \le 1$, $D \ge 3$ and $B_i = B$, $\forall\ 1 \le i \le D$, the NE of the CPU allocation game $\mathbb{G}_2$ $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is given by*

$$\boldsymbol{x}^* = \left[\mathbf{0}_{D\times 1} \quad \frac{1}{\left\lfloor \frac{2S_M}{D} \right\rfloor}\mathbf{1}_{D\times \left\lfloor \frac{2S_M}{D} \right\rfloor} \quad \mathbf{0}_{D\times \left(S_M - \left\lfloor \frac{2S_M}{D} \right\rfloor\right)}\right] \tag{4.17}$$

$$\boldsymbol{y}^* = \left[\left(1 - \frac{S_N}{S_M}\right)\mathbf{1}_{D\times 1} \quad \left(\frac{S_N}{S_M \left\lfloor \frac{2S_M}{D} \right\rfloor}\right)\mathbf{1}_{D\times \left\lfloor \frac{2S_M}{D} \right\rfloor}\right.$$

$$\left.\mathbf{0}_{D\times \left(S_M - \left\lfloor \frac{2S_M}{D} \right\rfloor\right)}\right]. \tag{4.18}$$

Proof. The CPU allocation game $\mathbb{G}_2$ can be formulated as a CBG with asymmetric players on D battlefields, where the defender (or attacker) chooses the probability density functions $\mathbf{x}$ (or $\mathbf{y}$) according to S_M (or S_N) resource budget, and the resources allocated to the i-th battlefield is M_i (or N_i).

By Theorem 2 in [150], the unique Nash equilibrium for the defender and the attacker with $2/D \le S_N/S_M \le 1$ is given by

$$\mathbf{x}(M_i^*) \sim \mathcal{U}\left(\left[0, \frac{2S_M}{D}\right]\right) \tag{4.19}$$

$$\mathbf{y}(N_i^*) \sim \left(1 - \frac{S_N}{S_M}\right)\delta(N_i^*) + \frac{S_N}{S_M}\mathcal{U}\left(\left[0, \frac{2S_M}{D}\right]\right). \tag{4.20}$$

Therefore, the CPU allocation of the i-th storage device M_i^* on NE is uniformly distributed on $[0, 2S_M/D]$, i.e.,

$$M_i^* \sim \mathcal{U}\left(\left\{0, 1, 2, ..., \left\lfloor \frac{2S_M}{D} \right\rfloor\right\}\right). \tag{4.21}$$

Thus, the NE strategy of the CPU allocation game $\mathbb{G}_2$ is given by

$$x_{i,j}^* = \Pr\left(M_i^* = j\right) = \frac{1}{\left\lfloor \frac{2S_M}{D} \right\rfloor}, \ \forall\ 1 \le j \le \left\lfloor \frac{2S_M}{D} \right\rfloor. \tag{4.22}$$

Thus, we have (4.17).

Similarly, we have

$$N_i^* \sim \left(1 - \frac{S_N}{S_M}\right)\delta(N_i^*) + \frac{S_N}{S_M}\mathcal{U}\left(\left\{0, 1, 2, ..., \left\lfloor \frac{2S_M}{D} \right\rfloor\right\}\right), \tag{4.23}$$

and thus

$$y_{i,j}^* = \Pr\left(N_i^* = j\right) = \begin{cases} 1 - \frac{S_N}{S_M}, & \text{if } j = 0 \\ \frac{S_N}{S_M\left\lfloor \frac{2S_M}{D} \right\rfloor}, & \text{if } 1 \le j \le \left\lfloor \frac{2S_M}{D} \right\rfloor. \end{cases} \tag{4.24}$$

Thus, we have (4.18). □

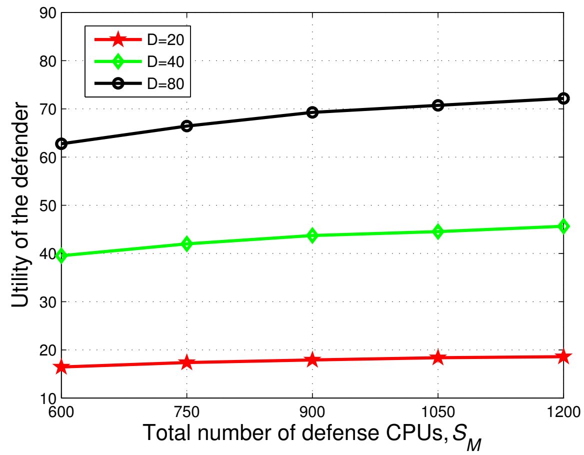
Corollary 4.4. *At the NE of the CPU allocation game $\mathbb{G}_2$, the expected data protection level is $1 - S_N/S_M$ and*

$$U_D = -U_A = \left(1 - \frac{S_N}{S_M}\right)\widehat{B}. \tag{4.25}$$

Proof. According to (4.3), (4.15), (4.17) and (4.18), as $B_i = B$, we have

$$\begin{aligned} E_{\mathbf{x}^*,\mathbf{y}^*}(R) &= \frac{1}{\widehat{B}}\sum_{i=1}^{D} B_i \left(\Pr\left(N_i^* < M_i^*\right) - \Pr\left(N_i^* > M_i^*\right)\right) \\ &= \frac{1}{\widehat{B}}\sum_{i=1}^{D} B_i \Big(\Pr\left(M_i^* > 0\right) + \Pr\left(N_i^* < M_i^* \middle| N_i^* \neq 0\right) \\ &\quad - \Pr\left(N_i^* > M_i^* \middle| N_i^* \neq 0\right)\Big) \\ &= \frac{1}{\widehat{B}}\sum_{i=1}^{D} B_i \left(1 - \frac{S_N}{S_M}\right) = 1 - \frac{S_N}{S_M}. \end{aligned} \tag{4.26}$$

Similarly, by (4.4) and (4.9), we have (4.25). □



Utility of the defender

Figure 4.2: APT defense performance of the CBG-based CPU allocation game $\mathbb{G}_2$ at the NE with D storage devices and S_M defense CPUs against an APT attacker with 150 CPUs.

Remark: The defender has to use more CPUs than the APT attackers to protect the data privacy of the cloud storage system. Therefore, a subset of the storage devices is safe from the attacker who has to match the defender on the other storage devices. In this case, the defender wins the game, and the utility increases with the total data size. The expected data protection level as shown in (4.26) increases with the resource advantage of the defender over the attacker, i.e., S_M/S_N.

The APT defense performance of the CPU allocation game $\mathbb{G}_2$ at the NE is presented in Fig. 4.2, in which the D storage devices are threatened by an APT attacker with 150 attack CPUs. If the defender uses 1200 CPUs instead of 600 CPUs to scan the 20 devices, the utility of the defender increases by 18.75%. The APT defense performance of the CBG game at the NE provides the optimal defense performance with known APT attack model and defense model and can be used as a guideline to design the CPU allocation scheme.

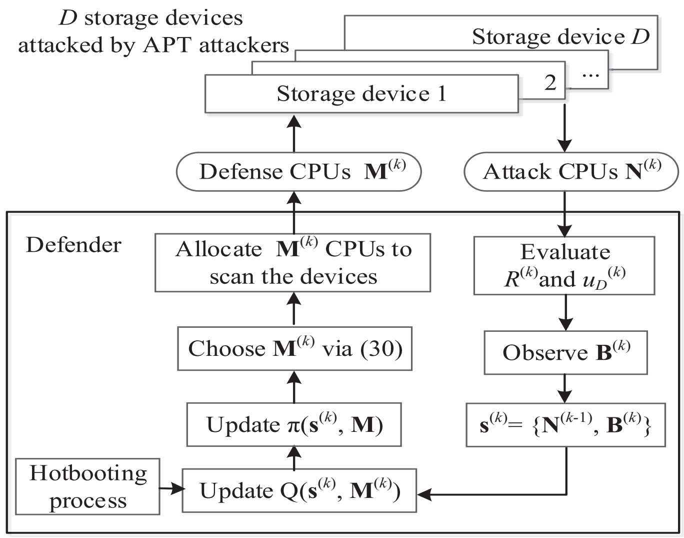


Figure 4.3: Illustration of the hotbooting PHC-based defense CPU allocation.

4.5 Hotbooting PHC-based CPU Allocation

Since the defender is usually unaware of the attack policy, we propose a hotbooting PHC-based CPU allocation scheme to scan D storage devices in the dynamic APT detection game, as illustrated in Fig. 4.3. At each time slot, the defender observes the amount of the data stored in each storage device $B_i^{(k)}$ and evaluates the previous APT detection results to estimate the last attack CPU allocation profile $\mathbf{N}^{(k-1)}$. The state $\mathbf{s}^{(k)}$ is chosen as $\mathbf{s}^{(k)} = \{\mathbf{N}^{(k-1)}, \mathbf{B}^{(k)}\}$, which is the basis to select the defense policy $\mathbf{M}^{(k)}$.

The Q-function for each action-state pair denoted by $Q(\mathbf{s}, \mathbf{M})$ is the expected discounted long-term reward of the defender, and is updated in each time slot according to the iterative Bellman equation as follows:

$$\begin{aligned} Q\left(\mathbf{s}^{(k)}, \mathbf{M}^{(k)}\right) \leftarrow & (1-\alpha) Q\left(\mathbf{s}^{(k)}, \mathbf{M}^{(k)}\right) \\ & + \alpha\left(u_D^{(k)} + \gamma V\left(\mathbf{s}'\right)\right), \end{aligned} \tag{4.27}$$

where the learning rate $\alpha \in (0, 1]$ is the weight of the current experience, the discount factor $\gamma \in [0, 1]$ indicates the uncertainty of the defender on the future reward, $\mathbf{s}'$ is the

Algorithm 1 CPU allocation with hotbooting PHC

Hotbooting defense process in Algorithm 2
Initialize α, γ, δ, $\mathbf{N}^{(0)}$ and $\mathbf{B}^{(1)}$
Set $\mathbf{Q} = \overline{\mathbf{Q}}$, $\pi = \overline{\pi}$
for $k = 1, 2, 3, ...$ **do**
    Observe the current data size $\mathbf{B}^{(k)}$
    $\mathbf{s}^{(k)} = \{\mathbf{N}^{(k-1)}, \mathbf{B}^{(k)}\}$
    Choose $\mathbf{M}^{(k)} \in \triangle_D$ with $\pi(\mathbf{s}^{(k)}, \mathbf{M})$ via (4.30)
    for $i = 1, 2, ...D$ **do**
        Allocate $M_i^{(k)}$ CPUs to scan storage device i
    end for
    Observe the compromised storage devices and estimate $\mathbf{N}^{(k)}$
    Obtain $u_D^{(k)}$ via (4.4)
    Update $Q(\mathbf{s}^{(k)}, \mathbf{M}^{(k)})$ via (4.27)
    Update $V(\mathbf{s}^{(k)})$ via (4.28)
    Update $\pi(\mathbf{s}^{(k)}, \mathbf{M})$ via (4.29)
end for

next state if the defender uses $\mathbf{M}^{(k)}$ CPUs at state $\mathbf{s}^{(k)}$, and the value function $V(\mathbf{s})$ maximizes $Q(\mathbf{s}, \mathbf{M})$ over the action set given by

$$V\left(\mathbf{s}^{(k)}\right) = \max_{\mathbf{M}' \in \triangle_D} Q\left(\mathbf{s}^{(k)}, \mathbf{M}'\right). \tag{4.28}$$

The mixed-strategy table of the defender denoted by $\pi(\mathbf{s}, \mathbf{M})$ provides the distribution of the number of CPUs $\mathbf{M}$ over the D storage devices under state $\mathbf{s}$ and is updated via

$$\begin{aligned} \pi(\mathbf{s}^{(k)}, \mathbf{M}) &\leftarrow \pi(\mathbf{s}^{(k)}, \mathbf{M}) \\ &+ \begin{cases} \delta, & \text{if } \mathbf{M} = \arg\max\limits_{\mathbf{M}' \in \triangle_D} Q\left(\mathbf{s}^{(k)}, \mathbf{M}'\right) \\ \frac{\delta}{1-|\triangle_D|}, & \text{o.w.} \end{cases} \end{aligned} \tag{4.29}$$

In this way, the probability of the action that maximizes the Q-function increases by δ, with $0 < \delta \leq 1$, and the probability of other actions decrease by $\delta / (|\triangle_D| - 1)$. The

Algorithm 2 Hotbooting defense process

Initialize ξ, K, α, γ, δ, $\mathbf{N}^{(0)}$ and $\mathbf{B}^{(1)}$
Set $\mathbf{Q} = \mathbf{0}_{(|\triangle_A| \times L^D) \times |\triangle_D|}$, $\mathbf{V} = \mathbf{0}_{(|\triangle_A| \times L^D) \times 1}$, $\pi = \frac{\mathbf{1}}{|\triangle_D|}$
for $i = 1, 2, 3, ..., \xi$ **do**
    Emulate a similar CPU allocation scenario for the defender to scan storage devices
    for $k = 1, 2, ..., K$ **do**
        Observe the current data size $\mathbf{B}^{(k)}$
        $\mathbf{s}^{(k)} = \{\mathbf{N}^{(k-1)}, \mathbf{B}^{(k)}\}$
        Choose $\mathbf{M}^{(k)} \in \triangle_D$ via (4.30)
        for $j = 1, 2, ...D$ **do**
            Allocate $M_j^{(k)}$ CPUs to scan storage device j
        end for
        Observe the compromised storage devices and estimate $\mathbf{N}^{(k)}$
        Obtain $u_D^{(k)}$ via (4.4)
        Update $\mathbf{Q}$ and π via (4.27)-(4.29)
    end for
end for
Output $\overline{\mathbf{Q}} \leftarrow \mathbf{Q}$, $\overline{\pi} \leftarrow \pi$

defender then selects the number of CPUs $\mathbf{M}^{(k)} \in \triangle_D$ according to the mixed strategy $\pi\left(\mathbf{s}^{(k)}, \mathbf{M}\right)$, i.e.,

$$\Pr\left(\mathbf{M}^{(k)} = \widehat{\mathbf{M}}\right) = \pi\left(\mathbf{s}^{(k)}, \widehat{\mathbf{M}}\right), \ \forall\, \widehat{\mathbf{M}} \in \triangle_D. \tag{4.30}$$

We apply the hotbooting technique to initialize both the Q-value and the strategy table π with the CPU allocation experiences in similar environments. The hotbooting PHC-based CPU allocation saves random explorations at the beginning stage of the dynamic game and thus accelerates the learning speed. As shown in Algorithm 2, ξ CPU allocation experiences are performed before the game. Each experiment lasts K time slots, in which the defender chooses the number of CPUs to scan the D storage devices according to the mixed-strategy table $\pi\left(\mathbf{s}^{(k)}, \mathbf{M}\right)$. The defender observes the attack CPU distribution and evaluates the utility $u_D^{(k)}$. Both the Q-function and π are updated via (4.27)-(4.29) in each time in the experiences.

The Q-values as the output of the hotbootng process based on the ξ experiences denoted by $\overline{\mathbf{Q}}$ is used to initialize the Q-values in Algorithm 1. Similarly, the mixed-strategy table as the output of Algorithm 2 based on the ξ experiences denoted by $\overline{\pi}$ is used to initialize π in Algorithm 1. The learning time of Algorithm 1 increases with the dimension of the action-state space $|\triangle_D| \times |\triangle_A| \times L^D$, which increases with the number of storage devices in the cloud storage system and the number of CPUs, yielding serious performance degradation.

4.6 Hotbooting DQN-Based CPU Allocation

In this section, we propose a hotbooting DQN-based CPU allocation scheme to improve the APT defense performance of the cloud storage system. This scheme applies deep convolutional neural network, a deep reinforcement learning technique, to compress the action-state space and thus accelerate the learning process. As illustrated in Fig. 4.4, the deep convolution neural network is a nonlinear approximator of the Q-value for each action. The CNN architecture allows a compact storage of the learned information between similar states [172].

The DQN-based CPU allocation as summarized in Algorithm 3 extends the system state $\mathbf{s}^{(k)}$ as in Algorithm 1 to the experience sequence at time k denoted by $\boldsymbol{\varphi}^{(k)}$ to accelerate the learning speed and improve the APT resistance. More specifically, the experience sequence consists of the current system state $\mathbf{s}^{(k)}$ and the previous W system state-action pairs, i.e., $\boldsymbol{\varphi}^{(k)} = \left(\mathbf{s}^{(k-W)}, \mathbf{M}^{(k-W)}, ..., \mathbf{s}^{(k-1)}, \mathbf{M}^{(k-1)}, \mathbf{s}^{(k)}\right)$.

