IN PURSUIT OF DESIRABLE EQUILIBRIA IN LARGE SCALE NETWORKED SYSTEMS

A Dissertation

by

JIAN LI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Srinivas Shakkottai
Committee Members,	Jean-Francois Chamberland
	Anxiao Jiang
	P. R. Kumar
	Vijay Subramanian
Head of Department,	Miroslav M. Begovic

December 2016

Major Subject: Computer Engineering

ABSTRACT

This thesis addresses an interdisciplinary problem in the context of engineering, computer science and economics: In a large scale networked system, how can we achieve a desirable equilibrium that benefits the system as a whole? We approach this question from two perspectives. On the one hand, given a system architecture that imposes certain constraints, a system designer must propose efficient algorithms to optimally allocate resources to the agents that desire them. On the other hand, given algorithms that are used in practice, a performance analyst must come up with tools that can characterize these algorithms and determine when they can be optimally applied. Ideally, the two viewpoints must be integrated to obtain a simple system design with efficient algorithms that apply to it.

We study the design of incentives and algorithms in such large scale networked systems under three application settings, referred to herein via the subheadings: *Incentivizing Sharing in Realtime D2D Networks: A Mean Field Games Perspective*, *Energy Coupon: A Mean Field Game Perspective on Demand Response in Smart Grids*, *Dynamic Adaptability Properties of Caching Algorithms*, and *Accuracy vs. Learning Rate of Multi-level Caching Algorithms*. Our application scenarios all entail an asymptotic system scaling, and an equilibrium is defined in terms of a probability distribution over system states. The question in each case is to determine how to attain a probability distribution that possesses certain desirable properties.

For the first two applications, we consider the design of specific mechanisms to steer the system toward a desirable equilibrium under self interested decision making. The environments in these problems are such that there is a set of shared resources, and a mechanism is used during each time step to allocate resources to

agents that are selfish and interact via a repeated game. These models are motivated by resource sharing systems in the context of data communication, transportation, and power transmission networks. The objective is to ensure that the achieved equilibria are socially desirable. Formally, we show that a Mean Field Game can be used to accurately approximate these repeated game frameworks, and we describe mechanisms under which socially desirable Mean Field Equilibria exist.

For the third application, we focus on performance analysis via new metrics to determine the value of the attained equilibrium distribution of cache contents when using different replacement algorithms in cache networks. The work is motivated by the fact that typical performance analysis of caching algorithms consists of determining hit probability under a fixed arrival process of requests, which does not account for dynamic variability of request arrivals. Our main contribution is to define a function which accounts for both the error due to time lag of learning the items' popularity, as well as error due to the inaccuracy of learning, and to characterize the tradeoff between the two that conventional algorithms achieve. We then use the insights gained in this exercise to design new algorithms that are demonstrably superior.

To my parents, sister and brother-in-law, for their love and support.

ACKNOWLEDGEMENTS

Working towards a Ph.D. was an extraordinary experience with many ups and downs along the way. Some moments were extremely elating, and others were deeply depressing, but all of them were very rewarding when looking back now. Many people guided me through critical times during this journey, and this dissertation cannot have been completed without the help from all of them. I am forever grateful to everyone.

First and foremost, I owe my deepest gratitude to my advisor, Professor Srinivas Shakkottai. Many memories come to mind, and I do not even know where to start and how to thank him. Without any exaggeration, the work presented here would not be possible without his guidance and constant encouragement. He was always ready to sit down with me to help with tackling problems, with seemly endless reserves of patience in tolerating my capricious views at times. He taught me how to find interesting research problems, conduct solid research, and present results to different audiences. I have benefited greatly from his insights, ideas and enthusiasm in research, and most importantly, his friendship. I am very lucky to have an advisor who would treat you not only as a student, but also as a friend. Srinivas, thank you.

I am also very grateful to Professor Vijay Subramanian from the University of Michigan. Serving effectively as my co-advisor, Vijay always had his door open to me and provided guidance during my graduate studies. I have been surprised on many occasions by Vijay's deep insights on the fundamental problems, and his incredible conscientiousness and attention to the mathematical details in every piece of work. I hope I have learned and inherited some characteristics from him through these years. Being a student of you and working with you is one of the luckiest experiences in my

life. Vijay, thank you.

It has been a great pleasure to work with both of you, and I look forward to continuing working together in the future.

I also benefited much from collaborations with Professor John C.S. Lui, with whom I have been working over multiple visits to the Chinese University of Hong Kong. John introduced me to caching problems that constitute the second part of this dissertation. I am fortunate to get a chance to work with and learn from John. He always kept reminding me to find interesting problems that could be applied in real systems, and find practical tools to solve them. John, thank you.

I would like to acknowledge my committee members, Professors P. R. Kumar, Jean-Francois Chamberland, and Anxiao Jiang. Thank you for comments and suggestions on my proposal and dissertation, support and encouragement you have shown on my work. I have learned a lot from all of you, both from your research and your courses. Professor Kumar taught me optimization and stochastic systems, and I have used ideas from these courses in multiple parts of my work. Professor Chamberland introduced coding theory to me. I learned analysis of algorithms from Professor Jiang's class, one of my favorite topics.

I would like to thank my previous and current group members Navid Abedini, Vinod Ramaswamy, Mayank Manjrekar, Bainan Xia, Rajarshi Bhattacharyya, Suman Paul, Vamseedhar Reddyvari, Ki-Yeob Lee, Kartic Bhargav, Adway Dogra for their discussions and comments on my work, as well as other students in Computer Engineering and Systems Group (CESG) for their friendship and enthusiasm. I am also equally indebted to many of my friends who have made my research and life much richer in my years at Texas A&M University. I cannot image a life without them. Thank you for being part of my life.

I would like to thank the administrative staff at Electrical and Computer Engi-

neering Department and CESG, especially Tammy Carda, Carolyn Warzon, Melissa Sheldon, and Anni Brunker for their patience, support and help over the past few years.

I would like to thank everyone who brought piano playing into my life. Special thanks go to Scott, who has been so patient and helpful over the years. I would also like to thank the TAMU Recreation center and everyone who has worked out together with me. You all have changed me in many aspects of my life: I started working out, running and swimming since I came to College Station.

Last, but not least, I would like to thank my family. My parents, you raised me, loved me, gave me all that I needed and supported me always. My sister and brother-in-law, I am very blessed to have you in my life. I owe my deepest gratitude to them.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

We have become increasingly dependent on large scale networked systems that are used to allocate shared resources so as to benefit the largest possible set of users. These systems include societal networks that are crucial to the functioning of society such as those used for data communication, transportation and power transmission. In each such network, resource allocation decisions have to be made based on the current state of the system, either in a distributed or centralized manner, and the net result is probability distribution over the states of the system. Typically, the decision makers are individual users who might make choices by using algorithms that maximize their individual utilities. In many problems, we have an asymptotic scaling regime in the number of decision makers that each try to maximize their own value over a set of choices. A fundamental question that we aim at answering in this thesis is whether it is possible to design incentive schemes such that the resulting equilibrium distribution of system states is a desirable one in terms of maximizing user utility.

We are also dependent on engineered systems in which the designer has the freedom to select a decision algorithm, which must then select between large number of choices each of whose value is unknown. An example of such a system is content caching wherein the popularity of different items of content is unknown apriori and continually changes. Here, as each request arrives, a caching algorithm must take a decision on which item to evict in order to make room for the newly cached item. The decision rule creates a distribution over the combination of content items in the cache, with different distributions resulting in different probabilities of finding a desired item in the cache. A basic problem that we wish to solve in this thesis is that

of determining a rule that will achieve a desired tradeoff between ensuring a high hit probability at equilibrium versus quickly converging to the equilibrium distribution.

While in the first problem, we have limited control over the distribution of decision makers' states and can only try to modify their behavior using incentive schemes, in the second we have full control over the decision rule but only have limited knowledge about the distribution that generates the arrival process of requests. An underlying theme in both trains of thought has to do with generation of equilibrium probability distributions, with system value being tied to some function of the resultant distribution. Attaining desirable equilibria is the goal of our work.

1.1 Overview and Main Contributions

In the first part of this thesis, we describe our results on achieving desirable equilibria under a repeated game framework in societal networks. The mean field game (MFG) framework is a promising approach towards studying societal networks, which typically have a large number of agents, and where any subset of agents has infrequent interactions. Here, agents model their opponents at any particular interaction through an assumed distribution over their action spaces, and play the best response action against this distribution. We say that the system is at a mean field equilibrium (MFE) if this best response action turns out to be a sample drawn from the assumed distribution. Our objective is to ensure that the achieved MFE is socially desirable. We consider two scenarios in which we wish to attain such a desirable MFE.

Realtime D2D Streaming Networks: In Section 2, we consider the problem of streaming live content to a cluster of co-located wireless devices that have both an expensive unicast base-station-to-device (B2D) interface, as well as an inexpensive broadcast device-to-device (D2D) interface, which can be used simultaneously. Our setting is

a streaming system that uses a block-by-block random linear coding approach to achieve a target percentage of on-time deliveries with minimal B2D usage. Our goal is to design an incentive framework that would promote such cooperation across devices, while ensuring good quality of service. Based on ideas drawn from truth-telling auctions, we design a mechanism that achieves this goal via appropriate transfers (monetary payments or rebates) in a setting with a large number of devices, and with peer arrivals and departures. Here, we show that a Mean Field Game can be used to accurately approximate our system. Furthermore, the complexity of calculating the best responses under this regime is low. We implement the proposed system on an Android testbed, and illustrate its efficient performance using real world experiments.

Societal Networks and Electricity Usage: In Section 3, we consider the problem of a Load Serving Entity (LSE) trying to reduce its exposure to electricity market volatility by incentivizing demand response in a Smart Grid setting. We focus on the day-ahead electricity market, wherein the LSE has a good estimate of the statistics of the wholesale price of electricity at different hours in the next day, and wishes its customers to move a part of their power consumption to times of low mean and variance in price. Based on the time of usage, the LSE awards a differential number of "Energy Coupons" to each customer in proportion to the customer's electricity usage at that time. A lottery is held periodically in which the coupons held by all the customers are used as lottery tickets.

Our study takes the form of a Mean Field Game, wherein each customer models the number of coupons that each of its opponents possesses via a distribution, and plays a best response pattern of electricity usage by trading off the utility of winning at the lottery versus the discomfort suffered by changing its usage pattern. The

system is at a Mean Field Equilibrium (MFE) if the number of coupons that the customer receives is itself a sample drawn from the assumed distribution. We show the existence of an MFE, and characterize the mean field customer policy as having a multiple-threshold structure in which customers who have won too frequently or infrequently have low incentives to participate. We then numerically study the system with a candidate application of air conditioning during the summer months in the state of Texas. Besides verifying our analytical results, we show that the LSE can potentially attain quite substantial savings using our scheme. Our techniques can also be applied to resource sharing problems in other *societal* networks such as transportation or communication.

In the second part of this thesis, we explore the construction of desirable equilibria for content distribution using caching algorithms. Caching algorithms typically follow Markovian dynamics, with a decision on what to cache and evict being made at each time based on the current cache content and arriving request. Hence, a caching algorithm generates a Markov process over the occupancy states of the cache. Performance analysis of caching replacement algorithms usually consists of determining the stationary distribution of this process, and using it to calculate the hit probability at the cache under either a synthetic request data, or by using a trace observed in a real system. However, this approach loses all notion of dynamically changing request popularities, and does not allow us to compare the performance of each algorithm with the best possible. We consider adaptability of such caching algorithms from two perspectives: the accuracy of learning a fixed popularity distribution; and the speed of learning items' popularity. We wish to study the adaptability of caching algorithms by defining a function that accounts for both the error due to time lag of learning items' popularity, as well as error due to the inaccuracy of learning. Our

goal is to obtain such a characterization over multiple existing algorithms, and to develop new ones.

Algorithms on Simple Caches: In Section 4, we analyze the performance of conventional caching algorithms such as LRU, FIFO and RANDOM, as applied to simple (single-stage) caches. We first determine the stationary distributions of these algorithms, and compute the distance between the stationary distributions of each algorithm with that of an algorithm that has knowledge of the true popularity ranking. We adopt the well known Wasserstein distance to compare the distance between distributions by taking the generalized Kendall's tau distance as the cost function. We call this metric as the τ-distance, which correctly represents the accuracy of learning the request distribution. We next use the *mixing time* to study the evolution of the Markov chain associated with caching algorithm to understand its rate of convergence to stationarity. We use a triangle inequality bound and combine the τ-distance and mixing time with appropriate normalization to obtain a new metric, called *learning error*, which represents both how quickly and how accurately an algorithm learns the optimal caching distribution. This allows us to determine how well each algorithm would perform after it has learned for a certain time interval.

Algorithms on Multi-level Caches: Multi-level caches have been shown to improve the hit probabilities of conventional caching algorithms through numerical studies. However, it is unclear how the number of levels and the partition of total cache size across these levels impacts the performance. In Section 5, we explore the value of multi-level caching by first considering a particular topology called a linear cache network. As the name suggests, the linear cache network consists of a stack of caches, potentially of different sizes and at different distances from the content requesting site. In such a network, an item enters via the first cache and moves up to a higher

cache whenever there is a cache hit on it, with a replacement algorithm determining which item should be replaced. We desire to understand the replacement algorithm from the perspective of how the division of cache levels impacts both the stationary hit probability and the rate of adaptation to a changing request distribution, through the τ-distance and mixing time metrics mentioned above. A main finding is that multi-level caches are a good way of increasing the accuracy of a caching algorithm for a given cache size, but at the expense of increasing the mixing time. Motivated by our analysis, we propose a novel hybrid algorithm, Adaptive-LRU (A-LRU) that learns both faster and better the change in popularity. We show numerically that it outperforms all other candidate algorithms when confronted with a dynamically changing synthetic request process, as well by using real world trace files.

We conclude with a brief summary of the main results of this thesis, and provide discussion on the future research directions in Section 6.

2. INCENTIVIZING SHARING IN REALTIME D2D NETWORKS: A MEAN FIELD GAME PERSPECTIVE

2.1 Introduction

There has recently been much interest in networked systems for collaborative resource utilization. These are systems in which agents contribute to the overall welfare through their individual actions. Usually, each agent has a certain amount of resources, and can choose how much to contribute based on the perceived return via repeated interactions with the system. An example is a peer-to-peer file sharing network, wherein each peer can contribute upload bandwidth by transmitting chunks to a peer, and receive downloads of chunks from that peer as a reward. Interactions are bilateral, and hence tit-for-tat type strategies are successful in preventing free-riding behavior [26]. More generally, collaborative systems entail multilateral interactions in which the actions of each agent affect and are affected by the collective behavior of a subset of agents. Here, more complex mechanisms are needed to accurately determine the value of the contribution of each individual to the group.

An example of a collaborative system with repeated multilateral interactions is a device-to-device (D2D) wireless network. Suppose that multiple devices require the same content chunk. The broadcast nature of the wireless medium implies that several agents can be simultaneously satisfied by a single transmission. However, they might each have different values for that particular chunk, and may have contributed different amounts in the past to the transmitting agent. Furthermore, D2D systems undergo "churn" in which devices join and leave different clusters as they move around. How then is an agent to determine whether to collaborate with others, and whether it has received a fair compensation for its contribution?

Our objective in this section is to design mechanisms for cooperation in systems with repeated multilateral interactions. As in earlier literature, we assume that there exists a currency to transfer utility between agents [7, 93], and our goal is to determine how much should be transferred for optimal collaboration. We focus on wireless content streaming as our motivating example. In particular, as shown in Figure 2.1, we assume that all devices are interested in the same content stream, and receive a portion of chunks corresponding to this stream via a unicast base-station-to-device (B2D) interface. The B2D interface has a large energy and dollar cost for usage, and the devices seek to mitigate this cost via sharing chunks through broadcast D2D communication.[1]

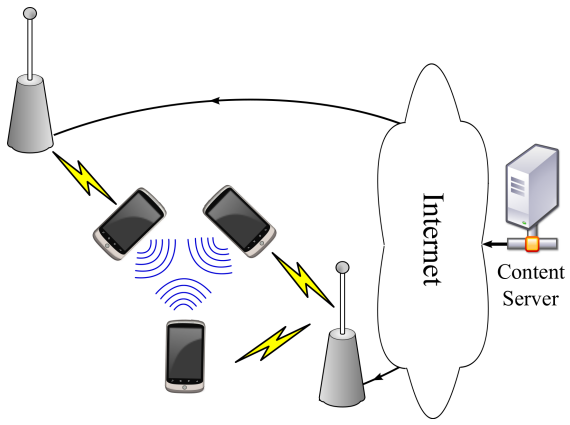

Figure 2.1: Wireless content distribution via multiple interfaces [4].

A content sharing system is described in [4], in which the objective is to achieve *live streaming* of content synchronously to multiple co-located devices. The system architecture of that work forms an ideal setting for studying mechanism design in

[1]Note that, as we describe in greater detail later in this section, it is possible to enable the usage of both the 3G (unicast) and WiFi (broadcast) interfaces simultaneously on Android smart phones.

which multilateral interactions occur. The setup is illustrated in Figure 2.2. Here, time is divided into *frames*, which are subdivided into T *slots.* A *block* of data is generated by the content server in each frame, and the objective is to ensure that this block can be played out by all devices two frames after its generation, *i.e.*, data block k is generated in frame $k-2$, and is to be played out in frame k. Such a strict delay constraint between the time of generation and playout of each data block ensures that the *live* aspect of streaming is maintained.

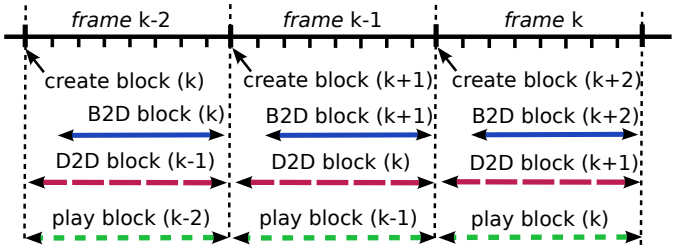

Figure 2.2: Streaming architecture [4] in which each block must be delivered within two frames after its creation.

Upon generation of block k (in frame $k-2$), the content server divides it into N *chunks* and performs random linear coding (RLC) over these chunks [28]. The server unicasts some of these coded chunks to each device using its B2D interface. This number is to be kept small to reduce B2D usage. Next, in frame $k-1$, the devices use the broadcast D2D network to disseminate these chunks among themselves. At the end of frame $k-1$ the devices attempt to decode block k. If a device i has received enough coded chunks to decode the block, it plays out that block during frame k. Otherwise, i will be idle during this frame. The use of RLC results in two desirable system features. *First*, the server can unicast a fixed number of chunks to the devices in each frame over a lossy channel (Internet plus B2D link) without any

feedback. *Second*, the devices do not need to keep track of what chunks each one possesses while performing D2D broadcasts.

The notion of quality of experience (QoE) here is *delivery-ratio* denoted by η, which is the average ratio of blocks desired to the blocks generated [42]. For instance, a delivery ratio of 95% would mean that it is acceptable if 5% of the blocks can be skipped. A device can keep track of its QoE thus far via the "deficit" incurred upto frame k, which is the difference between the actual number of number of blocks successfully decoded by frame k and the target value ηk. In [4], it was shown that, assuming complete cooperation by the participating devices, it is possible to design a chunk sharing scheme whereby all devices would meet their QoE targets with minimal usage of the B2D interface. *But how do we design a mechanism to ensure that the devices cooperate?*

The setting of interest in this section is that of a large number of D2D clusters, each with a fixed number of agents, and with all clusters interested in the same content stream. Examples of such settings are sports stadia, concerts or protest meetings, where a large number of agents gather together, and desire to receive the same live-stream (replays, commentary, live video *etc.*) Devices move between clusters as agents move around, causing churn. The objective of our work is to develop an incentive framework wherein each device truthfully reports the number of chunks that it receives via B2D and its deficit in each frame, so that a system-wide optimal allocation policy can be employed. Such an incentive framework should be lightweight and compatible with minimal amounts of history retention. Finally, we also desire to implement the system on Android smart phones and measure its real world performance.

2.1.1 Related Work

The question of how to assign value to wireless broadcast transmissions is intriguing. For instance, [43] considers a problem of repeated interaction with time deadlines by which each node needs to receive a packet. Each node declares its readiness to help others after waiting for a while; the game lies in choosing this time optimally, and the main result is to characterize the price of anarchy that results. However, decision making is myopic, *i.e.*, devices do not estimate future states while taking actions. In a similar fashion, [93] propose a scheme for sharing 3G services via WiFi hotspots using a heuristic scheme that possesses some attractive properties. Here too, decision making is myopic. The question of fair scheduling at a base station that uses the history of interactions with individual stations in order to identify whether they are telling the truth about their state is considered in [52]. However, since the devices in our network undergo churn and keeping track of device identities is infeasible, we desire a scheme that does not use identities or history to enable truthful revelation of state.

2.1.1.1 Perfect Bayesian and Mean Field Equilibria

The typical solution concept in dynamic games is that of Perfect Bayesian Equilibrium (PBE). Consider a strategy profile for all players, as well as beliefs about the other players' types at all information sets. This strategy profile and belief system is a PBE if: (i) *Sequential rationality*: Each player's strategy specifies optimal actions, given her beliefs and the strategies of other players; (ii) *Consistency of beliefs*: Each player's belief is consistent with the strategy profile (following Bayes' rule). PBE requires each agent to keep track of their beliefs on the future plays of all other agents in the system, and play the best response to that belief. The dynamic pivot mechanism [11] extends the truth-telling VCG idea [56] to dynamic games. It provides

a basis for designing allocation schemes that are underpinned by truthful reporting. Translating the model in [52] to the language of [11], it is possible to use the dynamic pivot mechanism to develop a scheme (say FiniteDPM) with appropriate transfers that will be efficient, dominant strategy incentive compatible and per-period individually rational; note that while this scheme would use the identities of the devices, it will not need to build up a history of interactions. We omit the details of this as it is a straight-forward application of the general theory from [11].

Computation of PBE becomes intractable when the number of agents is large. An accurate approximation of the Bayesian game in this regime is that of a Mean Field Game (MFG) [44, 49, 58]. In MFG, the agents assume that each opponent would play an action drawn *independently* from a static distribution over its action space. The agent chooses an action that is the best response against actions drawn in this fashion. The system is said to be at Mean Field Equilibrium (MFE) if this best response action is itself a sample drawn from the assumed distribution, *i.e.*, the assumed distribution and the best response action are consistent with each other [47,64,69]. Essentially, this is the canonical problem in game theory of showing the existence of a Nash equilibrium, as it applies to the regime with a large number of agents. We will use this concept in our setting where there are a large number of peer devices with peer churn.

To the best of our knowledge, there is no prior work that considers mechanism design for multilateral repeated games in the mean field setting. One of the important contributions of this section is in providing a truth-telling mechanism for a mean-field game. In the process of developing the mechanism we will also highlight the nuances to be considered in the mean-field setting. In particular, we will see that aligning two concepts of value—from the system perspective and from that of the agents—is crucial to our goal of truth-telling.

2.1.2 Organization and Main Results

We describe our system model in Section 2.2. Our system consists of a large number of clusters, with agents moving between clusters. The lifetime of an agent is geometric; an agent is replaced with a new one when it exits. Each agent receives a random number of B2D chunks by the beginning of each frame, which it then shares using D2D transmissions.

In Section 2.3, we present an MFG approximation of the system, which is accurate when the number of clusters is large. Here, the agents assume that the B2D chunks received and deficits of the other agents would be drawn independently from some distributions in the future, and optimize against that assumption when declaring their states. The objective is to incentivize agents to truthfully report their states (B2D chunks and deficit) such that a schedule of transmissions (called an "allocation") that minimizes the discounted sum of costs can be used in each frame. The mechanism takes the form of a scheme in which tokens are used to transfer utility between agents. A nuance of this regime is that while the system designer sees each cluster as having a new set of users (with IID states) in each time frame, each user sees states of all its competitors *but not itself* as satisfying the mean field distribution. Reconciling the two view points is needed to construct a cost minimizing pivot mechanism, whose truth-telling nature is shown in Section 2.4. This is our main contribution in this section. The allocation itself turns out to be computationally simple, and follows a version of a min-deficit first policy [4].

Next, in Sections 2.5–2.6, we present details on how to prove the existence of the MFE in our setting. Although this proof is quite involved, it follows in a high-level sense in the manner of [47, 69]. For the ease of exposition, the details of the proof are provided in Appendix A. We then turn to computing the MFE and the

value functions needed to determine the transfers in Section 2.8. The value iteration needed to choose allocation is straightforward.

We present details of our Android implementation of a music streaming app used to collect real world traces in Section 2.9. We discuss the viability of our system in Section 2.10, and illustrate that under the current price of cellular data access, our system provides sufficient incentives to participate. Finally, we conclude in Section 2.11.

2.2 Content Streaming Model

We consider a large number of D2D clusters, each with a fixed number of agents, and with all clusters interested in the same content stream. We assume that a cluster consists of M co-located peer devices denoted by $i \in \{1, \ldots, M\}$[2]. The data source generates the stream in the form of a sequence of blocks. Each block is further divided into N chunks for transmission. We use random linear network coding over the chunks of each block (with coefficients in finite field F_q of size q). We assume that the field size is very large; this assumption can be relaxed without changing our cooperation results. Time is divided into frames, which are further divided into slots. At each time slot τ, each device can simultaneously receive up to one chunk on each interface.

B2D Interface: Each device has a (lossy) B2D unicast channel to a base-station. For each device i, we model the number of chunks received using the B2D interface in the previous frame by a random variable with (cumulative) distribution ζ, independent of the other devices. The support of ζ is the set $\{0, 1, \cdots, T\}$, denoted by $\mathbb{T}$. The statistics of this distribution depend on the number of chunks transmitted by the server and the loss probability of the channel. In [4], a method for calcu-

[2]Our analysis is essentially unchanged when there are a random but finite number of devices in each cluster.

lating statistics based on the desired quality of service is presented. We take the distribution ζ as given.

D2D Interface: Each device has a zero-cost D2D broadcast interface, and only one device can broadcast over the D2D network at each time τ. For simplicity of exposition, we will assume that the D2D broadcasts are always successful; the more complex algorithm proposed in [4] to account for unreliable D2D is fully consistent[3] with our incentive scheme. Since each D2D broadcast is received by all devices, there is no need to rebroadcast any information. It is then straightforward to verify that the order of D2D transmissions does not impact performance. Thus, we only need to keep track of the number of chunks transmitted over the D2D interfaces during a frame in order to determine the final state of the system.

Allocation: We denote the total number of coded chunks of block k delivered to device i via the B2D network during frame $k-2$ using $e_i[k] \sim \zeta$. We call the vector consisting of the number of transmissions by each device via the D2D interfaces over frame $k-1$ as the "allocation" pertaining to block k, denoted by $\mathbf{a}[k]$. Also, we denote the number received chunks of block k by device i via D2D during frame $k-1$ using $g_i[k]$. Due to the large field size assumption, if $e_i[k] + g_i[k] = N$, it means that block k can be decoded, and hence can be played out. For simplicity of exposition, we develop our results assuming that the allocation is computed in a centralized fashion in each cluster. However, we actually implement a distributed[4] version on the testbed.

Quality of Experience: Each device i has a delivery ratio $\eta_i \in (0, 1]$, which is the minimum acceptable long-run average number of frames device i must playout. In the mobile agents model, we assume that all devices have the same delivery ratio η for

[3]We will discuss this at the end of Section 2.6.1.

[4]At the end of Section 2.6.1 we will argue that the distributed implementation is also consistent with our incentive scheme.

simplicity. It is straightforward to extend our results to the case where delivery ratios are drawn from some finite set of values. The device keeps track of the current deficit using a deficit queue with length $d_i[k] \in \mathbb{K}$. The set of possible deficit values is given by $\mathbb{K} = \{k\eta - m : k, m \geq 0, m \leq \lfloor k\eta \rfloor\}$, where for $x \in \mathbb{R}$, $\lfloor x \rfloor = \max\{k \in \mathbb{Z} : k \leq x\}$ is the largest whole number that x is greater than. Note that $\mathbb{K}$ is a countable set and the possible deficit values are all non-negative. In fact, by the well-ordering principle $\mathbb{K}$ can be rewritten as $\{d_n\}_{n\in\mathbb{N}}$ with d_n an increasing sequence (without bound) such that $d_1 = 0$. We will use this representation to enumerate the elements of $\mathbb{K}$. If a device fails to decode a particular block, its deficit increases by η, else it decreases by $1 - \eta$. The impact of deficit on the user's quality of experience is modeled by a function $c(d_i[k])$, which is convex, differentiable and monotone increasing. The idea is that user unhappiness increases more with each additional skipped block.

Transfers: We asume the existence of a currency (either internal or a monetary value) that can be used to transfer utility between agents [7, 93]. In our system, a negative transfer is a price paid by the agent, while a positive value indicates that the agent is paid by the system. Such transfer systems are well established; see for instance a review in [7]. Transfers are used by agents either to pay for value received through others' transmissions, or to be compensated for value added to others by transmitting a chunk. We assume that the transmissions in the system are monitored by a reliable device, which can then report these values to decide on the transfers. In practice we use the device that creates each ad-hoc network as the monitor.

An *allocation policy* maps the values of the B2D chunks received and deficits as revealed by agents, denoted by $\hat{\boldsymbol{\theta}}[k] := (\hat{\mathbf{e}}[k], \hat{\mathbf{d}}[k-1])$, to an allocation for that frame $\mathbf{a}[k]$. Given an allocation, agents have no incentive to deviate from it, since an agent that does not transmit the allocated number of chunks would see no benefit; those time slots would have no transmissions by other agents either. The fundamental

question is that of how to incentivize the agents to reveal their states truthfully so that the constructed allocation can maximize system-wide welfare.