The experience sequence $\boldsymbol{\varphi}^{(k)}$ is reshaped into a 5×5 matrix and then input into the CNN, as shown in Fig. 4.4. The CNN consists of two convolutional (Conv) layers and two fully connected (FC) layers, with parameters chosen to achieve a good performance according to the experiment results as listed in Table 4.2. The filter weights of the four layers in the CNN at time k are denoted by $\theta^{(k)}$ for simplicity. The first Conv layer includes 20 different filters. Each filter has size 2×2 and uses stride 1. The output of

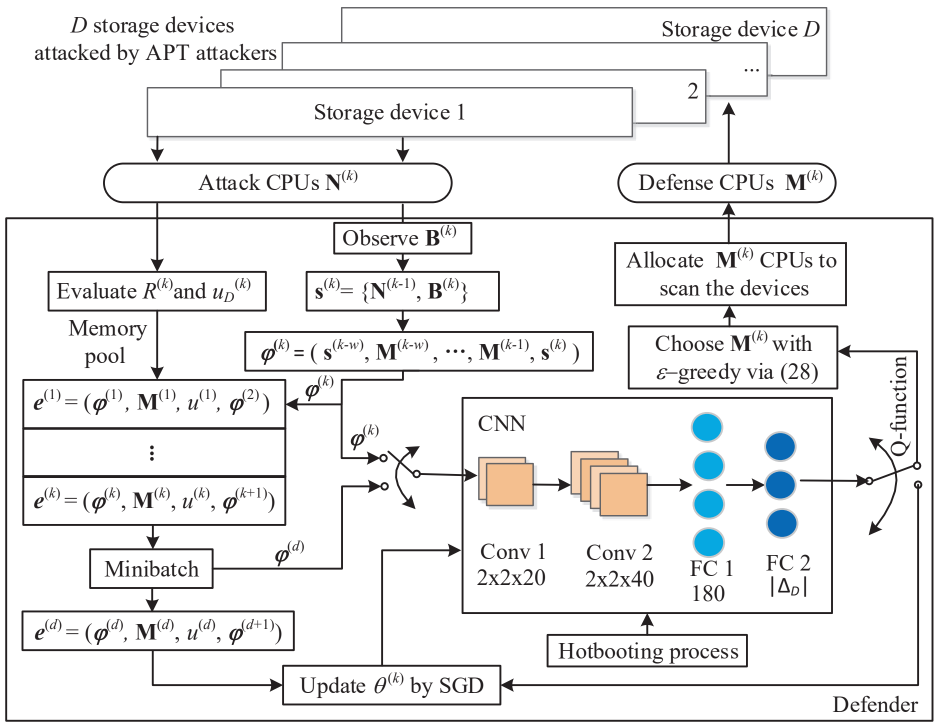


Figure 4.4: Hotbooting DQN-based defense CPU allocation.

the first Conv is 20 different 4×4 feature maps that are then passed through a rectified linear function (ReLU) as an activation function. The second Conv layer includes 40 different filters. Each filter has size 2×2 and stride 1. The outputs of the 2nd Conv layer are 40 different 3×3 feature maps, which are flattened to a 360-dimension vector and then will be sent to the two FC layers. The first FC layer involves 180 rectified linear units, and the second FC layer provides the Q-value for each CPU allocation policy $\mathbf{M} \in \triangle_D$ at the current system sequence $\boldsymbol{\varphi}^{(k)}$.

Table 4.2: CNN Parameters

Layer	Conv1	Conv2	FC1	FC2
Input	5×5	$4 \times 4 \times 20$	360	180
Filter size	2×2	2×2	/	/
Stride	1	1	/	/
# Filters	20	40	180	$\lvert\triangle_D\rvert$
Activation	ReLU	ReLU	ReLU	ReLU
Output	$4 \times 4 \times 20$	$3 \times 3 \times 40$	180	$\lvert\triangle_D\rvert$

Algorithm 3 Hotbooting DQN-based CPU allocation

Initialize $\mathbf{N}^{(0)}$, $\mathbf{B}^{(1)}$, W and H
Set $\theta = \overline{\theta}$, $\mathcal{D} = \emptyset$
for $k = 1, 2, 3, ...$ **do**
  Observe the current data size $\mathbf{B}^{(k)}$
  $\mathbf{s}^{(k)} = \{\mathbf{N}^{(k-1)}, \mathbf{B}^{(k)}\}$
  if $k \leq W$ **then**
    Choose $\mathbf{M}^{(k)} \in \triangle_D$ at random
  else
    $\boldsymbol{\varphi}^{(k)} = \left(\mathbf{s}^{(k-W)}, \mathbf{M}^{(k-W)}, ..., \mathbf{s}^{(k-1)}, \mathbf{M}^{(k-1)}, \mathbf{s}^{(k)}\right)$
    Set $\boldsymbol{\varphi}^{(k)}$ as the input of the CNN
    Observe the output of the CNN to obtain $Q\left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}\right)$
    Choose $\mathbf{M}^{(k)} \in \triangle_D$ via (4.32)
  end if
  for $i = 1, 2, ...D$ **do**
    Allocate $M_i^{(k)}$ CPUs to scan storage device i
  end for
  Observe the compromised storage devices and estimate $\mathbf{N}^{(k)}$
  Obtain $u_D^{(k)}$ via (4.4)
  Observe $\boldsymbol{\varphi}^{(k+1)}$
  $\mathcal{D} \leftarrow \mathcal{D} \bigcup \left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}^{(k)}, u_D^{(k)}, \boldsymbol{\varphi}^{(k+1)}\right)$
  for $d = 1, 2, 3, ..., H$ **do**
    Select $\left(\boldsymbol{\varphi}^{(d)}, \mathbf{M}^{(d)}, u_D^{(d)}, \boldsymbol{\varphi}^{(d+1)}\right) \in \mathcal{D}$ at random
    Calculate G via (4.34)
  end for
  Update $\theta^{(k)}$ via (4.35)
end for

The Q-function as the expected long-term reward for the state sequence $\boldsymbol{\varphi}$ and the action $\mathbf{M}$, is given by definition as

$$Q\left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}\right) = \mathbb{E}_{\boldsymbol{\varphi}'}\left[u_D^{(k)} + \gamma \max_{\mathbf{M}'} Q\left(\boldsymbol{\varphi}', \mathbf{M}'\right) | \boldsymbol{\varphi}^{(k)}, \mathbf{M}\right], \tag{4.31}$$

where $\boldsymbol{\varphi}'$ is the next state sequence by choosing defense CPU allocation $\mathbf{M}$ at state $\boldsymbol{\varphi}^{(k)}$.

To make a tradeoff between exploitation and exploration, the defense CPU allocation is chosen according to the ε-greedy policy [173]. More specifically, the CPU allocation $\mathbf{M}^{(k)}$ that maximizes the Q-function is chosen with a high probability $1 - \varepsilon$, and other actions are selected with a low probability to avoid staying in the local maximum, i.e.,

$$\Pr\left(\mathbf{M}^{(k)} = \widehat{\mathbf{M}}\right) = \begin{cases} 1-\varepsilon, & \widehat{\mathbf{M}} = \arg\max_{\mathbf{M}'} Q\left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}'\right) \\ \frac{\varepsilon}{|\triangle_D|-1}, & \text{o.w.} \end{cases} \tag{4.32}$$

Based on the experience replay as shown in Fig. 4.4, the CPU allocation experience at time k denoted by $\mathbf{e}^{(k)}$ is given by $\mathbf{e}^{(k)} = \left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}^{(k)}, u_D^{(k)}, \boldsymbol{\varphi}^{(k+1)}\right)$, and saved in the replay memory pool denoted by $\mathcal{D}$, with $\mathcal{D} = \left\{\mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(k)}\right\}$. An experience $\mathbf{e}^{(d)}$ is chosen from the memory pool at random, with $1 \le d \le k$. The CNN parameters $\theta^{(k)}$ are updated by the stochastic gradient descent (SGD) algorithm in which the mean-squared error between the network's output and the target optimal Q-value is minimized with the minibatch updates. The loss function denoted by L in the stochastic gradient descent algorithm is chosen as

$$L\left(\theta^{(k)}\right) = \mathbb{E}_{\boldsymbol{\varphi}, \mathbf{M}, u_D^{(k)}, \boldsymbol{\varphi}'}\left\{\left(G - Q\left(\boldsymbol{\varphi}, \mathbf{M}; \theta^{(k)}\right)\right)^2\right\}, \tag{4.33}$$

where the target value denoted by G approximates the optimal value $u_D^{(k)} + \gamma Q^*\left(\boldsymbol{\varphi}', \mathbf{M}'\right)$ based on the previous CNN parameters $\theta^{(k-1)}$, and is given by

$$G = u_D^{(k)} + \gamma \max_{\mathbf{M}'} Q\left(\boldsymbol{\varphi}', \mathbf{M}'; \theta^{(k-1)}\right). \tag{4.34}$$

The gradient of the loss function with respect to the weights $\theta^{(k)}$ is given by

$$\begin{aligned} \nabla_{\theta^{(k)}} L\left(\theta^{(k)}\right) &= \mathbb{E}_{\boldsymbol{\varphi}, \mathbf{M}, u_D, \boldsymbol{\varphi}'}\left[G \nabla_{\theta^{(k)}} Q\left(\boldsymbol{\varphi}, \mathbf{M}; \theta^{(k)}\right)\right] \\ &- \mathbb{E}_{\boldsymbol{\varphi}, \mathbf{M}}\left[Q\left(\boldsymbol{\varphi}, \mathbf{M}; \theta^{(k)}\right) \nabla_{\theta^{(k)}} Q\left(\boldsymbol{\varphi}, \mathbf{M}; \theta^{(k)}\right)\right]. \end{aligned} \tag{4.35}$$

This process repeats H times to update $\theta^{(k)}$ in Algorithm 3.

Similar to Algorithm 1, we apply the hotbooting technique to initialize the CNN parameters in the DQN-based CPU allocation rather than initializing them randomly to accelerate the learning speed. As shown in Algorithm 4, the defender stores the emulational experience $\left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}^{(k)}, u_D^{(k)}, \boldsymbol{\varphi}^{(k+1)}\right)$ in the database $\mathbb{E}$ and the resulting $\overline{\theta}$ based on ξ experiences are used to set θ as shown in Algorithm 3.

4.7 Simulation Results

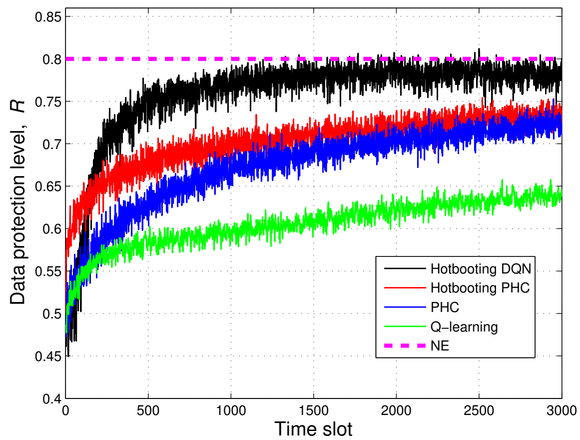
Simulations have been performed to evaluate the APT defense performance of the CPU allocation schemes in a cloud storage system, with the CNN parameters as listed in

Algorithm 4 Hotbooting process for Algorithm 3

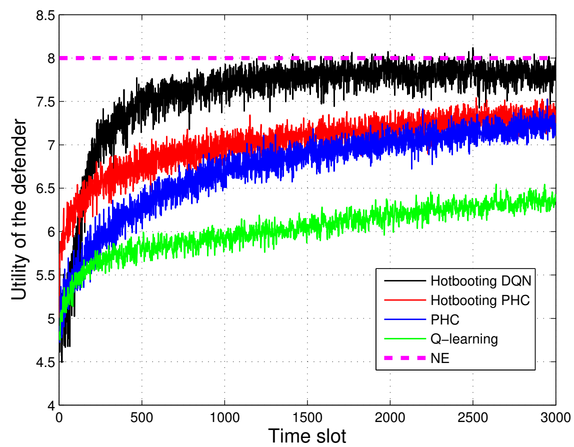
Initialize $\mathbf{N}^{(0)}$, $\mathbf{B}^{(1)}$, $\theta^{(0)}$, ξ, K and $\mathbb{E} = \emptyset$
for $i = 1, 2, 3, ..., \xi$ **do**
  Emulate a similar CPU allocation scenario for the defender to scan storage devices
  for $k = 1, 2, 3, ..., K$ **do**
    Observe the output of the CNN to obtain $Q\left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}\right)$
    Choose $\mathbf{M}^{(k)}$ via (4.32)
    for $i = 1, 2, ...D$ **do**
      Allocate $M_i^{(k)}$ CPUs to scan storage device i
    end for
    Observe the compromised storage devices and estimate $\mathbf{N}^{(k)}$
    Obtain $u_D^{(k)}$ via (4.4)
    Observe the resulting state sequence $\boldsymbol{\varphi}^{(k+1)}$
    $\mathbb{E} \leftarrow \mathbb{E} \bigcup \left(\boldsymbol{\varphi}^{(k)}, \mathbf{M}^{(k)}, u_D^{(k)}, \boldsymbol{\varphi}^{(k+1)}\right)$
    Perform minibatch update as steps 19-23 in Algorithm 3 to update $\theta^{(k)}$
  end for
end for
Output $\overline{\theta} \leftarrow \theta^{(k)}$

Table 4.2. Some APT attackers applied the ε-greedy algorithm to choose the number of CPUs to attack the D storage devices. Smarter APT attackers deliberately induced the storage defender to use a specific defense policy and then attacked it accordingly. We set $\alpha = 0.9$, $\gamma = 0.5$, $\delta = 0.02$, $W = 12$, and $H = 16$, if not specified otherwise, to achieve good security performance according to the experiments not presented in this chapter.

In the first simulation, the defender with 10 CPUs resisted the attacker with 2 CPUs over 10 storage devices, each with normalized data size. As shown in Fig. 4.5, the hotbooting DQN-based CPU allocation scheme achieves the optimal policy in a dynamic APT defense game after convergence, which matches the theoretical results of the NE given by Theorem 4.3. For example, the data privacy level almost converges to the NE given by (4.26), and the utility of the defender almost converge to the NE given by (4.25). Figure 4.5 also shows that the hotbooting technique accelerates the learning speed, improves the data protection level and increases the utility of the cloud storage system. For instance, the hotbooting PHC scheme saves 61.54% time to reach 0.67 data protection level, and improves the data protection level by 9.68% and increases



(a) Data protection level

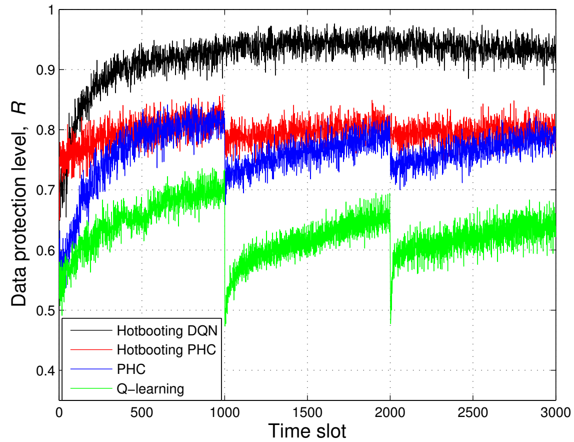


(b) Utility of the defender

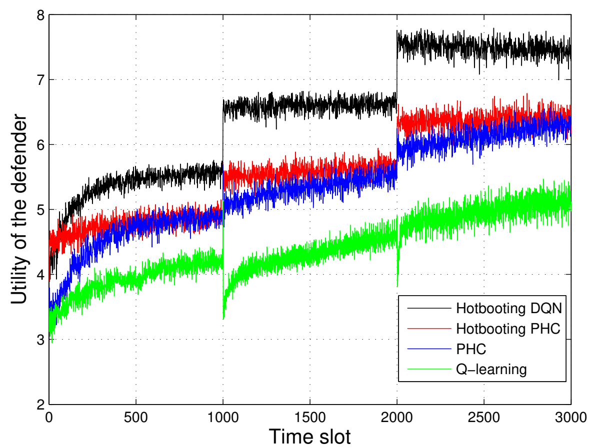
Figure 4.5: APT defense performance of the cloud storage system with 10 storage devices and 10 defense CPUs against an APT attacker with 2 attack CPUs. The size of data stored in each storage device is 1.

the utility of the cloud storage system by 9.59% at time slot 500, compared with the PHC scheme.

Moreover, the hotbooting DQN-based CPU allocation scheme outperforms the hotbooting PHC with a faster learning speed, a higher data protection level and a higher utility. The latter in turn exceeds both PHC and Q-learning. For instance, the data protection level of the hotbooting DQN-based scheme is 14.92% higher than the PHC-based scheme at time slot 1000, which is 30.51% higher than the Q-learning based scheme. As



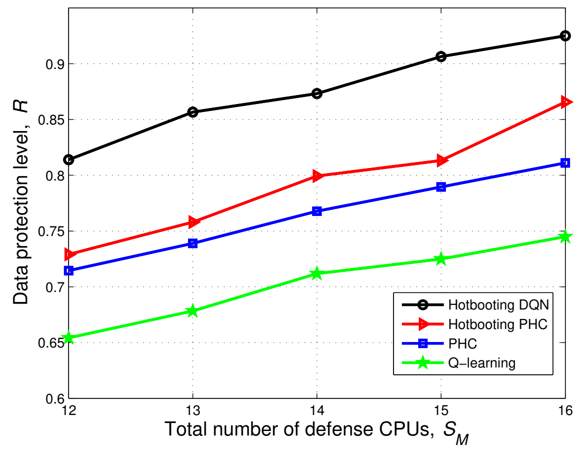
(a) Data protection level



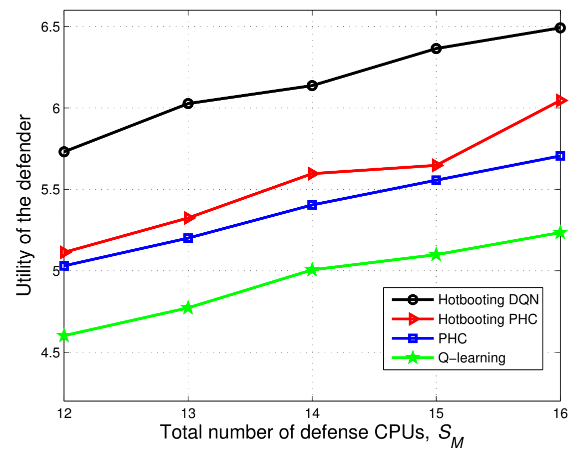
(b) Utility of the defender

Figure 4.6: APT defense performance of the cloud storage system with 3 storage devices and 16 defense CPUs against an APT attacker with 4 attack CPUs. Both the size of data stored on each device and the attack policy change every 1000 time slots.

a result, the hotbooting DQN-based scheme has a 14.89% higher utility than the PHC-based strategy at time slot 1000, which is 30.48% higher than the Q-learning based strategy. That is because the hotbooting DQN-based algorithm that uses CNN to compress the learning state space can accelerate the learning process and enhance the cloud security performance. The RL-based CPU allocation scheme keeps learning the APT attack profile in the dynamic CPU allocation game via trial-and-error and can achieve the optimal detection scheme after sufficient number of time slots. If the interaction time is long enough, the hotbooting PHC and Q-learning scheme can also converge



(a) Data protection level



(b) Utility of the defender

Figure 4.7: APT defense performance of the cloud storage system with S_M defense CPUs, and 3 storage devices that are attacked by 4 attack CPUs, averaged over 3000 time slots. The size of data stored in each storage device changes every 1000 time slots.

to the NE of the theoretical results in Theorem 4.3. The PHC-based scheme has less computation complexity than DQN. For example, the PHC-based strategy takes less than 94% of the time to choose the CPU allocation in a time slot compared with the DQN-based scheme.