2.3 Mean Field Model and Mechanism Design

Our system consists of JM agents (or users) organized into J clusters with M agents per cluster. As mentioned earlier, time is slotted into frames. At the end of a frame, any agent i can leave the system only to be replaced by a new agent (also denoted by i) whose initial deficit is drawn from a (cumulative) distribution Ψ with support $\mathbb{K}$. This event occurs with probability $\bar{\delta} = (1 - \delta)$ independently for each agent, so that the lifetimes of the agents are geometrically distributed. As described in the previous section, we assume that the number of chunks received via B2D for agent i in frame k, denoted by $e_i[k]$, is chosen in an *i.i.d.* fashion according to the (cumulative) distribution ζ, with support $\mathbb{T}$; one such distribution is the binomial distribution. In addition to the agents having geometrically distributed lifetimes, we also allow mobility in our set-up. In particular, in every frame we assume that all the agents are randomly permuted and then assigned to clusters such that there are exactly M agents in each cluster. Using this system as a starting point we will develop our mean-field model that will be applicable when the number of clusters J is extremely large.

The mean field framework in Figure 2.3 illustrates system relationships that will be discussed below. The blue/dark tiles apply to the value determination process for mechanism design, which will be discussed in this section. The beige/light tiles are relevant to showing the existence of an MFE on which the mechanism depends, which will be discussed in Sections 2.5–2.6.

The mean field model yields informational and computational savings, since otherwise each agent will need to not only be cognizant of the values and actions of all

agents, but also track their mobility patterns. Additionally, the mean field distribution accounts for regenerations, which do not have to be explicitly accounted for when determining best responses.

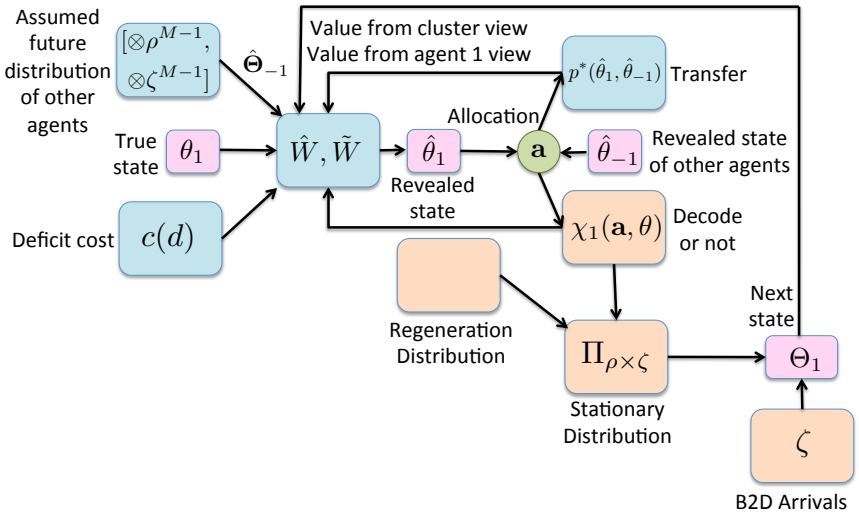

Figure 2.3: The mean field system from perspective of agent 1.

There is, however, an important nuance that the mean-field analysis introduces: when there are a large number of clusters, each cluster sees a different group of agents in every frame with their states drawn from the mean-field distribution, but even though each agent interacts with a new set of agents in every frame, it's own state is updated based on the allocations made to it, so that the differing viewpoints of the two entities need to be reconciled while providing any incentives.

The number of chunks received over the B2D interface and the deficit value constitute the state of an agent at the beginning of a frame. At frame k we collect together the state variables of all the agents in system as $\boldsymbol{\theta}[k] = (\mathbf{e}[k], \mathbf{d}[k-1])$. Our

mechanism then aims to achieve

$$W(\boldsymbol{\theta}[k]) = \min_{\{\mathbf{a}[l]\}_{l=k}^{\infty}} \mathbb{E}\left\{ \sum_{j=1}^{J} \sum_{l=k}^{\infty} \delta^{l-k} \sum_{i \in s_j[l]} v_i(\mathbf{a}_{s_j}[l], \theta_i[l]) \right\}, \tag{2.1}$$

where $j = 1, 2, \cdots, J$ is the number of clusters in the system, $s_j[k]$ is the set of agents in cluster j at frame k, $\mathbf{a}_{s_j}$ is the allocation in cluster j and $v_i(\mathbf{a}_{s_j}[l], \theta_i[l])$ is the value that agent i makes from the allocation in frame k. For agent i set $j_i[k]$ to be the cluster he belongs in during frame k, i.e., $i \in s_{j_i[k]}[k]$. Note that the probability of remaining in the system δ appears as the discount factor in the above expression.

Given the allocation in each cluster, if agent i does not regenerate, then his deficit gets updated as

$$d_i[k] = (d_i[k-1] + \eta - \chi_i(\mathbf{a}_{j_i[k]}[k], \theta_i[k]))^+, \tag{2.2}$$

where $(\cdot)^+ = \max(\cdot, 0)$, whereas if the agent regenerates, then $d_i[k] = \tilde{d}_i[k]$ where $\tilde{d}_i[k]$ is drawn *i.i.d.* with distribution Ψ. Here,

$$\chi_i(\mathbf{a}, \theta_i) = 1_{\{e_i + g_i(\mathbf{a}) = N\}} = \begin{cases} 1 & \text{if } e_i + g_i(\mathbf{a}) = N \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

where $\chi_i(.)$ is 1 if and only if agent i obtains all N coded chunks to be able to decode a block, $g_i(\mathbf{a})$ is the number of packets agent i can get during a frame under the allocation $\mathbf{a}$ (where we suppress the dependence of $\mathbf{a}$ on $\boldsymbol{\theta}$). We specialize to the case where the value per frame for agent i with system state $\boldsymbol{\theta}$ and vector of allocations $\mathbf{a}$ is given by $v_i(\mathbf{a}, \theta_i) = c\Big(\big(d_i + \eta - \chi_i(\mathbf{a}, \theta_i)\big)^+\Big)$ if there is no regeneration and $v_i(\mathbf{a}, \boldsymbol{\theta}_i) = c(\tilde{d}_i)$ otherwise, where $\tilde{d}_i$ is *i.i.d.* with distribution Ψ and $c(\cdot)$ is the holding cost function that is assumed to be convex and monotone increasing.

As there are a large number of clusters, in every frame there is a completely

different set of agents that appear at any given cluster. The revealed states of these agents will be drawn from the mean field distribution. Hence, from the perspective of some cluster l, the revealed state of the agents in that cluster $\hat{\mathbf{\Theta}}_l$ will be drawn according to the (cumulative) distributions $[\otimes\rho^M, \otimes\zeta^M]$, with ρ pertaining to the deficit, and ζ pertaining to the B2D transmissions received by that agent. Note that the support of ρ is $\mathbb{K}$ while the support of ζ is $\mathbb{T}$, and $\otimes$ indicates the *i.i.d* nature of the agent states. Whereas from the perspective a particular agent i, the revealed states of all the other agents in that cluster will be drawn according to $\hat{\mathbf{\Theta}}_{-i} \sim [\otimes\rho^{M-1}, \otimes\zeta^{M-1}]$. These facts will simplify the allocation problem in each cluster and also allow us to analyze the MFE by tracking a particular agent.

First, we consider the allocation problem as seen by the clusters. Pick any finite number of clusters. In the mean-field limit, the agents from frame to frame will be different in each cluster, therefore the allocation decision in each cluster can be made in an distributed manner, independent of the other clusters; this is one of the chaos hypotheses of the mean-field model. This then implies that the objective in (2.1) is achieved by individual optimization in each cluster, *i.e.*,

$$W(\hat{\boldsymbol{\theta}}[k]) = \sum_{j=1}^{J} W_j(\hat{\boldsymbol{\theta}}_{s_j}[k]), \tag{2.4}$$

where we recall that $\hat{\boldsymbol{\theta}}_{s_j}[k]$ is the revealed state of agents in cluster j at time k and

$$W_j(\hat{\boldsymbol{\theta}}_{s_j}[k]) = \min_{\{\mathbf{a}_{s_j}[l]\}_{l=k}^{\infty}} \sum_{l=k}^{\infty} \delta^{l-k} \sum_{i \in s_j[l]} v_i(\mathbf{a}_{s_j}[l], \hat{\theta}_i[l]). \tag{2.5}$$

Under mean field assumption, the method of determining value does not change from step-to-step. The value function in the mean-field is determined by the first solving

the following Bellman equation

$$\hat{W}(\hat{\boldsymbol{\theta}}) = \min_{\mathbf{a}} \sum_{i=1}^{M} v_i(\mathbf{a}, \hat{\theta}_i) + \delta \mathbb{E}\left\{\hat{W}(\hat{\boldsymbol{\Theta}})\right\} \tag{2.6}$$

to obtain function $\hat{W}(\cdot)$, where $\hat{\boldsymbol{\theta}}$ is the M-dimensional revealed state vector (with elements $\hat{\theta}_i$) and the future revealed state vector $\hat{\boldsymbol{\Theta}}$ is chosen according to $[\otimes\rho^M, \otimes\zeta^M]$, and thereafter setting $W_j(\hat{\boldsymbol{\theta}}_{s_j}[k]) = \hat{W}(\hat{\boldsymbol{\theta}}_{s_j}[k])$ for every $j = 1, 2, \ldots, J$. This observation then considerably simplifies the allocation in each cluster to be the greedy optimal, *i.e.*, determine (multi)function

$$\mathbf{a}^*(\hat{\boldsymbol{\theta}}) = \arg\min_{\mathbf{a}} \sum_{i=1}^{M} v_i(\mathbf{a}, \hat{\theta}_i), \tag{2.7}$$

and for $j = 1, 2, \ldots, J$ we set $\mathbf{a}^*_{s_j} = \mathbf{a}^*(\hat{\boldsymbol{\theta}}_j)$.

Next, we consider the system from the viewpoint of a typical agent i; w.l.o.g let $i = 1$. Any allocation results in the deficit changing according to (2.2) and the future B2D packets drawn according to ζ, whereas the state of every other agent that agent 1 interacts with in the future gets chosen according to the mean field distribution. Then the value function (of the cluster) from the perspective of agent 1 is determined using

$$\tilde{W}(1, (\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1})) = \min_{\mathbf{a}} \sum_{i'=1}^{M} v_{i'}(\mathbf{a}, \hat{\theta}_{i'}) + \delta \mathbb{E}\left\{\tilde{W}(1, (\hat{\Theta}_1, \hat{\boldsymbol{\Theta}}_{-1}))|\mathbf{a}, \hat{\theta}_1\right\}. \tag{2.8}$$

Here, $\hat{\boldsymbol{\theta}}_{-1}$ represents the revealed states of all the agents in cluster except 1, $\hat{\boldsymbol{\Theta}}_{-1} \sim [\otimes\rho^{M-1}, \otimes\zeta^{M-1}]$, and for $\hat{\Theta}_1$, the deficit term is determined via (2.2) (setting $\theta_i = \hat{\theta}_i$) while the B2D term follows ζ. This recursion yields a function $\tilde{W}(1, \cdot)$ which applies to all agents. Using this function, one can also determine the allocation that agent

1 expects his cluster to perform, namely,

$$\tilde{\mathbf{a}}(\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1}) = \arg\min_{\mathbf{a}} \sum_{i'=1}^{M} v_{i'}(\mathbf{a}, \hat{\theta}_{i'}) + \delta \mathbb{E}\left\{ \tilde{W}(1, (\hat{\Theta}_1, \hat{\boldsymbol{\Theta}}_{-1})) | \mathbf{a}, \hat{\theta}_1 \right\}. \tag{2.9}$$

Using the two allocations $\mathbf{a}^*$ and $\tilde{\mathbf{a}}$ we can write down the value of agent 1 from the system optimal allocation and the value of agent 1 in the allocation that the agent thinks that the system will be performing. For a given allocation function $\mathbf{a}(\cdot)$ (for the state of agents in the cluster where agent 1 resides at present), we determine the solution to the following recursion

$$V(\mathbf{a}(\hat{\boldsymbol{\theta}}), \tilde{\theta}_1) = v_1(\mathbf{a}, \tilde{\theta}_1) + \delta \mathbb{E}\left\{ V(\mathbf{a}(\hat{\Theta}_1, \hat{\boldsymbol{\Theta}}_{-1}), \tilde{\Theta}_1) \right\} \tag{2.10}$$

to get function $V(\cdot, \cdot)$, where $\tilde{\theta_1}$, is an arbitrary state variable, the deficit term of $\tilde{\Theta}_1$ follows (2.2) while the B2D term is generated independently (setting $\theta_i = \tilde{\theta}_i$), $\mathbf{a}$ is an arbitrary allocation, the B2D term is generated independently, and $\hat{\boldsymbol{\Theta}}_{-1}$ is chosen using the mean-field distribution. Notice that $\tilde{\theta}_1 = \theta_1$ would yield the true value of allocation $\mathbf{a}$ to agent 1. By the cluster optimal allocation (what the cluster actually does), agent 1 gets $V(\mathbf{a}^*(\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1}), \theta_1)$ whereas from the perception of agent 1 he thinks he should be getting $V(\tilde{\mathbf{a}}(\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1}), \theta_1)$ (based on what he thinks the cluster should be doing).

2.3.1 Transfer

We will use the different value functions to define the transfer for agent 1 depending on the reported state variable $\hat{\theta}_1$ such that the transfer depends on the difference between what he gets from the system optimal allocation and what he expects the system to do from his own perspective. Using this logic we set the transfer for agent

1 as

$$p^*(\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1}) = V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}), \hat{\theta}_1) - V(\tilde{\mathbf{a}}(\hat{\boldsymbol{\theta}}), \hat{\theta}_1) + H(\hat{\boldsymbol{\theta}}_{-1}) - (\tilde{W}(1, (\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1})) - V(\tilde{\mathbf{a}}(\hat{\boldsymbol{\theta}}), \hat{\theta}_1)). \tag{2.11}$$

where $H(\hat{\boldsymbol{\theta}}_{-1})$, following the Groves pivot mechanism, can be chosen using the recursion

$$H(\hat{\boldsymbol{\theta}}_{-1}) = \min_{\mathbf{a}_{-1}} \sum_{i \neq 1} v_i(\mathbf{a}_{-\mathbf{1}}, \hat{\theta}_i) + \delta \mathbb{E}\left\{ H(\hat{\boldsymbol{\Theta}}_{-1}) \right\}, \tag{2.12}$$

where $\hat{\boldsymbol{\Theta}}_{-1} \sim [\otimes \rho^{M-1}, \otimes \zeta^{M-1}]$, and $\mathbf{a}_{-\mathbf{1}}$ is used to denote an allocation in a system in which agent 1 is not present.

The Clarke pivot mechanism idea ensures that the net-cost of agent 1, $V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}), \hat{\theta}_1)$ $-p^*(\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1})$, equals $\tilde{W}(1, (\hat{\theta}_1, \hat{\boldsymbol{\theta}}_{-1})) - H(\hat{\boldsymbol{\theta}}_{-1})$. This is simply the value of the system as a whole from the viewpoint of agent 1, minus a function only of $\hat{\boldsymbol{\theta}}_{-1}$. As in the Vickrey-Clarke-Groves mechanism, such formulation of net-cost naturally promotes truth-telling as a dominant strategy at each step.

2.3.2 Allocation Scheme

The basic building block of our mechanism is the per-frame optimal allocations that solve (2.1). We will now spell out the allocation in greater detail. First, we observe that the allocation problem separates into independent allocation problems in each cluster that have the same basic structure. Therefore, it suffices to discuss the allocation problem for one cluster.

From (2.7), the objective in this cluster is

$$\min_{\mathbf{a}} \sum_{i=1}^{M} c((d_i[k-1] + \eta - \chi_i(\mathbf{a}[k], \theta_i[k]))^+). \tag{2.13}$$

An optimal allocation is determined using the following observations. First, we partition the agents into two sets, ones who cannot decode the frame even if they never transmit during the T slots of the D2D phase and the rest; the former agents are made to transmit first. After this we determine agents who have extra chunks (number of slots that they can transmit on such that there is still time to decode whole frame) and make these agents transmit their extra chunks. After all the extra chunks have been transmitted, it is easy to see using the properties of the holding cost function that agents are made to transmit in a minimum-deficit-first fashion in order to prioritize agents with large deficits. This is summarized in the follow lemma.

Lemma 1 *The algorithm delineated in Algorithm 1 provides an optimal greedy allocation.*

Algorithm 1 Optimal Mean Field D2D Allocation Rule

At the beginning of each frame $k-1$, given the arrivals $(e_1[k], ..., e_M[k])$:
Partition the devices into sets $\mathcal{S} = \{i \in \{1, ..., M\} : \ N - e_i[k] \leq T, e_i[k] + \sum_{j \neq i} e_j[k] \geq N\}$ and $\mathcal{S}^c$.
If $\mathcal{S} = \emptyset$, none of the agents can decode the block. Else,
Phase 1) Let all the agents in $\mathcal{S}^c$ transmit all that they initially received for the next $T_1 = \min\{\sum_{i \in \mathcal{S}^c} e_i[k], T\}$ slots.
If there exists time and a need for more transmissions,
Phase 2) Let each agent $i \in \mathcal{S}$ transmit up to $(e_i[k] + T - N)^+$ of its initial chunks.
Phase 3) While there exists time and a need for more transmissions, let devices in $\mathcal{S}$ transmit their remaining chunks in an increasing order of their deficit values.

2.4 Properties of Mechanism

2.4.1 Truth-telling as Dominant Strategy

Since we consider a mean-field setting, we will assume that deficit of agent i changes via the allocation while the deficits of all the other agents are drawn using the given distribution ρ. The $\mathbf{e}$ values are generated *i.i.d.* with distribution ζ. Based on the system state report $\boldsymbol{\theta}[k]$ at time k, we assume that the mechanism makes the optimal greedy allocation $\mathbf{a}^*[k]$ from (2.7) and levies transfers $\mathbf{p}^*[k]$ from (2.11) that uses the allocations from the agent's perspective from (2.9). We can then show that truthfully revealing the state, *i.e.,* (d, e) values at the beginning of every frame is incentive compatible.

Definition 1 *A direct mechanism (or social choice function)* $f = (a, p)$ *is* dominant strategy incentive compatible *if* θ_i *is a dominant strategy at* θ_i *for each* i *and* $\theta_i \in \Theta_i$*, where* $a(\cdot)$ *is a decision rule and* $p(\cdot)$ *is a transfer function.*

Theorem 1 *Our mechanism* $\{\mathbf{a}^*[k], \mathbf{p}^*[k]\}_{k=0}^{\infty}$ *is dominant strategy incentive compatible.*

2.4.2 Nature of Transfers

We now determine the nature of the transfers that are required to promote truth-telling. We will show that the transfers constructed in (2.11) are always non-negative, *i.e.,* the system needs to pay the agents in order to participate. In other words, each agent needs a subsidy to use the system, since it could simply choose not to participate otherwise. Thus, the system is not budget-balanced. We will show later how the savings in B2D usage that results from our system provides the necessary subsidy in Section 2.10. Given these transfers, we will also see that our mechanism is individually rational so that users participate in each frame.

Lemma 2 *The transfers defined in* (2.11) *are always non-negative.*

The proof of individual rationality follows along the same lines as Lemma 2.

Lemma 3 *Our mechanism* $\{\mathbf{a}^*[k], \mathbf{p}^*[k]\}_{k=0}^{\infty}$ *is individually rational, i.e., the voluntary participation constraint is satisfied.*

We remark that not participating in a frame is equivalent to free-riding, and our transfers ensure a lower cost is obtained when participating. However, as the net payment to the users is non-negative[5], we will not immediately have budget-balance. For the broader class of Bayes-Nash incentive-compatible mechanism, [8] shows that only under the assumption of "independent types" (the distribution of each agent's information is not directly affected by the other agents' information), budget can be balanced ex-interim. However, in our system, each agent's information will have an impact on the other agents' information through the allocation. Nevertheless, using the same technique of an initial sum being placed in escrow with the expectation that it would be returned at each stage (*i.e,.* interim), our system may be budget-balanced. Details using current prices of B2D service are provided in Section 2.10.

2.4.3 Value Functions and Optimal Strategies

We will now show that the value function given by the solution to (2.6) is well-defined and can be obtained using value iteration. Similarly, we will show that both the value function and the optimal allocation policy from a agent's perspective, given by (2.8) and (2.9) respectively, exist and can also be determined via value iteration.

[5]While we don't prove it, we expect the transfer to be positive if the agent transmits, but we also note that it need not be zero if he doesn't, owing to the translation of viewpoints mentioned earlier.

Define operators T_1 and T_2 by

$$T_1 w(\boldsymbol{\theta}) = \sum_{i=1}^{M} v_i(\mathbf{a}^*(\boldsymbol{\theta}), \theta_i) + \delta \mathbb{E}\left\{w(\boldsymbol{\Theta})\right\}, \tag{2.14}$$

$$T_2 \tilde{w}(1, (\theta_1, \boldsymbol{\theta}_{-1})) = \min_{\mathbf{a}} \sum_{i'=1}^{M} v_{i'}(\mathbf{a}, \theta_{i'}) + \delta \mathbb{E}\left\{\tilde{w}(1, (\Theta_1, \boldsymbol{\Theta}_{-1}))|\mathbf{a}, \theta_1\right\}, \tag{2.15}$$

using (2.6) and (2.8), respectively.

Theorem 2 *The following hold:*

1. *There exists a unique $W(\boldsymbol{\theta})$ such that $T_1 W(\boldsymbol{\theta}) = W(\boldsymbol{\theta})$, and given $\boldsymbol{\theta}$ for every $\mathbf{w} \in \mathbb{R}_+^M$, we have $\lim_{n\to\infty} T_1^n w = W(\boldsymbol{\theta})$;*

2. *There exists a unique $\tilde{W}(1, (\theta_1, \boldsymbol{\theta}_{-1}))$ such that $T_2 \tilde{W}(1, (\theta_1, \boldsymbol{\theta}_{-1}))$ $= \tilde{W}(1, (\theta_1, \boldsymbol{\theta}_{-1}))$, and given $(\theta_1, \boldsymbol{\theta}_{-1})$ for every $\mathbf{w} \in \mathbb{R}_+^M$, we have $\lim_{n\to\infty} T_2^n w$ $= \tilde{W}(1, (\theta_1, \boldsymbol{\theta}_{-1}))$; and*

3. *The Markov policy $\tilde{\mathbf{a}}((\theta_1, \boldsymbol{\theta}_{-1}))$ obtained from (2.9) is an optimal policy to be used in cluster $j_1[\cdot]$ from the viewpoint of agent 1.*

2.5 Mean Field Equilibrium

In the mean-field setting, assuming the state of every other agent is drawn *i.i.d.* with distribution $\rho \times \zeta$, the deficit of any given agent evolves as a Markov chain. We start by showing that this Markov chain has a stationary distribution. If this stationary distribution is the same as ρ, then the distribution ρ is defined as a mean-field equilibrium (MFE); we use the Schauder fixed point theorem to show the existence of a fixed point ρ. Using the regenerative representation of the stationary distribution of deficits given ρ and a strong coupling result, we prove that the mapping that takes ρ to the stationary distribution of deficits is continuous using a strong coupling

result. Finally, we show that the set of probability measures to be considered is convex and compact so that existence follows.

2.5.1 Stationary Distribution of Deficits

Fix a typical agent i and consider the state process $\{d_i[k]\}_{k=-1}^{\infty}$. This is a Markov process in the mean-field setting: if there is no regeneration, then the deficit changes as per the allocation and the number of B2D packets received, and is chosen via the regeneration distribution otherwise. The allocation is a function of the past d_i, the number B2D packets received and the state of the other agents. The number of B2D packets received and the state of the other agents are chosen $i.i.d$ in every frame. This Markov process has an invariant transition kernel. We construct it by first presenting the form given the past state and the allocations, namely,

$$\mathbb{P}(d_i[k] \in B | d_i[k-1] = d, e_i[k] = e, \mathbf{a}) = \delta 1_{\left\{\left(d+\eta_i-\chi_i(\mathbf{a},(d,e))\right)^+ \in B\right\}} + (1-\delta)\Psi(B), \tag{2.16}$$

where $B \subseteq \mathbb{R}^+$ is a Borel set and Ψ is the density function of the regeneration process for deficit. In the above expression, the first term corresponds to the event that agent i can either decode the packet using D2D transmissions or not, and the second term captures the event that the agent regenerates after frame k. Using (2.16) we can define the one-step transition kernel $\tilde{\Upsilon}$ for the Markov process as

$$\begin{aligned}\tilde{\Upsilon}(B,d) = \mathbb{P}(d_i[k] \in B | d_i[k-1] = d) = \delta \int 1_{\left\{\left(d+\eta_i-\chi_i(\mathbf{a}^*((d,e),\hat{\boldsymbol{\theta}}_{-i}),(d,e))\right)^+ \in B\right\}} \\ \times d(\otimes\rho^{M-1} \times \otimes\zeta^{M-1})(\hat{\boldsymbol{\theta}}_{-i}) d\zeta(e) + (1-\delta)\Psi(B).\end{aligned} \tag{2.17}$$

For later use we also define the transition kernel without regeneration but one obtained by averaging the states of the other users while retaining the state of user i,

i.e.,

$$\begin{aligned}\Upsilon(B|d,e) =&\mathbb{P}(d_i[k] \in B| \text{ no regeneration}, d_i[k-1]=d, e_i[k]=e)\\ =&\int 1_{\left\{\left(d+\eta_i-\chi_i(\mathbf{a}^*((d,e),\hat{\boldsymbol{\theta}}_{-i}),(d,e))\right)^+ \in B\right\}} \times d(\otimes\rho^{M-1}\times\otimes\zeta^{M-1})(\hat{\boldsymbol{\theta}}_{-i})d\zeta(e).\end{aligned} \tag{2.18}$$

The k fold iteration of this transition kernel is denoted by $\Upsilon^{(k)}$.

Lemma 4 *The Markov chain where the allocation is determined using* (2.7) *based on choosing the states of all users other than i* i.i.d. *with distribution $\rho\times\zeta$ and the number of B2D packets of user i independently with distribution ζ, and the transition probabilities in (2.16) is positive Harris recurrent and has a unique stationary distribution. We denote the unique stationary distribution for the deficit of a typical agent by $\Pi_{\rho\times\zeta}$; the dependence on Ψ is suppressed. The expression of this stationary distribution $\Pi_{\rho\times\zeta}$ in term of $\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E)$ is given as,*

$$\Pi_{\rho\times\zeta}(B)=\sum_{k=0}^{\infty}(1-\delta)\delta^k\mathbb{E}_\Psi(\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E)), \tag{2.19}$$

where $D=\{D_k\}_{k\in\mathbb{N}}$ is the deficit process, $E=\{E_k\}_{k\in\mathbb{N}}$ is the B2D packet reception process, and $\mathbb{E}_\Psi(\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E))=\int\Upsilon^{(k)}_{\rho\times\zeta}(B|d,e)d\Psi(d)d\zeta(e)$.

2.5.2 Agent and Cluster Decision Problems

Suppose that each agent has common information about the distribution for the deficit $\rho\in\mathcal{M}_1(\mathbb{K})$ (where $\mathcal{M}_1(\mathbb{K})$ is the set of probability measures on $\mathbb{K}$); this is one of the mean-field assumptions. We further assume that $\rho\in\mathcal{P}$ where

$$\mathcal{P}=\big\{\rho|\rho\in\mathcal{M}_1(\mathbb{K}) \text{ with finite mean}\big\}. \tag{2.20}$$

We will also assume that the regeneration distribution $\Psi \in \mathcal{P}$. From Section 2.4, the best strategy for each agent is to truthfully reveal its state based on the transfers suggested in each frame as per (2.11). Then each cluster simply maximizes the system value function by choosing the greedy optimal allocation based on (2.7).

2.5.3 Mean Field Equilibrium

Given the distribution for deficit ρ and the station distribution $\Pi_{\rho\times\zeta}$, we have the following definition.

Definition 2 *(Mean field equilibrium). Let ρ be the common cumulative distribution for deficit and telling-truth is the optimal policy for each agent in every frame. Then, we say that the given ρ along with the truth-telling behavior constitutes a mean field equilibrium if*

$$\rho(d) = \Pi_{\rho\times\zeta}(d), \forall d \in \mathbb{K}. \tag{2.21}$$

2.6 Existence of MFE

The main result showing the existence of MFE is as follows.

Theorem 3 *There exists an MFE of ρ and truth-telling policy such that $\rho(d) = \Pi_{\rho\times\zeta}(d)$, $\forall d \in \mathbb{K}$.*

As mentioned earlier, we will be specializing to the space $\mathcal{M}_1(\mathbb{K})$, its subset $\mathcal{P}$ and further subsets of $\mathcal{P}$. The primary topology on $\mathcal{M}_1(\mathbb{K})$ that we will consider is the uniform norm topology, i.e., using the l_∞ norm given by $\|\rho\| = \max_{d\in\mathbb{K}} \rho(d)$. Another topology on $\mathcal{M}_1(\mathbb{K})$ that we will use is the point-wise convergence topology, i.e., $\{\rho_n\}_{n=1}^{\infty} \subset \mathcal{M}_1(\mathbb{K})$ converges to $\rho \in \mathcal{M}_1(\mathbb{K})$ point-wise if $\lim_{n\to\infty} \rho_n(d) = \rho(d)$ for all $d \in \mathbb{K}$; it is easily verified that the convergence is the same as weak convergence of measures. Also, define the mapping Π^* that takes ρ to the invariant stationary

distribution $\Pi_{\rho\times\zeta}(\cdot)$. Let $\mathcal{P}' \subset \mathcal{P}$. We will use the Schauder fixed point theorem to prove existence which is given as follows.

Theorem 4 *(Schauder Fixed Point Theorem). Suppose $\mathcal{F}(\mathcal{P}') \subset \mathcal{P}'$, $\mathcal{F}$ is continuous and $\mathcal{F}(\mathcal{P}')$ is contained in a convex and compact subset of $\mathcal{P}'$, then $\mathcal{F}$ has a fixed point.*

Note that from the definition of $\mathcal{P}$, it is already convex. Then in the following section, we will prove that under the topology generated by the uniform norm, Π^* is continuous and the image of Π^* for a specific subset $\mathcal{P}'$ is pre-compact.

2.6.1 Steps to Prove MFE Existence

We first need to prove the continuity of Π^* with the uniform norm topology. For this we will start by showing that for any sequence $\rho_n \to \rho$ with $\rho_n, \rho \in \mathcal{P}$ in uniform norm, $\Pi^*(\rho_n) \Rightarrow \Pi^*(\rho)$ (where $\Rightarrow$ denotes weak convergence). Finally, using some properties of $\mathcal{M}_1(\mathbb{K})$ we will strengthen the convergence result to prove that $\Pi^*(\rho_n) \to \Pi^*(\rho)$ in uniform norm too.

2.6.1.1 Continuity of the Mapping Π^*

We will restrict our attention to subset of probability measures $\mathcal{P}(F) \subset \mathcal{M}_1(\mathbb{K})$ such that

$$\mathcal{P}(F) = \left\{ \rho \in \mathcal{M}_1(\mathbb{K}) : \sum_{d\in\mathbb{K}} d\rho(d) \leq F \right\}, \tag{2.22}$$

where F is a given non-negative constant; in other words, probability measures with a specified bound on the mean and not just a finite mean. We will assume that the regeneration distribution $\Psi \in \mathcal{P}(F')$ for some F'. Later on we will specify the values of F and F' to be used.

We start with the following preliminary result that establishes compactness of sets like $\mathcal{P}(F)$ in the uniform norm topology; note that convexity is immediate.

Lemma 5 *Given a sequence of non-negative numbers* $\{b_n\}_{n\in\mathbb{N}}$ *such that* $\lim_{n\to\infty} b_n = 0$*, then* $\mathcal{C} = \{x : |x_n| \leq b_n \ \forall n \in \mathbb{N}\}$ *is a compact subset of* l_∞ *and sequences of elements from* $\mathcal{C}$ *that converge point-wise also converge uniformly.*

One can also use the Cantor diagonalization procedure to show sequential compactness in the proof above.

We have an immediate corollary of this result.

Corollary 1 *The set of probability measures* $\mathcal{P}(F)$ *on* $\mathbb{K}$ *is a compact set of* l_∞ *for every* $F \in \mathbb{R}_+$*.*

Proof For any $\rho \in \mathcal{P}(F)$, $p(d_1) \leq 1$ and by Markov's inequality for $n > 1$

$$p(d_n) \leq \sum_{k=n}^{\infty} p(d_k) \leq \frac{F}{d_n}, \tag{2.23}$$

with $\lim_{n\to\infty} \frac{F}{d_n} = 0$. Using Lemma 5 the result follows.

Next, we present a coupling result from Thorisson [86, Theorem 6.1, Chapter 1]. This result will be used in proving continuity of the stationary distribution of the deficit process under the topology of point-wise convergence and in strengthening the convergence result.

Theorem 5 *Let* $\{\rho_n\}_{n=1}^{\infty} \in \mathcal{M}_1(\mathbb{K})$ *converge weakly to* $\rho \in \mathcal{M}_1(\mathbb{K})$*, then there exists a coupling, i.e., random variables* $\{X_n\}_{n=1}^{\infty}$*,* X *on a common probability space and a random integer* N *such that* $X_n \sim \rho_n$ *for all* $n \in \mathbb{N}$*,* $X \sim \rho$ *and* $X_n = X$ *for* $n \geq N$*.*

This result shows that weak convergence of probability measures on $\mathbb{K}$ is equivalent to convergence of probability measures in total variation norm, and hence, also in uniform norm.

Next we show that $\Pi_{\rho\times\zeta} \in \mathcal{P}(F)$ whenever $\rho \in \mathcal{P}(F)$.

Lemma 6 *If $\rho \in \mathcal{P}(F)$ for $F \geq \frac{\delta\eta}{1-\delta}$ and the regeneration distribution $\Psi \in \mathcal{P}(F')$ for $F' \leq F - \frac{\delta\eta}{1-\delta}$, then the stationary distribution of the deficit process of any specific user $\Pi_{\rho\times\zeta} \in \mathcal{P}(F)$.*

Next we show continuity properties of the mapping Π^*.

Theorem 6 *The mapping $\Pi^* : \mathcal{P}(F) \mapsto \mathcal{P}(F)$ is continuous in the uniform topology. In addition, Π^* has a fixed point in $\mathcal{P}(F)$.*

Theorem 7 *The MFE is unique.*

As mentioned earlier, we constrain our analysis to the case of D2D transmissions being error-free. We give a discussion on generalizing the D2D transmission model in [61].

2.7 Passage to the Mean Field Limit

We gave an overview of the finite agent system in Sections 2.1.1.1 (description of FiniteDPM) and 2.3. Here, we briefly discuss the passage between the finite agent system and the mean field model that we have used throughout the section. As in other literature on repeated games under the mean field setup [47, 69], we have considered the system with an infinitely large number of agents at finite time. It is straight-forward to follow the steps in [47, 69] to prove convergence of the finite agent system to the mean-field model in our context. However, to the best of our knowledge, the study of mean field games as time also becomes infinitely large is

currently open. There has been recent work in non-game-theoretical settings (using a fixed policy) studying the question of the conditions required to ensure that the mean field model is indeed the limiting case of the finite system when time becomes asymptotically large [10, 16]. In the case of our system, the set of measures that we consider is tight, since they are all stochastically dominated by a fictitious system in which no D2D transmissions happen and the agents' deficits simply increase and then they regenerate. Furthermore, we showed in Theorem 7 that the MFE, which is efficient, dominant strategy incentive compatible and per-period individually rational, is unique. We believe that these two properties might aid us in characterizing the equilibrium as time becomes large, and we defer this problem to future work.

2.8 Value Determination

We now turn to computing the system value from the viewpoint of a cluster and also a typical agent (say 1). Here, we suppose there are $M = 4$ agents in each cluster, and all have $\eta = 0.95$, $\delta = 0.9995$. Hence, each agent spends an average of 2000 frames in the system before leaving. A new agent has a deficit drawn uniformly at random from the interval $[0, 13]$. Each agent needs to receive $N = 10$ packets to decode the block, and there are $T = 8$ time slots in each frame. We wish to determine the value function from the perspective of the cluster and from the perspective of agent 1, using (2.6) and (2.8).

The following observation is useful to determine the allocations $\mathbf{a}^*$ and $\tilde{\mathbf{a}}$. It is straightforward to find $\mathbf{a}^*$, since it simply follows Algorithm 1. Now, consider $\tilde{\mathbf{a}}$. It is simple to see that it too would follow Phases 1 and 2 of Algorithm 1. Then, from the perspective of agent 1, after the completion of these two phases, there are only two classes of allocations–those in which he transmits and those in which he does not. Now, since all the other agents that agent 1 comes in contact with in the future

are drawn from $[\otimes\rho^{M-1}, \otimes\zeta^{M-1}]$, the allocation should follow a greedy minimization with respect to the other agents. Thus, we only need consider two allocations while conducting value iterations: min-deficit-first with agent 1 (identical to Phase 3 of Algorithm 1) and min-deficit-first without agent 1 (just set aside agent 1 in Phase 3 of Algorithm 1).

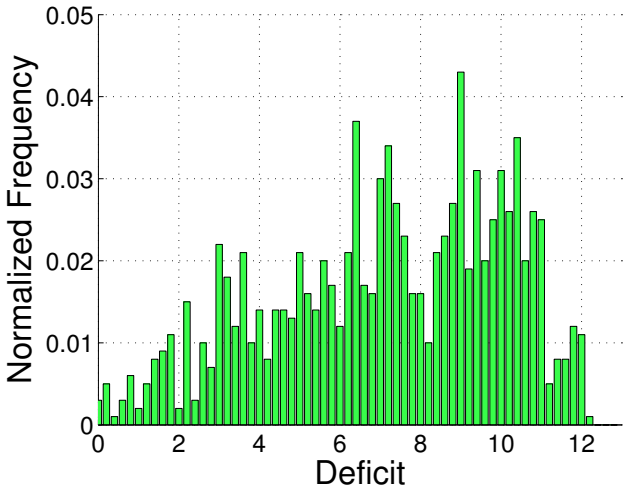

Figure 2.4: Deficit distribution.

We first run the system according to Algorithm 1, and use the results to find the empirical deficit distribution, denoted by R. This is identical to the Mean Field deficit distribution. The empirical distribution of deficit R, is shown in Figure 2.4. We find that deficit lies in the range $0 - 13$.

With $\eta = 0.95$, the (countable) deficit set is $\{0, 0.05, 0.1, 0.15, ...\}$. With a deficit range of $0 - 13$, there are totally 260 potential values for deficit. For the number of B2D chunks received e, we take values 3, 4 and 5 (uniformly). Therefore, there are totally $260^4 \times 3^4$ states in the system. Using R to represent the MF deficit

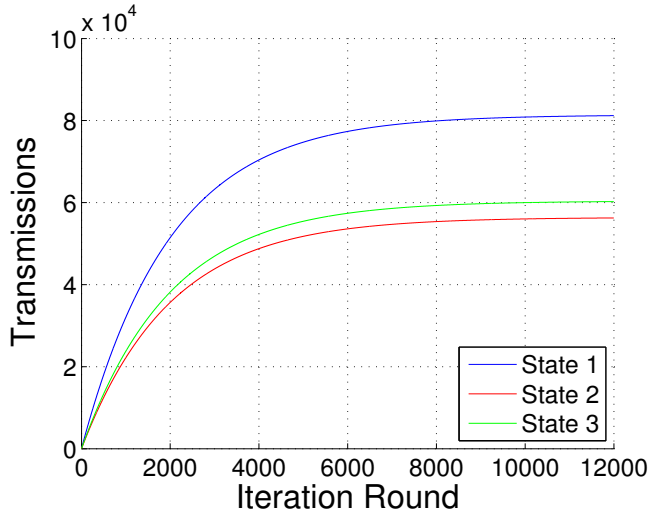

Figure 2.5: Convergence of value iteration.

distribution, and a linear holding cost function, we run value iteration; we present an example for a few states in Figure 2.5. We thus obtain the mean field value functions.

The empirical distribution of the average discounted transfers over the lifetime of each device is shown in Figure 2.6. The average transfer is 18039. We will discuss the economic implications of this observation after describing the Android experiments in the next section.

2.9 Android Implementation

We now describe experiments on an Android testbed using a cluster size of four Google Nexus 7 tablets. We modified the kernel of Android v 4.3 to simultaneously allow both WiFi and 3G interfaces to transmit and receive data.

Our system consists of a server application on a desktop that codes data and sends it to the tablets over the Internet, an Android app that receives data over Internet on a 3G interface and shares it over the WiFi interface, and a monitor that keeps track

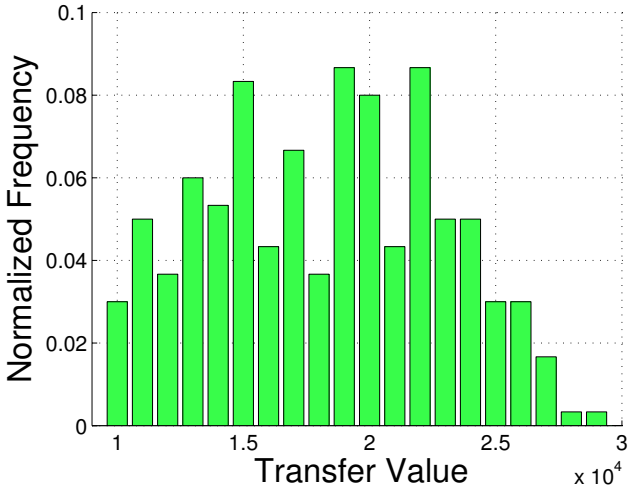

Figure 2.6: Transfer distribution.

of the state of the system and generates a trace of events. The server initializes each tablet in the system with a randomly selected number of chunks. Additionally, churn is emulated in the system by making the application on the tablet reset randomly with a probability $\bar{\delta} = 5 \times 10^{-4}$ (*i.e.*, $\delta = 0.9995$).

We set the frame duration as 500 ms. Since we have $\delta = 0.9995$, this means that the average duration that a device spends in the system is 1000 seconds. We use an MP3 music file as the data, and divide it into blocks, with the blocks being further divided into chunks. Chunks are generated using an open source random linear coding library [2], using field size 256 and 10 degrees of freedom per block. Hence, a block is decodable with high probability if 10 chunks are received successfully. Each chunk has an average size of 1500 Bytes, and has a header that contains the frame number it corresponds to as well as its current deficit. The system maintains synchronization by observing these frame numbers.

The allocation algorithm proceeds as suggested by Algorithm 1. We approximate the three phases by setting back-off times for D2D access. Devices that cannot com-

plete (*i.e.*, Phase 1 devices) should be the most aggressive in D2D channel access. We set them to randomly back-off between 1 and 5 ms before transmission. Devices that can afford to transmit some number of chunks (Phase 2) should be less aggressive, and transmit chunks by backing off between 1 and 15 ms. Finally, each device enters Phase 3, and modulates its aggressiveness based on deficits. Each device normalizes its deficit based on the values of deficits that it sees from all other transmissions, and backsoff proportional to this deficit within the interval of 5 to 15 ms. The average error in value due to a back-off based implementation is about $10 - 15\%$.

We conducted experiments to determine the stable delivery ratio achieved using D2D for different B2D initializations per frame. We present some sample deficit trajectories in Figure 2.7. The random resets emulating peer churn are visible as sharp changes in the deficit. We found that on average, B2D transfer of 4 chunks to each device is sufficient to ensure a delivery ratio of over 0.95. Hence, it is easy to achieve a 60% reduction in B2D usage, while maintaining a high QoE.

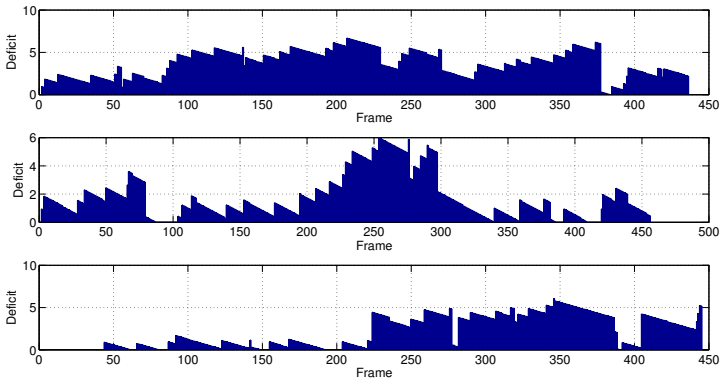

Figure 2.7: Sample deficit trajectories. We have used $\delta = 0.98$ in this run to illustrate frequent resets, which cause sharp decreases or increases.

2.10 System Viability

We saw in Section 2.8 that the average transfer to each agent is positive, meaning that the agents need to obtain some kind of subsidy in order to use the system. What kind of subsidy should they be given? The Android experiments indicate that each agent is able to save 60% of the B2D costs when participating in the system. Would this be sufficient?

The price of B2D service is currently $10 per GB across many US cellular providers. Suppose that we consider music streaming at a rate of 250 kbps corresponding to our Android system. If each device uses only B2D communication (no D2D at all), the cost of spending 1000 seconds in the sytem is 31.25 cents. The per frame communication cost is 0.0156 cents, and we can consider this to be the value of each frame to the agent.

The experiments in Section 2.9 indicate that the agents have to utilize their B2D connection for at least 40% of the chunks to maintain the desired QoE. Hence, the value that can potentially be received by participating in the D2D system is $0.6 \times 0.0156 = 0.00936$ cents per frame. Let us assume a linear deficit cost function that takes a value of 0.00936 cents at deficit value of 15. In other words, if the agent were to experience a deficit of 15 or above in a frame, it gets no payoff from that frame. Using this linear transformation, we can translate the average transfer of 18039 (the value found in Section 2.8) over the entire 1000 seconds into a total of 11.26 cents. Thus, if each agent saves at least 11.26 cents, it has an incentive to participate in the D2D system. The actual saving is $0.6 * 31.25 = 18.75$ cents (60% of the B2D costs) per agent, which is well above the minimum required saving.

The situation is still better for video streaming at a rate of 800 kbps. A similar calculation indicates that a 16 minute video costs about $1 using pure B2D, while

the B2D cost in the hybrid system is only 40 cents, yielding a savings of 60 cents per agent. However, a saving of about 36 cents per agent is all that is needed to incentivize them to participate.

In a full implementation, each agent would place an amount (*eg.* 36 cents for a 16 minute average lifetime) in escrow with the monitor upon connecting. Each agent would receive transfers according to our mechanism, and, on average, would receive its amount back from the monitor for its contributions. Hence, the system would then be *ex-ante* budget balanced.

2.11 Conclusion

We studied the problem of providing incentives for cooperation in large scale multi-agent systems, using wireless streaming networks as an example. The objective was to incentivize truth telling about individual user states so that a system wide cost minimizing allocation can be used. We showed how a mean field approximation for large systems yields a low-complexity framework under which to design the mechanism. Finally, we implemented the system on Android devices and presented results illustrating its viability using the current price of cellular data access as the basis for transfers.

3. ENERGY COUPON: A MEAN FIELD GAME PERSPECTIVE ON DEMAND RESPONSE IN SMART GRIDS

3.1 Introduction

There has recently been much interest in understanding *societal networks,* consisting of interconnected communication, transportation, energy and other networks that are important to the functioning of human society. These systems usually have a shared resource component, and participants have to periodically take decisions on when and how much to utilize such resources. Research into these networks often takes the form of behavioral studies on decision making by the participants, and whether it is possible to provide incentives to modify their behavior in such a way that the society as a whole benefits [72, 78].

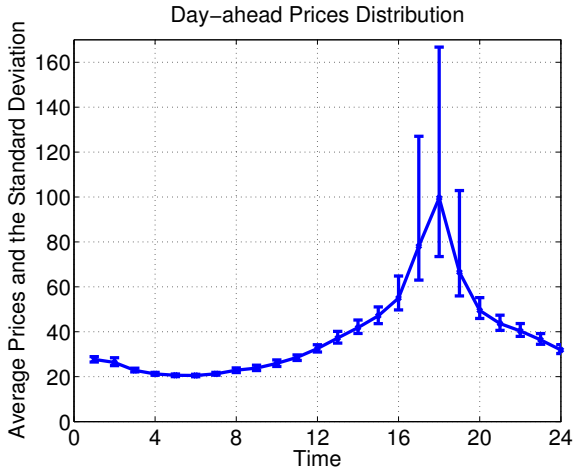

Figure 3.1: Day-ahead electricity market prices in dollars per MWh on an hourly basis between 12 AM to 12 PM, measured between June–August, 2013 in Austin, TX. Standard deviations above and below the mean are indicated separately.

Our candidate application in this section is that of a Load Serving Entity (LSE) or a Load Aggregator (LA) (*e.g.,* a utility company) trying to reduce its exposure to daily electricity market volatility by incentivizing demand response in a Smart Grid setting. The reason for our choice is the ready availability of data and reliable models for the cost and payoff structure that enables a realistic study. For instance, consider Figure 3.1, which shows the (wholesale) price of electricity at different hours of day during the summer months in Texas. The data was obtained from the Electric Reliability Council of Texas [1], an organization that manages the deregulated wholesale energy market in the state. The price shows considerable variation during the day, and peaks at about 5 PM, which is the time at which maximum demand occurs. A major source of this demand in Texas is air conditioning, which in each home is of the order of 30 kWh per day by [3]. Incentivizing customers to move a few kWh of peak-time usage to the sides of the peak each day could lead to much reduced risks of peak price borne by the LSE. When coupled with a reduction in energy usage, such demand shaping could also have a positive effect on environmental impact of power plant emissions.

As an example, we take the baseline temperature setpoint as 22.5°C, and consider a customer that every day increases the setpoint by 1°C in $5-6$ PM and decreases the setpoint by 0.5°C in the off-peak times. We will see later that even such a small change of the setpoint of AC can yield substantial savings of the order of a hundred dollars per week to the LSE when conducted over a group of fifty homes. This result is under the implicit assumption that the LSE in question is a price-taker so that changes in its demand profile are assumed not to perturb the prices. The shifting of daily energy usage could potentially cause a small increase in the mean and deviation of the internal home temperature, which is a discomfort cost borne by the customer. In our system (an actual system that we are currently developing,

and using which we intend to conduct user trials soon), the LSE awards a number of "Energy Coupons" to the customer in proportion to his usage at the non-peak times, and these coupons are used as tickets at a lottery conducted by the LSE. A higher number of coupons would be obtained by choosing an option that potentially entails more discomfort, and would also imply a higher probability of winning at the lottery. Since the customers do not observe the impact of day-ahead prices on a day-to-day basis nor do they see the aggregate demand at the LSE, the lottery scheme serves as a mechanism to transfer some of this information over to the customers by coupling them.

In our analytical model, each agent has a set of actions that it can take in each play of a repeated game, with each action having a corresponding cost. Higher cost actions yield a higher number of coupons. At the end of each play, the agents participate in a lottery in which they are randomly permuted into groups, and one or more prizes are given in each group. The state of each agent is measured using his surplus, which captures the history of plays experienced by the agent, and is a proxy to capture his interest in participating in the incentive system. Each win at the lottery increases the surplus, and each loss decreases it. Furthermore, we assume that the agent has a prospect utility function that is increasing and concave for positive surplus and convex for negative surplus. This prospect theory model captures the decision making under risk and uncertainty for agents. Any agent could depart from the system with a fixed probability, and a departing agent is replaced by a new entrant with a randomly drawn surplus. The question we answer in this section then is how would agents decide on what action to take at each play?

3.1.1 Prospect Theory

Most previous studies account for uncertainty in agent payoffs by means of the *expected utility theory* (EUT). Under EUT, the objective of the decision maker is to maximize the probabilistically weighted average utilities under different outcomes. However, EUT does not incorporate observed behavior of human agents, who can take decisions deviating from this conventional norm. For example, empirical studies have shown that agents ascribe higher weights to rare, positive events (such as winning at a lottery) [50].

Prospect theory (PT) [50, 51, 87, 88] is perhaps the most well-known alternative theory to EUT. It was originally developed for binary lotteries [50] and later refined to deal with issues related to multiple outcomes and valuations [88]. This Nobel-prize-in-economics-winning theory has been observed to provide a more accurate description of decision making under risk and uncertainty than EUT. There are three key characteristics of PT. First, the value function is concave for gains, convex for losses, and steeper for losses than for gains. This feature is due to the observation that most decision makers prefer avoiding losses to achieving gains. Thus, the value function is usually S-shaped. Second, a nonlinear transformation of the probability scale is in effect, *i.e.*, decision makers will overweight low probability events and underweight high probability events. The weighting function usually has an inverted S-shape, *i.e.*, it is steepest near endpoints and shallower in the middle of the range, which captures the behaviors related to risk seeking and risk aversion. Finally the third, the framing effect is accounted for, *i.e.*, the decision maker takes into account the relative gains or losses with respect to a reference point rather than the final asset position. As PT fits better in reality than EUT based on many empirical studies, it has been widely used in many contexts such as social sciences [32,39], communication

networks [24, 65, 92] and smart grids [90, 91]. Since we study equilibria that arise through agents' repeated play in lotteries, we use prospect theory as opposed to expected utility theory to account for agent-perceived value while taking decisions.

3.1.2 Mean Field Games

The problem described is an example of a dynamic Bayesian game with incomplete information, wherein each player has to estimate the actions of all his potential opponents in the current lottery (and in the future) without knowing their surpluses, play a best response, and update his beliefs about their states of surplus based on the outcome of the lottery. However, since the set of agents is large and, from the perspective of each agent, each lottery is conducted with a randomly drawn finite set of opponents, an accurate approximation for any agent is to assume that the states of his opponents (and hence actions) are independent of each other. This is the setting of a Mean Field Game (MFG) [44, 49, 58], which we will use as a framework to study equilibria in societal networks. Here, the system is viewed from the perspective of a single agent, who assumes that each opponent's action would be drawn independently from an assumed distribution, and plays a best response action. We say that the system is at a Mean Field Equilibrium (MFE) if this best response action turns out to be a sample drawn from the assumed distribution. We will use such a MFG model to model dynamics in societal networks.

In our analysis, we can prove that regardless of the exact form of the utility function, a MFE exists in our system. In our numerical study, we use the prospect utility function for study, we can observe further properties of the value function based on this special utility function.

3.1.3 Demand Response in Deregulated Markets

Demand Response is the term used to refer to the idea of customers being incentivized in some manner to change their normal electricity usage patterns in response to peaks in the wholesale price of electric power [5]. Many methods of achieving demand response exist, including an extreme one of turning off power for short intervals to customers a few times a year if the price is very high. In any method of achieving demand response, customers expect a subsidy in return, often in terms of a reduced electricity bill.

The idea is particularly relevant in deregulated electricity markets that exist in several US states, such as in Texas, wherein the firm that serves customer demand might have no infrastructure of its own, and merely buys on the wholesale market and sells to the home consumer. Customers have a choice between many different LSEs that they can obtain service from. For instance, many urban neighborhoods in Texas are served by $5 - 10$ LSEs and customers can periodically choose to sign contracts of 1, 6 and 12 months with them.

3.1.4 Main Results

Our objective in this section is to design a system that would incentivize the convergence of user action profiles to one that would result in large savings to the LSE. Our contributions in this section towards such an objective are as follows:

1. We propose a mean field model to capture the dynamics in societal networks. Our model is well suited to large scale systems in which any given subset of agents interact only rarely. This kind of system satisfies a chaos hypothesis that enables us to use the mean field approximation to accurately model agent interactions. The state of the mean field agent is his surplus, which forms a Markov process that increases by winning and decreases by losing at the lottery.

Our mean field model of societal networks is quite general, and can be applied to different incentive schemes that are currently being proposed in the field of public transportation and communication network usage.

2. We develop a characterization of a lottery in which multiple rewards can be distributed, but with each participant getting at most one by withdrawing the winner in each round. Each lottery is played amongst a cluster of M agents drawn from a random permutation of the set of all agents. While the exact form of the lottery is not critical to our results, we present it for completeness.

3. We characterize the best response policy of the mean field agent, using a dynamic programming formulation. We find that under our assumptions the value function is continuous in the action distribution. Further, we show using this result that given our ordering in which higher cost actions result in a higher probability of winning the lottery (due to more coupons being given), the choice of one action versus another depends on thresholds in the surplus, i.e., we obtain a threshold policy for the action choices.

4. The probability of winning the lottery defines the transition kernel (along with the regeneration distribution) of the Markov process of the surplus, and hence maps an assumed distribution across competitors states to a resultant stationary distribution. We show the existence of a fixed point of this kernel, which is the MFE, by using Kakutani's fixed point theorem. Our proof of the existence of MFE doesn't depend on the shape of the utility function, which is quite general. Since we have a discrete action and state space, showing a fixed point in the space of stationary distributions is quite intricate.

5. We develop an accurate model of the daily usage of electricity in each hour,

using available measurements over several months in Texas. We also use the data on wholesale electricity prices during the interval to calculate what times of day would yield the best returns to rewards. We show that if customers are willing to change the setpoints of AC as small as 1°C each day then each week the LSE gains a benefit of the order of a $100 over a cluster of 50 homes. Further, we show that such behavior can be incentivized by offering a weekly prize of $40 at the lottery.

3.1.5 Related Work

In terms of the mean field game, our framework is based on work such as [46, 60, 62, 69]. In [46] the setting is that of advertisers bidding for spots on a webpage, and the focus is on learning the value of winning (making a sale though the advertisement) as time proceeds. In [69], apps on smart phones bid for service from a cellular base station, and the goal is to ensure that the service regime that results has low per-packet delays. In both works, the existence of an MFE with desired properties is proved. In [60, 62], even though the state-space and deficit dynamics are similar, the reward structure is determined using a resource allocation problem necessitating a different proof technique and the exploration of truthful dynamic mechanisms.

Nudge systems are typically designed and used to encourage socially beneficial behaviors and individually beneficial behaviors. For instance, lottery schemes are widely used in practice to incentivize good behavior, e.g., to combat (sales) tax evasion in Brazil [76], Portugal [77], Taiwan [22], and for Internet congestion management [67]. Similarly, [72, 78] provide experimental results on designing lottery based "nudge engines" to provide incentives to participants to modify their behaviors in the context of evenly distributing load on public transportation. In another scheme, [6] study the impact of nudging on social welfare by sending one-year home

energy reports to participants and using multiple price lists to determine participants' willingness to stay in the system for the next year. Our system is a form of nudge engine, but our focus is on analytical characterization of system behavior and attained equilibria with large number of customers with repeated decision-making. We aim to design incentive schemes to modify customer behavior such that the system as the whole benefits from the attained equilibrium.

Our idea of offering coupons for electricity usage at certain times of day is based on one presented in [94], which suggests offering such incentives to coincide with the predicted realtime price peaks. An experimental trial based on a similar idea is described in [15], in which the focus is on designing algorithms to coordinate demand flexibility to enable the full utilization of variable renewable generation. In [38], this kind of system is modeled as a Stackelberg game with two stages: setting the coupon values followed by consumer choice. The decision making model in all the above research is myopic. [83] study demand-response as trading off the cost of an action (such as modifying energy usage) against the probability of winning at a lottery in terms of a mean field game. However, the game is played in a single step according to their model, and there is no evolution of state or dynamics based on repeated play. Further, their conception of the mean field equilibrium is that the mean value of the action distribution (not the distribution itself) is invariant. Unlike these models, we are interested in characterizing repeated consumer choice with state evolution when the number of customers is large, and identifying the action distribution and benefits (if any) of the resulting equilibrium.