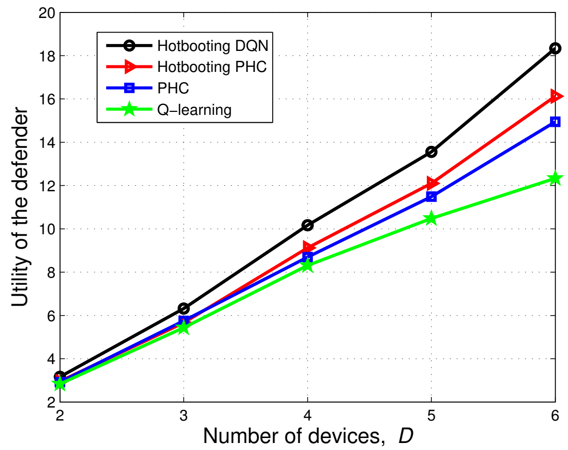
In the second simulation, the size of the data stored in each of the 3 storage devices of the cloud storage system changed every 1000 time slots. The total data size increases 1.167 times at the 1000^{th} time slot and then increases 1.143 times at the 2000^{th} time slot. The cloud storage system used 16 CPUs to scan the storage devices and the APT attacker used 4 CPUs to attack them. Besides, the attack policy changed every 1000 time slots. The APT attacker estimated the defense CPU allocation due to the learning algorithm and launched an attack specifically against the estimated defense strategy at time slot 1000 and 2000 to steal data from the cloud storage system. As shown in Fig. 4.6, the hotbooting DQN-based CPU allocation is more robust against smart APTs and the time-variant cloud storage system. For example, the data protection level of the hotbooting DQN-based scheme is 30.98% higher than that of the PHC-based scheme at time slot 1000, which is 97.87% higher than that of the Q-learning based scheme. As a result, the hotbooting DQN-based scheme has a 30.69% higher utility than the PHC-based strategy at time slot 1000, which is 96.97% higher than the Q-learning based strategy.

As shown in Fig. 4.7, both the data protection level and the utility increase with the number of defense CPUs. For instance, if the defender has 16 CPUs instead of 12 and applies the hotbooting DQN-based APT defense algorithm, its data protection level and utility averaged over 1000 time slots increase by 14.20% and 14.03%, respectively. In the dynamic game with $S_M = 16$, $D = 3$ and $S_N = 4$, the data protection level of the hotbooting DQN-based scheme is 14.65% higher than that of PHC, which is 25.67% higher than Q-learning, and the utility of the hotbooting DQN-based CPU allocation scheme is 14.34% higher than PHC, which is 25.38% higher than Q-learning.

It is also shown in Fig. 4.8 that the utility of the defender increases with the number of storage devices, as more quantity of data has been protected. The performance gain of the hotbooting DQN-based CPU allocation scheme over the hotbooting PHC-based scheme increases with the number of storage devices in the cloud storage system.

4.8 Conclusion

In this chapter, we have formulated a CBG-based CPU allocation game to protect cloud storage and cyber systems against APT. We have provided the NEs of the game to show how the number of storage devices, the data sizes of the storage devices and the total number of CPUs impact on the data protection level of the cloud storage system and the defender's utility. A hotbooting DQN-based CPU allocation strategy has been proposed



Utility of the defender

Figure 4.8: APT defense performance of the cloud storage system with D storage devices and 21 defense CPUs against an APT attacker with 4 attack CPUs, averaged over 3000 time slots. The size of data stored in each storage device changes every 1000 time slots.

for the defender to scan the storage devices without being aware of the attack model and the data storage model in the dynamic game. The proposed scheme can improve the data protection level with a faster learning speed and is more robust against smart APT attackers that choose the attack policy based on the estimated defense learning scheme. For instance, in a scenario with 3 storage devices with 16 defense CPU and 4 attack CPUs, the data protection level of the cloud storage system and the utility of the defender increases by 25.67% and 25.38%, respectively, in comparison with the Q-learning based scheme. The hotbooting PHC-based CPU allocation scheme can reduce the computation complexity of DQN.

Chapter 5

Colonel Blotto Game (CBG) Formulation for Inter-Network Dynamic Spectrum Allocation

5.1 Overview and Motivation

Traditional wireless networks operate as parallel, independent infrastructures with little to no inter network coordination [174]. This can lead to poor utilization of the licensed spectrum, inability to address load imbalance, and high interference in the case of unlicensed spectrum. Prime examples of such scenarios include multiple co-located Wi-Fi hotspots, and cellular networks with mismatched data demand and spectrum availability. To address better usage of licensed and unlicensed spectrum, several new notions of spectrum usage are being promoted, including licensed assisted access (LAA) [175], licensed shared access [176], co-primary sharing [177], etc. While these solutions are aimed at a more harmonious use of the available spectrum through coordination and cooperation among the network service providers (NSPs), better spectrum utilization can also be realized through competition among the NSPs. Such an environment can be created by allowing users to choose their NSPs on an on-demand basis, without them committing to monthly subscription plans. NSPs are then required to compete with one another to provide service to these untethered users by competitively allocating the available radio resources. While such a model enables users to choose the best available service without committing to a single NSP, it also forces the NSPs to constantly employ all available resources, thus promoting better spectrum utilization.

As an example of such an architecture, consider the concept behind Googles Project Fi [59], where mobile users have no dedicated NSPs, and instead opportunistically connect to the cellular service or WiFi hotspot offering them the best service. Such a setup unlocks spectrum to be used in an opportunistic manner, with the ancillary benefits of better interference management and higher spectral efficiency. A key component in such a system is the competitive bidding by the multiple NSPs to be chosen as service providers to the different users in the pool. In such a setting, it is of interest to analyze the competitive allocation of spectrum by multiple NSPs among a group of users. Naturally, to analyze such competitive scenarios, game-theoretic tools [178] are known to be a suitable framework.

In particular, we consider two NSPs, each with a fixed amount of non-overlapping bandwidth, competing over a common pool of users. The users can either be of equal or different value to the NSPs. Depending on the total available bandwidth and the value of the user, each user receives an offer of service from the two NSPs, quantified by the amount of bandwidth the NSP is willing to support that user with. The user then picks the offer that maximizes its utility. The NSPs aim to serve as many users as possible so as to maximize their revenue. This process is constantly repeated as users move in and out of the system and when more bandwidth becomes available.

Each NSP must deal with the challenging task of strategically allocating spectrum so as to out-bid the other NSP while adhering to the constraints on the total available bandwidth. The competitive allocation of spectrum by the NSPs is closely related to the Colonel Blotto game (CBG) [150, 179] − a multidimensional problem on strategic resource allocation. The classical CBG is a two-person constant-sum game in which two players (colonels) are tasked with allocating a limited resource (troops) over multiple fronts (battlefields), with the player allocating the most resources to a front being declared the winner, and the overall payoff being proportional to the number fronts won. The classical CBG and its variants are known to be challenging problems due to the complex strategy space. Yet, recent progress by Roberson [150] has provided valuable insight on equilibrium-achieving mixed strategies in such problems. Characterizing equilibrium-achieving mixed strategies of other variants of the CBG is an active area of research [180, 181].

The main contribution of this chapter is to introduce a novel approach to the inter-network spectrum allocation problem using the framework of CBG in both the discrete and continuous domains. In the continuous case, we assume spectrum to be an infinitely divisible resource and establish parameter settings under which equilibrium achieving mixed strategies to internetwork spectrum allocation are known. We then proceed to consider spectrum as a quantized resource and shift focus to the discrete CBG. Since the discrete CBG is a 2-player constant-sum matrix game, we propose a learning algorithm based on fictitious play [182] to numerically compute the mixed strategies that achieve Nash equilibrium (NE). We compare the numerically obtained strategies to those predicted by the theoretical results in the continuous case and further proceed to consider parameter settings for which no theoretical results are available.

5.2 Inter-Network Spectrum Allocation

Consider two independent NSPs R_1 and R_2 with non-overlapping bandwidths W_1 and W_2, respectively. R_1 and R_2 compete to provide service to a pool of N users labeled U_1, U_2 , . . . , U_N. We let p_i denote the payoff/revenue to the NSP that is chosen to provide service to user U_i. The two NSPs strategically divide the available bandwidth W_i among the pool of N users so as to maximize their payoff. Each user U_i thus receives a *bid* of w_{1k} and w_{2k} from the two NSPs, indicating an intention to provide service using an amount, w_{ik} of bandwidth. Using an estimate of the spectral efficiency σ_{ik} that can be achieved when served by NSP R_i, User U_K chooses the NSP maximizing the total rate achieved, i.e., users choose the NSP that maximizes $\sigma_{ik}w_{ik}$[1]. It is assumed that information regarding the spectral efficiencies is relayed to both the NSPs. Spectral efficiency for the link between R_i and U_k is obtained by measuring the signal-to-noise ratio (SNR_{ik}) and setting $\sigma_{ik} = log(1 + SNR_{ik})$. If spectral efficiencies are not estimated, they are assumed to be 1 and such a scenario is termed SNR-agnostic spectrum allocation. Note that since the payoff does not incentivize bandwidth conservation, it can be assumed without loss of generality that all available bandwidth is used in the bidding process, i.e., the N bids by R_i satisfy $\sum_{k=1}^{N} w_{ik} = W_i$. The total payoff to NSP R_1 from such a

[1]Tie resolution depends on whether bandwidth is treated as a continuous or a discrete parameter. In the continuous case, ties are always resolved in favor of NSP R_1, while in the discrete case they are resolved using a coin toss.

process, assuming no ties, is given by

$$c_1 = \sum_{k:\sigma_{1k}w_{1k}>\sigma_{2k}w_{2k}} p_k \tag{5.1}$$

After this process, the users get served by their NSP of choice using the promised amount of bandwidth. Since it is unlikely that all users choose to associate with an NSP, the unused bandwidth at each NSP is rolled over to the next session or time slot when the NSPs compete again to serve a new pool of users. This chapter restricts focus to a single instance of such a bidding process, with the design and behavior of the repetitive bidding process being of interest in the future. Note that not all the N bids made by an NSP are likely to be accepted and this results in some unallocated bandwidth. We assume that this residual bandwidth is not reassigned as the users have already agreed to be saved and further, there is an economic incentive to save this bandwidth for a subsequent bidding session.

For our model, one important consideration is whether or not bandwidth can be treated as an infinitely divisible resource. In theory, while it is indeed possible to treat bandwidth as an infinitely divisible resource, in practice, bandwidth is typically assigned in certain preset quantized values. This subtle but important distinction in how this resource is treated leads to two different problem formulations. Treating bandwidth as an infinitely divisible resource allows us to take recourse to well-known results in the context of continuous CBG.

In particular, we view inter-network spectrum allocation as a strategic game $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ between two non-cooperating NSPs and aim to characterize strategies for bandwidth allocation that achieve NE. As we note in the next section, pure strategies, i.e., strategies that allocate a predetermined amount of bandwidth to each of N users, achieve NE only under rare circumstances. This turns our attention to mixed strategies where bandwidth allocation to the N users is governed by an underlying probability distribution. Let the set of all possible mixed strategies of NSP R_i for the game $\mathcal{G}$ be denoted by $\mathcal{S}_i$, where $\mathcal{S}_i$ consists of all $\mathcal{N}$-variate probability density functions $f_i\left(w_{i1}, w_{i2}, \ldots, w_{iN}\right)$ with support $\Delta_i = \{\{w_{ik}\}_{k=1}^{N} : \sum_{k=1}^{N} w_{ik} = W_i\}$. Characterizing equilibrium achieving mixed strategies for $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ requires establishing a pair of $\mathcal{N}$-variate probability density functions $f_i(\cdot) \in \mathcal{S}_i$ and$f_j \in \mathcal{S}_j$ that satisfy

$$c_i\left(f_i{}^*, f_j{}^*\right) \geq c_i\left(f_i, f_j{}^*\right) \quad \forall f_i \in \mathcal{S}_i, i \neq j, \tag{5.2}$$

where $c_i(f_i, f_j)$ denotes the expected payoff to R_i when $f_i{}^*$ and $f_j{}^*$ are chosen as the strategies by R_i and R_j, respectively. Since the game $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ is a constant-sum game with compact pure strategy spaces and has a semi-continuous payoff function, we can apply a result by Dasgupta and Maskin [183] establish the following proposition.

Proposition 5.1. *For the spectrum allocation game $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$, there always exists a pair of mixed strategies that achieve the NE.*

Since this is a constant-sum game, due to the minimax theorem [184], it is not necessary to specify optimal strategy profiles $(f_1{}^*, f_2{}^*)$ as a pair, and instead it suffices to establish equilibrium-achieving mixed strategies for each individual NSP which can then be paired in any manner to obtain an optimal strategy profile $(f_1{}^*, f_2{}^*)$. Optimal mixed strategies for certain parameter settings of $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ are described in the next section.

From a practical standpoint, it is of interest to also study the discrete version of the spectrum allocation problem. Although analytical results are difficult to obtain when treating bandwidth as a quantized resource, such a formulation is more amenable to well-known numerical techniques such as fictitious play [182]. Section 5.4 discusses the 2-player constant-sum matrix game that results when bandwidth is treated as a quantized resource and numerically computes the optimal mixed strategies.

5.3 Inter-Network Spectrum Allocation As a Continuous Colonel Blotto Game

Proposed as early as 1921 by Borel, CBG is one of the best examples of resource allocation in a competitive environment. It closely mirrors the spectrum allocation problem that is of interest here, but is presented in the context of a war between two colonels over multiple battlefields. The canonical CBG involves two colonels (players) B_1 and B_2 engaged in a war over N battlefields with a total of T_1 and T_2 troops (assume $T_1 \leq T_2$) at their disposal. The colonels strategically assign the available troops among the N battlefields, with the winner of each battlefield determined to be the colonel assigning the greater number of troops to that battlefield. Assuming the k^{th} battlefield to have a payoff of q_i, the goal for each colonel is to assign troops in such a manner that the total payoff is maximized. Denoting t_{ik} as the troops assigned by B_i to the k^{th} battlefield, the troop assignments must satisfy $\sum_{k=1}^{N} t_{ik} \leq T_i$. CBG is typically studied as a continuous game with the troops T_1 and T_2 being treated as infinitely divisible. This is a constant-sum game and a NE in mixed strategies exists due to the result by Dasgupta and Maskin [183]. Early studies on CBG [179] assumed symmetric colonels ($T_1 = T_2$) and symmetric battlefields ($q_i = q_j \forall \quad i, j$), a setup called doubly-symmetric CBG. More recently, CBG with symmetric colonels but asymmetric battlefields is studied in [185], while CBG with asymmetric colonels but symmetric battlefields is studied in [150]. To the best of our knowledge, there are no known results when both symmetries are broken. While other variants of the CBG have also been studied, they are not immediately relevant to the spectrum allocation problem.

The analogy between CBG and inter-network spectrum allocation is immediate once we note that the NSPs play the role of colonels, with bandwidth as their constrained resource (troops), and users serving as the $\mathcal{N}$-battlefields. Thus, denoting the CBG as $\mathcal{B} = (N, \{T_i\}, \{q_i\})$, it is straightforward to establish the following proposition.

Proposition 5.2. *When spectral efficiencies σ_{ik} satisfy $\sigma_{1k} = \sigma_{2k} \quad \forall k$, the inter-network spectrum allocation game $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ is equivalent to the colonel Blotto game $\mathcal{B} = (N, \{W_i\}, \{p_i\})$.*

This equivalence allows us to re-frame equilibrium strategies for the CBG in the context of inter-network spectrum allocation. Despite its relatively simple formulation, equilibrium achieving mixed strategies for the CBG are only known for certain parameter settings. Similar to the spectrum allocation game, a mixed strategy for the CBG is an $\mathcal{N}$-variate density function with support contained in the set of feasible allocations of the troops. Typically, characterizing the mixed strategies that achieve NE is split into two parts, one focused on specifying the N univariate marginal distributions of the $\mathcal{N}$-variable equilibrium distribution, and the other on constructing an $\mathcal{N}$-variate distribution that has the appropriate univariate marginal distributions. Proposition 5.2 is most relevant to a SNR-agnostic spectrum allocation game where the spectral efficiencies are not immediately available and assumed to be 1. In such a setting, the results on equilibrium mixed strategies in [150, 185] can be immediately adapted to establish results of the following form:

Theorem 5.1. *(based on Proposition 1 in [185]): For a spectrum allocation game $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ that satisfies (a) $W_1 = W_2$ (symmetric colonels), (b) $\sigma_{1k} = \sigma_{2k} \quad \forall k$ (SNR-agnostic) and (c) $p_k < \sum_{j \neq k} p_j \quad \forall k$ (no dominant user), any $\mathcal{N}$-variate probability density function with support δ_i where the k^{th} univariate marginal density function is uniformly distributed on $[0, \frac{2W_i p_i}{\sum p_i}]$ constitutes an equilibrium-achieving mixed strategy providing equal payoffs to both the NSPs.*

Theorem 5.2. *(based on Theorem 2 in [150]): For a spectrum allocation game $\mathcal{G} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ that satisfies (a) $\frac{2}{N} \leq \frac{W_1}{W_2} \leq 1$ (asymmetric colonels), (b) $\sigma_{1k} = \sigma_{2k} \quad \forall k$ (SNR-agnostic) and (c) $p_i = p_j \quad \forall i, j$ (symmetric users), the equilibrium univariate marginal density functions $f_{1i}()$ and $f_{2i}()$ are given by*

$$f_{1i}(w_{1i}) \sim (1 - \tfrac{W_1}{W_2})\delta(w_{1i}) + \tfrac{W_1}{W_2}\mathcal{U}([0, \tfrac{2W_2}{N}]) \tag{5.3}$$

$$f_{2i}(w_{2i}) \sim \mathcal{U}([0, \tfrac{2W_2}{N}]) \tag{5.4}$$

where $\delta(\cdot)$ denotes the unit impulse function and $\mathcal{U}(\cdot)$ denoted the uniform density function over a specified interval. The equilibrium payoff to NSP 1 is $\frac{W_1}{2W_2}$.