A rich literature studies lottery schemes, and here we can only hope to cover a fraction of them that we see most relevant. In this section, we model lotteries as choosing a random permutation of the M agents participating in it, and picking the first K of them as winners, with the distribution on the symmetric group of

permutations of $\{1, \cdots, M\}$ being a function of the coupons assigned to the different actions. Assuming that different actions yield different numbers of coupons, we will choose the distribution such that more coupons results in a higher probability of winning. There are various probabilistic models on permutations in the ranking literature [68,81], Here we use the popular Plackett-Luce model [45] to implement our lotteries. While the Plackett-Luce model is used for concreteness, other probabilistic models on permutations such as the Thurstone model [68] can also be used with the number of coupons as parameters of the distribution as long as more coupons results in a higher probability of winning.

3.1.6 Organization

This section is organized as follows. In Section 3.2, we introduce our mean field model. In Section 3.3 we develop a characterization of a lottery in which multiple rewards can be distributed, but with each participant getting at most one by withdrawing the winner in each round. We discuss the basic property of the optimal value function in Section 3.4. The existence of MFE is considered in Section 3.5. We characterize the best response policy of the mean field agent, using a dynamic programming formulation in Section 3.6. We then conduct numerical studies in Section 3.7, on utilizing our framework to the context of electricity markets. We conclude in Section 3.8. To ease exposition of our results, all proofs are relegated to the Appendix B.

3.2 Mean Field Model

We consider a general model of a societal network in which the number of agents is large. Agents have a discrete set of actions available to them, and must take one of these actions at each discrete time instant. The actions result in the agents receiving coupons, with higher cost actions resulting in more coupons. The agents are then

randomly permuted into clusters of size M and a lottery is held using the coupons to win real rewards. Thus, agents must take their actions under some belief about the likely actions, and hence the likely coupons held by their competitors in the auctions.

Figure 3.2 illustrates the mean field approximation of our model, which is an accurate representation of the Bayesian system when the number of agents is large [36,46]. The diagram is drawn from the perspective of a single agent (w.l.o.g, let this be agent 1), who assumes that the actions played by each of his opponents would be drawn independently of each other from the probability mass function ρ. In this section, we will introduce the notation, costs and payoffs of the agent, and provide a brief description of the policy space and equilibrium.

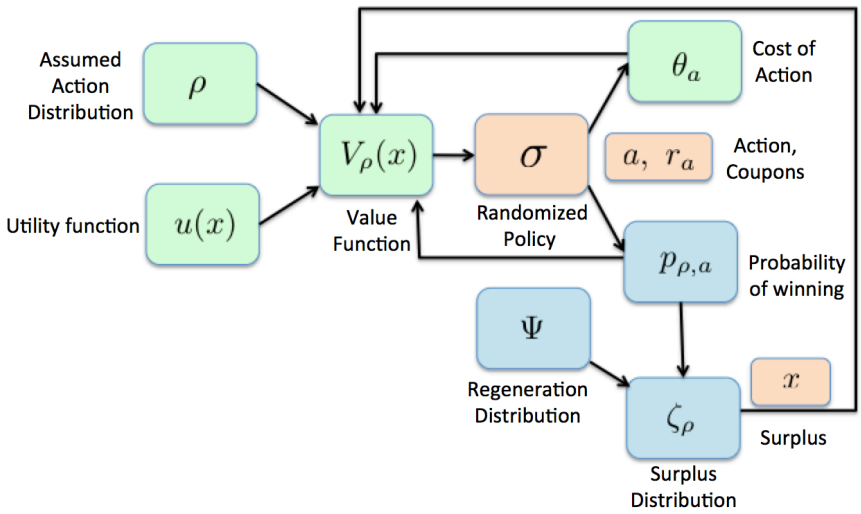

Figure 3.2: Mean field game.

Time: Time is discrete and indexed by $k \in \{0, 1, \cdots\}$.

Agents: As discussed above, the total number of agents is infinite, and in the MFG, we consider a generic agent 1 who in each lottery will be paired with $M - 1$

other agents drawn randomly from the infinite population.

Actions: We suppose that each agent has the same action space denoted as $\mathcal{A} = \{1, 2, \cdots, |\mathcal{A}|\}$. Hence, the action that this agent takes at time k is $a[k] \in \mathcal{A}$. Under the mean field assumption, the actions of the other agents would be drawn independently from the p.m.f. $\rho = [b_1, b_2, \cdots, b_{|\mathcal{A}|}]$, where b_a is the probability mass associated with action a. We call ρ as the assumed action distribution.

Costs: Each action $a \in \mathcal{A}$ taken at time k has a corresponding cost θ_a. This cost is fixed and represents the discomfort suffered by the agent in having to take that action.

Coupons: When agent takes an action a, it is awarded some fixed number of coupons r_a for playing that action. These coupons are then used by the agents as lottery tickets.

Lottery: We suppose that there are only K rewards for agents in one cluster, where K is a fixed number less than M. The probability of winning is based on the number of coupons that each agent possesses. We model each lottery as choosing a permutation of the M agents participating in it, and picking the first K of them as winners.

States: The agent keeps track of his history of wins and losses in the lotteries by means of his net surplus at time k, denoted $x[k]$. The value of surplus is the state of the agent, and is updated in a Markovian fashion as follows:

$$x[k+1] = \begin{cases} x[k] + w, & \text{if agent 1 wins the lottery} \\ x[k] - l, & \text{if agent 1 loses the lottery} \end{cases} \tag{3.1}$$

where w and l is the impact of winning or losing on surplus. Effectively, the assumption is that the agent expects to win at least an amount l at each lottery. Not

receiving this amount would decrease his surplus. Similarly, if the prize money at the lottery is $w + l$, the increase in surplus due to winning is w. Surplus values are discrete, and the set of possible values is given by a countable $\mathbb{X}$, that ranges from $(-\infty, +\infty)$.

Value function for prospect: The impact of surplus on the agent's happiness is modeled by an S-shaped utility function $u(x[k])$, which is monotone increasing, concave for a positive surplus and convex for a negative surplus. Moreover, the impact of loss is usually larger than that of gain of the same absolute value. Following [88], we use the value function for prospect

$$u(x) = \begin{cases} u^+(x) = x^\gamma, & x \geq 0 \\ u^-(x) = -\varphi(-x)^\gamma, & x < 0, \end{cases} \tag{3.2}$$

where $\varphi > 1$ is the loss penalty parameter and $0 < \gamma < 1$ is the risk aversion parameter. A larger φ means that the operator is more loss averse, while a smaller γ (*i.e.*, the value function is more concave) indicates that the operator is more risk seeking. From past empirical studies [51, 88], realistic values are $\varphi = 2.25$ and $\gamma = 0.88$.

Weighting function for prospect: In earlier studies [79], it has been observed that people tend to subjectively weight uncertain outcomes in real-life decision making. In the proposed game, this weighting factors capture the agent's subjective evaluation on the mixed strategy of its opponents. Thus, under PT, instead of objectively observing the probability of winning the lottery $p_{\rho,a}$, each user perceives a weighted version of it, $\phi(p_{\rho,a})$. Here, $\phi(\cdot)$ is a nonlinear transformation that maps the objective probability to a subjective one, which is monotonic increasing in probability. It has been shown in many PT studies that, people usually overweight low

probability outcomes and underweight high probability outcomes. Following [79], we use the weighting function

$$\phi(p) = \exp(-(-\ln p)^{\xi}), \quad \text{for} \quad 0 \leq p \leq 1 \tag{3.3}$$

where $\xi \in (0, 1]$ is the objective weight that characterizes the distortion between subjective and objective probability. Note that under the extreme case of $\xi = 1$, (3.3) reduces to the conventional EUT probability, i.e., $\phi(p) = p$.

Regeneration: We assume that an agent may choose to quit the system at any time. This event occurs with probability $1-\beta$, where $\beta \in (0, 1)$. When this happens, a new agent takes the place of the old one, and his state is drawn from a probability mass function Ψ.

Best Response Policy: The agent must choose an action at each time; we including staying with the status-quo/baseline as an action too. The green/light tiles in Figure 3.2 relate to the problem of the agent determining his best response policy. The agent assumes that the actions taken by each of his $M-1$ opponents are drawn independently from probability mass function ρ. Given this assumption, the state of his surplus is x and current utility is $u(x)$, the agent must calculate the probability of winning at the lottery $p_{\rho,a}(x)$, if he were to take action $a(x) \in \mathcal{A}$, incurring a cost $\theta_{a(x)}$ and gaining $r_{a(x)}$ coupons. Since the agent must take this decision repeatedly, he must solve a dynamic program to determine his optimal policy. There could be many best response actions, and we assume that the agent chooses a randomized policy $\sigma(x) \triangleq [\sigma_1(x), \sigma_2(x), \cdots, \sigma_a(x), \cdots, \sigma_{|\mathcal{A}|}(x)]$, in which $\sigma_a(x)$ specifies the probability of playing action a when the agent's surplus is x. The action taken by the agent is a random variable $A \sim \sigma(x)$. The details of the lottery and how to calculate the probability of success are given in Section 3.3. The properties of the best response

policy are described in detail in Section 3.6.

Stationary Surplus Distribution: The assumed action distribution ρ, and the best-response randomized policy $\sigma(x)$ yield the state transition kernel of the Markov chain corresponding to the surplus, via the probability of winning the lottery $p_{\rho,a}(x)$. This is illustrated by means of the blue/dark tiles in Figure 3.2. The transition kernel also is influenced by the regeneration distribution Ψ. The stationary distribution of surplus associated with the transition kernel is denoted as ζ_ρ. This stationary distribution of the single mean field agent is equivalent to the one-step empirical state distribution of infinite agents who all assume that the actions of their competitors would be drawn from ρ.

Mean Field Equilibrium: The triple of an assumed action distribution ρ, randomized policy σ and stationary surplus distribution ζ gets mapped into a triple of action distribution $\tilde{\rho}$, best-response randomized policy $\tilde{\sigma}$ and a stationary surplus distribution $\tilde{\zeta}$ via the operations described above. A fixed point of the resulting map is called an MFE. For a formal definition and details of the proof of existence see Section 3.5.

3.3 Lottery Scheme

We first construct the lottery scheme that will be used in our mean field game. We permute all the agents into clusters, where there are exactly M agents in each cluster, and conduct a lottery in each such cluster. Suppose there are K rewards for all agents in one cluster, where K is a fixed number less than M. When an agent takes an action, he/she will receive the credit (number of coupons) associated with that action. Then the probability of winning is based on the number of coupons that each agent possesses. We will model the lotteries as choosing a permutation of the M agents participating in it, and picking the first K of them as winners. Then

different lottery schemes can be interpreted as choosing different distributions on the symmetric group of permutations on M. In particular, we will use ideas from the Plackett-Luce model to implement our lotteries.

Without loss of generality, we assume that the actions are ordered in decreasing order of the costs so that $\theta_1 \geq \cdots \geq \theta_A$. In order to incentivize agents to take the more costly actions we will insist that the vector of coupons obtained for each action is also in decreasing order of the index, i.e., $r_1 \geq \cdots \geq r_A$.

The specific lottery procedure we consider is the following: for every agent m that takes action $a[m]$ and receives coupons $r_{a[m]} > 0$, we choose an exponential random variable with mean $1/r_{a[m]}$ and then pick the first K agents in increasing order of the realizations of the exponentials. Note the abuse of notation only in this section to use $a[m]$ to refer to the action of agent m. Since we consider only one lottery, we do not consider time k. Let the agent $m = 1, \ldots, M$ receive $r_{a[m]}$ number of coupons. The set of winners is a permutation over the agent indices, and we denote such a permutation by $\sigma = [\sigma_1, \sigma_2, \cdots \sigma_M]$. We then have the probability of the permutation σ given by

$$\mathbb{P}(\sigma | r_{a[1]}, \ldots, r_{a[M]}) = \prod_{n=1}^{M-1} \frac{r_{a[\sigma_n]}}{\sum_{j=n}^{M} r_{a[\sigma_j]}}. \tag{3.4}$$

Essentially, after each agent is chosen as a winner, he is removed and the next lottery is conducted just as before but with fewer agents.

We now analyze the probability of winning in our lottery. For analysis under the mean field assumption, it suffices to consider agent 1 with the coupons it gets by taking action a being denoted as $r_{a[1]}$. Let $\mathcal{M} := \{2, \ldots, M\}$, which is the set of opponents of agent 1. For these agents, suppose there are υ_n agents that choose action n, where $\sum_{n \in \mathcal{A}} \upsilon_n = M - 1$. We denote the vector of these actions by

$\vec{v} = (v_1, \ldots, v_A)$.

The conditional probability of agent 1 failing to obtain a reward is given by

$$p^L_{1,\vec{v}} = \sum_{\kappa_1 \in \mathcal{M}_1} \cdots \sum_{\kappa_K \in \mathcal{M}_K} \frac{\prod_{l=1}^K r_{a[\kappa_l]}}{\prod_{l=1}^K (r_{a[1]} + \sum_{m \in \mathcal{M}_l} r_{a[m]})},$$

where L refers to the fact that agent 1 "loses," $\mathcal{M}_1 = \mathcal{M}$, and for $l \geq 2$ we have $\mathcal{M}_l = \mathcal{M}_{l-1} \setminus \{\kappa_{l-1}\}$. Essentially, the above looks at the lottery process round by round, and is a summation of the probabilities of all permutations in which agent 1 does not appear in the first spot in any round.

The above expression considerably simplifies if the summations are instead taken over the actions $\tilde{\kappa}_l$ that the lottery winner κ_l at round $l \in \{1, \ldots, K\}$ can take. Note that we assume that we can distinguish the actions based on the number of coupons given out. If this were not true, then we could further simplify the expression by summing over the coupon space. Given a coupon/action profile $\vec{v}$, let $\mathcal{J}(\vec{v})$ denote the actions that have non-zero entries. Additionally, by $\vec{v} - \vec{1}_{\tilde{\kappa}}$ for $\tilde{\kappa} \in \mathcal{J}(\vec{v})$ denote the resulting coupon profile obtained by removing one entry at location $\tilde{\kappa}$, and by $r_{\vec{v}}$ the sum of all the coupons in profile $\vec{v}$, i.e., $\sum_{\tilde{\kappa} \in \mathcal{J}(\vec{v})} r_{\tilde{\kappa}} v_{\tilde{\kappa}}$. Then

$$p^L_{1,\vec{v}} = \sum_{\tilde{\kappa}_1 \in \mathcal{J}(\vec{v}^1)} \cdots \sum_{\tilde{\kappa}_K \in \mathcal{J}(\vec{v}^K)} \frac{\prod_{l=1}^K v^l_{\tilde{\kappa}_l} r_{\tilde{\kappa}_l}}{\prod_{l=1}^K (r_{a[1]} + r_{\vec{v}^l})}, \tag{3.5}$$

where $\vec{v}^1 = \vec{v}$, for $l = 2, \ldots, K$, $\vec{v}^l = \vec{v}^{l-1} - \vec{1}_{\tilde{\kappa}_l}$ and $v^l_{\tilde{\kappa}}$ is the number of entries at location $\tilde{\kappa}$ for coupon profile $\vec{v}^l$. Note that $p^L_{1,\vec{v}}$ is a decreasing function of $r_{a[1]}$ for every $\vec{v}$. Therefore, agent 1 comparing two actions i and j that have $r_{1,i} > r_{1,j}$ will find $p^L_{1,\vec{v}}(i) < p^L_{1,\vec{v}}(j)$ for all $\vec{v}$. Also by taking the limit of $r_{a[1]}$ going to 0, having an action with 0 coupons results in a loss probability of 1 for every $\vec{v}$.

To determine the probability of winning in the lottery we need to account for

the fact that the actions of the opponents are drawn from the distribution ρ (under the mean field assumption). Hence, the probability of obtaining the coupon profile (equivalently action profile) of the opponents $\vec{v} = (v_1, \ldots, v_A)$ is given by the multinomial formula, *i.e.,*

$$\mathbb{P}_\rho(\vec{v}) = \frac{(M-1)! \prod_{i \in \mathcal{A}} b_i^{v_i}}{\prod_{i \in \mathcal{A}} v_i!}. \tag{3.6}$$

Using (3.5) and (3.6), we obtain the winning probability for the mean field agent 1 when taking action a as

$$p_{\rho,a} = 1 - \sum_{\vec{v}:|\mathcal{J}(\vec{v})|=M-1} p_{1,\vec{v}}^L \mathbb{P}_\rho(\vec{v}). \tag{3.7}$$

By lower bounding each term in the conditional probability of not obtaining a reward we get $p_{\rho,a} \leq 1 - \frac{M-K}{M}(\frac{r_A}{r_1})^K =: \overline{p}_W \in (0,1)$. If we ran the lottery without removing the winners (and any of their coupons), we obtain a lower bound on the probability of winning that has a simpler expression. Using this simpler expression we can obtain the lower bound $p_{\rho,1} \geq 1 - (1 - \frac{r_A}{r_A + (M-1)r_1})^K =: \underline{p}_W \in (0,1)$. Note that both bounds are independent of ρ. If we allow an action that yields 0 coupons, then the above bounds become trivial with $\overline{p}_W = 1$ and $\underline{p}_W = 0$.

An important feature of our lottery scheme is that the probability of winning increases with the number of coupons given out. For simplicity we assumed a fixed reward for any win. However, we can extend the lotteries to ones where different rewards are given out at different stages, and also where the rewards are dependent on the number of coupons of the winner. For the latter, we will insist on the rewards being an increasing function of the number of coupons of the winner. Finally, we can also extend to scenarios where we choose the number of stages K is an (exogenous)

random fashion in $\{1, \ldots, M-1\}$. Since the analysis carries through unchanged except with more onerous notation, we only discuss the simplest setting.

3.4 Optimal Value Function

As discussed in Section 3.2, the mean field agent must determine the optimal action to take, given his surplus x and the assumed action distribution ρ. We follow the usual quasi-linear combination of prospect function and cost consistent with Von Neumann-Morgenstern utility functions, and under which the impact of winning or losing at a lottery is on the surplus of the agent (and not simply a one-step myopic value change). Thus, the dynamic program that the agent in prospect theory needs to solve is

$$V_\rho(x) = \max_{a(x)\in\mathcal{A}} \{u(x) - \theta_{a(x)} + \beta[\phi(p_{\rho,a}(x))V(x+w) + \phi(1-p_{\rho,a}(x))V(x-l)]\}. \tag{3.8}$$

Note that $p_{\rho,a}(x)$ is a result of a lottery that we described in detail in Section 3.3, and $\phi(\cdot)$ is the weighting function, which overweights small probabilities (win the lottery) and underweights moderate an high probabilities (loss the lottery). Here we use the weighting function defined in (3.3).

First, we need to define a set of functions as

$$\Phi = \left\{ f : \mathbb{X} \to \mathbb{R} : \sup_{x\in\mathbb{X}} \left| \frac{f(x)}{\Omega(x)} \right| < \infty \right\},$$

where $\Omega(x) = \max\{|u(x)|, 1\}$. Note that Φ is a Banach space with $\Omega-$norm,

$$||f||_\Omega = \sup_{x\in\mathbb{X}} \left| \frac{f(x)}{\Omega(x)} \right| < \infty.$$

Also define the Bellman operator T_ρ as

$$T_\rho f(x) = \max_{a(x)\in\mathcal{A}} \{u(x) - \theta_{a(x)} + \beta[\phi(p_{\rho,a}(x))f(x+w), +\phi(1-p_{\rho,a}(x))f(x-l)]\}, \quad (3.9)$$

where $f \in \Phi$.

We now show that the optimal value function $V_\rho(x)$ exists and it is continuous in ρ.

Lemma 7 *1) There exists a unique $f^* \in \Phi$, such that $T_\rho f^*(x) = f^*(x)$ for every $x \in \mathbb{X}$, and given $x \in \mathbb{X}$, for every $f \in \Phi$, we have $T_\rho^n f(x) \to f^*(x)$, as $n \to \infty$.*
2) The fixed point f^* of operator T_ρ is the unique solution of Equation (3.8)*, i.e. $f^* = V_\rho^*$.*

Lemma 8 *The value function $V_\rho(\cdot)$ is Lipschitz continuous in ρ.*

3.4.1 Stationary Distributions

For a generic agent, w.l.o.g., say agent 1, we consider the state process $\{x_1[k]\}_{k=0}^\infty$. It's a Markov chain with countable state-space $\mathbb{X}$, and it has an invariant transition kernel given by a combination of the randomized policy $\sigma(x)$ at each surplus x for any $a(x) \in \mathcal{A}$, and the lottery scheme from Section 3.3. By following this Markov policy, we get a process $\{W[k]\}_{k=0}^\infty$ that takes values in $\{\text{win}, \text{lose}\}$ with probability $p_{\rho,a}(x)$ for the win, drawn conditionally independent of the past (given $x_1[k]$). Then the transition kernel conditioned on $W[k]$ is given by

$$\mathbb{P}(x_1[k] \in B | x_1[k-1] = x, W[k]) = \beta 1_{\{x+w1_{\{W[k]=\text{win}\}} - l1_{\{W[k]=\text{lose}\}} \in B\}} + (1-\beta)\Psi(B), \quad (3.10)$$

where $B \subset \mathbb{X}$ and Ψ is the probability measure of the regeneration process for surplus. The unconditioned transition kernel is then

$$\begin{aligned}\mathbb{P}(x_1[k] \in B | x_1[k-1] = x) &= \beta \sum_{a(x)\in\mathcal{A}} \sigma_a(x) p_{\rho,a}(x) 1_{x+w\in B} \\ &+ \beta\big(1 - \sum_{a(x)\in\mathcal{A}} \sigma_a(x) p_{\rho,a}(x)\big) 1_{x-l\in B} + (1-\beta)\Psi(B).\end{aligned} \tag{3.11}$$

Lemma 9 *The Markov chain where the action policy is determined by $\sigma(x)$ based on the states of the users and the transition probabilities in (3.11) is positive recurrent and has a unique stationary surplus distribution. We denote the unique stationary surplus distribution as $\zeta_{\rho\times\sigma}$. Let $\zeta^{(k)}_{\rho\times\sigma}(B|x)$ be the surplus distribution at time k induced by the transition kernel (3.11) conditioned on the event that $X[0] = x$ and there is no regeneration until time k. $\zeta_{\rho\times\sigma}(\cdot)$ and $\zeta^{(k)}_{\rho\times\sigma}(\cdot)$ are related as follows:*

$$\begin{aligned}\zeta_{\rho\times\sigma}(B) &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}_\Psi\left(\zeta^{(k)}_{\rho\times\sigma}(B|X)\right) \\ &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \int \zeta^{(k)}_{\rho\times\sigma}(B|x) d\Psi(x).\end{aligned} \tag{3.12}$$

Thus $\zeta_{\rho\times\sigma}(B)$ in terms of $\zeta^{(k)}_{\rho\times\sigma}(B|x)$ is simply based on the properties of the conditional expectation. And note that in $\mathbb{E}_\Psi\left(\zeta^{(k)}_{\rho\times\sigma}(B|X)\right)$, the random variable X is the initial condition of the surplus, generated by Ψ. For $x \in \mathbb{X}$, the only possible one-step updates are the increase of the surplus to $x + w$ or a decrease to $x - l$, i.e. $B = \{x + w, x - l\}$.

3.5 Mean Field Equilibrium

The action distribution ρ is a probability mass function on the action set $\mathcal{A}$: let b_i be the probability of choosing action i. Note that ρ lives in the probability simplex on $\mathbb{R}^{|\mathcal{A}|}$, which is compact and convex, denote it as Γ_ρ. Let ζ be the stationary

surplus distribution and the set of all such possible surplus distributions is denoted as Γ_ζ, which is compact and convex. For a given surplus x, let $\sigma(x)$ be the action distribution at x. Denote Γ_σ as the set of all possible distributions over the action space for each x, which is compact and convex. We further assume that $\rho \in \Gamma_\rho$, $\zeta \in \Gamma_\rho$ and $\sigma(x) \in \Gamma_\sigma$ for each x.

Definition 3 *Consider the action distribution ρ, the randomized policy σ and the stationary surplus distribution ζ_ρ: (i), Given the action distribution ρ, determine the success probabilities in the lottery scheme using (3.7) and then compute the value function in (3.8). Taking the best response given by (3.8) results in an action distribution $\tilde{\sigma}$; (ii), Given action distribution ρ, following the randomized policy σ yields transition kernels for the surplus Markov chain and stationary surplus distribution $\tilde{\zeta}_\rho$, (with each transition kernel having a unique stationary distribution); and (iii), Given the stationary surplus distribution ζ_ρ, applying the randomized policy $\sigma(x)$ at each surplus x yields the distribution of actions $\tilde{\rho}$. Define the best response mapping Π^* that maps $\Gamma_\rho \otimes \Gamma_\sigma^{|\mathbb{X}|} \otimes \Gamma_\zeta$ into itself. Then we say that the assumed action distribution ρ, randomized policy σ and stationary surplus distribution ζ_ρ constitute a mean field equilibrium (MFE) if $\Pi^* : \rho \otimes \sigma \otimes \zeta_\rho \mapsto \tilde{\rho} \otimes \tilde{\sigma} \otimes \tilde{\zeta}_\rho$ has $(\rho, \sigma, \zeta_\rho)$ as a fixed point.*

3.5.1 Existence of MFE

Theorem 8 *There exists an MFE of ρ, the randomized policy $\sigma(x)$ at each surplus x and ζ, such that $\rho \in \Gamma_\rho$, $\sigma(x) \in \Gamma_\sigma$ and $\zeta \in \Gamma_\zeta$, $\forall a \in \mathcal{A}$ and $\forall x \in \mathbb{X}$.*

We will be specializing to the spaces $\Gamma_\rho, \Gamma_\sigma, \Gamma_\zeta$ and define the topologies being used in the following proofs first.

1. For the assumed action distribution $\rho \in \Gamma_\rho$ on the finite set $\mathcal{A}$, all norms are

equivalent, we will consider the topology of uniform convergence, i.e., using the l_∞ norm given by $||\rho|| = \max_{a\in\mathcal{A}} \rho(a)$.

2. For the randomized policy $\sigma \in \Gamma_\sigma^{|\mathbb{X}|}$, we enumerate the elements in $\mathbb{X}$ as $1, 2, \cdots$, and consider the topology with norm $||\sigma|| = \sum_{j=1}^{\infty} 2^{-j}|\sigma(x_j)|$, where $|\sigma(x)| = \max_{a\in\mathcal{A}} \sigma(x, a)$. We consider any sequence $\{\sigma_n\}_{n=1}^{\infty}$ converges to σ in this topology space.

3. For the surplus distribution ζ on the countable set $\mathbb{X}$, we consider the topology of pointwise convergence.

Note that from the definition of Γ_ρ, Γ_σ and Γ_ζ, they are already non-empty, convex and compact. Furthermore, they are jointly convex. Then in order to show that the mapping Π^* satisfies the conditions of Kakutani fixed point theorem, we only need to verify the following three lemmas.

Lemma 10 *Given ρ, by taking the best response given by (3.8), we can obtain the action distribution $\sigma(x)$ for every x, which is upper semicontinuous in ρ.*

Lemma 11 *Given ρ and $\sigma(x)$, there exists a unique stationary surplus distribution $\zeta(x)$, which is continuous in ρ and $\sigma(x)$.*

Lemma 12 *Given $\zeta(x)$ and $\sigma(x)$, there exists a stationary action distribution ρ, which is continuous in $\zeta(x)$ and $\sigma(x)$.*

3.6 Characteristics of the Best Response Policy

In this section, we characterize the best response policy under the assumption that V_ρ in (3.8) has some properties. Then we discuss the relations between the utility function $u(x)$ and the optimal value function V_ρ.

3.6.1 Existence of Threshold Policy

We assume that given the action distribution ρ, $V_\rho(x)$ is increasing and submodular in x when $x \leq -l$; increasing and linear in x when $-l \leq x \leq w$; and increasing and supermodular in x, when $x \geq w$.

In Section 3.3, our lotteries will be constructed such that the probability of winning monotonically increases with the cost of the action. This when combined with the monotonicity, submodularity (decreasing differences) for positive argument and supermodularlity (increasing differences) for negative argument of V_ρ yields the following characterization of the best response policy.

Lemma 13 *For any two action, say actions a_1 and a_2, suppose that $\theta_{a_1} > \theta_{a_2}$, so that $p_{\rho,a_1} > p_{\rho,a_2}$, i.e., $\phi(p_{\rho,a_1}) > \phi(p_{\rho,a_2})$, then there is a threshold value of the surplus queue for user such that preference order for the actions changes from one side of the threshold to the other.*

Similarly, if we simply assume that $V_\rho(x)$ is increasing and submodular in $x \in (-\infty, \infty)$, or increasing and supermodular in $x \in (-\infty, \infty)$, it will also yield the existence of threshold policy.

3.6.2 Relations Between Utility Function $u(x)$ and the Optimal Value Function V_ρ

3.6.2.1 S-shaped Prospect Utility Function

Consider the S-shaped prospect utility function $u(x)$, which satisfied the following conditions: (i) $u(x)$ is a concave increasing function of x when $x \geq w$; (ii) $u(x)$ is a linear increasing function of x when $-l \leq x \leq w$; (iii) $u(x)$ is a convex increasing function of x when $x \leq -l$; and (iv) $u(x)$ is continuous on $(-\infty, \infty)$.

From (3.9), it's clear that the Bellman operator will take concave functions into concave functions. Since the limit of any sequences of functions is the value function,

all we need to prove is that the limit of a sequence of concave functions must be concave because we start from our iterations with a concave function, but note that our concave functions are defined by a weak inequality. Thus the set defining it must be closed, so the value function is concave. However, in our case, that set is not a closed set given $x \geq -l$. It's only closed for $x \geq 0$.

Therefore, we cannot prove theoretically that $V_\rho(\cdot)$ is increasing, convex in $(-\infty, -l]$, linear in $[-l, w]$ and concave in $[w, \infty)$, but intuitively $V_\rho(\cdot)$ should also be convex, linear and then concave as x increases from $-\infty$ to ∞. Indeed, we observed this property from the numerical studies in Section 3.7.

In other words, we conjecture that the optimal value function is supermodular, linear and then submodular as x increases from $-\infty$ to ∞, which will yield an optimal threshold policy.

3.6.2.2 Concave/Convex Utility Function

Now if we simply assume that the utility function $u(x)$ is concave/convex and monotone increasing in x, then we can obtain the following useful properties of the optimal value functions, which we can use to characterize the optimal threshold policy.

Next, we show that the value function $V_\rho(x)$ is increasing, submodular (i.e., decreasing differences) in x if $u(x)$ is concave in x, and increasing, supermodular (i.e., increasing differences) in x if $u(x)$ is convex in x.

Lemma 14 *Given the distribution of action ρ, $V_\rho(x)$ is an increasing and submodular function of x if $u(x)$ is a concave function of x, supermodular function of x if $u(x)$ is a convex function of x.*

3.7 Numerical Study

We first conduct an empirical data-based simulation in the context of electricity usage for home air conditioning to illustrate the performance our system. Besides the data on electricity prices available from [1], we also used a data set from [3] containing the ambient temperature over June–August, 2013, as well as customer electricity usage with a 15 minute resolution for 40 homes in Austin, TX. The data-set differentiates between air conditioning and other energy consumption, and hence is a good resource to validate usage models. Our first step is to model the usage of electricity for air conditioning by an average home in Texas over the course of a day. While we present the case of homogeneous homes that all have identical parameters and an identical baseline, it is straightforward to extend our results to the case where there are a finite set of classes of homes, and the participating homes are drawn randomly from these classes.

3.7.1 Home Model

A standard continuous time model [17, 37] for describing the evolution of the internal temperature $\tau(t)$ at time t of an air conditioned home is

$$\dot{\tau}(t) = \begin{cases} -\dfrac{1}{RC}(\tau(t) - \tau_a) - \dfrac{\eta}{C}P_m, & \text{if } q(t) = 1, \\ -\dfrac{1}{RC}(\tau(t) - \tau_a), & \text{if } q(t) = 0. \end{cases} \tag{3.13}$$

Here, τ_a is the ambient temperature (of the external environment), R is the thermal resistance of the home, C is the thermal capacitance of the home, η is the efficiency, and P_m is the rated electrical power of the AC unit. The state of the AC is described by the binary signal $q(t)$, where $q(t) = 1$ means AC is in the ON state at time t and in the OFF state if $q(t) = 0$. The state is determined by the crossings of user

specified temperature thresholds as follows:

$$\lim_{\epsilon \to 0} q(t+\epsilon) = \begin{cases} q(t), & |\tau(t) - \tau_r| < \Delta, \\ 1, & \tau(t) > \tau_r + \Delta, \\ 0, & \tau(t) < \tau_r - \Delta, \end{cases} \tag{3.14}$$

where τ_r is the temperature setpoint and Δ is the temperature deadband.

Table 3.1: Parameters for a residential AC unit

Parameter	Value
C, Thermal Capacitance	10 kWh/°C
R, Thermal Resistance	2 °C/kW
P_m, Rated Electrical Power	6.8 kW
η, Coefficient of Performance	2.5
τ_r, Temperature Setpoint	22.5 °C
Δ, Temperature Deadband	0.3 °C

A number of studies investigate the thermal properties of typical homes. We use the parameters shown in Table 3.1 for our simulations. These are based on the derivations presented in [17] for conditioning a 250 m^2 home (about 2700 square feet), which is a common mid-size home in many Texas neighborhoods.

In order to determine the energy usage for AC in our typical home, we need to have an estimate of how the ambient temperature varies over a day in Texas during the summer months. These values are available in the Pecan Street data set, and we plot the values of 3 days which are arbitrarily chosen over three months for Austin, TX in Figure 3.3.

Next, we calculate the ON-OFF pattern of our typical air conditioner based on the ambient temperature variation over the course of the day. We do this by simulating

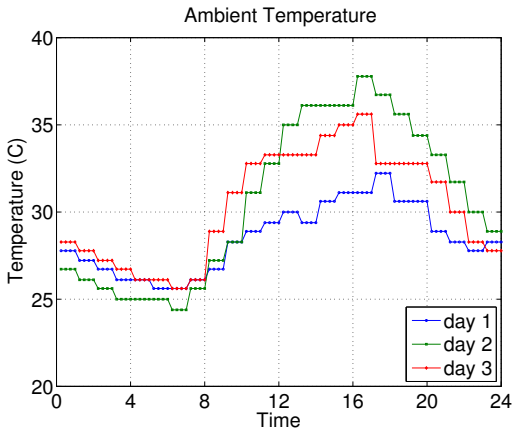

Figure 3.3: Ambient temperature of 3 arbitrary days from June–August, 2013 in Austin, TX. Measurements are taken every 15 minutes from 12 AM to 12 PM.

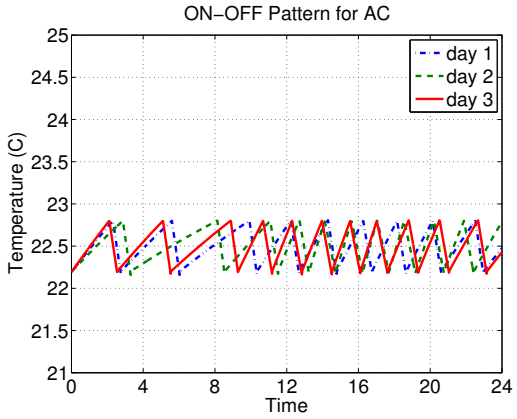

Figure 3.4: Simulated ON/OFF state of AC over a 24 hour period in a home and the corresponding interior temperature. The interior temperature falls when the AC comes on, and rises when it is off.

the controller in (3.14) with the appropriate ambient temperature values taken from Figure 3.3. The pattern is presented in Figure 3.4. We see that there is higher energy usage during the hotter times of the day, as is to be expected. This also corresponds to the peak in wholesale electricity prices shown for the same period in Figure 3.1. The total energy used each day corresponding to our 5 ton AC (= 6.8 kW; see Table 3.1) is 32.83 kWh.

For comparison, we use the Pecan Street data set to provide the measured daily average energy usage for AC during the same period for 4 homes that have parameters in the same ballpark as our typical home. These values are shown in Table 3.2. The table shows the close match of our home model with real AC usage patterns.

3.7.2 Actions and Costs

The customer action space in our problem consists of choosing when to turn ON and OFF the AC, and is uncountably infinitely large. We need to pick a reasonable

Table 3.2: Daily AC usage for four homes

ID	Square feet	Age	AC (tons)	Energy (kWh)
93	2934	20	5	28.2
545	2345	6	3.5	41.5
4767	2710	5	4	31.7
3967	2521	5	3.5	37.8

discrete subset of the action space for our study. From Figure 3.1 it is clear that the consumption during maximum price period $5-6$ PM has the maximum impact on the overall energy cost of the LSE. The LSE would like to incentivize the shift of some of this usage, without excessively affecting the internal home temperature. We assume that the actions available to the customer involve setting different setpoints during each period from $2-8$ PM. We take the setpoint 22.5°C as the baseline. Each action can now be identified with a vector $(y_1, y_2, y_3, y_4, y_5, y_6)$, where y_j indicates the setpoint in the period j. Hence, the vector $(22.5, 22.5, 22.5, 22.5, 22.5, 22.5)$ would indicate the baseline in which the customer does not change the original setpoint in each period. This defines the action set $\mathcal{A}$, and we define the action with index $a = 0$ to be the no-change action.

Our next step is to calculate the cost of taking each action $a \in \mathcal{A}$, which corresponds to the discomfort of having a potentially higher mean and standard deviation in the home temperature, and higher energy consumption due to that action. We measure the state of the home under action $a \in \mathcal{A}$ by the tuple consisting of the mean temperature, the standard deviation and energy usage, denoted $[\bar{\tau}_a, \sigma_a, \mathrm{E}_a]$. The baseline state of these parameters is under action 0, denoted by $[\bar{\tau}_0, \sigma_0, \mathrm{E}_0]$ Then we define the cost of taking any action a as

$$\theta_a = |\bar{\tau}_0 - \bar{\tau}_a| + \lambda|\sigma_0 - \sigma_a| - \varsigma(\mathrm{E}_0 - \mathrm{E}_a), \tag{3.15}$$

where we choose $\lambda = 10$ to make the numerical values of the mean and standard deviation comparable to each other and $\varsigma = 10$ cents/kWh as the fixed energy price. Note that the calculation of cost for each action involves simulating the home under that action to determine $[\bar{\tau}_a, \sigma_a, \mathrm{E}_a]$. However, this has to be done only once to create a look-up table, which can be used thereafter.

3.7.3 Coupons, Lottery and Surplus

We now consider the incentives provided to the customers by the LSE, which wishes to generate an MFE that has most of its mass on actions that are beneficial to it. We measure the day-ahead price of electricity experienced by the LSE in dollars/MWh and denote the price at time period j in day i as $\pi_{i,j}$, where $i = \{1, 2, \cdots, 92\}$ and $j = \{1, \cdots, 6\}$. Each action vector of a customer would impose a net price on the LSE in proportion to the usage. We define the *differential price* measured in dollars imposed by an action $y = (y_1, y_2, y_3, y_4, y_5, y_6)$ versus $z = (z_1, z_2, z_3, z_4, z_5, z_6)$ as

$$H(y, z) = \sum_j \left(k(y_{i,j}) - k(z_{i,j})\right) \pi_{i,j}, \tag{3.16}$$

where k converts the setpoints into electricity usage in each period, which is measured in MWh. Setting y as the baseline action $(22.5, 22.5, 22.5, 22.5, 22.5, 22.5)$ presents a way of measuring the reduction/increase in hazard due to the incentive scheme. We will use this metric to quantify the value of the MFE achieved.

Now, the baseline action $a = 0$ corresponds to a setpoint of 22.5°C in period 3 at which $\pi_{i,3}$ is highest (Figure 3.1) for any day i. Hence, the LSE should incentivize actions that are likely to reduce the risks of peak day-ahead price borne by the LSE by offering Energy Coupons in proportion to the usage during the corresponding

periods. The problem of optimally selecting these coupons is a hard one in general. However, in the limited context of our simulation, it is intuitively clear that coupons must be placed at periods of lower price. Our coupon choices are shown in Table 3.3, where x_1 and x_6 are energy usage in the corresponding periods (measured in kWh) and the day-ahead price shown is that of one day randomly drawn from the three months. Note that the number of coupons are not necessarily integer, although making them integer quantities will not have any impact on our results.

Table 3.3: Day-ahead price and energy coupons

Index	Period	Day-ahead Price/MWh	Coupons/kWh
1	$2-3$ PM	\$47	107 if $x_1 > 2.464$; 1.8 otherwise
2	$3-4$ PM	\$55	5.4
3	$4-5$ PM	\$78	1.8
4	$5-6$ PM	\$99.6	0
5	$6-7$ PM	\$66.5	3.6
6	$7-8$ PM	\$49.5	54 if $x_6 > 2.24$; 1.8 otherwise

Given the coupon placement by the LSE, the customers need to determine the costs and number of coupons resulting from each action, and use these values to estimate the utility that they would attain. We identified 6 actions that appeared to have the most promise of being used, and these shown in Table 3.4 with their attendant costs and number of coupons received. Note that action 0 is the one in which the customer does not participate in the system.

The LSE conducts an auction each week across clusters of $M = 50$ homes in each auction. For each cluster, there is $K = 1$ prize for winning the lottery with a value of \$40 (we will show in the next subsection that this choice is viable). We assume that the customers choose the same action on each day of the week, and then participate

Table 3.4: Actions, costs and energy coupons

Index	Action Vector	Cost	Coupons
0	$(22.5, 22.5, 22.5, 22.5, 22.5)$	0	37.4
1	$(21.5, 21.5, 22.25, 23.5, 23.75, 21.25)$	3.68	715
2	$(21.5, 21.5, 22.25, 24, 23.5, 21.75)$	3.51	713
3	$(21.5, 21.5, 22.25, 24, 23.5, 22)$	3.50	704
4	$(21.5, 21.5, 22.25, 23.5, 23.25, 22.25)$	3.146	693
5	$(21.5, 21.5, 22.25, 24, 23, 22.5)$	2.68	577

in the lottery.

The final few parameters of our simulated system need to be determined by experiment, but in the absence of data until we conduct user trials, we make the following reasonable assumptions. We assume that the customer expects to win at least \$1 on average by participating. Hence, the value of decrease in surplus due to losing is $l = 1$, while the value of increase in surplus by winning is $w = 40 - 1 = 39$. The customer expects nothing if he does not participate (action 0), and there is no change of surplus in this case. We assume that customers are likely to remain in the system with probability 0.98, *i.e.,* the average customer participates for 50 time steps, which roughly parallels the fact that many home users sign a new contract once a year. Further, a newly entering customer has a surplus that is an integer uniformly drawn from $[10, 30]$. Finally, for the customer utility, which maps surplus (in dollars) to utility units, we use the value function of prospect model, $u(x) = x^\gamma$ if $x \geq 0$ or $u(x) = -\varphi(-x)^\gamma$ if $x < 0$, where $\varphi = 2.25$ and $\gamma = 0.88$ according to the empirical studies conducted in [51, 88].

Under this model, we expect a user who has lost a number of lotteries to stop participating in the system, since his surplus becomes negative and he is not receiving enough of an incentive to stay, given the cost he bears each day. Similarly, a user

who has won too many times would have a large surplus, and would also not be keen on participating since the marginal utility he gets may not be high enough for him. While we expect the latter event to occur very infrequently, the former is something that we have to watch out for, since it would result in a poor customer response to our system and potentially less savings to the LSE. In what follows, we will see that our selected value of \$40 reward appears to be sufficient to ensure a good level of participation.

3.7.4 Mean Field Equilibrium

We now are ready to determine the properties of the MFE generated by our system. We start with a uniform action distribution as the initial condition. We run the system over 50 iterations, determining the steady state action distribution at each step and using that as the input for the next iteration. We find that convergence occurs quite quickly and reaches within 0.1% of the final value within 10 iterations.

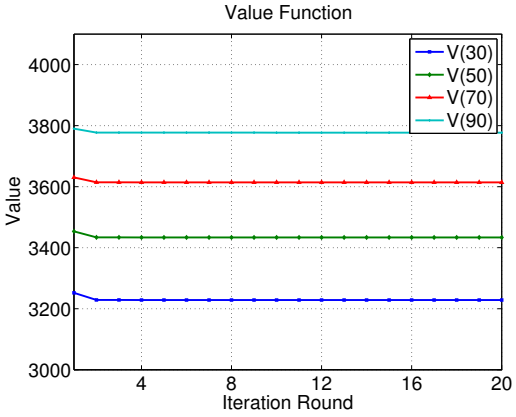

Figure 3.5: Value function

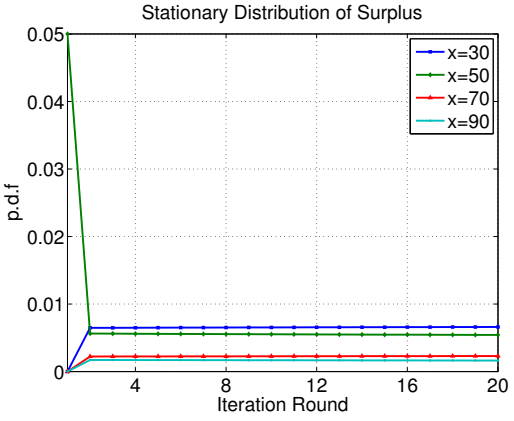

Figure 3.6: Convergence of surplus distribution

Figure 3.5 illustrates the convergence of the values associated with a few candidate

states (surplus). Each point on the graph is obtained by value iteration over the Bellman equation describing value, keeping the action distribution of other players fixed. The value iteration converges within about 50 steps in each case. Figure 3.6 shows the convergence of the stationary probability of having certain surplus values for a few examples. The eventual values to which they converge is the mean field surplus distribution. The complete mean field distribution of surplus is shown in Figure 3.7. It indicates that customers win at a lottery between 1 and 2 times over an average lifetime of 50 time intervals, as is to be expected with a cluster size of 50 customers at each lottery.

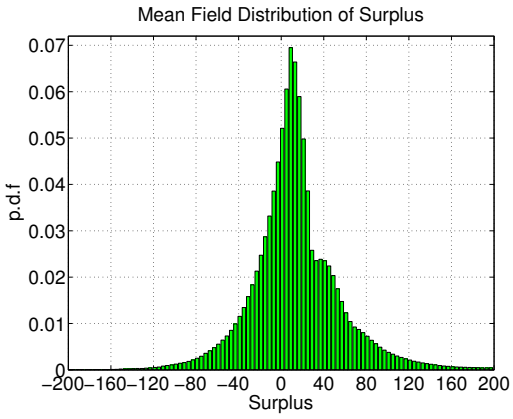

Figure 3.7: Mean field distribution of surplus

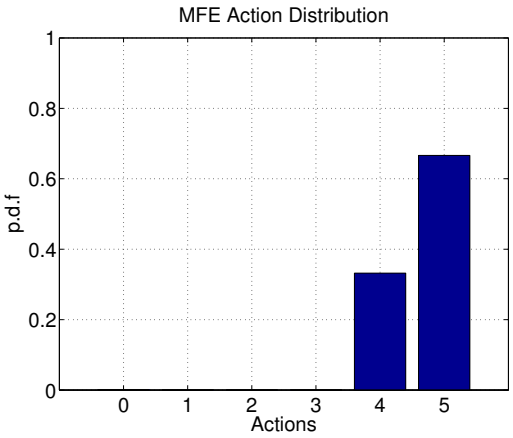

Figure 3.8: Action distribution

Finally, Figure 3.8 illustrates the mean field action distribution. For example, the best action from the LSE's perspective is action 5, which is chosen with probability 0.68. Figure 3.9 shows the interior temperature under actions 0, 4, 5 and mean field action in a home in an arbitrary day. Figure 3.10 shows the comparison of energy consumption between action 0 and the mean field action. We use the mean field

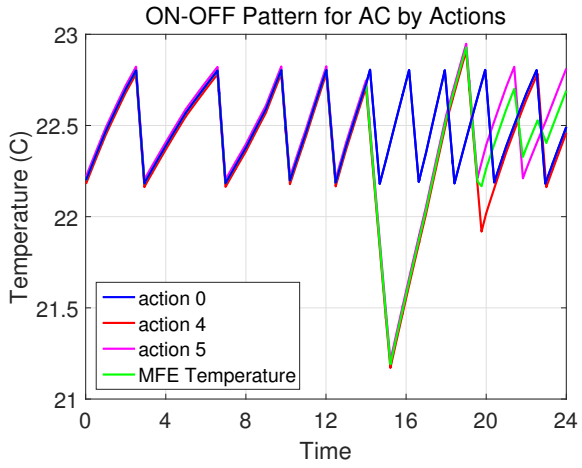

Figure 3.9: Simulated ON/OFF state of AC over a 24 hour period in a home under actions 0, 3 and the mean field action on an arbitrary day and the corresponding interior temperature. The temperature graph is slightly offset for actions $4, 5$ and the mean field action for ease of visualization.

action distribution to find that the net reduction in price over 50 homes is \$78 each week. Thus, incentivizing customers by offering a prize of \$40 each week is certainly feasible. The MFE illustrates that even as small as 1°C change of the setpoint of AC each day over several homes can yield significant benefits.

3.7.5 Reward, Saving and Profit

We assumed in the above simulations that the customer expects to win at least \$1 on average by participating, and hence the decrease in surplus due to losing at the lottery is $l = 1$, while the increase in surplus due to winning is $w = 40 - 1 = 39$ (since the reward for winning the lottery is \$40). We saw that the total net reduction (savings to the LSE) over 50 homes is \$78 each week, and hence \$40 reward is sustainable.

We now numerically determine the relation between the reward to customers, savings to the LSE and profit to the LSE, shown in Figure 3.11 for $l = \$1$ and \$5.

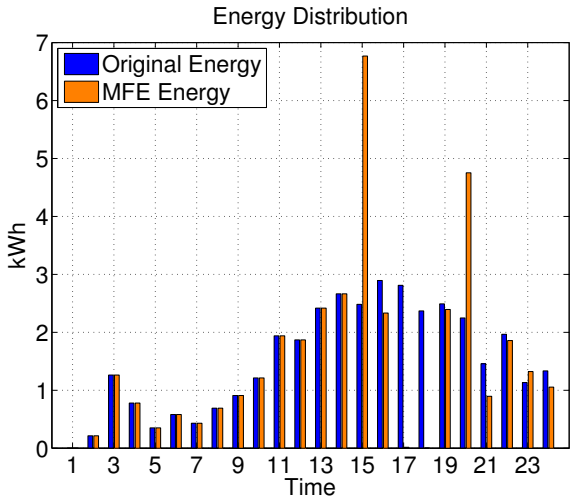

Figure 3.10: Energy distribution

We plot the total savings to the LSE as well as its profit (savings minus reward) as a function of the reward offered for winning the lottery. From the left figure ($l = 1$), most of the savings can be achieved by giving a reward of \$40. Also at this point almost all the customers will participate in the system, i.e., the probability of choosing action 0 is 0. The maximum profit at $l = 1$ is achieved when the reward is \$20. The break-even point is about \$80 reward.

In the right figure ($l = 5$), the profit does not change much, but the savings increases as we increase the offered reward. This is because with a large penalty (decrease in surplus due to losing) more customers will participate in the system by choosing actions 4 and 5 only if a large reward is offered. The break-even point is about \$70 reward. We also consider other cases like $l = 3$, which exhibit the same trends and hence are omitted here. In all cases, the total rewards are bounded by \$80, and the mean field action distribution is similar to that in Figure 3.8. As we increase the decrease in surplus due to losing l, the number of customers choosing action 0 will increase if the reward is small, i.e., customers become less risk-seeking.

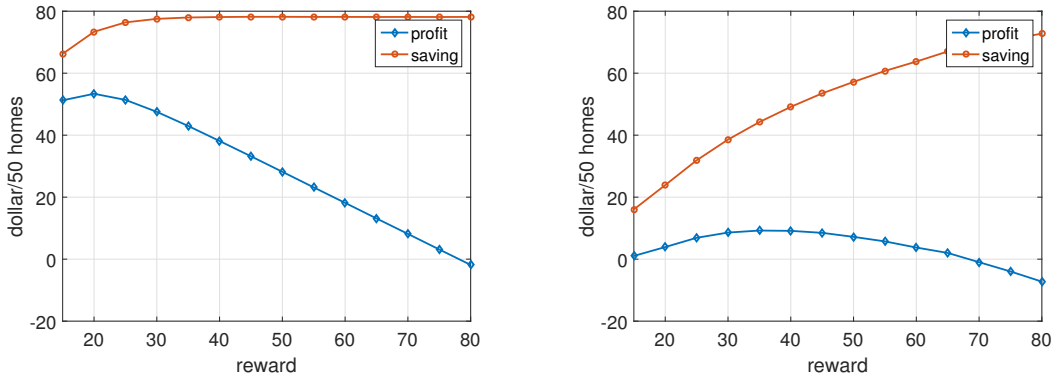

Figure 3.11: The relation between offered reward, LSE savings and LSE profit. Left: $l = 1$. Right: $l = 5$.

We also studied the impact of changing the mapping of actions to coupons that we showed in Tables 3.3 and 3.4, and conducted the above experiment with $l = 1, 3, 5$ again. We found that our results are robust to the mapping of actions to coupons. For example, when we set $l = 1$, most savings can be achieved by giving a \$40 reward and the break-even point is about \$80, as was seen above. The total rewards are bounded by \$80 in all cases. We present details of other mappings and the corresponding results in the Appendix B.4.

3.8 Conclusion

In this section we developed a general framework for analyzing incentive schemes to promote desirable behavior in societal networks by posing the problem in the form of a Mean Field Game (MFG). Our incentive scheme took the form of awarding coupons in such that higher cost actions would correspond to more coupons, and conducting a lottery periodically using these coupons as lottery tickets. Using this framework, we developed results in the characteristics of the optimal policy and showed the existence of the MFE.

We used the candidate setting of an LSE trying to promote demand-response in

the form of setting high setpoints in higher price time of the day in order to transfer energy usage from a higher to a lower price time of day for an air conditioning application. We conducted data driven simulations that accurately account for electricity prices, ambient temperature and home air conditioning usage. We showed how the prospect of winning at a lottery could potentially motivate customers to change their AC usage patterns sufficiently that the LSE can more than recoup the reward cost through a likely reduced expenditure in electricity purchase.

Our setup is general enough to capture population behavior in other societal networks. For example, it applies in an essentially unchanged manner to an experiment conducted on a bus transportation system of an IT firm in India, described in [72]. Here, employees have a choice of an early morning bus that experiences low traffic congestion or a later one that experiences more. Providing incentives to employees in the form of lottery tickets for taking the earlier bus was shown to increase its attractiveness, while simultaneously reducing costs to the firm by running a smaller number of buses at higher fuel efficiency.

In the future we intend to conduct user trails of the Energy Coupon system. This is something that we are actively working on, and such trials would both validate the idea of using incentive schemes to promote cooperation, as well as suport our analytical prediction of being able to run a viable societal system with desired behavior using the MFE framework.

4. DYNAMIC ADAPTABILITY PROPERTIES OF CACHING ALGORITHMS

4.1 Introduction

The dominant application in today's Internet is streaming of content such as video and music. This content is typically streamed by utilizing the services of a content distribution network (CDN) provider such as Akamai or Amazon [48]. Streaming applications often have stringent conditions on the acceptable latency between the content source and the end-user, and CDNs use caching as a mean of reducing access latency and bandwidth requirements at a central content repository. The fundamental idea behind caching is to improve performance by making information available at a location close to the end-user. Managing a CDN requires policies to route requests from end-users to near-by distributed caches, as well as algorithms to ensure the availability of the requested content in the cache that is polled.

While the request routing policies are optimized over several economic and technical considerations, they end up creating a request arrival process at each cache. Caching algorithms attempt to ensure content availability by trying to learn the distribution of content requests in some manner. Typically, the requested content is searched for in the cache, and if not available, a miss is declared, the content is retrieved from the central repository (potentially at a high cost in terms of latency and transit requirements), stored in the cache, and served to the requester. Since the cache is of finite size, some content may need to be evicted in order to cache the new content, and caching algorithms are typically described by the eviction method employed.

Some well known content eviction policies are Least Recently Used (LRU) [34], k-LRU [70], First In First Out (FIFO), RANDOM [25], CLIMB [25,85] and Adaptive

Replacement Cache (ARC) [71]; these will be described in detail later on. Performance analysis typically consists of determining the hit probability at the cache under either a synthetic arrival process (usually with independent draws of content requests following a fixed Zipf popularity distribution, referred to as the Independent Reference Model (IRM)), or using a data trace of requests observed in a real system. It has been noted that performance of an eviction algorithm under synthetic versus real data traces can vary quite widely [70]. For instance, 2-LRU usually does better than LRU when faced with synthetic traffic, but LRU often outperforms it with a real data trace. The reason for this discrepancy is usually attributed to the fact that while the popularity distribution in a synthetic trace is fixed, real content popularity changes with time [18, 95]. Thus, it is not sufficient for a caching algorithm to learn a fixed popularity distribution accurately, it must also learn it quickly in order to track the changes on popularity that might happen frequently.

Since each known caching algorithm generates a Markov process over the occupancy states of the cache, the typical performance analysis approach is to determine the stationary distribution of this process, and to use it to calculate the hit probability. However, this approach loses all notion of time and does not allow us to compare the performance of each algorithm with the best possible. A major goal of this section is to define function that accounts for both the error due to time lag of learning, as well as the error due to the inaccuracy of learning. Such an error function would allow us to better understand the performance of existing algorithms, as well as decide how to develop new ones.

Our first requirement to attain this goal is a refinement of the hit probability metric to characterize the nearness of the stationary distribution of an algorithm to the best-possible cache occupancy. If the statistics of the cache request process are known, the obvious approach to maximizing the hit probability is to simply cache the

most popular items as constrained by the cache size, creating a fixed vector of cached content. How do we compare the stationary distribution generated by a caching algorithm with this vector? A well known approach to comparing distributions is to determine the Wasserstein distance between them [89]. However, since we are dealing with distributions of permutations of vectors, we need to utilize a notion of a cost that depends on the ordering of elements. Such a notion is provided by a metric called the generalized Kendall's tau [57]. Coupling these two notions together, we define a new metric that we call the τ-distance, which correctly represents the accuracy of learning the request distribution. The τ-distance can also be mapped back to hit probability or any other performance measure that depends on learning accuracy.

Our second requirement is to study the evolution of the Markov chain associated with caching algorithm to understand its rate of convergence to stationarity. The relevant concept here is that of *mixing time*, which is the time required for a Markov process to reach within ϵ distance (in Total Variation (TV) norm) of the eventual stationary distribution. To our knowledge, no existing work has characterized the mixing time of caching algorithms. However, this metric is crucial to understanding caching algorithm performance, as it effectively characterizes the speed of learning.

Once we have both the τ-distance and the mixing time characterized for a caching algorithm, we can determine how well algorithm would perform after it has learned for a certain time interval. Using a triangle inequality bound and combining the τ-distance and mixing time with appropriate normalization, we can come up with a metric that provides an upper bound on this tradeoff at any given time, and we call this metric as the *learning error*.

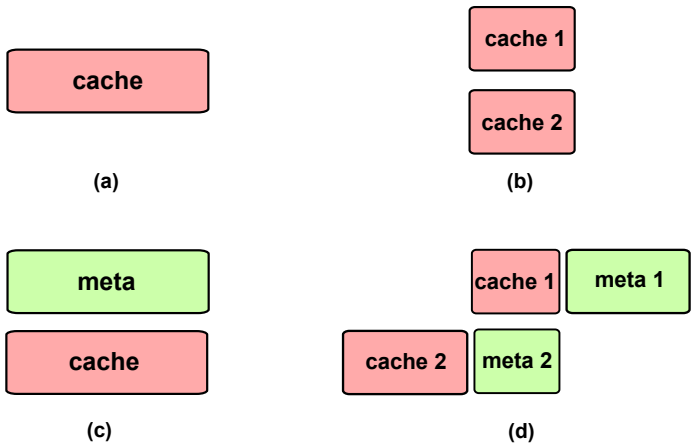

Figure 4.1: Different dimensions of caching paradigms.