Constructing $\mathcal{N}$-variate distributions that satisfy the above univariate marginals is non-trivial. Geometric methods of construction are proposed in [179, 185–187], while other approaches are suggested in [150]. For brevity, we omit the exact details.

As an illustration of these results, suppose two NSPs with 10 MHz each compete to serve a set of 5 users, each offering the same payoff, then according to Theorem 3, the optimal strategy is to offer each user a bandwidth chosen at random from 0 to 4 MHz, while satisfying the bandwidth constraint. Suppose instead, the first NSP only has 5 MHz of bandwidth, then by Theorem 5.2, a user is allocated non-zero bandwidth only 50% of the time. This in effect halves the total number of users for whom NSP 1 allocates a non-zero bandwidth. Thus, it can be observed that bandwidth-constrained

NSPs tend to adopt a strategy whereby they only compete over a random subset of users while bandwidth-rich NSPs tend to spread out the available bandwidth among all the users in the pool.

Extending these results to incorporate spectral efficiencies is a challenging problem and requires further research. However, spectral efficiencies can be naturally factored in when treating bandwidth as a quantized resource, where we rely on computational methods to design equilibrium strategies. This is discussed further in the next section.

5.4 Inter-Network Spectrum Allocation: The Discrete Case

The discrete spectrum allocation game is the same as its continuous version, except that the total bandwidth is now specified in terms of the number of orthogonal channels owned by the NSP and the bandwidth allocation is a non-negative integer vector specifying the number of channels allocated to the N users in the pool. Due to the integer constraints on bandwidth allocation, the resulting game is a constant-sum two-player matrix game with finite number of strategies. Such a game is known to have NE in mixed strategies, with all such strategies yielding the same payoff. As before, the equilibrium achieving strategies of the two NSPs can be paired in any manner to obtain an equilibrium strategy profile.

The discrete CBG is an immediate analogue of such a game and Proposition 5.2 also applies here. However, due to the combinatorial nature of the strategy space, the discrete CBG is not as extensively studied as the continuous CBG. The best result in this context is by Hart [188], who studied the discrete CBG with a primary focus on the doubly symmetric case.

Rather than pursuing analytical results, this section focuses on numerical techniques for computing equilibrium mixed strategies under general parameter settings. In particular, we adopt fictitious play [182], a well-known learning algorithm, to compute the equilibrium mixed strategies. Fictitious play is a belief based learning rule that is commonly used in the context of 2-player matrix games. Fictitious play for two player game simulates a repeated game where the two players play an action/strategy in each round and try to learn the best strategy from the cumulative outcome of all the previous rounds.

Let $\mathcal{G}_{\mathcal{D}} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ denote the discrete spectrum allocation game where $W_i,\ w_{ik} \in \mathbb{Z}^+ \quad \forall i, k$, Denote the set of all possible bandwidth allocations of NSP R_i as $\boldsymbol{\mathcal{A}}_i = \{\mathbf{a}_{i1}, \mathbf{a}_{i2}, \ldots, \mathbf{a}_{iH_i}\}$ where the allocation $\mathbf{a}_{ik}$ represents an $\mathcal{N}$-integer tuple $\{w_{i1}, w_{i2}, \ldots, w_{iN}\}$ satisfying $\sum_{k=1}^{N} w_{ik} = W_i$. The size of the set $\boldsymbol{\mathcal{A}}_i$, denoted as H_i, is equal to $\binom{W_i+N-1}{W_i}$.

Fictitious play for the game $\mathcal{G}_{\mathcal{D}} = (N, \{W_i\}, \{\sigma_{ik}\}, \{p_k\})$ simulates an iterated spectrum allocation game between the two NSPs, where after the k^{th} iteration, NSP R_i holds the belief that its opponent is playing this iterative game using a stationary (possibly mixed) strategy that is characterized by the belief vector $\mathbf{q}_i^{(k)} = \left[q_{i1}^{(k)}, q_{i2}^{(k)}, \ldots, q_{iH_i}^{(k)}\right]$ where $q_{il}^{(k)}$ represents the belief held by NSP R_i, after the k^{th} iteration, that R_j's $(j \neq i)$ mixed strategy plays the l^{th} action with probability $q_{il}^{(k)}$. In every iteration of this game, R_i updates this belief based on the strategy played by R_j. Thus, if R_j plays the l^{th} strategy at the $(k+1)^{\text{th}}$ iteration, R_i updates the belief vector as follows:

$$q_{im}^{(k+1)} = \begin{cases} \frac{k}{k+1} q_{im}^{(k)} + \frac{1}{k+1} & m = l \\ \frac{k}{k+1} q_{im}^{(k)} & m \neq l \end{cases} \tag{5.5}$$

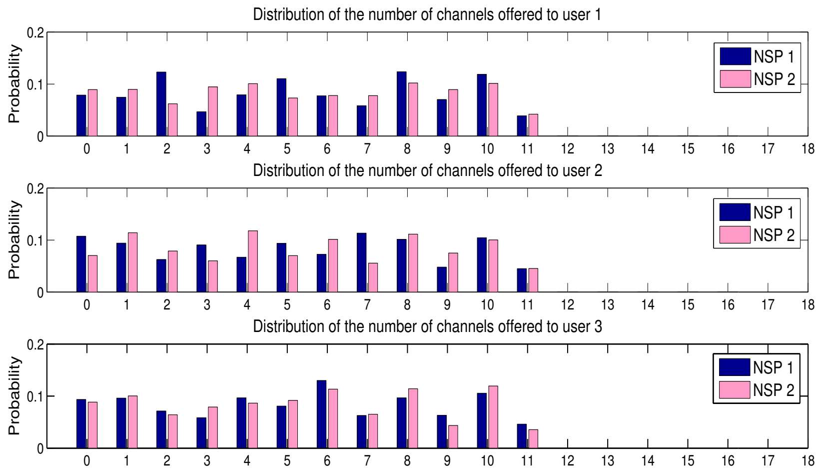
Now, the strategy chosen by R_i at the $(k+1)^{\text{th}}$ iteration is based on its beliefs at the end of the k^{th} iteration. In particular, R_i chooses the action that maximizes its payoff in response to a mixed strategy of R_j governed by $\mathbf{q}_i^{(k)}$, i.e.,

$$\underset{a_{il} \in A_i}{\arg\max} \quad C_i\left(a_{il}, \mathbf{q}_i\left(k\right)\right), \tag{5.6}$$

where $C_i(\cdot)$ denotes the expected payoff, i.e., $E_{a_{il}, \mathbf{q}_i^{(k)}}[c_i]$.

Starting from a random initialization of the belief vectors, this process is repeated until convergence (Convergence to a mixed strategy equilibrium is guaranteed in constant sum games [189]). At convergence, $\mathbf{q}_i^{(k)}$ represents an equilibrium strategy of R_j.

We use this process to numerically compute the equilibrium mixed strategies to the discrete inter-network spectrum allocation problem under general parameter settings.



Parameters	
N	3
W_1	16
W_2	16
ρ_{1k}	$[1, 1, 1]$
ρ_{2k}	$[1, 1, 1]$
p_i	$[4, 4, 4]$

Payoffs	
c_1	6
c_2	6

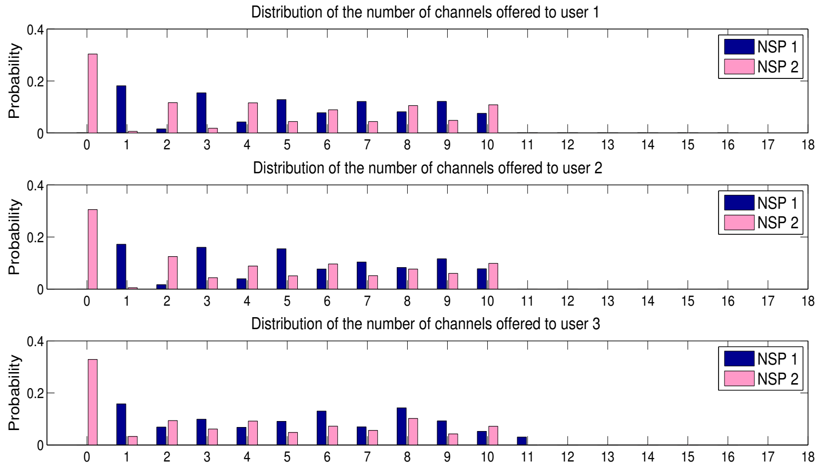
Figure 5.1: Optimal mixed strategy marginal distribution in a 2-NSP 3-user internetwork spectrum allocation problem with symmetric bandwidth availability and equal payoffs across users

5.5 Numerical Results

We use fictitious play on a network with two NSPs competing to provide service to three users. We consider four different parameter settings and highlight the important features of the resulting equilibrium mixed strategies.

Case (i): This case considers SNR-agnostic spectrum allocation with symmetric NSPs and equal payoffs for all users. The two NSPs are assumed to have a total of 10 MHz of bandwidth that can only be assigned in multiples of 1.25 MHz (16 channels to be assigned). The mixed strategy obtained for such a scenario is presented in Fig. 5.1, where it is seen that the resulting univariate marginals randomly allocate up to 11 channels to a user. Interestingly, the marginal distributions are not uniform distributions as predicted by theory in the continuous case [150, 179]. However, the support of the marginal distributions is in line with that predicted by theory $\left(\frac{2W_i}{N}\right)$. Since all three users are identical from an NSPs perspective, the marginal distributions suggest that the NSPs compete to provide service to all users, with no preference given to any of them. By symmetry the two NSPs receive equal payoffs.

Case (ii): This case is similar to the previous case, except that the second NSP is



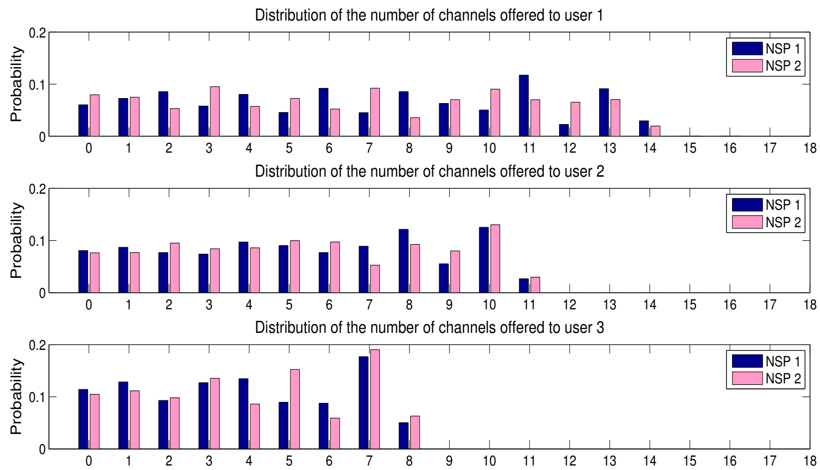
Parameters	
N	3
W_1	16
W_2	12
ρ_{1k}	$[1, 1, 1]$
ρ_{2k}	$[1, 1, 1]$
p_i	$[4, 4, 4]$

Payoffs	
c_1	7.5
c_2	4.5

Figure 5.2: Optimal mixed strategy marginal distribution in a 2-NSP 3-user inter-network spectrum allocation problem with asymmetric bandwidth availability and equal payoffs across users

now assumed to have only 12 channels. It is seen from Fig. 5.2 that the bandwidth-constrained NSP now tries to compete only over a random subset of users. This is inferred by noting that NSP 2 chooses to allocate no channels to user i with a probability ≈ 0.5. Interestingly, the bandwidth-rich NSP is cognizant of this behavior and ensures that all three users are allocated at least one channel, thus enabling it to win over users that receive no channel allocations from NSP 2, while expending the least amount of channels to win over these users. The resulting payoffs suggest that NSP 2 is likely to serve only one of the three users. These results are in close agreement with those predicted by Theorem 5.2 for the continuous case, except for non-uniformity of the marginal distributions.

Case (iii): This case uses the same parameters as the first case, except that the users now have unequal payoffs. It can be observed from Fig. 5.3 that the support of the univariate marginal distributions of the equilibrium mixed strategies is proportional to the users value. With the NSPs having the same amount of bandwidth at their disposal they compete for all the three users, with a higher interest in winning over the users with larger payoffs. Due to the symmetry among the two NSPs, the net payoff remains equal.



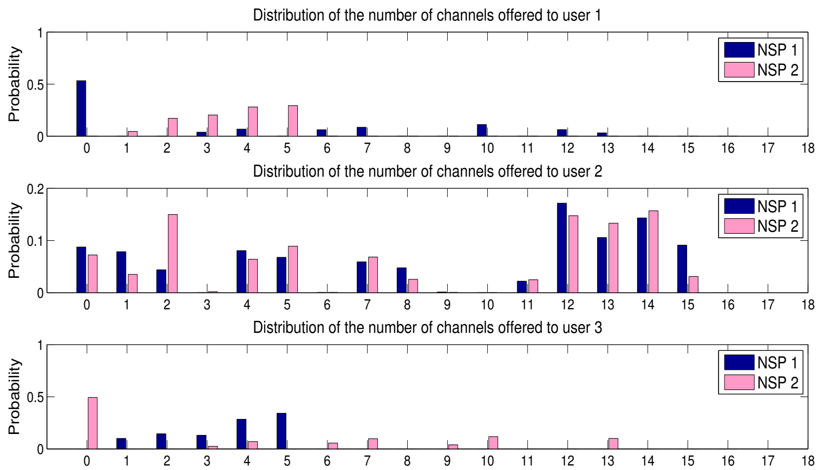
Parameters	
N	3
W_1	16
W_2	16
ρ_{1k}	$[1, 1, 1]$
ρ_{2k}	$[1, 1, 1]$
p_i	$[5, 4, 3]$

Payoffs	
c_1	6
c_2	6

Figure 5.3: Optimal mixed strategy marginal distribution in a 2-NSP 3-user inter-network spectrum allocation problem with symmetric bandwidth availability and unequal payoffs across users

Case (iv): Unlike the previous three cases, this case considers different spectral efficiencies for the user-NSP links, while offering the same payoff for all the users. As seen in Fig. 5.4 the equilibrium mixed strategies allocate more channels to users with better channel conditions, i.e., higher spectral efficiency. This can be observed by noting NSP 1 (NSP 2) avoids competing for user 1 (user 3) and prefers to not allocate any bandwidth to this user with a probability of 0.5. This feature has a clear practical significance it shows that such a competitive approach to inter-network spectrum allocation can also capture the salient aspects of user-base-station association in traditional networks, thereby contributing to an increase in the overall throughput across all NSPs. Interestingly, due to similar match-ups between the differences in spectral efficiency, the payoffs get equally divided among the two NSPs.

Case (v): Finally, a completely general setup is considered in Fig. 5.5 where in addition to spectral efficiencies, unequal payoffs are also factored in. The NSPs appear to back off from competing for users where they have a significant disadvantage in terms of spectral efficiency despite a higher payoff and instead direct their resources towards winning over users with more favorable spectral efficiencies.



Parameters	
N	3
W_1	16
W_2	16
ρ_{1k}	$[1, 2, 3]$
ρ_{2k}	$[3, 2, 1]$
p_i	$[4, 4, 4]$

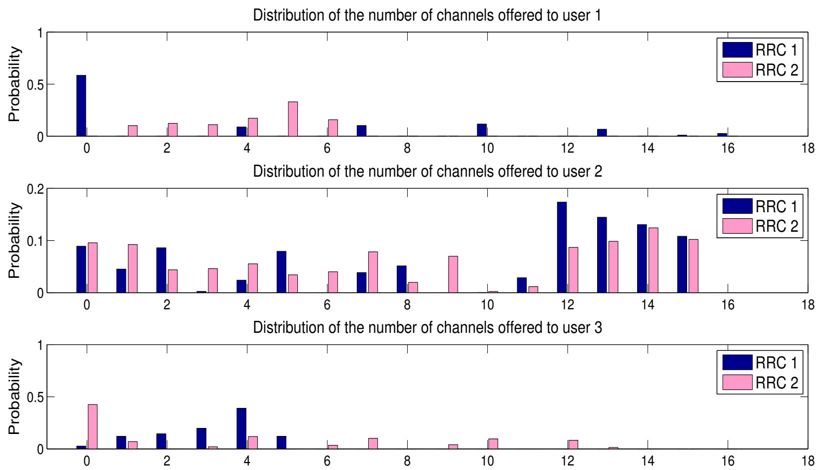
Payoffs	
c_1	6
c_2	6

Figure 5.4: Optimal mixed strategy marginal distribution in a 2-NSP 3-user internetwork spectrum allocation problem with symmetric bandwidth availability and equal payoffs across users but with a different spectral efficiency for each link.

These results illustrate the broad applicability of fictitious play to compute equilibrium mixed strategies of the internetwork spectrum allocation for any set of parameters. The results further illustrate that the numerically computed strategies are physically meaningful and promote better use of the available spectrum.