4.1.1 Structure of Caching Paradigms

A cache is fundamentally a block of memory that can be used to store data items that are frequently requested. Over the years, different paradigms have evolved on how best to utilize the available memory. Most conventional caching algorithms, such as LRU, RANDOM and FIFO, have been designed and analyzed on an simple (isolated) cache, as shown in Figure 4.1 (a). The model here is that any time a request arrives, the corresponding data item is fetched and cached if it is not already in the cache. Where to place this item, and which item to evict if needed determines the nature of the caching algorithm. New caching algorithms have been proposed in the past few years, which have been shown to have better performance than classical model, often through numerical studies. The different dimensions that have been explored are two fold. One the one hand, the memory block can be divided into two or more levels, with a hierarchical algorithm attempting to ensure that more popular content items get cached in the higher levels. For example, a simple 2-level cache is shown in Figure 4.1 (b), and it has been empirically observed that under an appropriate caching algorithm, it could display a higher hit probably than that of a

simple cache of the same size. On the other hand, a meta-cache that simply stores content identities an approach that can be used to better learn popularity without wasting memory to cache the actual data item. The idea is illustrated in Figure 4.1 (c) with one level of meta caching. Meta-caches are an efficient way of ensuring that only popular items are ever cached, and empirical observations suggest that when coupled with an appropriate caching algorithm, they too are quite effective at increasing the hit rate. However, in both cases, it is not clear how the multi-level caches and meta-caches enhance the hit probability, and what impact they have on the convergence to stationarity of the caching scheme. In this section, we first characterize the performance of an isolated cache through τ-distance and mixing time to study the adaptability of these algorithms. In the next section, we use this technique to study how the number of cache levels and cache partitions impact the performance. Finally, we use the insights gained in this process to design a new caching paradigm algorithm that combines ideas from using multi-level cache and meta-caches, as shown in Figure 4.1 (d). We design an algorithm to be applied to this cache structure, and name the resulting algorithm as Adaptive-LRU (A-LRU).

4.1.2 Related Work

Caching algorithms have mostly been analytically studied under the IRM Model. Explicit results for stationary distribution and hit probability for LRU, FIFO, RANDOM, CLIMB [9, 25, 35, 55, 85] have been derived under IRM, however, these results are only useful for small caches due to the computational complexity of solving for the stationary distribution. Several approximations have been proposed to analyze cache of a reasonably large size [27,82], and a notable one is the Time-To-Live (TTL) approximation, which was first introduced for LRU under IRM [20]. It has been further generalized to other situations [12, 30, 34, 70, 82]. Theoretical support on the

accuracy of TTL approximation was presented in [12]. A rich literature also studies the performance of caching algorithms in terms of hit probability based on real trace simulations, e.g., [63, 70, 71, 95], and we do not attempt to provide an overview here.

4.1.3 Organization

The next section contains some technical preliminaries and representative caching algorithms. We derive the steady state distributions of the algorithms in Section 4.3 and identify hit probabilities in Section 4.4. We consider our new notion of τ-distance in Section 4.5 and mixing time in Section 4.6. We join the two notions and investigate the learning error in Section 4.7. We conclude in Section 4.8.

4.2 Technical Preliminaries

4.2.1 Traffic Model

To compare various caching algorithms, it is necessary to define a model of how we specify the reference items first. For most of our analysis, we consider the simplest and most widely used stochastic model which is called the *Independent Reference Model* (IRM) [25]. In our numerical investigations, we will also consider three more realistic request processes: a Markov-modulated request process, a YouTube request trace [95], and one request trace from the IRCache project [13]. In IRM, the request process $\{r_1, r_2, \cdots\}$ is given by a sequence of independent, identically distributed random variables with a fixed probability distribution

$$\mathbb{P}(r_t = i) = p_i, \quad i \in \{1, \cdots, n\}, \quad t \in \{1, 2, \cdots\}, \tag{4.1}$$

where r_t is the item referenced by the t-th request, and there are n different items. Without loss of generality (w.l.o.g.), we assume that the reference items are numbered so that the probabilities are in a non-increasing order, i.e., $p_1 \geq p_2 \geq \cdots \geq p_n$.

4.2.2 Popularity Law

Whereas our analytical results are not for any specific popularity law, for our numerical investigations we will use a Zipf-like distribution as this family has been frequently observed in real traffic measurements, and is widely used in performance evaluation studies in the literature [19]. For a Zipf-like distribution, the probability to request the i-th most popular item is $p_i = A/i^\alpha$, where α is the Zipf parameter that depends on the application considered [31], and A is the normalization constant so that $\sum_{i=1}^{n} p_i = 1$ if there are totally n unique items to be considered in the system.

4.2.3 Caching Algorithms

There exist a large number of caching algorithms, with the difference being in their choice of insertion or eviction rules. In this section, we consider the following representative algorithms.

LRU: [34] When there is a request for item i, there are two cases: (1) i is not in the cache (cache miss), then i is inserted in the first position in the cache, all other items move back one position, and the item that was in the last position of the cache is evicted; (2) i is in position j of the cache (cache hit), then i moves to the first position of the cache, and all other items that were in positions 1 to $j-1$ move back one position.

FIFO: The difference between FIFO and LRU is when a cache hit occurs on an item that was in position j. In FIFO, this item does not change its position.

RANDOM: The difference between RANDOM and FIFO is when a cache miss occurs, the item is inserted in a random position, and the item that was in this randomly selected position is evicted.

CLIMB: [25,85] The difference between CLIMB and LRU is when a cache hit occurs on an item that was in position j. In CLIMB, this item is inserted in position $j+1$,

and the item that was in position $j+1$ moves to position j.

Remark 1 *LRU has been widely used due to its good performance and ease of implementation. FIFO and RANDOM have been used to replace LRU in some scenarios since they are easier to implement with a reasonable good performance. CLIMB has been numerically shown to have a higher hit ratio than LRU, at the expense of longer time to reach this steady state than LRU.*

4.3 Steady State Distribution

We first consider the question of determining the stationary distribution of the contents of a cache based on the caching algorithm used. Each (known) caching algorithm A under any Markov modulated request arrival process (including IRM) results in a Markov process over the occupancy states of the cache. Suppose there are a total of n content items in a library, and the cache size is $m < n$. Then each state $\boldsymbol{x}$ is a vector of size m indicating the content in each cache spot; we call the state space of all such vectors $\mathcal{S}$. Then one can potentially determine the stationary distribution of this algorithm, denoted π^*_A. This procedure is well established in the literature [25], but results for the algorithms of interest are not available in the desired form (viewed in terms of permutations), and we first derive these as as a foundation for novel performance metrics in the following sections.

For simplicity, we denote x_j as the identity of the item at position j in the cache, i.e., $\boldsymbol{x} = (x_1, \cdots, x_m)$. Next we present the steady state probabilities of this Markov chain for FIFO, RANDOM, CLIMB, and LRU.

Theorem 9 *Under the IRM, the steady state probabilities $\pi^*_{FIFO}(\boldsymbol{x})$, $\pi^*_{RANDOM}(\boldsymbol{x})$, $\pi^*_{CLIMB}(\boldsymbol{x})$, and $\pi^*_{LRU}(\boldsymbol{x})$, with $\boldsymbol{x} \in \mathcal{S}$ are as follows:*

$$\pi^*_{FIFO}(\boldsymbol{x}) = \frac{\Pi_{i=1}^m p_{x_i}}{\sum_{\boldsymbol{x}\in\mathcal{S}} \Pi_{i=1}^m p_{x_i}},$$

$$\pi^*_{RANDOM}(\boldsymbol{x}) = \frac{\Pi_{i=1}^m p_{x_i}}{\sum_{\boldsymbol{x}\in\mathcal{S}'} \Pi_{i=1}^m p_{x_i}},$$
$$\pi^*_{CLIMB}(\boldsymbol{x}) = \frac{\Pi_{i=1}^m p_{x_i}^{m-i+1}}{\sum_{\boldsymbol{x}\in\mathcal{S}} \Pi_{i=1}^m p_{x_i}^{m-i+1}},$$
$$\pi^*_{LRU}(\boldsymbol{x}) = \frac{\Pi_{i=1}^m p_{x_i}}{(1-p_{x_1})(1-p_{x_1}-p_{x_2})\cdots(1-p_{x_1}-\cdots-p_{x_{m-1}})}, \tag{4.2}$$

where $\mathcal{S}'$ denotes the set of all combinations of elements of $\{1,\cdots,n\}$ taken m at a time. Note that elements of $\mathcal{S}'$ are subsets of $\{1,\cdots,n\}$, while elements of $\mathcal{S}$ are ordered subset of $\{1,\cdots,n\}$, satisfying $\sum_{\boldsymbol{x}\in\mathcal{S}} \Pi_{j=1}^m p_{x_j} = m! \sum_{\boldsymbol{x}\in\mathcal{S}'} \Pi_{j=1}^m p_{x_j}$.

The proofs for FIFO, RANDOM and CLIMB follow directly by constructing an auxiliary Markov chain on the set $\mathcal{S}$ and verifying that $\pi^*_{\text{FIFO}}(\boldsymbol{x})$, $\pi^*_{\text{RANDOM}}(\boldsymbol{x})$ and $\pi^*_{\text{CLIMB}}(\boldsymbol{x})$ satisfy detailed balance equations (and reversibility [53]). The result for LRU is obtained by using a probabilistic argument following [40], we present the details of this proof in Appendix C for completeness. These are the well-known steady state probabilities for FIFO, RANDOM, CLIMB and LRU [9, 25, 35, 55, 85].

4.4 Hit Probability

A primary performance measure in caching systems is the hit probability. One can derive the hit probability once we have the stationary distribution. We illustrate how to compute this standard performance measure in this section, and will present hit probability as a special case of our new more general metrics that we will develop in the next few sections.

Denote the hit probability of algorithm A as H_m^A and let $F_m^A = 1 - H_m^A$ be the miss probability under IRM.

By the ergodic theorem, the miss probability F_m^A is equal to the stationary probability of a miss,

$$F_m^A = \sum_{\boldsymbol{x}} \pi^*_A(\boldsymbol{x}) \sum_{\boldsymbol{x}'} q(\boldsymbol{x}, \boldsymbol{x}'), \tag{4.3}$$

where summation in the outer sum is with respect to all the states in $\mathcal{S}$, and the inner summation denotes a summation only over those states $\boldsymbol{x}'$ satisfying the condition $|\boldsymbol{x}' \setminus \boldsymbol{x}| = 1$, i.e., differ from $\boldsymbol{x}$ in precisely one element, and $q(\cdot, \cdot)$ is the transition probability.

Theorem 10 *Under IRM, we have*

$$F_m^A = \sum_{\boldsymbol{x} \in \mathcal{S}} \left(1 - \sum_{j=1}^{m} p_{x_j}\right) \pi_A^*(\boldsymbol{x}), \tag{4.4}$$

where $\pi_A^*(\boldsymbol{x})$ is given in (4.2).

4.5 Permutation Distance

The hit probability does not immediately allow us to compare the performance of an algorithm with the best possible. We seek a refinement that would allow us to determine "how close" the stationary performance of an algorithm is to the best-possible.

If we have full knowledge of the popularity distribution at any time, we could simply cache the most popular items in the available cache spots, placing the most popular element in first cache spot, and then proceeding onwards until the m-th spot. This approach would maximize the hit probability, as well as any other metric that yields better performance when more popular items are cached. We denote this ideal occupancy vector as $\mathbf{c}^*$. We first need a method of comparing the distance between this occupancy vector and any other permutation over the possible cache occupancy states.

4.5.1 Generalized Kendall's Tau Distance

Let $[n] = \{1, \cdots, n\}$ be a library of items and $[n]_m$ be the set of m items randomly chosen from $[n]$. Let S_n^m be the set of permutations on $[n]_m$. Consider a permutation

$\sigma \in S_n^m$, we interpret $\sigma(i)$ as the position of item i in σ, and we say that i is ahead of j in σ if $\sigma(i) < \sigma(j)$. W.l.o.g, we take $\sigma(i) = 0$ for $i \in [n]/[n]_m$.

The classical Kendall's tau distance [29] [1] is given by

$$K(\sigma_1, \sigma_2) = \sum_{(i,j):\sigma_1(i)>\sigma_1(j)} 1_{\{\sigma_2(i)<\sigma_2(j)\}}, \tag{4.5}$$

where $1_{\mathcal{A}}$ is the indicator function and $1_{\mathcal{A}} = 1$ if the condition $\mathcal{A}$ holds true, otherwise $1_{\mathcal{A}} = 0$.

However, this conventional definition does not take into account the item relevance and positional information, which are crucial to evaluating the distance metric in a permutation. Since we wish to compare with $\mathbf{c}^*$, in which the most popular items are placed in lower positions, the errors in lower positions in the permutation need to be penalized more heavily than those in higher positions. There are several alternative distance measures which have been proposed to address the above shortcomings of the conventional distance. In the following, we consider the generalized Kendall's tau distance proposed in [57] that captures the importance of each item as well as the positions of the errors.

Let $w_i > 0$ be the *element weight* for $i \in [n]$. For simplicity, we assume that $w_i \in \mathbb{Z}^+$; all the following results hold for non-integral weights as well. In addition to the element weight, as discussed earlier, we wish to penalize inversions early in the permutation more than inversions later in the permutations. In order to achieve this, we define the *position weights* to differentiate the importances of positions in the permutation. We first consider the cost of swapping between two adjacent positions.

[1] We consider $p = 0$ for the definition given in [29], which is an "optimistic approach" that corresponds to the intuition that we assign a nonzero penalty to the pair $\{i, j\}$ only if we have enough information to know that i and j are in the opposite order in the two permutations σ_1 and σ_2.

Let $\zeta_j \geq 0$ be the cost of swapping an item at position $j-1$ with an item at position j, and let $p_1 = 1$ and $p_j = p_{j-1} + \zeta_j$ for $1 < j \leq m$. Define $\bar{p}^i_{\sigma_1,\sigma_2} = \frac{p_{\sigma_1(i)} - p_{\sigma_2(i)}}{\sigma_1(i) - \sigma_2(i)}$ to be the average cost that item i encountered in moving from position $\sigma_1(i)$ to position $\sigma_2(i)$. In particular, $\bar{p}_i = 1$ if $\sigma_1(i) = \sigma_2(i)$. Similarly for $\bar{p}^j_{\sigma_1,\sigma_2}$. Now, we are ready to define the generalized Kendall's tau distance:

$$K_{w,\zeta}(\sigma_1, \sigma_2) = \sum_{\sigma_1(i) < \sigma_1(j)} w_i w_j \bar{p}^i_{\sigma_1,\sigma_2} \bar{p}^j_{\sigma_1,\sigma_2} 1_{\{\sigma_2(i) > \sigma_2(j)\}}. \tag{4.6}$$

4.5.2 Wasserstein Distance

While the generalized Kendall's tau is a way of comparing two permutations, the algorithms that we are interested in do not converge to a single permutation, but yield stationary distributions over permutations. Hence, we should compare the stationary distribution π^*_A of an algorithm A, with $\mathbf{c}^*$ using a distance function that accounts for the ordering of content in each state vector. A general way of comparing distributions on permutations, given a distance function between any two permutations is the Wasserstein distance.

Let $(\mathcal{S}, d)$ be a Polish space, and consider any two probability measures μ and ν on $\mathcal{S}$, then the Wasserstein distance [89] [2] between μ and ν is defined as

$$W(\mu, \nu) = \inf_{P_{X,Y}(\cdot,\cdot)} \Big\{ \mathbb{E}[d(X, Y)], \quad P_X(\cdot) = \mu, P_Y(\cdot) = \nu \Big\}, \tag{4.7}$$

which is the minimal cost between μ and ν induced by the cost function d.

[2] W.l.o.g., we are interested in the L^1-Wasserstein distance, which is also commonly called the Kantorovich-Rubinstein distance [89]. For convenience, we express Wasserstein distance by means of couplings between random variables.

4.5.3 τ-distance

We are now ready to define the specific form of Wasserstein distance between distributions on permutations that is appropriate to our problem. We define the τ-distance as the Wasserstein distance taking the generalized Kendall's distance in (4.6) as the cost function in (4.7).

Since the ideal occupancy vector $\mathbf{c}^*$ is unique and fixed, the infimum in (4.7) over all the couplings is trivial. Therefore, we have

$$|\pi_A^* - \mathbf{c}^*|_\tau = \sum_{\boldsymbol{x}} K_{w,\zeta}(\boldsymbol{x}, \mathbf{c}^*)\pi_A^*(\boldsymbol{x}), \tag{4.8}$$

where $K_{w,\zeta}(\cdot,\cdot)$ is the generalized Kendall's tau distance defined in (4.6).

4.5.4 Model Validation and Insights

Since the τ-distance characterizes how accurately an algorithm learns the popularity distribution, a smaller τ-distance should correspond to a larger hit probability. Computation of the τ-distance is complex, since it is a function of all possible permutations over the content items. But we can illustrate how different algorithms perform using a content library size of $n = 20$. Figure 4.2 compares the τ-distance and hit probabilities of various caching algorithms. The points on each curve correspond to cache size of $2, 3, 4, 5$ from left to right. From Figures 4.2, we can see that the τ-distance and hit probability follow the same rule, i.e., a smaller τ-distance corresponds to a larger hit probability, which is as expected. We also observe that the hit probabilities of the different algorithms are consistent with established results that indicate that the hit probability of CLIMB is superior to LRU, which in turn is

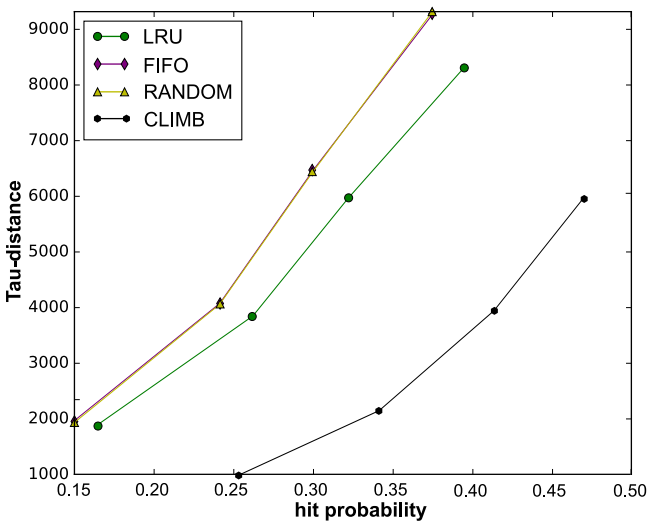

Figure 4.2: τ-distance vs. hit probability for various caching algorithms with IRM arrivals.

superior to FIFO and RANDOM. In summary, in terms of hit probability

$$H_m^{\text{CLIMB}} \quad \geq \quad H_m^{\text{LRU}} \geq \quad H_m^{(\text{FIFO, RANDOM})}.$$

Remark 2 *For the parameters in (4.6), we consider a Zipf-like popularity distribution with* $\alpha = 0.8$. *For simplicity, we set the element weights as* $w_i = n - i + 1$, *and the swapping cost* $\zeta_i = \log i$ *for* $i > 1$, *and* $\zeta_1 = 0.1$. *Different choices of the parameters result in different values of the* τ*-distance, therefore, the y-axis value in Figure 4.2 is only used to show the relative difference between different algorithms.*

4.6 Mixing Time

While identifying the τ-distance does provide some insight into the algorithm's accuracy of learning, it says nothing about how quickly the algorithm can respond to changes in the request distribution—a critical shortcoming in developing and characterizing the ideal algorithm for a given setting. How does one come up with

a metric that accounts for both accuracy and speed of learning? It seems clear that one ought to study the evolution of the caching process with time to understand how quickly the distribution evolves. The metric of relevance in this context is called *mixing time*, which is the time required for a Markov process to reach within ϵ distance (in Total Variation (TV) norm) of the eventual stationary distribution. If we denote the row corresponding to state $\boldsymbol{x} \in \mathcal{S}$ of the t-step transition matrix of algorithm A by $\pi_A(\boldsymbol{x}, t)$, then the mixing time is the smallest value of t such that

$$\sup_{\boldsymbol{x} \in \mathcal{S}} |\pi_A(\boldsymbol{x}, t) - \pi_A^*|_{TV} \leq \epsilon,$$

for a given $\epsilon > 0$ [59]. Denote it as $t_{\text{mix}}(\epsilon)$. As mentioned earlier, we will also think of $\pi_A(\boldsymbol{x}, t)$ and π_A^* as distributions on permutations of the n objects.

Mixing time can be characterized from different perspectives. Here, we use *conductance* to characterize rapid mixing, which also builds bounds on the mixing time through *Cheeger's inequality*. In the rest of this section, we first introduce these techniques and then characterize the mixing time of various caching algorithms.

4.6.1 Spectral Gap and Mixing Time

Let γ^* be the *spectral gap* of a Markov chain with transition matrix P, and denote π_P^* as its corresponding stationary distribution. We first define the spectral gap of a Markov chain through Rayleigh quotient and Dirichlet form.

Definition 4 *[23, 75] For $f, g : \mathcal{S} \to \mathbb{R}$, let $\mathcal{E}(f, g) = \mathcal{E}_P(f, g)$ denote the Dirichlet form,*

$$\mathcal{E}(f, g) = \langle f, (I - P)g \rangle_{\pi_P^*} = \sum_{\boldsymbol{x}, \boldsymbol{y}} f(\boldsymbol{x})(g(\boldsymbol{x}) - g(\boldsymbol{y}))P(\boldsymbol{x}, \boldsymbol{y})\pi_P^*(\boldsymbol{x}). \tag{4.9}$$

If $f = g$, then

$$\mathcal{E}(f,f) = \frac{1}{2}\sum_{\boldsymbol{x},\boldsymbol{y}}(f(\boldsymbol{x}) - f(\boldsymbol{y}))^2 P(\boldsymbol{x},\boldsymbol{y})\pi_P^*(\boldsymbol{x}). \tag{4.10}$$

The Rayleigh quotient for any $f : \mathcal{S} \to \mathbb{R}$, is defined as follows [23],

$$R(f) = \frac{\mathcal{E}(f,f)}{\sum_{\boldsymbol{x}} |f(\boldsymbol{x})|^2 \pi_P^*(\boldsymbol{x})}. \tag{4.11}$$

The spectral gap of the Markov chain with transition matrix P is defined as [23]

$$\gamma^* = \inf_{\substack{f \\ \sum_{\boldsymbol{x}} f(\boldsymbol{x})\pi_P^*(\boldsymbol{x})=0}} \frac{R(f)}{2}. \tag{4.12}$$

Then the upper bound of mixing time in terms of spectral gap and the stationary distribution of the chain is given as follows [59, 75]:

$$t_{\text{mix}}(\epsilon) < 1 + t_{\text{rel}} \ln\left(\frac{1}{\epsilon\pi_{\min}}\right), \tag{4.13}$$

where $t_{\text{rel}} = 1/\gamma^*$ is the *relaxation time* of the Markov chain with transition matrix P, and $\pi_{\min} = \min_{\boldsymbol{x}\in\mathcal{S}} \pi_P^*(\boldsymbol{x})$.

4.6.2 Reversibility and Mixing Time

Reversibility is a significant concept in studying the properties of Markov chains. Many current results of mixing time are shown in the context of a reversible Markov chain. However, a recent survey [75] shows that some of these results hold even without reversibility. In this subsection, we discuss the difference between reversible and non-reversible Markov chains, and then show how to bound the mixing time of a non-reversible Markov chain through constructing a reversible Markov chain. We will use the result later to obtain a bound on the mixing time of the LRU algorithm,

which is associated with a non-reversible Markov chain.

Suppose that P is the transition matrix of a non-reversible chain, and π_P^* is its corresponding stationary distribution. Consider the time-reversal P^*, which is defined by

$$\pi_P^*(\boldsymbol{x})P^*(\boldsymbol{x},\boldsymbol{y}) = \pi_P^*(\boldsymbol{y})P(\boldsymbol{y},\boldsymbol{x}), \tag{4.14}$$

where $\boldsymbol{x},\boldsymbol{y} \in \mathcal{S}$.

Then it is easy to check that the additive Markov chain with transition matrix $\frac{P+P^*}{2}$ is reversible [75]. From Kelly [54], we know $\forall \boldsymbol{x} \in \mathcal{S}$, $\pi_P^*(\boldsymbol{x}) = \pi_{P^*}^*(\boldsymbol{x})$, where $\pi_{P^*}^*$ is the stationary distribution of Markov chain with transition matrix P^*. Therefore, we obtain

$$\pi_P^*(\boldsymbol{x}) = \pi_{P^*}^*(\boldsymbol{x}) = \pi_{\frac{P+P^*}{2}}^*(\boldsymbol{x}). \tag{4.15}$$

From Equation (4.10), we immediately have

$$\mathcal{E}_P(f,f) = \mathcal{E}_{P^*}(f,f) = \mathcal{E}_{\frac{P+P^*}{2}}(f,f), \tag{4.16}$$

Furthermore, by the definition of Rayleigh quotient in Equation (4.11) and the spectral gap of a Markov chain in Equation (4.12), we have

$$\gamma_P^* = \gamma_{P^*}^* = \gamma_{\frac{P+P^*}{2}}^*. \tag{4.17}$$

Therefore, for any non-reversible Markov chain with transition matrix P, we can construct a reversible Markov chain with transition matrix $\frac{P+P^*}{2}$. Since these two Markov chains have the same stationary distribution (4.15) and spectral gap (4.17), by (4.13), we can equivalently use the reversible Markov chain $\frac{P+P^*}{2}$ to bound the mixing time of the non-reversible Markov chain P through applying the existing

results on reversible Markov chains. This procedure will be discussed in the following subsections.

4.6.3 Conductance and Mixing Time

For a pair of states $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$, we define the transition rate $Q(\boldsymbol{x}, \boldsymbol{y}) = \pi(\boldsymbol{x})P(\boldsymbol{x}, \boldsymbol{y})$. Let $Q(S_1, S_2) = \sum_{\boldsymbol{x}\in S_1} \sum_{\boldsymbol{y}\in S_2} Q(\boldsymbol{x}, \boldsymbol{y})$, for two sets $S_1, S_2 \in \mathcal{S}$. Now, for a given subset $S \in \mathcal{S}$, we define its conductance as $\Phi(S) = \frac{Q(S,\bar{S})}{\pi(S)}$, where $\pi(S) = \sum_{i\in S} \pi_i$. Note that $Q(S, \bar{S})$ represents the "ergodic flow" from S to $\bar{S}$. Finally, we define the *conductance of the chain* P to capture the conductance of the "worst" set as

$$\Phi = \min_{S\subset\mathcal{S}, \pi(S)\leq\frac{1}{2}} \Phi(S), \tag{4.18}$$

The relationship between the conductance and the mixing time of a Markov chain (the spectral gap) is given by the *Cheeger inequality* [21]:

$$\frac{\Phi^2}{2} \leq \gamma^* \leq 2\Phi. \tag{4.19}$$

Combining (4.19) with the previous result in (4.13), we can relate the conductance directly to the mixing time as follows:

$$t_{\text{mix}}(\epsilon) \leq \frac{2}{\Phi^2}\left(\ln\frac{1}{\pi_{\min}} + \ln\frac{1}{\epsilon}\right). \tag{4.20}$$

While the spectral gap and conductance of a Markov chain can provide close bounds on the mixing time of the chain, these values are often difficult to calculate. If we are more interested in proving rapid mixing[3], we can provide a lower bound

[3] A family of ergodic, reversible Markov chain with state space of size $|\mathcal{S}|$ and conductance $\Phi_{|\mathcal{S}|}$ is *rapidly mixing* if and only if $\Phi_{|\mathcal{S}|} \geq \frac{1}{\mathcal{P}(|\mathcal{S}|)}$ for some polynomial $\mathcal{P}$ [74]. This result is commonly used to show rapid mixing of Markov chains.

on the conductance. Canonical path and congestion can be useful in this regard as they are easier to compute, and can be used to bound the conductance from below. For any pair $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$, we can define a canonical path $\psi_{\boldsymbol{xy}} = (\boldsymbol{x} = \boldsymbol{x}_0, \cdots, \boldsymbol{x}_l = \boldsymbol{y})$ running from $\boldsymbol{x}$ to $\boldsymbol{y}$ through adjacent states in the state space $\mathcal{S}$ of the Markov chain. Let $\Psi = \{\psi_{\boldsymbol{xy}}\}$ be the family of canonical paths running between all pairs of states. The *congestion* of the Markov chain is then defined as

$$\rho = \rho(\Psi) = \max_{(\boldsymbol{u},\boldsymbol{v})} \left\{ \frac{1}{\pi(\boldsymbol{u})P_{\boldsymbol{uv}}} \sum_{\substack{\boldsymbol{x},\boldsymbol{y}\in\mathcal{S} \\ \psi_{\boldsymbol{xy}}\text{uses}(\boldsymbol{u},\boldsymbol{v})}} \pi(\boldsymbol{x})\pi(\boldsymbol{y}) \right\}, \tag{4.21}$$

where the maximum runs over all pairs of states in the state space. . Therefore, high congestion corresponds to a lower conductance, as demonstrated in [84]

$$\Phi \geq \frac{1}{2\rho}. \tag{4.22}$$

Note that the above result applies to all possible choices of canonical paths, for example, no requirement was ever made that the shortest path between two states has be chosen.

4.6.4 Analysis of Mixing Time

In this subsection, we characterize the mixing time of LRU, FIFO, RANDOM, and CLIMB. To ease exposition of our results, all proofs are relegated to the Appendix C.

4.6.4.1 Mixing Time of LRU

We consider the IRM arrival process and denote the probability of requesting item i by p_i. It is easy to verify that the Markov chain associated with the LRU algorithm is non-reversible, for instance using the Kolmogorov condition. Hence, as discussed in Section 4.6.2, we first need to construct the time reversal $P^{\text{LRU},*}$ satis-

fying Equation (4.14), given the transition matrix P^{LRU} of LRU. Then the Markov chain with transition matrix $\frac{P^{\mathrm{LRU}}+P^{\mathrm{LRU},*}}{2}$ is reversible. Therefore, we can adapt the results of mixing time of reversible Markov chain to show that the Markov chain of LRU is rapidly mixing, i.e., we can show that the congestion of $\frac{P^{\mathrm{LRU}}+P^{\mathrm{LRU},*}}{2}$ is polynomial in the size of state space. We have the following result.

Theorem 11 *The Markov chain of LRU is rapidly mixing.*

Once we have characterized the congestion, we consider the reversible Markov chain with transition matrix $\frac{P^{\mathrm{LRU}}+P^{\mathrm{LRU},*}}{2}$ and use a conductance argument to give an upper bound on the mixing time of LRU.

Theorem 12 *The mixing time of LRU satisfies*

$$t_{mix}^{LRU}(\epsilon) = O(n^{4\alpha m+2} \ln n).$$

4.6.4.2 Mixing Time of RANDOM and FIFO

We next show that the congestion of RANDOM/FIFO are polynomial in the size of state space, i.e., they are rapidly mixing. Since is easy to verify that these two algorithms have reversible Markov chains, we can use the traditional approach of using a conductance argument on the transition matrix P to bound the mixing times.

Theorem 13 *The Markov chains of RANDOM and FIFO are rapidly mixing.*

We then derive a bound on the mixing time of both algorithms.

Theorem 14 *The mixing times of RANDOM and FIFO satisfy*

$$t_{mix}^{RANDOM}(\epsilon) = O(n^{6\alpha m+2} \ln n).$$

4.6.4.3 Mixing Time of CLIMB

We now turn to the CLIMB algorithm. It is easy to verify that it too generates a reversible Markov chain. We show that the congestion of CLIMB is polynomial in the size of the state space, i.e., it is rapidly mixing.

Theorem 15 *The Markov chain of CLIMB is rapidly mixing.*

We now have the following bound on the mixing time of CLIMB.

Theorem 16 *The mixing time of CLIMB satisfies*

$$t_{mix}^{CLIMB}(\epsilon) = O(n^{3\alpha m(m+1)+2} \ln n).$$

4.6.5 Comparison of Mixing Times

We are now in a position to compare the bounds on mixing times of all our candidate algorithms for the simple cache system. While the results are all upper bounds on mixing time, they should allow us to make a judgement on the worst case behaviors of each algorithm, and are a conservative estimate on likely performance in practice. From Theorems 12 and 14, the upper bound of the mixing time of LRU is smaller than the upper bound of the mixing time of RANDOM. Similar results hold for FIFO. Thus, LRU mixes faster than RANDOM or FIFO. Similarly, by Theorems 12 and 16, the upper bound of the mixing times of LRU, FIFO and RANDOM are all smaller than the upper bound of the mixing time of CLIMB. Thus, they are all likely to mix faster than CLIMB. The phenomenon of LRU mixing faster than CLIMB has been numerically observed in [41]. In summary, the expected ordering in mixing times from smallest to largest is likely to be

$$t_{\text{mix}}^{\text{LRU}} \quad \leq \quad t_{\text{mix}}^{(\text{RANDOM, FIFO})} \quad \leq \quad t_{\text{mix}}^{\text{CLIMB}}.$$

4.7 Learning Error

Although we have succeeded in deriving the mixing time of a caching algorithm, how do we combine it with the notion of τ-distance to obtain a figure of merit for an algorithm's performance? What we really desire is a notion of error that accounts for the tradeoff between accuracy and speed of learning. Clearly, a figure of merit of this kind is the distance $\delta(t) = \sup_{x \in \mathcal{S}} |\pi_A(x,t) - \mathbf{c}^*|_\tau$ for some given time t. We could then argue that if the time constant of the change in the request distribution is t, the caching algorithm A would have attained some fraction of optimality by that time.

Fortunately, since the space of all permutations on n objects is finite, it has a finite diameter in terms of the generalized Kendall's tau distance. Let this diameter be denoted κ. Therefore, using the coupling definition of both the Total Variation distance and the Wasserstein metric, the product of the diameter and the Total Variation distance bounds the τ-distance [89]. In this context both are variants of the Wasserstein distance [89], and are therefore equivalent ways of measuring distance between distributions. Thus, we may use the triangle inequality to obtain

$$\begin{aligned}
\delta_A(t) &= \sup_{x \in \mathcal{S}} |\pi_A(x,t) - \mathbf{c}^*|_\tau \\
&\leq \sup_{x \in \mathcal{S}} |\pi_A(x,t) - \pi_A^*|_\tau + |\pi_A^* - c^*|_\tau \\
&\leq \kappa \sup_{x \in \mathcal{S}} |\pi_A(x,t) - \pi_A^*|_{TV} + |\pi_A^* - c^*|_\tau \\
&\triangleq e_A(t). \qquad (4.23)
\end{aligned}$$

The first term above indicates the error due to time lag of learning, while the second indicates the error due to (eventual) accuracy of learning. Hence, we refer to $e_A(t)$

as the *learning error* of algorithm A at time t.

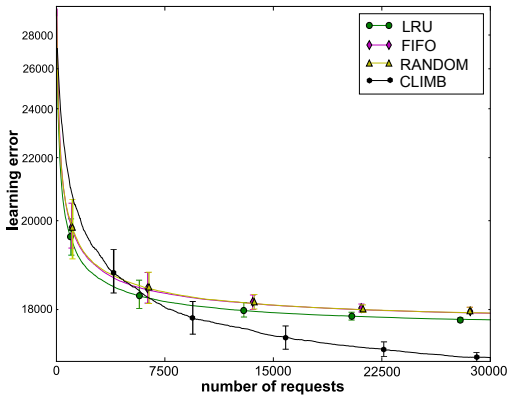

Figure 4.3: Learning error of various caching algorithms under the IRM arrival process.

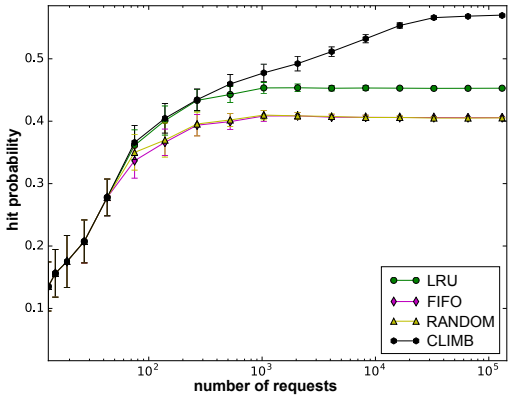

Figure 4.4: Hit probability of various caching algorithms under the IRM arrival process.

4.7.1 Model Validation and Insights

We illustrate the *learning error* of different caching algorithms in Figure 4.3. We use a small cache size for these simulations, since computing all permutations becomes prohibitively complex quickly. However, $(n = 20, m = 4)$ serves to illustrate the main insights. The learning error of various algorithms as a function of the number of requests received is shown in Figure 4.3, where the y-axis is shown with a logarithmic scale. We see immediately that FIFO and RANDOM have higher learning errors than the other algorithms, regardless of the number of requests. This shows why their performance is poor however long they are trained. LRU decreases fast initially and then levels off, but with a larger learning error than CLIMB. CLIMB has a good performance eventually, but it has the slowest decay rate. This corresponds to the slowest mixing of CLIMB, which is consistent with our analysis above.

The same effects are visible in Figure 4.4, where the x-axis is shown with a logarithmic scale.

4.8 Conclusion

In this section, we attempted to characterize the adaptability properties of different caching algorithms when confronted with non-stationary request arrivals. To begin with, we first considered the stationary distributions of various caching algorithms under a stationary request process, and computed the τ-distance between each one and the optimal content placement in the cache. We then analyzed the mixing time of each algorithm with stationary arrivals, to determine how long each one takes to attain stationarity. By combining both of these metrics, we constructed the *learning error,* which characterizes the tradeoff between speed and accuracy of learning. The learning error provides insight into the likely performance of each algorithm under non-stationary requests. In terms of prescriptive solutions, our result was that LRU achieves a good tradeoff between accuracy of learning versus the speed of learning the arrival process. However, the only parameter in all these algorithms is the cache size m, which is a constant. Hence, none of the algorithms described can be parametrically modified based on the application (how quickly the arrival distribution changes). In the next section, we consider using the dimensions of layering and addition of meta-caches to provide parameters that can be adjusted to obtain a desired tradeoff between learning rate and accuracy.

5. ACCURACY VS. LEARNING RATE OF MULTI-LEVEL CACHING ALGORITHMS

5.1 Introduction

In Section 4, we characterized the performance of caching algorithms on an isolated cache using the τ-distance and mixing time. As discussed earlier, the ideas of using multi-level caches have been explored to improve the performance of an isolated cache through numerical studies. However, it is not clear how the number of levels and the partition of total cache size across these levels will impact performance. In this section, we first characterize the performance of multi-level caches and then combine the insights we obtained to design new caching algorithm, which outperforms all the conventional algorithms we have considered.

First, we focus on the particular topology of multi-level cache network: a linear cache network. As the name suggests, such a cache network consists of a stack of caches, potentially of different sizes and at possibly different distances from the content requesting site. Linear stacks of caches can be used at a single node, such as a CDN content node, where they could have a higher hit probability than a single cache, or in a microprocessor where the delay of the cache responses increases with increasing distance from the core. Linear cache networks are also a basic building block of more complex cache hierarchies across a CDN [30], and hence can be thought of as the simplest CDN graph.

In a linear cache network, a content item enters the cache network via the first cache and is advanced to a higher index cache whenever there is a cache hit on it. An advancement could necessitate an eviction if the target cache is full, and a replacement algorithm determines the item to be evicted. We are particularly

interested in replacement algorithms that operate on a total available cache of size m, partitioned into h levels represented by $\boldsymbol{m} = (m_1, \cdots, m_h)$. In particular, we focus on well known policies such as RANDOM($\boldsymbol{m}$), First-in-First-Out (FIFO($\boldsymbol{m}$)) and Least-Recently-Used (LRU($\boldsymbol{m}$)); these will be described in detail later on.

Our first objective in this section is to derive a fundamental characterization of the accuracy versus convergence tradeoff across different caching algorithms in the case of a linear cache network. We wish to explore this tradeoff as parametrized by (i) the number of cache levels, and (ii) the partitioning of the total cache space across these levels. Towards this goal, we first characterize the stationary distributions of our candidate replacement algorithms in the linear cache network under the IRM model. Based on these stationary distributions, we derive explicit expressions for the hit probabilities. We then use the "τ-*distance*" to study the accuracy of learning the request distribution in the context of cache networks. We find that under IRM requests, the accuracy of an algorithm increases both with the number of cache levels and the space allocated to higher caches. Essentially, accuracy lies in favoring higher level caches.

Next, we characterize the mixing time of cache replacement algorithms in cache networks. We show that under IRM requests, the mixing time of an algorithm increases both with the number of cache levels and the space allocated to higher caches. Hence, learning accuracy and speed are exactly at odds with each other. We provide a numerical study on how to partition the available cache space in a linear cache network using both synthetic traces under the IRM model and trace-based simulations using traces from YouTube. These results provide guidelines on how to select a caching algorithm among these candidate replacement algorithms such that a good tradeoff is obtained between the cache size, the number of caches in the network and the request characteristics.

Final, motived by our analysis, we propose a novel hybrid algorithm which combines the ideas from LRU and 2-LRU in such a way that the learning error is minimized for a dynamic arrival process. We name the resulting algorithm as Adaptive-LRU (A-LRU), and are able to ensure that its learning error at a given time can be made less than either LRU or 2-LRU. We also show that it has the highest hit probability over a class of algorithms that we compare it with using both synthetic requests generated using a Markov-modulated process, as well as trace-based simulations using traces from YouTube and the IRCache project.

5.1.1 Organization

This section is organized as follows. In Section 5.2, we study the multi-level caching algorithms in the context of linear cache networks. Some technical preliminaries and representative replacement algorithms are introduced in Section 5.2.1. We derive the steady-state distributions of these algorithms in Section 5.2.2, and identify hit probabilities in Section 5.2.3. We consider the τ-distance in Section 5.2.4, and mixing time in Section 5.2.5. We provide trace-based numerical results of multi-level caching algorithms in Section 5.2.6. Finally, we propose A-LRU and analyze its performance in Section 5.3. We conclude in Section 5.4.

5.2 Performance of Multi-level Caching Algorithms

5.2.1 Preliminaries

Traffic Model and Popularity Law

We consider the *Independent Reference model* (IRM) [25] and Zipf-like law for content popularity for most of our analysis. Details are given in Section 4.2 of Section 4 and hence are omitted here.

Linear Cache Network

We consider a general linear cache network, as illustrated in Figure 5.1, which

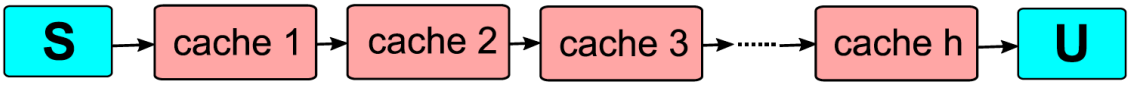

Figure 5.1: Linear cache network: "S" and "U" stands for the server and user, respectively.

is composed of h caches labeled as $1, \cdots, h$, each with size $m_i \in \mathbb{Z}_+$, $i = 1, \cdots, h$. The total cache size is denoted as $m = \sum_{i=1}^{h} m_i$. There are no exogenous requests at caches $2, \cdots, h$. An item enters the cache network via cache 1, and will be promoted to a higher index cache whenever there is a cache hit on it.

Replacement Algorithms

We consider a class of replacement algorithms based on the linear cache network in Figure 5.1. We denote the members of these classes as RANDOM($\boldsymbol{m}$), FIFO($\boldsymbol{m}$), and LRU($\boldsymbol{m}$), where $\boldsymbol{m} = (m_1, \cdots, m_h)$.

RANDOM($\boldsymbol{m}$): When item k is requested, there are three cases: (1) k is not in the cache network (cache miss), then k is inserted in a *random* position in cache 1, and this randomly selected item is evicted; (2) k is in position j of cache $i < h$ (cache hit), then k is inserted in a *random* position in cache $i+1$, while the item that was in this randomly selected position moves to position j of cache i; and (3) k is in cache h (cache hit), no change is made.

FIFO($\boldsymbol{m}$): The differences between FIFO($\boldsymbol{m}$) and RANDOM($\boldsymbol{m}$) are from two perspectives. First, when a cache miss happens, item k is inserted into the first position of cache 1, items in higher position move back one position, and the last item is evicted. Second, when there is a cache hit on item k in cache $i < h$, say in position j of cache i, item k moves to the first position of cache $i+1$, all other items in cache $i+1$ move back one position. The last item that was in cache $i+1$ moves to position j of cache i.

LRU($\boldsymbol{m}$): The differences between LRU($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) are from two perspectives. First, when there is a cache hit on item k that was in position j of cache $i < h$, as before, k moves to the first position of cache $i+1$, and all other items in cache $i+1$ move back one position, with the difference that the last item that was in cache $i+1$ moves to the first position of cache i, and all items that were in positions 1 to $j-1$ of cache i move back one position. Second, when there is a cache hit in position j of cache h, item k moves to the first position of cache h and all other items move back one position.

Remark 3 *We obtain RANDOM($\boldsymbol{m}$), FIFO($\boldsymbol{m}$), and LRU($\boldsymbol{m}$), described above, by slightly modifying the algorithms introduced in [9, 71].*

5.2.2 Steady State Distribution

Our first objective is to determine the stationary distributions of various replacement algorithms in the linear cache network. Suppose there are a total of n content items in a library, the total cache size is $m < n$, and there are h caches in the network. Let $\mathcal{S}$ contain all the vectors of m distinct integers taken from the set $\{1, \cdots, n\}$. It is easy to see that each replacement algorithm A under any Markov modulated request arrival process (which includes IRM as well) results in a Markov process on the state space $\mathcal{S}$. Denote $\pi^*_A(\boldsymbol{x})$ as the stationary probability of state $\boldsymbol{x} = (x_1, \cdots, x_m)$.

For simplicity, we denote $x(i,j)$ as the identity of the item at position j in cache i, where $i = 1, \cdots, h$ and $j = 1, \cdots, m_i$. Next, we present the steady state probabilities of this Markov chain for FIFO($\boldsymbol{m}$) and RANDOM($\boldsymbol{m}$).

Theorem 17 *Under the IRM, the steady state probabilities $\pi^*_{FIFO(\boldsymbol{m})}(\boldsymbol{x})$ and*

$\pi^*_{RANDOM(\boldsymbol{m})}(\boldsymbol{x})$, *with* $\boldsymbol{x} \in \mathcal{S}$ *are as follows*

$$\pi^*_{FIFO(\boldsymbol{m})}(\boldsymbol{x}) = Z(\boldsymbol{m})^{-1} \prod_{i=1}^{h} \left(\prod_{j=1}^{m_i} p_{x(i,j)} \right)^i,$$
$$\pi^*_{RANDOM(\boldsymbol{m})}(\boldsymbol{x}) = G(\boldsymbol{m})^{-1} \prod_{i=1}^{h} \left(\prod_{j=1}^{m_i} p_{x(i,j)} \right)^i,$$

where $Z(\boldsymbol{m})$, *and* $G(\boldsymbol{m})$ *are the normalizing constants, satisfying*

$$Z(\boldsymbol{m}) = \sum_{\boldsymbol{x} \in \mathcal{S}} \prod_{i=1}^{h} \left(\prod_{j=1}^{m_i} p_{x(i,j)} \right)^i,$$
$$G(\boldsymbol{m}) = \sum_{\boldsymbol{x} \in \mathcal{S}'} \prod_{i=1}^{h} \left(\prod_{j=1}^{m_i} p_{x(i,j)} \right)^i, \tag{5.1}$$

where $\mathcal{S}'$ *denotes the set of all combinations of the elements of* $\{1, \cdots, n\}$ *taken* m *at a time. Note that the elements of* $\mathcal{S}'$ *are subsets of* $\{1, \cdots, n\}$, *while elements of* $\mathcal{S}$ *are ordered subsets of* $\{1, \cdots, n\}$, *satisfying*

$$\sum_{\boldsymbol{x} \in \mathcal{S}} \prod_{i=1}^{h} \left(\prod_{j=1}^{m_i} p_{x(i,j)} \right)^i = m! \sum_{\boldsymbol{x} \in \mathcal{S}'} \prod_{i=1}^{h} \left(\prod_{j=1}^{m_i} p_{x(i,j)} \right)^i.$$

This result indicates a small inaccuracy in [33] (Theorem 1), which ignores the difference between $\mathcal{S}'$ and $\mathcal{S}$. Our results contain the well-known steady state probabilities for FIFO and RANDOM on an isolated cache (i.e. $h = 1$) [9, 25, 35, 55].

5.2.3 Hit Probability

Once we have the stationary distribution, we can easily characterize the hit probability of RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) using Equation (4.4) in Section 4.4 of Section 4.

We then compare the hit probability of RANDOM($\boldsymbol{m}$), FIFO($\boldsymbol{m}$) and LRU($\boldsymbol{m}$)

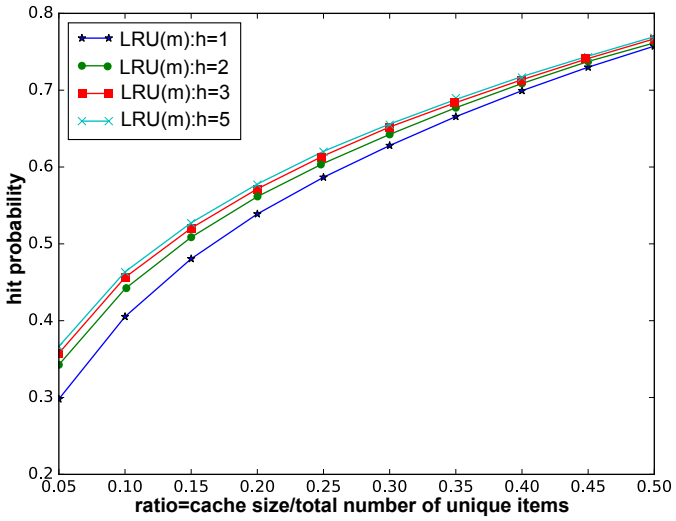

Figure 5.2: Hit probability of LRU($\boldsymbol{m}$) with $m_{i+1} = 0.5m_i$ for $i = 1, \cdots, h-1$ and $\sum_i m_i = m$.

via simulations. Unless otherwise specified, we will always simulate a request arrival process using the IRM with a Zipf-like popularity distribution with exponent $\alpha = 0.8$. We consider a large linear cache network, where the tuple (n, m) is comparable to real cache networks. We illustrate how different algorithms perform using a content library size of $n = 3,000$ with sufficiently long runs (i.e., enough number of requests to make sure that the system has reach steady state, e.g., about 6×10^6 requests). The hit probability of the algorithms are calculated as Hit probability = Total No. of Hit Counts/Total No. of Request Counts.

We first study how hit probability varies with the total cache size and the number of cache levels. Figure 5.2 shows the hit probabilities achieved by LRU($\boldsymbol{m}$), where we assume that $m_{i+1} = 0.5m_i$ for $i = 2, \cdots, h$, satisfying $\sum_{i=1}^{h} m_i = m$. The hit probabilities increase with total cache size m, as well as the number of caches in the linear cache network. However, the gain becomes limited when $h \geq 5$. In other words, most caching gain can be obtained by using a small number of cache levels. Similar

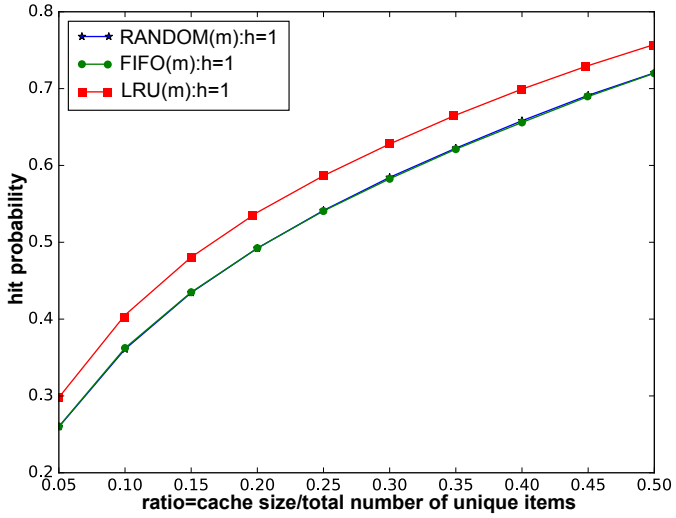

Figure 5.3: Hit probabilities of various replacement algorithms with $m_{i+1} = 0.5m_i$ for $i = 1, \cdots, h-1$ and $\sum_i m_i = m$: $h = 1$.

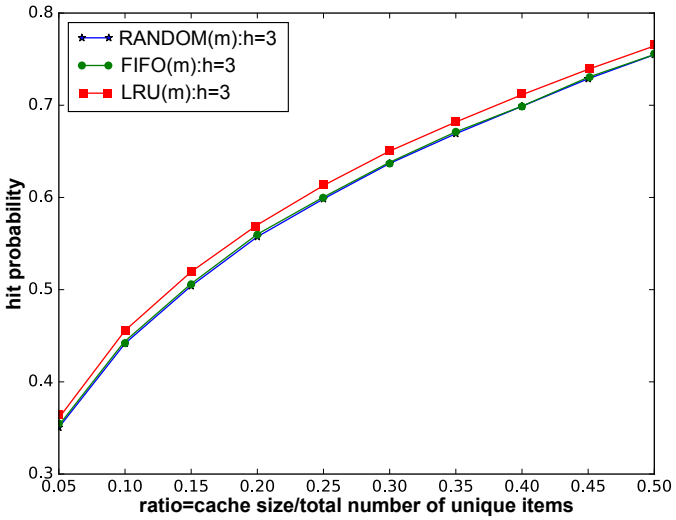

Figure 5.4: Hit probabilities of various replacement algorithms with $m_{i+1} = 0.5m_i$ for $i = 1, \cdots, h-1$ and $\sum_i m_i = m$: $h = 3$.

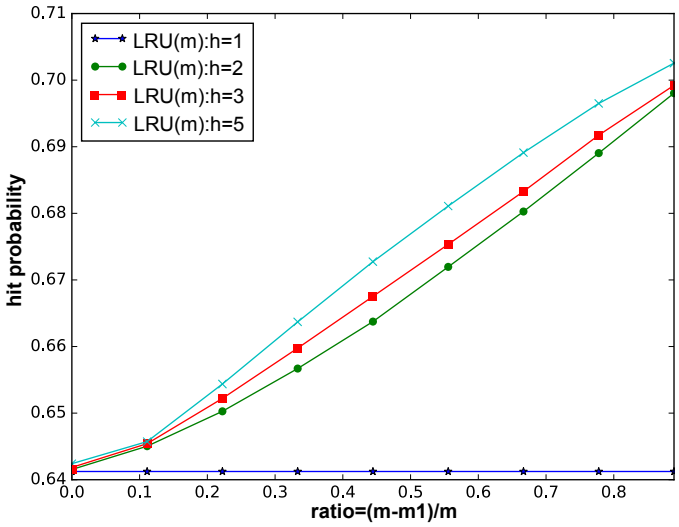

Figure 5.5: Hit probabilities of LRU($\boldsymbol{m}$) with $m_i = (m - m_1)/(h-1)$ for $i = 2, \cdots, h$ and $\sum_i m_i = m$.

trends hold for RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$), and we omit them here. Figures 5.3 and 5.4 compare the performance of different algorithms under the same setting; here we omitted the cases for $h = 2, 5$. We note that LRU($\boldsymbol{m}$) outperforms FIFO($\boldsymbol{m}$) and RANDOM($\boldsymbol{m}$). Again, the gain decreases as we further increase h.

We next consider the impact of the cache partitioning policy on performance. In particular, we focus on the division between cache 1 and all the others. Hence, we vary m_1 and divide the remaining cache size evenly among the remaining $h - 1$ caches, given a fixed total cache size $m = 900$. Figure 5.5 shows the hit probabilities of LRU($\boldsymbol{m}$) as a function of the cache partitions (decreasing value of m_1). Similarly, Figures 5.6 and 5.7 compare the performance of the different candidate algorithms. We see that the hit probability increases as the higher level caches are assigned more space. In summary, the hit probability of a caching policy increases by favoring more levels and assigning more resources to higher level caches.

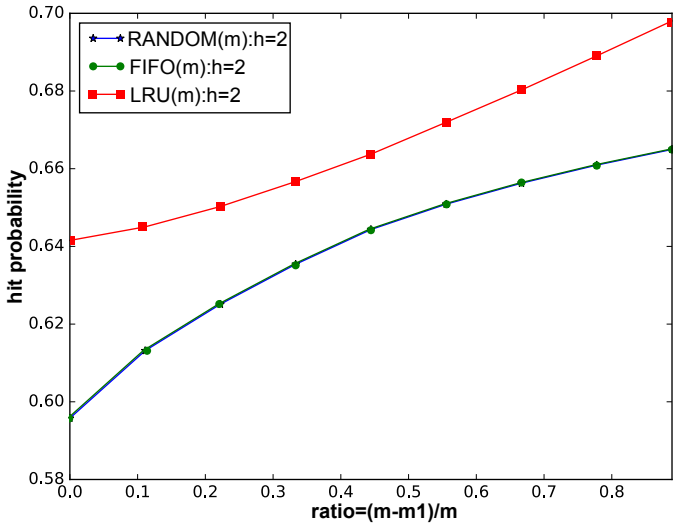

Figure 5.6: Hit probabilities of various replacement algorithms with $m_i = (m - m_1)/(h-1)$ for $i = 2, \cdots, h$: $h = 2$.

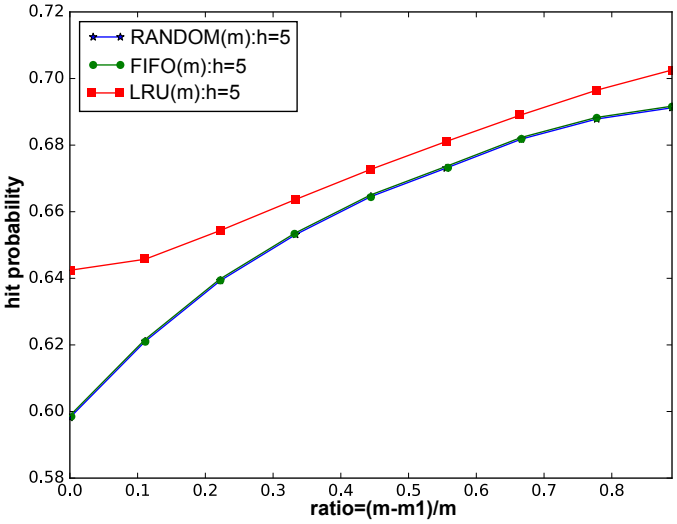

Figure 5.7: Hit probabilities of various replacement algorithms with $m_i = (m - m_1)/(h-1)$ for $i = 2, \cdots, h$ and $\sum_i m_i = m$: $h = 5$.

5.2.4 Permutation Distance

While we observed above using the hit probability metric that most of the caching gain can be obtained by using a small number of cache levels, what we really desire is to know "how close" the stationary performance of a replacement algorithm is to the best-possible one. In order to do this, we need to refine the hit probability metric to a measure of distance. We utilize the τ-distance defined in Section 4.5 of Section 4 to characterize the performance of replacement algorithms in the context of linear cache networks.