5.6 Discussion

This chapter considered the problem of spectrum allocation in a network architecture where users are free to choose their network service providers (NSPs) in an opportunistic manner. The NSPs are assumed to compete over a common pool of users by competitively allocating the available bandwidth. Spectrum allocation in such a setup is shown to be closely related to the Colonel Blotto gamea multidimensional resource allocation problem that is well studied in game theory. We cast the inter-network spectrum allocation problem as a CBG and studied it in the case of discrete as well as continuous spectrum (bandwidth) allocation. For the continuous case, we adapted the existing theoretical results for CBG, while a computational approach using fictitious play is used to numerically compute equilibrium mixed strategies in the discrete case. The resulting strategies were analyzed and shown to promote better utilization of available resources across the networks. In summary, the CBG is shown to provide



Parameters	
N	3
W_1	16
W_2	16
ρ_{1k}	$[1, 2, 3]$
ρ_{2k}	$[3, 2, 1]$
p_i	$[5, 4, 3]$

Payoffs	
c_1	6
c_2	6

Figure 5.5: Optimal mixed strategy marginal distribution in a 2-RRC 3-user inter-network spectrum allocation problem under general parameter settings.

a valuable framework to study competitive spectrum allocation and warrants further investigation.

Chapter 6

Dynamic Colonel Blotto Game Model for Spectrum Sharing in Wireless Networks

6.1 Overview and Motivation

It is widely believed that a scalable solution for dynamic spectrum assignment can also be realized through decentralized architecture and competition among the NSPs [58]. Such an environment can be created by allowing users to choose their NSPs on an on-demand basis, without committing to any specified provider. NSPs are competing with one another to provide service to these users by competitively allocating the available radio resources. Here, the term "users" has a broad meaning which can range from a single mobile user to a complete campus-wide WiFi network. Google's Project Fi [59] is an example of such an architecture, where a pool of mobile users with no dedicated cellular service provider opportunistically connect to the cellular service provider which offering them the best service. The key component in such a system is the competitive bidding by the multiple NSPs to attract more users in the specific region. In this regard, game theory [190] proves quite useful by providing a rich literature to model and analyze the competitive allocation of spectrum by multiple NSPs among a group of users.

In this chapter, we study the problem of dynamic spectrum allocation by considering two NSPs, each with a limited amount of non-overlapping bandwidth, competing repeatedly to provide wireless connectivity to a set of users. At the beginning of each time slot, the NSPs can lease all or portion of their promised bandwidth to compete with one another to serve as many users as possible. Each NSP strategically offers its available resources to the users so as to out-bid the other NSP while adhering to its bandwidth constraints. The competitive allocation of spectrum by the NSPs at each time slot is closely related to the Colonel Blotto Game (CBG) [89], a multidimentional problem on strategic resource allocation. The classical CBG is a two person constant-sum game in which two colonels are tasked with allocating a limited troops over multiple battlefields, with the player allocating the most troops to a front being declared the winner, and the overall payoff being proportional to the number of fronts won. In our problem, one can view the users as the battlefields where the NSPs compete over them, and the available bandwidth resources as the troops. This process is constantly repeated as users move in and out of the system and when more bandwidth becomes available. In particular, it is assumed that the available bandwidth at different time slots changes based on a certain dynamic which itself depends on the past and current allocation strategies of the NSPs.

As one of the main contributions, we introduce a dynamic noncooperative repeated game as the decentralized approach for the NSPs to determine optimal strategies for NSPs over a finite time horizon. Since at each stage of this dynamic game the NSPs play a CBG game (which is strategically equivalent to a zero-sum game), the problem of dynamic bandwidth allocation game can be cast as a zero-sum dynamic game (ZSDG) [190]. In particular, obtaining the optimal equilibrium strategies for the NSPs reduces to finding the saddle point strategies of such a ZSDG. However, unlike standard ZSDGs with linear dynamics and quadratic payoff functions where the saddle point strategies can be obtained by solving so-called *Ricatti* equations [191], the payoff functions in our dynamic CBG game has a complicated piecewise structure. This makes the problem of finding closed form solutions for the saddle point strategies in dynamic CBG game much more difficult. Instead, we approximate the instantaneous payoff functions using simple smooth functions, and use a dynamic programming (DP) approach to obtain closed form solutions for the value function and the saddle point strategies of the approximated problem in certain range of parameters. In particular, we envision that such an approximation closely mimics the structure of the dynamic CBG game which we validate using our simulations results based on recently developed techniques for solving more general DP problems [192].

6.2 Related Work

DSA has been extensively studied to enhance the spectrum efficiency and encourage more flexible services in the spectrum market in the context of cognitive radio [193], cloud radio access networks (CRAN) [194], and heterogeneous networks [195]. Dynamic spectrum allocation through coordination and cooperation has been studied in [55, 56], and [196]. Moreover, it has been shown in [197] that competitive interactions between network operators can always bring higher payoffs to the users as opposed to coordination. In [198], the problem of dynamic spectrum sharing in cognitive radio among primary and secondary users is formulated as a dynamic Cournot game. [199] considers a duopoly situation, where two wireless service providers participate in bandwidth competition in spectrum purchasing and price competition to attract end users. A framework for competition of future operators under the regulation of a spectrum policy server (SPS) is developed in [200]. A truthful spectrum double auction which allow different providers to dynamically buy and sell spectrum to each other has been studied in [201]. In fact, one of the main differences of our work compared with the previous literature is that we introduce a novel approach to the inter-network spectrum allocation problem using the framework of dynamic CBG and show that under specific conditions, one can obtain a pair of optimal saddle point strategies for the NSPs.

6.3 System Model and Problem Formulation

6.3.1 Inter-Network Spectrum Allocation

We consider two independent, non-cooperating NSPs (Fig.6.1) which are promised a certain amount of non-overlapping bandwidths. They are able to lease all or a portion of that bandwidth at each time slot. The NSPs compete to provide service to a pool of n users by knowing the amount of bandwidth that every NSP has at its disposal. The two NSPs strategically divide the available bandwidth among the pool of n users so as to maximize their own payoffs. Each user thus receives a *bid* from the two NSPs at each time slot, indicating the amount of bandwidth that the NSPs are willing to offer for that time period. The users choose the NSP with a better offer (ties are broken using a

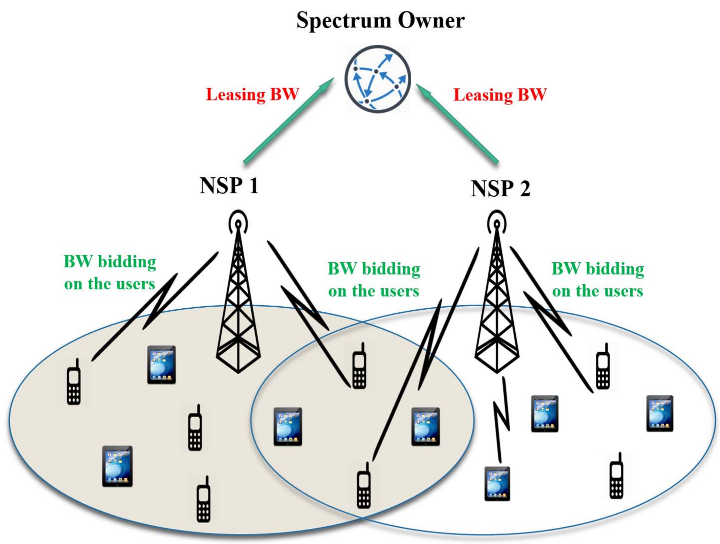


Figure 6.1: Network model.

coin toss). The payoff of each NSP at a given time slot is proportional to the number of users subscribed to that NSP, and its total payoff equals the sum of its payoffs over the entire horizon. Note that since the payoff does not incentivize bandwidth conservation, without loss of generality it can be assumed that all leased bandwidth is used through the bidding process.

Subsequent to this process, we assume that the rest of the bandwidth which is not leased by the NSPs is rolled over to the next time slot. On the other hand, a portion of the bandwidth allocated to the users will be available again for the next session, as some users may unsubscribe from the service in which case their released bandwidth can be reused by the NSPs.[1] Therefore, the NSPs start to lease the bandwidth and compete again to serve a new pool of users. Note that, the maximum amount of bandwidth available to NSPs at each time slot highly depends on the amount of bandwidth that they have leased on the previous sessions. To model this problem, CBG provides a solid framework to capture resource allocation in competitive environment when the players have limited resources. It closely mirrors the spectrum allocation problem that is of interest here. Next, we provide the problem of spectrum allocation in a single stage

[1]As an example, in a Starbucks coffee shop, the customers (users) use Internet service while they frequently come and leave.

using CBG.

6.3.2 Static Single Stage CBG Game

A single stage CBG is characterized by two non-cooperative rational players who compete for the same set of n battlefields. Each player has a fixed number of troops (budget) who can distribute them among the battlefields. A player wins a battlefield if he allocates more budget than his opponent to that battlefield, with a payoff proportional to his number of winning battlefields. Denoting the budgets of the players by u and w, respectively, it is known that the single stage CBG admits a mixed-strategy Nash equilibrium in which the maximum payoff of the first player given that his opponent distributes his budget optimally is given by [150]:

$$U_1(u,w) = \begin{cases} 0 & \text{if} \quad \frac{w}{u} \leq \frac{1}{n}, \\ \frac{2\theta-2}{\theta n^2} & \text{if} \quad \frac{1}{n} \leq \frac{w}{u} \leq \frac{1}{n-1}, \\ \frac{2}{n} - \frac{2u}{n^2 w} & \text{if} \quad \frac{1}{n-1} \leq \frac{w}{u} \leq \frac{2}{n}, \\ \frac{w}{2u} & \text{if} \quad \frac{2}{n} \leq \frac{w}{u} \leq 1, \\ 1 - \frac{u}{2w} & \text{if} \quad 1 \leq \frac{w}{u} \leq \frac{n}{2}, \\ 1 - \frac{2}{n} + \frac{2w}{n^2 u} & \text{if} \quad \frac{n}{2} \leq \frac{w}{u} \leq n-1, \\ 1 - \frac{2\theta'-2}{\theta' n^2} & \text{if} \quad n-1 \leq \frac{w}{u} \leq n, \\ 1 & \text{if} \quad n \leq \frac{w}{u}, \end{cases} \tag{6.1}$$

where $\theta := \lceil \frac{\frac{w}{u}}{1-(n-1)\frac{w}{u}} \rceil$ and $\theta' := \lceil \frac{\frac{u}{w}}{1-(n-1)\frac{u}{w}} \rceil$. In other words, the utility function (6.1) is the one which is obtained by player 1 in the Nash equilibrium of the single stage CBG. Note that CBG is a constant sum game[2] meaning that the sum of the utilities of both players equals to a constant which is the total reward of all battlefields. Note that in (6.1) we have normalized the total reward obtain from the battlefields to 1 so that wining each battlefield is worth $\frac{1}{n}$ to a player. Therefore, the utility of the second player is simply given by $U_2(u,w) = 1 - U_1(u,w)$.

[2]Constant sum games are strategically equivalent to zero sum games.

6.3.3 Dynamic CBG Game

Now let us consider two NSPs (players) which are allocating bandwidth to n user over a period of discrete times $k = 1, 2, \dots, K$. At each time instant k, each customer $i \in \{1, 2, \dots, n\}$ receives two bandwidth offers $u_i(k), w_i(k) \in \mathbb{R}^{\geq 0}$ from the NSPs and must accept one of them. Since each user selfishly tends to accept the better bandwidth offer, depending on whether $u_i(k)$ is greater than $w_i(k)$, the user will accept $u_i(k)$ from the first NSP rather than $w_i(k)$ from the second NSP. Denoting the total bandwidth allocated by the two NSPs at time instant k by $u_k = \sum_{i=1}^{n} u_i(k)$ and $w_k = \sum_{i=1}^{n} w_i(k)$, respectively, one can easily see that the maximum received payoff for player 1, assuming that player 2 distributes its w_k bandwidth units optimally, equals to $U_1(u_k, w_k)$, where $U_1(\cdot)$ is the function given by (6.1). In particular, the payoff for player 2 equals $U_2(u_k, w_k) = 1 - U_1(u_k, w_k)$.

If players were playing a single stage, then clearly the Nash equilibrium strategies for the single stage CBG would be the optimal strategies for both players. But when the game is played for $K > 1$ stages, it becomes very critical for each player on how to allocate his budget at different stages. Thus, at each stage k, players decide on the amount of budgets u_k, w_k to lease and announce it to the system, in which case they will play a CBG with budgets u_k, w_k. Here, we note that the actions of players in the dynamic bandwidth allocation are the sequence of allocated budgets over the horizon. Once these decisions are made, players' payoffs at stage k equal to the optimal payoffs of a single stage CBG with initial budgets u_k, and w_k. In order to capture the dynamics of the game over different stages, let us denote the initial budgets of the players by $x_1(0)$ and $x_2(0)$. Denoting the available budget at stage k for players 1 and 2 by $x_1(k)$, and $x_2(k)$, respectively, we let their budgets at the next time step $k + 1$ be equal to

$$\begin{aligned} x_1(k+1) &= x_1(k) - \alpha u_k + c_1, \\ x_2(k+1) &= x_2(k) - \alpha w_k + c_2, \end{aligned} \tag{6.2}$$

where $(1 - \alpha) \in (0, 1)$ is the rate of release of bandwidth from the earlier users, and $c_1, c_2 > 0$ are constant additional bandwidth for players 1 and 2, respectively. The idea for the expressions in (6.2) is that each player, let say player 1, has some budget $x_1(k)$

at the beginning of stage k, and it decides to spend u_k units in that stage. However, the released bandwidth for next stage is equal to $(1-\alpha)u_k$, which together with the constant bandwidth c_1 gives its available budget at the next time step:

$$\begin{aligned} x_1(k+1) &= x_1(k) - u_k + (1-\alpha)u_k + c_1 \\ &= x_1(k) - \alpha u_k + c_1. \end{aligned}$$

Finally, denoting the decisions of the players over the entire horizon by $\boldsymbol{u} := (u_1, \ldots, u_K)$ and $\boldsymbol{w} := (w_1, \ldots, w_K)$, the total payoffs received by players 1 and 2 equals

$$\begin{aligned} \hat{U}_1(\boldsymbol{u}, \boldsymbol{w}) &= \sum_{k=1}^{K} U_1(u_k, w_k), \\ \hat{U}_2(\boldsymbol{u}, \boldsymbol{w}) &= \sum_{k=1}^{K} U_2(u_k, w_k) = K - \hat{U}_1(\boldsymbol{u}, \boldsymbol{w}). \end{aligned} \tag{6.3}$$

Therefore, each player seeks to maximize its total accumulated payoff (6.3) subject to its own budget dynamics given by (6.2). This naturally defines a dynamic game between two service providers which we study next.

6.4 Dynamic Programming Solution for Saddle Point Strategies

First, we note that the dynamic CBG game is a special case of a 2-player zero-sum dynamic game.[3] This is because due to the structure of the CBG utility functions, any income by one of the players can be viewed as a loss for the other player.

Definition 6.1. *A pair of strategies $\boldsymbol{u}^* = (u_1^*, \ldots, u_K^*)$ and $\boldsymbol{w}^* = (w_1^*, \ldots, w_K^*)$ are called a saddle point strategy for the NSPs in the dynamic CBG game if*

$$\hat{U}_1(\boldsymbol{u}, \boldsymbol{w}^*) \le \hat{U}_1(\boldsymbol{u}^*, \boldsymbol{w}^*) \le \hat{U}_1(\boldsymbol{u}^*, \boldsymbol{w}),$$

where player 1 aims to maximize $\hat{U}_1$ while player 2 wants to minimize it.

Therefore, our goal in the remainder of this chapter is to find (approximate) the saddle point strategies for the NSPs (i.e., find Nash equilibrium points of the zero-sum dynamic CBG). In this regard, it has been shown earlier [202, Corrollary 6.2] that a

[3] More precisely, a 2-player dynamic constant-sum game.

pair of strategies form a saddle point for the dynamic CBG if and only if there exists a sequence of functions $V_k(\cdot)$ such that $V_{K+1}(x) = 0, \forall x$, and we have

$$\begin{aligned} &V_k(x_1(k), x_2(k)) \\ &\quad = \min_w \max_u \Big\{ U_1(u, w) + V_{k+1}(x_1(k+1), x_2(k+1)) \Big\} \\ &\quad = \max_u \min_w \Big\{ U_1(u, w) + V_{k+1}(x_1(k+1), x_2(k+1)) \Big\}, \end{aligned} \tag{6.4}$$

in which case the corresponding solution vectors $\boldsymbol{u}^*$ and $\boldsymbol{w}^*$ will form saddle point strategies for the player, and $V_0(x_1(0), x_2(0))$ will be the value of the game. Therefore, in order to find the value function $V_k(\cdot), k = 1, 2, \dots, K$, we can use dynamic programming (DP) as is shown in Algorithm 5.

Algorithm 5 Saddle point strategy (DP)

Initialize: $V_{K+1}(.) = 0$

for each step $k = K$ downto 1 **do**

Fix $x_1(k)$ and $x_2(k)$. Find the optimal actions u_k^*, w_k^*, such that

$$\begin{aligned} &U_1(u^*, w^*) + V_{k+1}\Big(x_1(k) - \alpha u_k^* + c_1, x_2(k) - \alpha w_k^* + c_2\Big) \\ &= \min_w \max_u \Big\{ U_1(u, w) + V_{k+1}\Big(x_1(k) - \alpha u + c_1, x_2(k) - \alpha w + c_2\Big) \Big\} \\ &= \max_u \min_w \Big\{ U_1(u, w) + V_{k+1}\Big(x_1(k) - \alpha u + c_1, x_2(k) - \alpha w + c_2\Big) \Big\} \end{aligned} \tag{6.5}$$

and the corresponding value function,

$$\begin{aligned} V_k(x_1(k), x_2(k)) &= U_1(u^*, w^*) \\ &\quad + V_{k+1}\Big(x_1(k) - \alpha u_k^* + c_1, x_2(k) - \alpha w_k^* + c_2\Big). \end{aligned}$$

Output: sequence $u_K^*, w_K^*, V_K(.), ..., u_0^*, w_0^*, V_0(.)$.

end for

Although Algorithm 5 provides a systematic way of finding saddle point strategies for the players, however, it has some computational limitations. For example, due to the complex structure of utility function $U_1(\cdot)$ given by (6.1), any time that we are solving the dynamic programming equation (6.5) backward, one need to consider 8

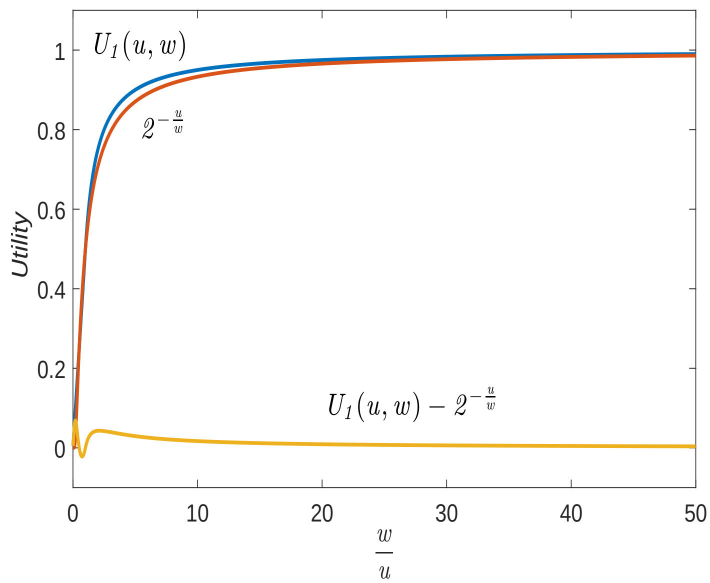


Figure 6.2: Illustrative of $U_1(u, w)$ (blue curve), its approximation $2^{-\frac{u}{w}}$ (red curve), and their difference error (yellow curve). As it can be seen $2^{-\frac{u}{w}}$ is very close to $U_1(u, w)$, and asymptotically they are identical.

different possibilities depending on what range $\frac{w}{u}$ lies in, which in turns requires 8^K case analysis if we want to solve the DP for a horizon of length K. Therefore, in the next section, our goal is to address this issue and provide a practical approach for finding (approximating) the saddle point strategies.

6.5 Approximating Saddle Point Strategies

A closer look at the utility function (6.1) shows that this function only depends on the ratio of the budgets $\frac{w}{u}$, and not their actual values u or w. In particular, for large number of customers $n >> 1$, one can easily see that only the fourth and fifth criteria in this function play an important role, i.e., when $\frac{2}{n} \leq \frac{w}{u} \leq \frac{n}{2}$. For other ranges of $\frac{w}{u}$, the function $U_1(u, w)$ is either very close to 0 or very close to 1. Therefore, instead of working directly with utility function $U_1(\cdot)$, we approximate it using the exponential function $2^{-\frac{u}{w}}$. Figure 6.2 shows the approximation for $n = 100$. In fact, one can easily check that this function is a very good approximation of the utility function $U_1(\cdot)$ not only in the critical range of $\frac{2}{n} \leq \frac{w}{u} \leq \frac{n}{2}$, but also outside of this range, and asymptotically matches $U_1(\cdot)$, as $\frac{w}{u} \to \infty$. We have illustrated this smooth function using the red curve in Figure 6.2.

Therefore, instead of $U_1(u,w)$, we use the smooth function $2^{-\frac{u}{w}}$ in our DP analysis, by only loosing a small fractional error which becomes negligible as the number of customers increases. This in turn eliminates the case dependent analysis of dealing with non-smooth function $U_1(u,w)$. As a result, the saddle point strategies of the approximated DP with $U_1(u,w)$ replaced by $2^{-\frac{u}{w}}$ can be solved efficiently after at most $O(K^2)$ iterations. In particular, in the following theorem we characterize in a closed form this approximated saddle point strategies of the NSPs given that the rate of released bandwidth $(1-\alpha)$ is not very large.

Theorem 6.2. *If $c_i \leq (2\alpha-1)x_i(0), i=1,2$, then the unique approximated saddle point strategies of the players are given by*

$$u_k^* = \frac{x_1(k-1)+(K-k)c_1}{(K-k+1)\alpha}, \quad w_k^* = \frac{x_2(k-1)+(K-k)c_2}{(K-k+1)\alpha}.$$

In particular, the value of the game equals to

$$V_0(x_1(0),x_2(0)) = K \times 2^{-\frac{x_1(0)+(K-1)c_1}{x_2(0)+(K-1)c_2}}.$$

Proof. Let assume that $K=1$, i.e., the game has only one stage. In this case, one can easily see that if the initial budgets are $x_1(0)$ and $x_2(0)$, its is the best for both players to use all of their bandwidth, in which case the payoff of the first player equals to $2^{-\frac{x_1(0)}{x_2(0)}}$ (consequently, the payoff of the second player equals $1-2^{-\frac{x_1(0)}{x_2(0)}}$). Now for $K=2$, i.e., when the game is played for two stages, let assume that the first and second players invest u and w units of their budget in the first time step $k=1$, and the remaining $x_1(0)-\alpha u+c_1$ and $x_2(0)-\alpha w+c_2$ budgets on the second time step $k=2$. Therefore, the maximum guaranteed payoff for both players is given by $\max_w \min_u\{2^{-\frac{u}{w}}+2^{-\frac{x_1(0)-\alpha u+c_1}{x_2(0)-\alpha w+c_2}}\}$, which must be equal to $\min_u \max_w\{2^{-\frac{u}{w}}+2^{-\frac{x_1(0)-\alpha u+c_1}{x_2(0)-\alpha w+c_2}}\}$, where $u \in [0,x_1(0)]$, and $w \in [0,x_2(0)]$. In order to solve this max-min problem, by taking the partial derivatives of the objective function with respect to u and w and letting them equal to zero, we obtain

$$\begin{aligned} \frac{\alpha u}{x_1(0)-\alpha u+c_1} &= 2^{(\frac{x_1(0)-\alpha u+c_1}{x_2(0)-\alpha w+c_2}-\frac{u}{w})}, \\ \frac{\alpha u^2(x_2(0)-\alpha w+c_2)}{w(x_1(0)-\alpha u+c_1)^2} &= 2^{(\frac{x_1(0)-\alpha u+c_1}{x_2(0)-\alpha w+c_2}-\frac{u}{w})}. \end{aligned} \tag{6.6}$$

Since their right-hand sides in (6.6) are identical, we get $\frac{u}{w} = \frac{x_1(0)-\alpha u+c_1}{x_2(0)-\alpha w+c_2}$. Finally, replacing this relation into (6.6) and solving for u, and w, we get $u_1^* = \frac{x_1(0)+c_1}{2\alpha}$, and $w_1^* = \frac{x_2(0)+c_2}{2\alpha}$. Now since $c_i \leq (2\alpha-1)x_i(0), i=1,2$, we get $u^* \leq x_1(0)$, and $w^* \leq x_2(0)$. This shows that indeed the unique solution of (6.6) lies in the valid range of budgets (see Figure 6.3). Hence, the game admits a unique saddle point strategy given by $(u_1^*, x_1(0)-\alpha u_1^*+c_1)$ and $(w_1^*, x_1(0)-\alpha w_1^*+c_2)$, and the game value equals $2\times 2^{-\frac{u^*}{w^*}} = 2\times 2^{-\frac{x_1(0)+c_1}{x_2(0)+c_2}}$.

To complete the proof using induction, let assume that the theorem statement is correct for the K stage game. Consider a $K+1$ stage game and denote the investments of the players at the first stage by u_1, and w_1, respectively. Using induction hypothesis, the value of the K stage game with initial budgets u and w is given by $K \times 2^{-\frac{u+(K-1)c_1}{w+(K-1)c_2}}$. Therefore, using the principle of optimality, u_1^* and w_1^* constitute the first stage actions of the saddle point strategies if and only if they are the solution of the following min-max equation:

$$\min_u \max_w \{2^{-\frac{u}{w}} + K \times 2^{-\frac{x_1(0)-\alpha u+Kc_1}{x_2(0)-\alpha w+Kc_2}}\}$$
$$= \max_u \min_w \{2^{-\frac{u}{w}} + K \times 2^{-\frac{x_1(0)-\alpha u+Kc_1}{x_2(0)-\alpha w+Kc_2}}\}.$$

By taking the partial derivative of the above function and using the same argument as in (6.6), one can obtain the unique solution of $u_1^* = \frac{x_1(0)+Kc_1}{(K+1)\alpha}$, and $w_1^* = \frac{x_2(0)+Kc_2}{(K+1)\alpha}$, with the value function being equal to $V_0(x_1(0), x_2(0)) = (K+1) \times 2^{-\frac{x_1(0)+Kc_1}{x_2(0)+Kc_2}}$.

We are only left to show that the sequence of u_k^*, w_k^* satisfy their budget constraints, i.e., $u_k^* \in [0, x_1(k-1)]$, and $w_k^* \in [0, x_2(k-1)]$, for all $k = 1, \ldots, K-1$. This can be shown using induction on K and the assumption on α. For $k = 1$ we must have $\frac{x_1(0)+(K-1)c_1}{K\alpha} \leq x_1(0)$ and $\frac{x_2(0)+(K-1)c_2}{K\alpha} \leq x_2(0)$, or equivalently $\frac{c_1}{x_1(0)} \leq \frac{\alpha K-1}{K-1}$, and $\frac{c_2}{x_2(0)} \leq \frac{\alpha K-1}{K-1}$. By repeating this process for every k, we get the following chain of inequalities:

$$\frac{c_2}{x_2(0)}, \frac{c_1}{x_1(0)} \leq \alpha + \frac{\alpha-1}{K-k}, \ \forall k = 1, \ldots, K-1.$$

Since the right hand side of these inequalities are minimized for $k = K-1$, thus it is sufficient to have

$$\frac{c_2}{x_2(0)}, \frac{c_1}{x_1(0)} \leq 2\alpha - 1,$$

which clearly holds by the assumption. This completes the proof. □

As a corollary of Theorem 6.2, if the constant reward budget (c_i) is proportional to the initial budget divided by the number of game stages, i.e., $c_i = \frac{x_i(0)}{K}$, then for $\alpha \in [\frac{1}{2} + \frac{1}{2K}, 1]$, the game admits a unique saddle point strategies which are given in Theorem 6.2.

Remark 6.3. *The optimal policies in Theorem 6.2 are written based on feedback policies. However, one can rewrite these policies in an open loop recursively using the dynamics* (6.2)*, to show that the budget investment for all the stages must be the same and equal to* $u_k^* = \frac{x_1(0)+(K-1)c_1}{K\alpha}$ *and* $w_k^* = \frac{x_2(0)+(K-1)c_2}{K\alpha}$, $\forall k = 2, \ldots, K-1$ *and for* $k = 1$, $u_k^* = x_1(0)$ *and* $w_k^* = x_2(0)$.

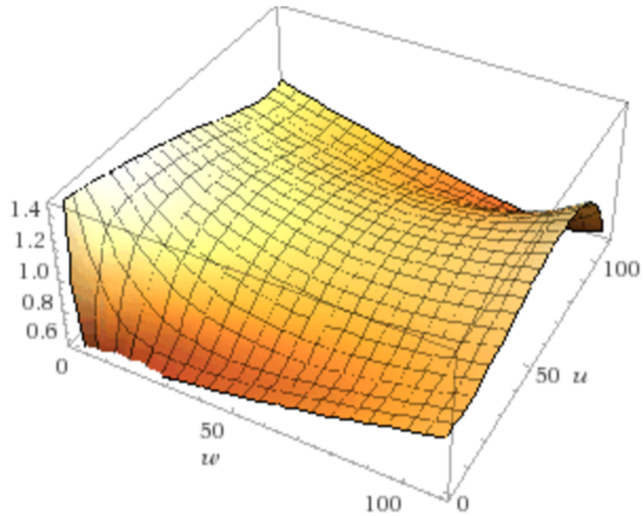


Figure 6.3: Illustration of the max-min argument $2^{-\frac{u}{w}} - 2^{-(\frac{x_1(0)-\alpha u+c_1}{x_2(0)-\alpha w+c_2})}$, for $\alpha = 0.8$, $c_1 = 2.5, c_2 = 1$, and initial budgets $x_1(0) = 90$, and $x_2(0) = 100$. As it can be seen this function has a unique saddle point in the budget range $[0, 90] \times [0, 100]$.

6.6 Numerical Results

In this section, we provide several simulation results to validate the dynamic programming approach for the approximated utility function and determine the value function and optimal policies for the NSPs. We consider a network with two NSPs competing to provide service wireless service to $n = 200$ users.

The value function for the Dynamic CBG for $\alpha = 0.8$ (the rate of releasing BW $= 0.2$), $c_1 = 3$, $c_2 = 1$ after $K = 10$ iteration is depicted in Fig.6.4. It shows that the value function follows an exponential structure as the utility function also takes an exponential form in terms of the NSPs' relative action $\frac{u}{w}$. If one of the NSPs dominates the other one with more initial resources (spectrum), it can have the privilege of playing aggressively and winning more users over time. On the other hand, a balanced initial budget makes the two NSPs to manage their budget more carefully in order to compete over the users in the next time slots.

Fig. 6.5, shows the value function for the NSP 1 after $K = 10$ stages of game play when the utility function is following equation (6.1) comparing to approximated utility function $2^{-\frac{u}{w}}$. As it can be seen, our approximation scheme is very accurate when the initial budgets of the NSPs are relatively close to each other.

Fig. 6.6 illustrates the effects of the number of game stages and the rate of release

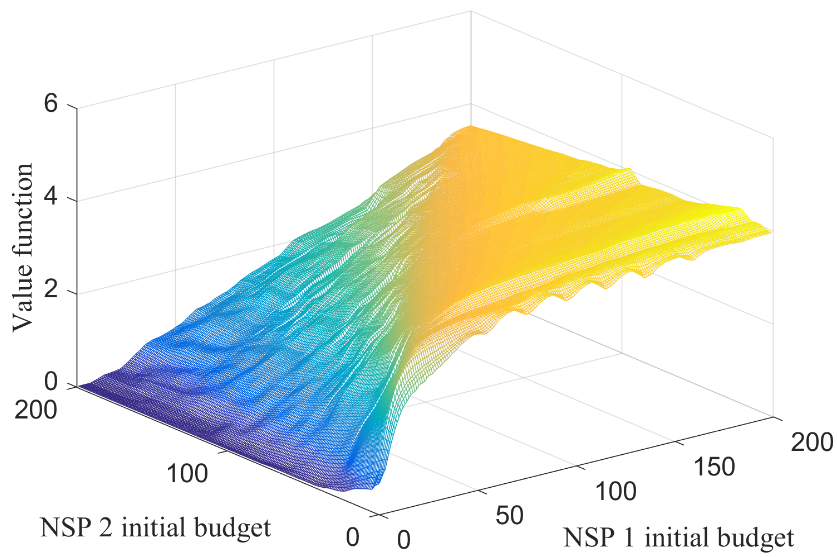


Figure 6.4: Illustrative of the value functions for exact utility $U_1(u,w)$.

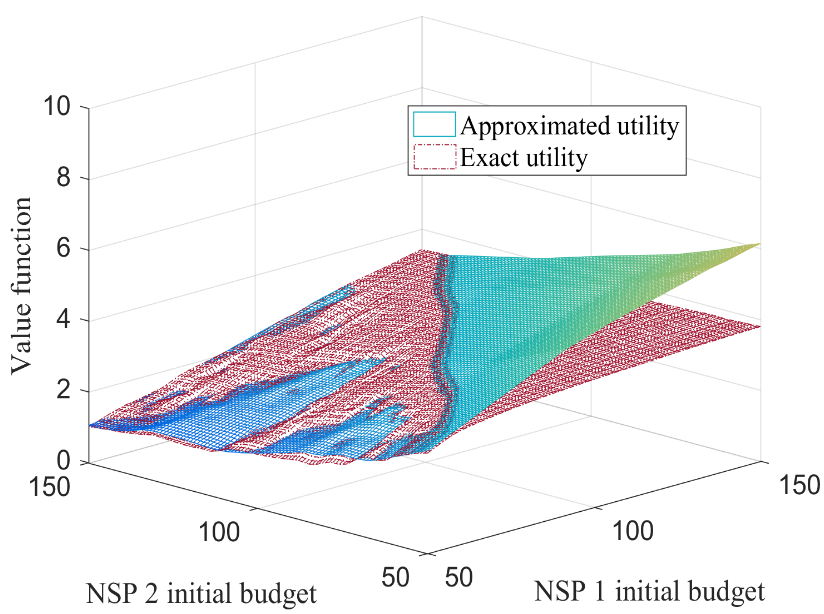


Figure 6.5: Illustrative of the value functions for exact utility $U_1(u,w)$ and its approximation $2^{-\frac{u}{w}}$.

of bandwidth $(1-\alpha)$ on the NSP 1 optimal policy when the NSP 1's initial budget is 190 and NSP 2's is 200. As mentioned in Remark 1, the budget investment for all the

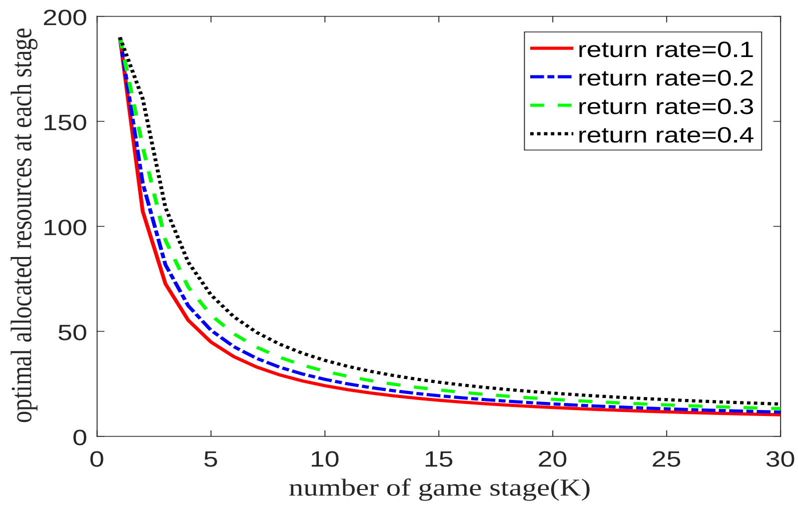


Figure 6.6: NSP 1 optimal policy versus number of game stages (K) for bandwidth return rate equal to 0.1, 0.2, 0.3, 0.4.

stages must be the same, but it depends on the number of the stages that the game is played. By looking at the dynamics of the NSPs (equation 6.2), we can find out that the NSPs' budget decrease over the horizon. So, when the number of game stages increases, the NSPs try to adjust their optimal policies by decreasing the number of invested budget at each step. On the other hand, when the NSP's rate of release of bandwidth increases, i.e, more users give back their leased bandwidth at each time slot, the NSPs are willing to invest more of their promised bandwidth at each time slots.

6.7 Conclusion

This chapter considered the problem of competitive spectrum allocation under certain dynamics where users are free to choose their network service providers (NSPs) in an opportunistic manner. We studied the scenario of two network service providers (NSPs) who are competing over a period of time to allocate spectrum among a pool of users seeking wireless connectivity. We showed that the dynamic process of spectrum allocation can be described by a zero-sum dynamic game with a cost function based on Colonel Blotto game–a multidimensional strategic resource allocation game. We leveraged this dynamic game formulation to determine optimal strategies for NSPs over a finite time horizon. In particular, we obtained saddle point strategies for the NSPs by approximating the instantaneous payoff functions using simple smooth functions, and used a dynamic programming (DP) approach to provide closed form solutions for the optimal strategies. We envisioned that such an approximation closely mimics the structure of the dynamic CBG game which we validated using our simulations results. As a future direction of research, one can consider the extension of our results to the scenarios where the number of users dynamically changes over the horizon. Also, learning saddle point strategies when the return bandwidth rates follow certain stochastic processes is another interesting research direction.

Chapter 7

Conclusion and future work

7.1 Summary of Research

This thesis focused on the design of information centric Networks and spectrum sharing using game theoretical approaches. The results from this can be broadly categorized into three parts. First, we have highlighted the importance of ICN as the potential candidate for the design of future Internet and provided a business model for distributing the popular content throughout the network. Second, we have proposed a defense strategy against Advanced Persistent Threat (APT) for cloud storage systems using Colonel Blotto Game (CBG). Thirdly, we have studied the dynamics of competitive spectrum allocation in wireless networks.

7.1.1 Joint Caching and Pricing Strategies for Information Centric Networks

We explored various aspects of the ICN including business model. We developed a game theoretical framework for distribution of popular content in a scenario consisting of access ICNs, a transit ICN and a content provider. By assuming that the caching cost of the access ICNs and transit ICNs is inversely proportional to popularity, which follows a generalized Zipf distribution, we first showed that at the NE, the caching strategies turn out to be 0-1 (all or nothing). Further, for the case of symmetric access ICNs, we showed that a unique *NE* exists and the caching policy (0 or 1) is determined by a threshold on the popularity of the content, i.e., all content more popular than the threshold value is cached. Moreover, we showed that the resulting threshold indices and prices can be obtained by a decomposition of the joint caching and pricing problem into two independent caching only and pricing only problems. We also highlighted the importance of having a monetary incentive mechanism in order to bring all the network components in cooperation for having better performance. In this project, we discussed a hierarchical scenario with K access ICNs, one transit ICN and one content provider and global popularity throughout the network. Content caching by considering local popularity for different regions is a challenging problem that needs to be addressed using machine learning approaches.

7.1.2 Defense Against Advanced Persistent Threat

Specifically, we modeled the computational resource (or CPU) allocation problem under APT attacks as a two-player zero-sum game, where the defender aims at maximizing the data protection level of the cloud system by randomizing the amounts of CPUs allocated to each cloud device, which is converted to a Colonel Blotto Game. We derived the NE of the static APT defense game, and investigated the impacts of the data size, the number of devices and the CPU resource constraints on the data protection level and the defender's utility. We also proposed a policy hill-climbing based APT defense strategy for the defender to scan the devices while the attack model is unknown in the dynamic defense game.

7.1.3 Dynamic Competitive Spectrum Allocation in Wireless Networks

At last, we introduced a novel approach to the inter-network spectrum allocation problem using the framework of Colonel Blotto Game. We considered two NSP with limited resources which are competing to serve the users in a geographical region. We proposed a learning algorithm based on fictitious play to numerically compute the mixed strategies that achieve Nash equilibrium (NE). Moreover, we introduced a dynamic noncooperative repeated game as the decentralized approach for the Network service providers to determine optimal strategies over a finite time horizon. This problem is been modeled as a zero-sum dynamic game (ZSDG) and the optimal equilibrium strategies for the NSPs reduces to finding the saddle point strategies. We used dynamic programming to approach to obtain closed form solutions for the value function and the saddle point strategies. A scenario of multidimensional resources (e.g, bandwidth and power) can be of interest for future research.

References

[1] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, 2009.

[2] J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Müller, M.-S. Li *et al.*, *The Future Internet-Future Internet Assembly 2011: Achievements and Technological Promises.* Springer, 2011, vol. 6656.

[3] "Internet live stat." [Online]. Available: http://www.internetlivestats.com/

[4] F. Bronzino, K. Nagaraja, I. Seskar, and D. Raychaudhuri, "Network service abstractions for a mobility-centric future internet architecture," in *Proceedings of the eighth ACM international workshop on Mobility in the evolving internet architecture.* ACM, 2013, pp. 5–10.

[5] S. Paul, R. Yates, D. Raychaudhuri, and J. Kurose, "The cache-and-forward network architecture for efficient mobile content delivery services in the future internet," in *Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference.* IEEE, 2008, pp. 367–374.

[6] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 181–192.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies.* ACM, 2009, pp. 1–12.

[8] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 3, pp. 2–13, 2012.

[9] S. Gopinath, S. Jain, S. Makharia, and D. Raychaudhuri, "An experimental study of the cache-and-forward network architecture in multi-hop wireless scenarios," in *Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on.* IEEE, 2010, pp. 1–6.

[10] J. Pan, R. Jain, S. Paul, and C. So-In, "Milsa: A new evolutionary architecture for scalability, mobility, and multihoming in the future internet," *IEEE journal on selected areas in communications*, vol. 28, no. 8, pp. 1344–1362, 2010.

[11] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, "Mobilityfirst future internet architecture project," in *Proceedings of the 7th Asian Internet Engineering Conference*. ACM, 2011, pp. 1–3.

[12] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Communications Magazine*, vol. 49, no. 7, 2011.

[13] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, 2010.

[14] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future internet research and experimentation: the fire initiative," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 89–92, 2007.

[15] A.-M. K. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report*, vol. 4, 2007.

[16] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, 2012.

[17] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[18] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 26–33, 2010.

[19] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf)–an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721–735, 2013.

[20] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "Lipsin: line speed publish/subscribe inter-networking," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 195–206, 2009.

[21] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *IEEE Communications Magazine*, vol. 49, no. 3, 2011.

[22] B. Ahlgren, M. Dambrosio, C. Dannewitz, A. Eriksson, J. Golic, B. Grönvall, D. Horne, A. Lindgren, O. Mämmelä, M. Marchisio *et al.*, "Second netinf architecture description," *4WARD EU FP7 Project, Deliverable D-6.2 v2. 0*, 2010.

[23] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone, "Design considerations for a network of information," in *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008, p. 66.

[24] A. Anand, F. Dogar, D. Han, B. Li, H. Lim, M. Machado, W. Wu, A. Akella, D. G. Andersen, J. W. Byers *et al.*, "Xia: An architecture for an evolvable and trustworthy internet," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. ACM, 2011, p. 2.

[25] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," *IEEE Communications Magazine*, vol. 50, no. 7, 2012.

[26] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.

[27] W. Wang, Y. Sun, Y. Guo, D. Kaafar, J. Jin, J. Li, and Z. Li, "Crcache: Exploiting the correlation between content popularity and network topology information for icn caching," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3191–3196.

[28] Y. Li, H. Xie, Y. Wen, and Z.-L. Zhang, "Coordinating in-network caching in content-centric networks: Model and analysis," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 62–72.

[29] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[30] K. Suksomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, N. Motonori, M. Aoki, S. Urushidani, and S. Yamada, "Popcache: Cache more or less based on content popularity for information-centric networking," in *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*. IEEE, 2013, pp. 236–243.

[31] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in cyberspace: defining tomorrow's internet," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 347–356.

[32] S. Shakkottai and R. Srikant, "Economics of network pricing with multiple isps," *IEEE/ACM Transactions On Networking*, vol. 14, no. 6, pp. 1233–1245, 2006.

[33] X.-R. Cao, H.-X. Shen, R. Milito, and P. Wirth, "Internet pricing with a game theoretical approach: concepts and examples," *IEEE/ACM transactions on networking*, vol. 10, no. 2, pp. 208–216, 2002.

[34] L. He and J. Walrand, "Pricing and revenue sharing strategies for internet service providers," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1. IEEE, 2005, pp. 205–216.

[35] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. V. Vazirani, "How many tiers?: pricing in the internet transit market," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 194–205.

[36] Y. Wu, H. Kim, P. H. Hande, M. Chiang, and D. H. Tsang, "Revenue sharing among isps in two-sided markets," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 596–600.

[37] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma, "Incentive-compatible caching and peering in data-oriented networks," in *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008, p. 62.

[38] K. Seo and S. Kent, "Security architecture for the internet protocol," 2005.

[39] C. Douligeris and A. Mitrokotsa, "Ddos attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, vol. 44, no. 5, pp. 643–666, 2004.

[40] E. Skoudis and T. Liston, *Counter hack reloaded: a step-by-step guide to computer attacks and effective defenses*. Prentice Hall Press, 2005.

[41] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant, "Semantics-aware malware detection," in *Security and Privacy, 2005 IEEE Symposium on*. IEEE, 2005, pp. 32–46.

[42] J. Hong, "The state of phishing attacks," *Communications of the ACM*, vol. 55, no. 1, pp. 74–81, 2012.

[43] J. Schonwalder, M. Fouquet, G. D. Rodosek, and I. C. Hochstatter, "Future internet= content+ services+ management," *IEEE Communications Magazine*, vol. 47, no. 7, 2009.

[44] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1441–1454, 2015.

[45] C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Network security*, vol. 2011, no. 8, pp. 16–19, 2011.

[46] J. Vukalović and D. Delija, "Advanced persistent threats-detection and defense," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*. IEEE, 2015, pp. 1324–1330.

[47] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. IEEE, 2012, pp. 257–260.

[48] J. Zander, "Radio resource management in future wireless networks: Requirements and limitations," *IEEE Communications magazine*, vol. 35, no. 8, pp. 30–36, 1997.

[49] J. Zander, S.-L. Kim, M. Almgren, and O. Queseth, *Radio Resource Management for Wireless Networks*. Norwood, MA, USA: Artech House, Inc., 2001.

[50] Z. Han and K. R. Liu, *Resource allocation for wireless networks: basics, techniques, and applications*. Cambridge university press, 2008.

[51] E. Hossain, D. Niyato, and Z. Han, *Dynamic spectrum access and management in cognitive radio networks*. Cambridge university press, 2009.

[52] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE signal processing magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[53] C. Santivanez, R. Ramanathan, C. Partridge, R. Krishnan, M. Condell, and S. Polit, "Opportunistic spectrum access: Challenges, architecture, protocols," in *Proceedings of the 2nd annual international workshop on Wireless internet.* ACM, 2006, p. 13.

[54] O. Ileri and N. B. Mandayam, "Dynamic spectrum access models: Toward an engineering perspective in the spectrum debate," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 153–160, 2008.

[55] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "Dimsumnet: new directions in wireless networking using coordinated dynamic spectrum," in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a.* Ieee, 2005, pp. 78–85.

[56] V. Brik, E. Rozner, S. Banerjee, and P. Bahl, "Dsap: a protocol for coordinated spectrum access," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on.* IEEE, 2005, pp. 611–614.

[57] S. Sengupta, M. Chatterjee, and S. Ganguly, "An economic framework for spectrum allocation and service pricing with competitive wireless service providers," in *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN).* IEEE, 2007, pp. 89–98.

[58] Y. Xie, B. Armbruster, and Y. Ye, "Dynamic spectrum management with the competitive market model," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2442–2446, 2010.

[59] "Project Fi," *[Online] Available: http://fi.google.com/.*

[60] M. J. Osborne, *An introduction to game theory.* Oxford university press New York, 2004, vol. 3, no. 3.

[61] M. Felegyhazi and J.-P. Hubaux, "Game theory in wireless networks: A tutorial," Tech. Rep., 2006.

[62] D. Fudenberg and J. Tirole, "Game theory," The MIT press, Tech. Rep., 1991.

[63] R. Gibbons, *A primer in game theory.* Harvester Wheatsheaf, 1992.

[64] M. J. Osborne and A. Rubinstein, *A course in game theory.* MIT press, 1994.

[65] H. A. Simon, *The sciences of the artificial.* MIT press, 1996.

[66] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter, "A survey on networking games in telecommunications," *Computers & Operations Research*, vol. 33, no. 2, pp. 286–311, 2006.

[67] A. Ozdaglar, "Networks challenge: Where game theory meets network optimization," in *International Symposium on Information Theory*, 2008.

[68] Z. Han, *Game theory in wireless and communication networks: theory, models, and applications.* Cambridge University Press, 2012.

[69] L. Dasilva, H. Bogucka, and A. Mackenzie, “Game theory in wireless networks,” *IEEE Communications Magazine*, vol. 49, no. 8, pp. 110–111, 2011.

[70] A. A. Lazar, A. Orda, and D. E. Pendarakis, “Virtual path bandwidth allocation in multiuser networks,” *IEEE/ACM transactions on networking*, vol. 5, no. 6, pp. 861–871, 1997.

[71] D. Niyato and E. Hossain, “Radio resource management games in wireless networks: an approach to bandwidth allocation and admission control for polling service in ieee 802.16 [radio resource management and protocol engineering for ieee 802.16],” *IEEE Wireless Communications*, vol. 14, no. 1, pp. 27–35, 2007.

[72] H. Yaïche, R. R. Mazumdar, and C. Rosenberg, “A game theoretic framework for bandwidth allocation and pricing in broadband networks,” *IEEE/ACM Transactions On Networking*, vol. 8, no. 5, pp. 667–678, 2000.

[73] M. Chiang, P. Hande, T. Lan, C. W. Tan *et al.*, “Power control in wireless cellular networks,” *Foundations and Trends® in Networking*, vol. 2, no. 4, pp. 381–533, 2008.

[74] T. Alpcan, T. Başar, R. Srikant, and E. Altman, “Cdma uplink power control as a noncooperative game,” *Wireless Networks*, vol. 8, no. 6, pp. 659–670, 2002.

[75] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, “Pricing and power control in a multicell wireless data network,” *IEEE Journal on selected areas in communications*, vol. 19, no. 10, pp. 1883–1892, 2001.

[76] ——, “Efficient power control via pricing in wireless data networks,” *IEEE transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[77] S. Kunniyur and R. Srikant, “End-to-end congestion control schemes: Utility functions, random losses and ecn marks,” *IEEE/ACM Transactions on networking*, vol. 11, no. 5, pp. 689–702, 2003.

[78] T. Alpcan and T. Basar, “A game-theoretic framework for congestion control in general topology networks,” in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2. IEEE, 2002, pp. 1218–1224.

[79] T. Roughgarden, *Selfish routing and the price of anarchy.* MIT press Cambridge, 2005, vol. 174.

[80] E. Altman, T. Basar, T. Jimenez, and N. Shimkin, “Competitive routing in networks with polynomial costs,” *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 92–96, 2002.

[81] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, “Selfish caching in distributed systems: a game-theoretic analysis,” in *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing.* ACM, 2004, pp. 21–30.

[82] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on.* IEEE, 2010, pp. 1–10.

[83] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 25, 2013.

[84] M. Tambe, *Security and game theory: algorithms, deployed systems, lessons learned.* Cambridge University Press, 2011.

[85] M. Hajimirsadeghi, N. B. Mandayam, and A. Reznik, "Joint caching and pricing strategies for information centric networks," in *Global Communications Conference (GLOBECOM), 2015 IEEE.* IEEE, 2015, pp. 1–6.

[86] ——, "Joint caching and pricing strategies for popular content in information centric networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 654–667, 2017.

[87] M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, "Defense against advanced persistent threats: A colonel blotto game approach," in *Communications (ICC), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1–6.

[88] ——, "Defense against advanced persistent threats in dynamic cloud storage: A colonel blotto game approach," *arXiv preprint arXiv:1801.06270*, 2018.

[89] M. Hajimirsadeghi, G. Sridharan, W. Saad, and N. B. Mandayam, "Inter-network dynamic spectrum allocation via a Colonel Blotto game," in *Proc. IEEE Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, Mar. 2016, pp. 252–257.

[90] M. Hajimirsaadeghi and N. B. Mandayam, "A dynamic colonel blotto game model for spectrum sharing in wireless networks," in *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on.* IEEE, 2017, pp. 287–294.

[91] P. Pandey, M. Hajimirsadeghi, and D. Pompili, "Region of feasibility of interference alignment in underwater sensor networks," *IEEE journal of oceanic Engineering*, vol. 39, no. 1, pp. 189–202, 2014.

[92] A. A. A. Abass, M. Hajimirsadeghi, N. B. Mandayam, and Z. Gajic, "Evolutionary game theoretic analysis of distributed denial of service attacks in a wireless network," in *Information Science and Systems (CISS), 2016 Annual Conference on.* IEEE, 2016, pp. 36–41.

[93] M. Yousefvand, M. Hajimirsadeghi, and N. B. Mandayam, "Impact of end-user behavior on user/network association in hetnets," *arXiv preprint arXiv:1806.03213*, 2018.

[94] A. Afanasyev, J. Burke, L. Zhang, K. Claffy, L. Wang, V. Jacobson, P. Crowley, C. Papadopoulos, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[95] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[96] S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "Comparison of naming schema in icn," in *Local and Metropolitan Area Networks (LANMAN), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1–6.

[97] K. K. Leung, "Mobile ip mobility agent standby protocol," Feb. 27 2001, uS Patent 6,195,705.

[98] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe, "A survey of mobility in information-centric networks," *Communications of the ACM*, vol. 56, no. 12, pp. 90–98, 2013.

[99] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.

[100] "NSF named data networking project." [Online]. Available: [Online].Available: http://www.named-data.net/

[101] "FP7 pursuit project." [Online]. Available: [Online].Available:http://www.fp7-pursuit.eu/PursuitWeb/

[102] L. A. Adamic and B. A. Huberman, "Zipf's law and the internet." *Glottometrics*, vol. 3, no. 1, pp. 143–150, 2002.

[103] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*. IEEE, 2008, pp. 229–238.

[104] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 15–28.

[105] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *ACM SIGOPS Operating Systems Review*, vol. 40, no. 4. ACM, 2006, pp. 333–344.

[106] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1357–1370, 2009.

[107] L. Shi, Z. Gu, L. Wei, and Y. Shi, "An applicative study of zipfs law on web cache," *International Journal of Information Technology*, vol. 12, no. 4, pp. 49–58, 2006.

[108] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 126–134.

[109] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, “Impact of traffic mix on caching performance in a content-centric network,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012, pp. 310–315.

[110] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, “Tussle in cyberspace: defining tomorrow’s internet,” *IEEE/ACM transactions on networking*, vol. 13, no. 3, pp. 462–475, 2005.

[111] T. X. Tran, A. Hajisami, and D. Pompili, “Cooperative hierarchical caching in 5g cloud radio access networks,” *IEEE Network*, vol. 31, no. 4, pp. 35–41, 2017.

[112] T. X. Tran and D. Pompili, “Octopus: A cooperative hierarchical caching strategy for cloud radio access networks,” in *Mobile Ad Hoc and Sensor Systems (MASS), 2016 IEEE 13th International Conference on*. IEEE, 2016, pp. 154–162.

[113] T. X. Tran, F. Kazemi, E. Karimi, and D. Pompili, “Mobee: Mobility-aware energy-efficient coded caching in cloud radio access networks,” in *Mobile Ad Hoc and Sensor Systems (MASS), 2017 IEEE 14th International Conference on*. IEEE, 2017, pp. 461–465.

[114] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, “Collaborative multi-bitrate video caching and processing in mobile-edge computing networks,” in *Wireless On-demand Network Systems and Services (WONS), 2017 13th Annual Conference on*. IEEE, 2017, pp. 165–172.

[115] M. Ji, G. Caire, and A. F. Molisch, “Wireless device-to-device caching networks: Basic principles and system performance,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 176–189, 2016.

[116] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5g wireless networks,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[117] K. Hamidouche, W. Saad, M. Debbah, and H. V. Poor, “Mean-field games for distributed caching in ultra-dense small cell networks,” in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 4699–4704.

[118] G. Zhang, B. Tang, X. Wang, and Y. Wu, “An optimal cache placement strategy based on content popularity in content centric network,” *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, vol. 11, no. 8, pp. 2759–2769, 2014.

[119] S. Guo, H. Xie, and G. Shi, “Collaborative forwarding and caching in content centric networks,” *NETWORKING 2012*, pp. 41–55, 2012.

[120] J. Tadrous, A. Eryilmaz, and H. El Gamal, “Joint pricing and proactive caching for data services: Global and user-centric approaches,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, 2014, pp. 616–621.

[121] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, “Efficient video pricing and caching in heterogeneous networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8744–8751, 2016.

[122] F. Shen, K. Hamidouche, E. Bastug, and M. Debbah, “A stackelberg game for incentive proactive caching mechanisms in wireless networks,” in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[123] P. K. Agyapong and M. Sirbu, “Economic incentives in information-centric networking: Implications for protocol design and public policy,” *IEEE Communications Magazine*, vol. 50, no. 12, 2012.

[124] M. Arifuzzaman, K. Yu, and T. Sato, “Collaboration between network players of information centric network: An engineering-economic analysis,” *Journal of ICT Standardization*, vol. 2, no. 3, pp. 201–222, 2015.

[125] T.-M. Pham, S. Fdida, and P. Antoniadis, “Pricing in information-centric network interconnection,” in *IFIP Networking Conference, 2013*. IEEE, 2013, pp. 1–9.

[126] E. Bastug, M. Bennis, and M. Debbah, “Anticipatory caching in small cell networks: A transfer learning approach,” in *1st KuVS Workshop on Anticipatory Networks*, 2014.

[127] J. Tadrous and A. Eryilmaz, “On optimal proactive caching for mobile networks with demand uncertainties,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2715–2727, 2016.

[128] P. Blasco and D. Gunduz, “Learning-based optimization of cache content in a small cell base station,” in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1897–1903.

[129] A. Sengupta, S. Amuru, R. Tandon, R. M. Buehrer, and T. C. Clancy, “Learning distributed caching strategies in small cell networks,” in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*. IEEE, 2014, pp. 917–921.

[130] E. Baştuğ, M. Bennis, and M. Debbah, “A transfer learning approach for cache-enabled wireless networks,” in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*. IEEE, 2015, pp. 161–166.

[131] M. Chen, W. Saad, C. Yin, and M. Debbah, “Echo state networks for proactive caching in cloud-based radio access networks with mobile users,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3520–3535, 2017.

[132] B. Bharath, K. Nagananda, and H. V. Poor, “A learning-based approach to caching in heterogenous small cell networks,” *IEEE Transactions on Communications*, vol. 64, no. 4, pp. 1674–1686, 2016.

[133] B. B. Nagaraja and K. G. Nagananda, “Caching with unknown popularity profiles in small cell networks,” in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[134] E. Baştuğ, M. Bennis, E. Zeydan, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, “Big data meets telcos: A proactive caching perspective,” *Journal of Communications and Networks*, vol. 17, no. 6, pp. 549–557, 2015.

[135] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[136] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[137] P. Giura and W. Wang, "Using large scale distributed computing to unveil advanced persistent threats," *Science*, vol. 1, no. 3, pp. 93–105, 2013.

[138] Y. Han, T. Alpcan, J. Chan, C. Leckie, and B. I. Rubinstein, "A game theoretical approach to defend against co-resident attacks in cloud computing: Preventing coresidence using semi-supervised learning," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 3, pp. 556–570, Dec. 2016.

[139] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, May 2011.

[140] X. Cheng, L. Fang, L. Yang, and S. Cui, "Mobile big data: The fuel for data-driven wireless," *IEEE Internet of Things J.*, vol. 4, no. 5, pp. 1489–1516, Oct. 2017.

[141] C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Network Security*, vol. 2011, no. 8, pp. 16–19, Aug. 2011.

[142] R. B. Sagi, "The Economic Impact of Advanced Persistent Threats," IBM Research Intelligence, May 2014, http://www-01.ibm.com.

[143] M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "Flipit: The game of "stealthy takeover"," *J. of Cryptology*, vol. 26, no. 4, pp. 655–713, Oct. 2013.

[144] M. Zhang, Z. Zheng, and N. B. Shroff, "Stealthy attacks and observable defenses: A game theoretic model under strict resource constraints," in *Proc. IEEE Global Conf. Signal and Inf. Processing (GlobalSIP)*, Atlanta, GA, Dec. 2014, pp. 813–817.

[145] L. Xiao, D. Xu, Y. Li, N. B. Mandayam, and H. V. Poor, "Cloud storage defense against advanced persistent threats: A prospect theoretic study," *IEEE J. Sel. Areas Commun*, in press.

[146] L. Xiao, D. Xu, N. B. Mandayam, and H. V. Poor, "Cumulative prospect theoretic study of a cloud storage defense game against advanced persistent threats," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), BigSecurity Workshop*, Atlanta, GA, May 2017, pp. 1–6.

[147] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 25:1–25:39, Jul. 2013.

[148] M. Zhang, Z. Zheng, and N. B. Shroff, "A game theoretic model for defending against stealthy attacks with limited resources," in *Proc. Int. Conf. Decision and Game Theory for Security*, London, UK, Nov. 2015, pp. 93–112.

[149] X. Cheng, L. Fang, X. Hong, and L. Yang, "Exploiting mobile big data: Sources, features, and applications," *IEEE Network*, vol. 31, no. 1, pp. 72–79, Feb. 2017.

[150] B. Roberson, "The Colonel Blotto game," *Economic Theory*, vol. 29, no. 1, pp. 1–24, Sep. 2006.

[151] M. Labib, S. Ha, W. Saad, and J. H. Reed, "A Colonel Blotto game for anti-jamming in the Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, Dec. 2015, pp. 1–6.

[152] N. Namvar, W. Saad, N. Bahadori, and B. Kelley, "Jamming in the Internet of Things: A game-theoretic perspective," *arXiv preprint arXiv:1607.06255*, 2016.

[153] M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, "Defense against advanced persistent threats: A Colonel Blotto game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[154] M. Bowling and M. Veloso, "Rational and convergent learning in stochastic games," in *Proc. Int. Joint Conf. Artificial Intelligence*, Seattle, Washhington, Aug. 2001, pp. 1021–1026.

[155] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[156] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Jan. 2015.

[157] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, Dec. 2013.

[158] K. D. Bowers, M. Van Dijk, R. Griffin, A. Juels, A. Oprea, R. L. Rivest, and N. Triandopoulos, "Defending against the unknown enemy: Applying FlipIt to system security," in *GameSec*. Springer, 2012, pp. 248–263.

[159] A. Laszka, B. Johnson, and J. Grossklags, "Mitigating covert compromises," in *Int. Conf. Web and Internet Econ.*, Cambridge, MA, Dec. 2013, pp. 319–332.

[160] Z. Zheng, N. B. Shroff, and P. Mohapatra, "When to reset your keys: Optimal timing of security updates via learning," in *Proc. AAAI Conf. Artificial Intelligence*, San Francisco, CA, Feb. 2017, pp. 1–7.

[161] J. Pawlick, S. Farhang, and Q. Zhu, "Flip the cloud: Cyber-physical signaling games in the presence of advanced persistent threats," in *Proc. Int. Conf. Decision and Game Theory for Security*, London, UK, Nov. 2015, pp. 289–308.

[162] A. Alabdel Abass, L. Xiao, N. B. Mandayam, and Z. Gajic, "Evolutionary game therotic analysis of advanced persistent threats against cloud storage," *IEEE Access*, vol. 5, pp. 8482–8491, Apr. 2017.

[163] P. Hu, H. Li, H. Fu, D. Cansever, and P. Mohapatra, "Dynamic defense strategy against advanced persistent threat with insiders," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, HK, May 2015, pp. 747–755.

[164] Y. Wu, B. Wang, K. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 4–15, Jan. 2012.

[165] P. Chia and J. Chuang, "Colonel Blotto in the phishing war," *Decision and Game Theory for Security*, pp. 201–218, 2011.

[166] R. Brewer, "Advanced persistent threats: Minimising the damage," *Network Security*, vol. 2014, no. 4, pp. 5–9, Apr. 2014.

[167] S. Siddiqui, M. S. Khan, K. Ferens, and W. Kinsner, "Detecting advanced persistent threats using fractal dimension based machine learning classification," in *Proc. ACM on Int. Workshop Security and Privacy Analytics*, New Orleans, LA, Mar. 2016, pp. 64–69.

[168] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Computers & Security*, vol. 48, pp. 35–57, Feb. 2015.

[169] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, Apr. 2011.

[170] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Int. Conf. Acous., Speech, Signal Processing (ICASSP)*, New Orleans, LA, Mar. 2017, pp. 1–5.

[171] C. Thomas, "N-dimensional Blotto game with heterogeneous battlefield values," *Economic Theory*, pp. 1–36, Jan. 2017.

[172] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[173] E. Rodrigues Gomes and R. Kowalczyk, "Dynamic analysis of multiagent Q-learning with ε-greedy exploration," in *Proc. ACM Annual Int'l. Conf. Mach. Learn. (ICML)*, Montreal, Canada, Jun. 2009, pp. 369–376.

[174] T. Rosowski, J. Kronander, T. Irnich, O. Queseth, M. Tercero, K. Chatzikokolakis, R. Arapoglou, K. Koufos, A. Widaa, J. Markendahl, H. Schotten, M. Uusitalo, L. Campoy, and C. Markwart, "Description of the spectrum needs and usage principles," METIS, Tech. Rep., Sep 2014. [Online]. Available: https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D5.3_v1.pdf

[175] R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "LTE in unlicensed spectrum using licensed-assisted access," *Proceedings of IEEE Global Communication Conference (GLOBECOM)*, pp. 746–751, Dec. 2014.

[176] M. Palola, M. Matinmikko, J. Prokkola, M. Mustonen, M. Heikkila, T. Kippola, S. Yrjola, V. Hartikainen, L. Tudose, A. Kivinen, J. Paavola, and K. Heiska, "Live field trial of licensed shared access (LSA) concept using LTE network in 2.3 ghz band," *Proceedings of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pp. 38–47, Apr. 2014.

[177] Y. Teng, Y. Wang, and K. Horneman, "Co-primary spectrum sharing for denser networks in local area," *Proceedings of EAI International Conference on Cognitive Radio Oriented Wireless Network and Communication (CROWNCOM)*, pp. 120–124, Jun. 2014.

[178] T. Basar, "Control and game-theoretic tools for communication networks," *Appl. Comput. Math*, vol. 6, no. 2, pp. 104–125, 2007.

[179] O. Gross and R. Wagner, "A continuous Colonel Blotto game," DTIC Document, Tech. Rep., 1950.

[180] B. Roberson and D. Kvasov, "The non-constant-sum colonel Blotto game," *Economic Theory*, vol. 51, no. 2, pp. 397–433, 2012.

[181] R. Golman and S. E. Page, "General blotto: games of allocative strategic mismatch," *Public Choice*, vol. 138, no. 3-4, pp. 279–299, 2009.

[182] G. W. Brown, "Iterative solution of games by fictitious play," *Activity analysis of production and allocation*, vol. 13, no. 1, pp. 374–376, 1951.

[183] P. Dasgupta and E. Maskin, "The existence of equilibrium in discontinuous economic games, i: Theory," *The Review of economic studies*, vol. 53, no. 1, pp. 1–26, 1986.

[184] M. Sion, "On general minimax theorems," *Pacific Journal of mathematics*, vol. 8, no. 1, pp. 171–176, 1958.

[185] C. Thomas, "N-dimensional Colonel Blotto game with asymmetric battlefield values," Working paper, The University of Texas at Austin. 2012.

[186] J.-F. Laslier and N. Picard, "Distributive politics and electoral competition," *Journal of Economic Theory*, vol. 103, no. 1, pp. 106–130, 2002.

[187] J. Weinstein, "Two notes on the blotto game," *The BE Journal of Theoretical Economics*, vol. 12, no. 1, 2012.

[188] S. Hart, "Discrete Colonel Blotto and General Lotto games," *International Journal of Game Theory*, vol. 36, no. 3-4, pp. 441–460, 2008.

[189] D. Monderer and L. S. Shapley, "Fictitious play property for games with identical interests," *Journal of Economic Theory*, vol. 68, no. 1, pp. 258–265, 1996.

[190] T. Başar, "Control and game-theoretic tools for communication networks," *Appl. Comput. Math*, vol. 6, no. 2, pp. 104–125, 2007.

[191] M. Pachter and K. Pham, "Discrete-time linear-quadratic dynamic games," *Journal of Optimization Theory and Applications*, vol. 146, no. 1, pp. 151–179, 2010.

[192] Q. Wei, D. Liu, Q. Lin, and R. Song, "Adaptive dynamic programming for discrete-time zero-sum games," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[193] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[194] D. Pompili, A. Hajisami, and H. Viswanathan, "Dynamic provisioning and allocation in cloud radio access networks (C-RANs)," *Ad Hoc Networks*, vol. 30, pp. 128–143, 2015.

[195] D. B. Rawat, S. Shetty, and C. Xin, "Stackelberg-game-based dynamic spectrum access in heterogeneous wireless systems," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1494–1504, 2016.

[196] N. Mandayam, "Cognitive algorithms and architectures for open access to spectrum," in *The Conference on the Economics, Technology and Policy of Unlicensed Spectrum, East Lansing, MI*, 2005.

[197] L. Duan, J. Huang, and B. Shou, "Duopoly competition in dynamic spectrum leasing and pricing," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1706–1719, 2012.

[198] D. Niyato and E. Hossain, "A game-theoretic approach to competitive spectrum sharing in cognitive radio networks," in *Wireless Communications and Networking Conference, WCNC*. IEEE, 2007, pp. 16–20.

[199] J. Jia and Q. Zhang, "Competitions and dynamics of duopoly wireless service providers in dynamic spectrum market," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2008, pp. 313–322.

[200] O. Ileri, D. Samardzija, and N. B. Mandayam, "Demand responsive pricing and competitive spectrum allocation via a spectrum server," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*. IEEE, 2005, pp. 194–202.

[201] W. Dong, S. Rallapalli, L. Qiu, K. Ramakrishnan, and Y. Zhang, "Double auctions for dynamic spectrum allocation," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2485–2497, 2016.

[202] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, Second Edition, 1998.