5.2.4.1 Model Validation and Insights

As the τ-distance is a metric that characterizes the accuracy of an algorithm learning the popularity distribution, a lager hit probability is expected for an algorithm with a smaller τ-distance. The computational complexity of the τ-distance is high, due to the explosion of the state space as the cache size and number of items increases. However, we can still shed light on how different algorithms perform using a relatively small content library size of $n = 15$.

We explore the impact of the number of cache levels h on the performance when using a small cache size $m = 5$ to illustrate the main insights. Since the cache size should be an integer, we explore a range of cache partitions under each h. We compare the lower envelope of achievable τ-distance of various replacement algorithms, shown in Figure 5.8, while the corresponding upper envelope of achievable hit probabilities are shown in Figure 5.9. We observe that given a total cache size, increasing the number of caches in the linear network can improve the performance, however, most gain can be obtained with a small number of caches in the linear cache network. This observation can be confirmed from the synthetic request data simulations shown in Section 5.2.3, and trace based simulations that will be presented in Section 5.2.6.

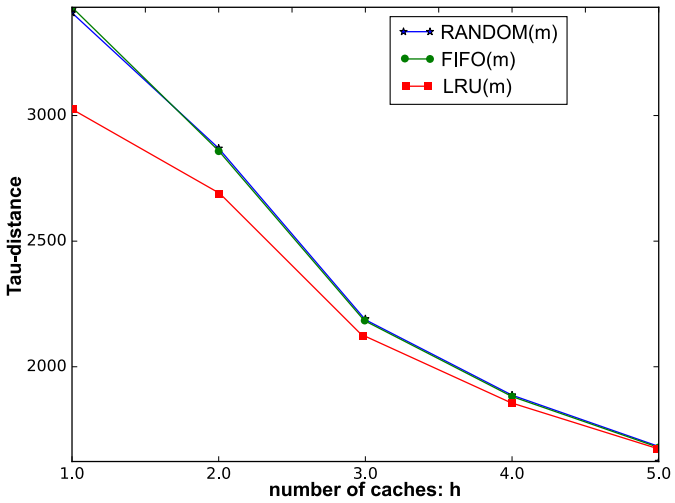

Figure 5.8: τ-distance vs. number of caches h for various replacement algorithms with IRM arrivals.

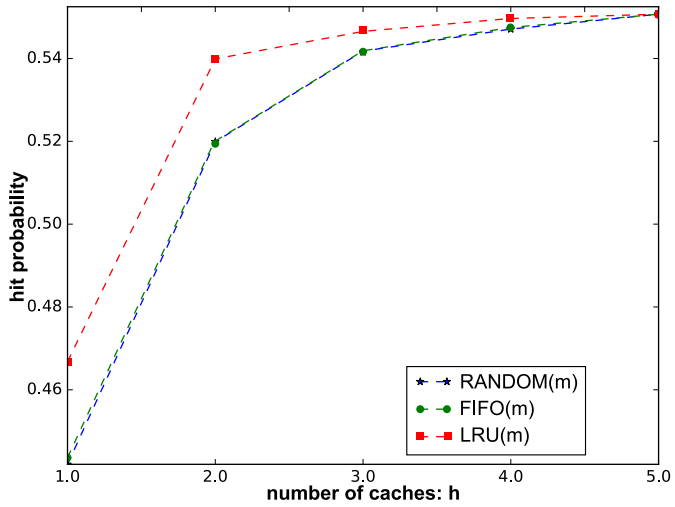

Figure 5.9: Hit probability vs. cache number h for various replacement algorithms with IRM arrivals.

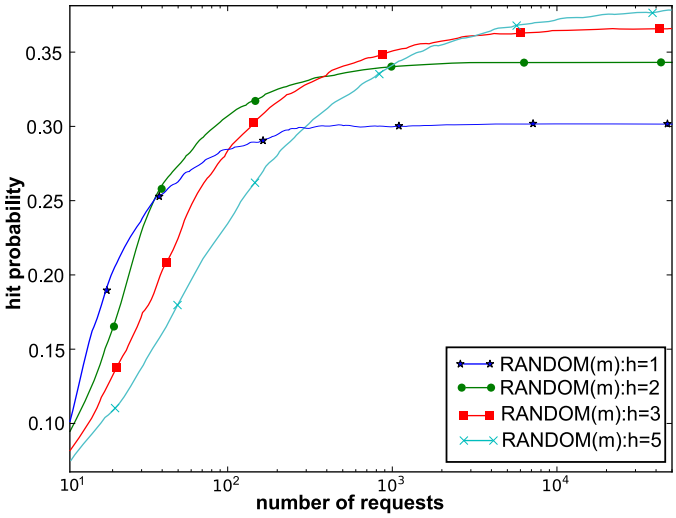

Figure 5.10: Hit probability vs. number of requests for RANDOM($\boldsymbol{m}$) replacement algorithm with IRM arrivals.

These effects become more pronounced as the cache size and the size of content library become large. Figures 5.8 and 5.9 show that the τ-distance and hit probability follow the same rule, i.e., a smaller τ-distance corresponds to a larger hit probability, which is as expected.

Remark 4 *The parameters that we used in (4.6) are as follows. We consider a Zipf-like popularity distribution with exponent* $\alpha = 0.8$. *For the ease of calculations, we take* $w_i = n - i + 1$ *as the element weights, and* $\zeta_i = \log i$ *for* $i > 1$, *and* $\zeta_1 = 0.1$, *as the swapping cost. Different values of* τ*-distance can be obtained by different choices of the parameters, hence, the y-axis value in Figure 5.8 only represents the relative difference between different algorithms.*

5.2.5 Mixing Time of Multi-level Caching Algorithms

While the τ-distance allowed us to obtain insights on how the number of levels in a linear cache network impact the algorithm's accuracy of learning, it does not

indicate how long it takes to reach this eventual accuracy. In this section, we wish to understand how quickly a replacement algorithm can respond to external changes: changes in the number of caches, as well as the change in the request distribution. In other words, we need to study how a caching process evolves with respect to time to understand the evolution of the distribution. The relevant metric here is called "*mixing time*," which is defined in Section 4.6 of Section 4. In this section, we utilize the same techniques introduced in Section 4.6 of Section 4 to characterize the mixing time of cache placement algorithms in linear cache networks.

5.2.5.1 Analysis of Mixing Time

We are now ready to characterize the mixing time of RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$). We show that RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) are rapid mixing by the conductance argument, which results in an explicit form of the upper bound of the mixing time. Details of the proof are available in Appendix D.

Rapid mixing of RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$)

We show that the congestion of RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) are polynomial in the size of state space, i.e., they are rapidly mixing.

Theorem 18 *The Markov chains of RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) are rapid mixing.*

Theorem 19 *The mixing time of RANDOM($\boldsymbol{m}$) satisfies*

$$t_{mix}^{RANDOM(\boldsymbol{m})}(\epsilon) = O(n^{6\alpha(m_1+2m_2+\cdots+hm_h)+2} \ln n), \tag{5.2}$$

where $m = m_1 + m_2 + \cdots + m_h$.

Similar results hold for FIFO($\boldsymbol{m}$).

Remark 5 *For a given total cache size m, we can determine the impact of multiple cache levels and partitions by analyzing the term in the exponent in (5.2), which we*

define as $\mu(\boldsymbol{m}) \triangleq m_1+2m_2+\cdots+hm_h$. *Now, suppose we choose a harmonic sequence of integers* $q_1 = q, q_2 = q/2, q_3 = q/3, \cdots q_h = q/h$, *where* $q = m/\sum_{i=1}^{h} \frac{1}{i}$. *Suppose that we partition the total cache space* m *into levels as* $\boldsymbol{m}_{\mathcal{A}} = \{m_1 = q_1, m_2 = q_2, \cdots m_h = q_h\}$, *then we have a decreasing sequence of partitions, and* $\mu(\boldsymbol{m}_{\mathcal{A}}) = hq$. *We see immediately that the mixing time bound is increasing in the number of levels* h.

Now, to determine how the cache partitions themselves impact the mixing time, we instead choose $\boldsymbol{m}_{\mathcal{B}} = \{m_1 = q_h, m_2 = q_{h-1}, \cdots m_h = q_1\}$. *Then we have an increasing sequence of partitions and the value of* $\mu(\boldsymbol{m}_{\mathcal{B}}) = q/h + 2q/(h-1) + \cdots + hq$. *Since* $\mu(\boldsymbol{m}_{\mathcal{A}}) < \mu(\boldsymbol{m}_{\mathcal{B}})$, *the mixing time bound is smaller for a decreasing sequence of cache sizes than for an increasing sequence, with the sequences identified yielding the minimum and maximum values.*

We noted in Section 5.2.4 that learning accuracy favors more cache levels with more space allocated to higher levels. However, we have just seen that mixing time favors exactly the complementary case. Figure 5.10 illustrates this tradeoff between learning accuracy and speed by depicting the hit probability as a function of the number of requests, where we take $(n, m) = (100, 15)$. *Note that the x-axis is plotted in a logarithmic scale. We see that increasing the number of caching levels promotes accuracy at the expense of a longer time to attain that accuracy. The results of FIFO(*$\boldsymbol{m}$*) exhibit similar trends, and hence are omitted here.*

5.2.6 Trace-based Simulations Using Youtube Traces

The analytical insights that we have obtained thus far on learning rate and accuracy were obtained under a fixed IRM request model. We next consider arrivals that follow a dynamic request process by conducting trace-based simulations. The trace that we use is publicly available [95] and was extracted from a 2-week YouTube re-

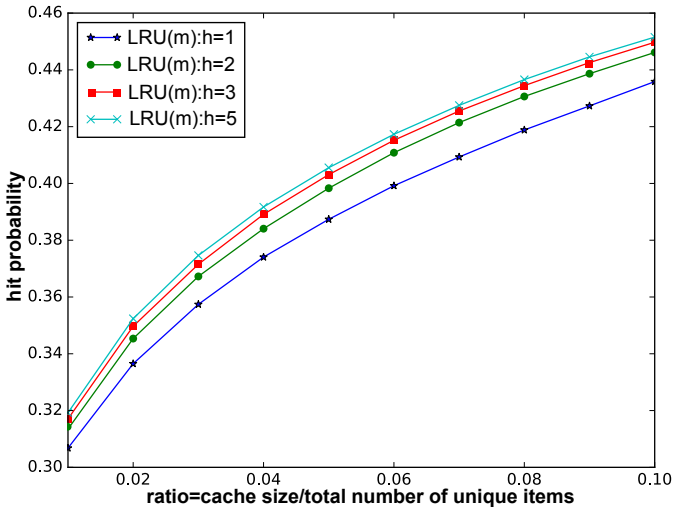

Figure 5.11: Trace-based hit probabilities of LRU($\boldsymbol{m}$) with $m_{i+1} = 0.5m_i$ for $i = 1, \cdots, h-1$ and $\sum_i m_i = m$.

quest traffic dump between June 2007 and March 2008. There are a total of $611,968$ requests for $303,331$ different videos in this trace. About 75% of those videos were requested only once during the trace. There is no information on the video sizes. We therefore assume that the cache size is expressed as the number of videos that can be stored in it. This assumption should have a low impact if the correlation between video popularity and video size is low.

We first compare the performance of different replacement algorithms with the total cache size m and the number of levels h. We make use of the ratio $m/n = 5\%, 10\%, \cdots$ under a different number of cache levels h in the linear cache network. We still consider $m_{i+1} = 0.5m_i$ and $\sum_i m_i = m$ for $i = 1, \cdots, h$; we will explore other partitions later in this section. Figure 5.11 reports the hit probabilities of LRU($\boldsymbol{m}$) as a function of total cache size m. We observe that under the selected partitioning scheme, increasing the number of caching levels can improve performance. This is consistent with the observations under the synthetic simulation in IRM model. In

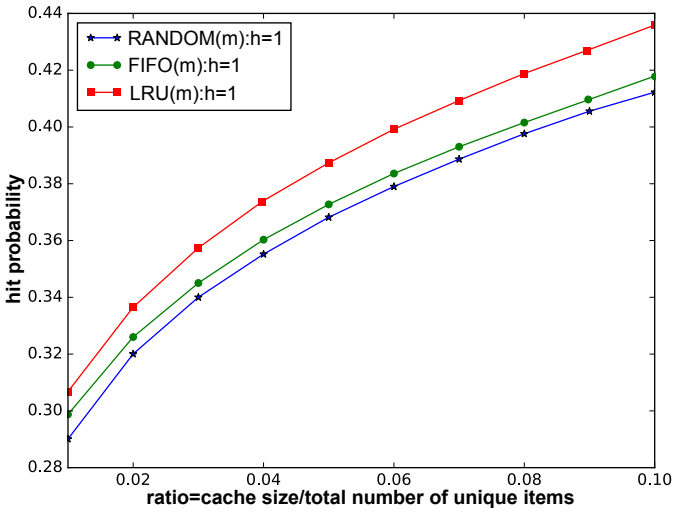

Figure 5.12: Trace-based hit probabilities of various replacement algorithms with $m_{i+1} = 0.5m_i$ for $i = 1, \cdots, h-1$ and $\sum_i m_i = m$: $h = 1$.

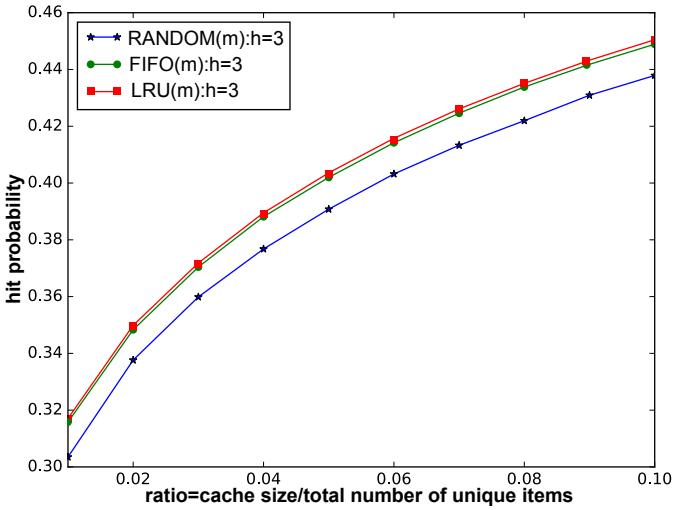

Figure 5.13: Trace-based hit probabilities of various replacement algorithms with $m_{i+1} = 0.5m_i$ for $i = 1, \cdots, h-1$ and $\sum_i m_i = m$: $h = 3$.

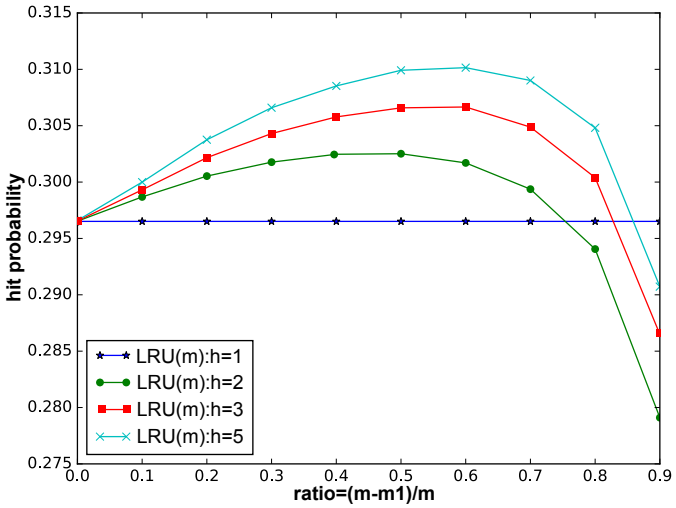

Figure 5.14: Trace-based hit probabilities of LRU($\boldsymbol{m}$) with $m_i = (m - m_1)/(h - 1)$ for $i = 2, \cdots, h$.

fact, most of the gain can be obtained with a small number of caches in the linear cache network. The performance of RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) exhibit similar trends to LRU($\boldsymbol{m}$), and are omitted here.

Figures 5.12 and 5.13 compares hit probabilities between different algorithms, we omit the cases for $h = 2, 5$. We note that LRU($\boldsymbol{m}$) does outperform FIFO($\boldsymbol{m}$) and RANDOM($\boldsymbol{m}$) in all cases, but the gain becomes very limited when h is large. Furthermore, although FIFO($\boldsymbol{m}$) and RANDOM($\boldsymbol{m}$) have the same performance under the IRM model, FIFO($\boldsymbol{m}$) outperforms RANDOM($\boldsymbol{m}$) when confronted with a real data trace.

Next, we investigate the impact of the cache partitions on performance. We fix the total cache size $m = 2,000$, and vary m_1, and evenly divide the remaining cache size among the remaining $h - 1$ caches. Hence, $m_i = \frac{m-m_1}{h-1}$ for $i = 2, \cdots, h$ and $\sum_{i=1}^{h} m_i = m$. Figure 5.14 depicts the hit probability of LRU($\boldsymbol{m}$) as a function of the cache size assigned to m_1, and we illustrate the cases where $h = 1, 2, 3, 5$.

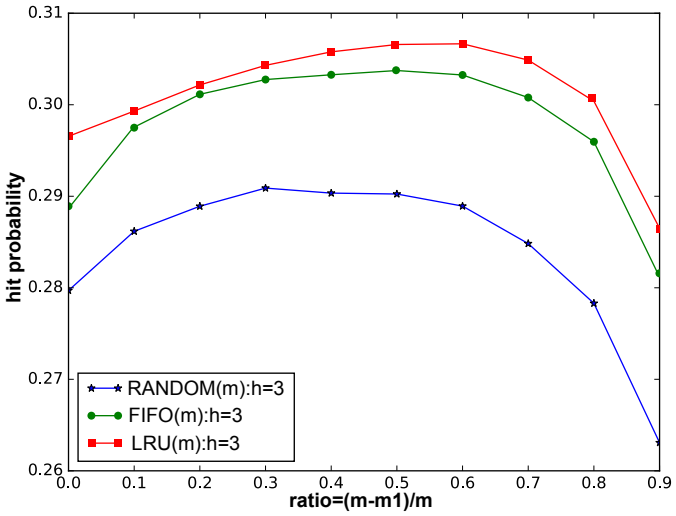

Figure 5.15: Trace-based hit probabilities of various replacement algorithms with $m_i = (m - m_1)/(h - 1)$ for $i = 2, \cdots, h$: $h = 3$.

Figures 5.15 and 5.16 compare the performance of different algorithms. These three figures clearly illustrate the tradeoff between accuracy and speed of learning. As we increase the caching resources allocated to the higher levels, the hit probability first rises due to increased accuracy of learning, and then falls due to increasing learning time. The observation remains consistent with an increase in the number of caching levels, as well as across the different algorithms. Secondary observations are that LRU($\boldsymbol{m}$) has a better performance than the other two, and that the enhanced hit probability obtained through using multiple levels tapers off quite quickly.

5.3 A-LRU Algorithm

From the analysis of multi-level caching algorithm in the context of linear cache networks in Section 5.2, we find that under IRM requests, the accuracy and mixing time of an algorithm increases with both the number of cache levels and the space allocated to higher caches. In this section, we numerically study the functionality

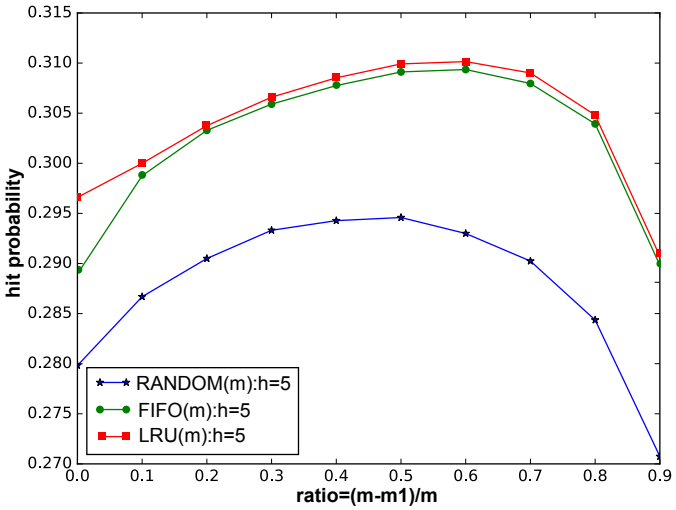

Figure 5.16: Trace-based hit probabilities of various replacement algorithms with $m_i = (m - m_1)/(h - 1)$ for $i = 2, \cdots, h$: $h = 5$.

of meta-cache through the analysis of 2-LRU, we find that under IRM model, meta cache would improve hit probability at the expense of longer time to achieve the stationarity. Then a natural question arises: can we design a new caching algorithms that combine the ideas from meta-cache and multi-level caches which will achieve a better tradeoff between the accuracy of learning items' popularity and the speed of learning? We combine the ideas from the analysis of LRU and 2-LRU to design a new caching algorithm, A-LRU. We characterize its performance in this section.

5.3.1 Caching Algorithms

Adaptive Replacement Cache: ARC [71] uses the history of recently evicted items to change its recency or frequency preferences. Specifically, ARC splits the cache into two parts, $\mathcal{T}_1$ and $\mathcal{T}_2$, which cache items that only have been accessed once, and many times, respectively. Furthermore, ARC maintains two additional lists, $\mathcal{B}_1$ and $\mathcal{B}_2$, to record (LRU-based) eviction history of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. Recency or frequency preferences are adjusted by dynamically changing target sizes

of $\mathcal{T}_1$ and $\mathcal{T}_2$ according to eviction histories recorded in $\mathcal{B}_1$ and $\mathcal{B}_2$. In this way, ARC traces changes in traffic patterns and adjusts the replacement policy to emphasize frequency or recency accordingly.

k-LRU: k-LRU [70] manages a cache of size m by making use of $k-1$ virtual caches, which only store meta-data. Each cache is ordered such that the item in the j-th position of cache l is the j-th most-recently-used item among all items in cache l. When item i is requested, two events occur: (1) For each cache l in which item i appears, say in position j for cache l, then item i moves to the first position of cache l and items in positions 1 to $j-1$ move back one position; (2) For each cache l in which item i does not appear but appears in cache $l-1$, item i is inserted in the first position of cache l, all other items of cache l move back one position, and the last item is evicted.

Adaptive-LRU (A-LRU): We define the quantities $c_1 = \min(1, \lfloor(1-\beta)m\rfloor)$, $c_2 = \lfloor(1-\beta)m\rfloor$, $c_3 = \lfloor(1-\beta)m\rfloor + 1$ and $c_4 = \max(m, \lfloor(1-\beta)m\rfloor + 1)$, where $\beta \in [0,1]$ is a parameter. We partition the cache into two parts with $C2$ defined as the positions from $c_1 \cdots c_2$ and $C1$ as the positions from $c_3 \cdots c_4$. We also define the quantities $m_1 = \min(1, \lfloor\beta m\rfloor)$, $m_2 = \lfloor\beta m\rfloor$, $m_3 = \lfloor\beta m\rfloor + 1$ and $m_4 = \max(m, \lfloor\beta m\rfloor + 1)$. We associate positions $m_1 \cdots m_2$ with meta cache $M2$ and $m_3 \cdots m_4$ with meta cache $M1$. Note that value $m_1 = m_2 = 0$ is an extreme point that yields behavior similar to 2-LRU, while $m_3 = m_4 = m+1$ yields LRU. The cache partitions are shown in Figure 5.17.

Let us denote the meta data associated with a generic item i by $M(i)$. The operation of A-LRU is as follows if item i is requested. Different cases are illustrated in Figure 5.17. There are two possibilities:

(1) **Cache miss**, then there are three cases to consider:

(1a) $M(i) \notin M1 \cup M2$: If $c_3 \neq m+1$, i is inserted into cache position $l = c_3$, else

(extreme case similar to 2-LRU) $M(i)$ is inserted into meta cache position $l = m_3$. Cache/meta cache items in positions greater than l move back one position, and the last meta-data is evicted;

(1b) $M(i) \in M1$: Item i is inserted into position c_1, all other items in $C2$ move back one position, the meta data of item in cache position c_2 is placed in position m_1, all other meta-data items move back one position, and the meta data in position m_2 moves to position m_3;

(1c) $M(i) \in M2$: If $c_1 = 1$, item i is inserted into position $l = c_1$. All other items in $C2$ move back one position, and the meta data of item in cache position c_2 is placed in position m_1. Note that this situation cannot occur in the extreme case of LRU, since $M2$ is always empty for LRU;

(2) **Cache hit**, then there are two cases to consider:

(2a) $i \in C1$ (suppose in position j): If $c_1 = 1$, then item i moves to cache position $l = c_1$, else (extreme case of LRU) item i moves to cache position $l = c_3$. If $l = c_1$, all other items in $C2$ move back one position, the item in cache position c_2 is placed in position c_3, all other items in $C1$ upto position j move back one position. If $l = c_3$ (extreme case of LRU), all other items in $C1$ upto position j move back one position.

(2b) $i \in C2$ (suppose in position j): Item i moves to cache position c_1, and all other items in positions $\min(2, c_2)$ to $j - 1$ move back one position.

Remark 6 *Note that the A-LRU setup can be generalized to as many levels as desired by simply "stacking up" sets of real and meta caches, and following the same caching and eviction policy outlined above (where (1a) would apply to the top level, while (1c) and (2b) would apply to the bottom level). Since most of the possible cache gain has already been achieved by 2-level cache network, here, we focus on analysis the two-level A-LRU algorithm, shown in Figure 5.17. These results can be generalized*

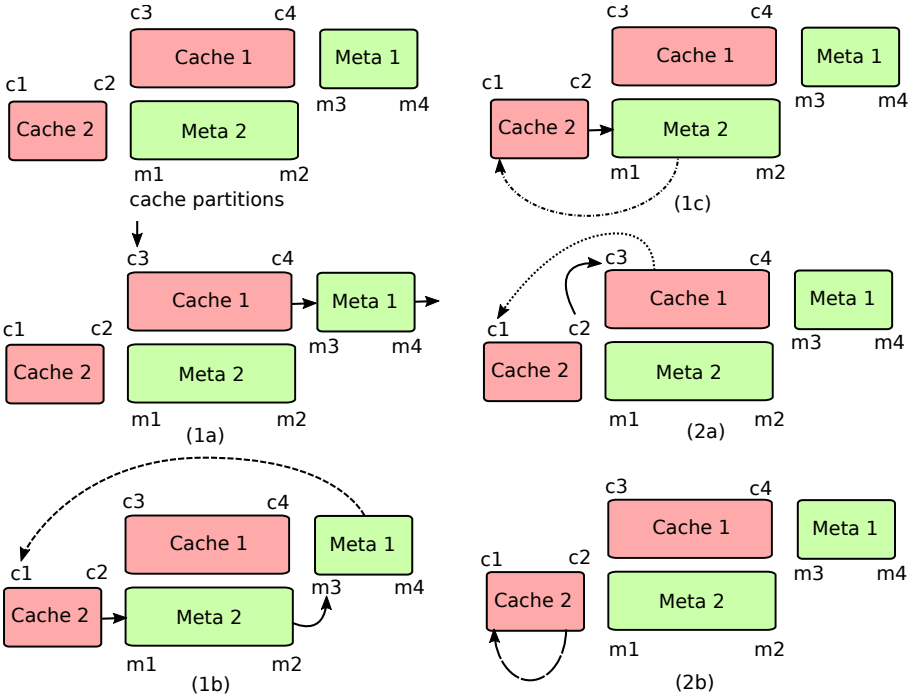

Figure 5.17: Operation of the A-LRU algorithm.

to multi-level A-LRU.

Remark 7 *ARC is an online algorithm with a self-tuning parameter, which has a good performance in some real systems but the implementation is complex. k-LRU has a relatively low complexity, which requires just one parameter, i.e., the number of meta caches $k-1$. We will see that these meta caches will provide a significant improvement over LRU even for small number k. In fact, most of the gain can be achieved by $k=2$. A-LRU captures advantages of LRU and 2-LRU, i.e., learns both faster and better about the changes in the popularity.*

Now we are ready to characterize the performance of A-LRU with respect to τ-distance and mixing time. We studied the performance of LRU, RAMDOM, FIFO and CLIMB in Section 4, where we only provided a partial comparison between these

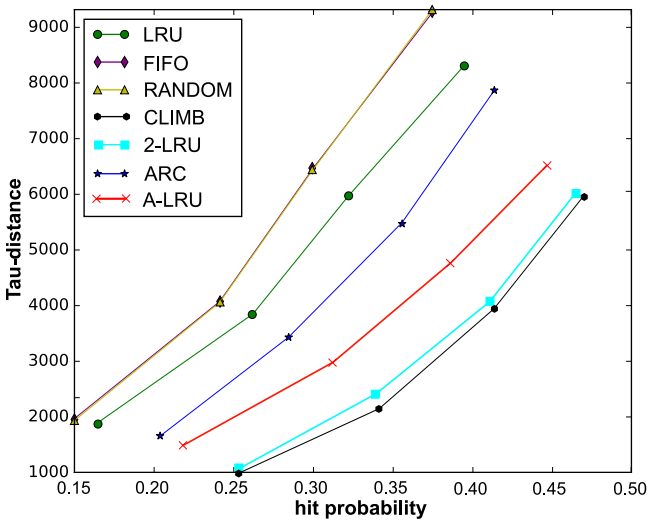

Figure 5.18: τ-distance vs. hit probability for various caching algorithms with IRM arrivals.

algorithms on a single cache. In this section, we provide a comprehensive comparison between A-LRU and these algorithms.

5.3.2 Hit Probability and Permutation Distance

Since the τ-distance characterizes how accurately an algorithm learns the popularity distribution, a smaller τ-distance should correspond to a larger hit probability. Computation of the τ-distance is complex, since it is a function of all possible permutations over the content items. But we can illustrate how different algorithms perform using a content library size of $n = 20$. Figure 5.18 compares the τ-distance and hit probabilities of various caching algorithms. The points on each curve correspond to cache size of $2, 3, 4, 5$ from left to right. Since the cache size should be an integer, we partition the cache for A-LRU such that the size of cache 1 is always 1, and the remaining cache size is allocated to cache 2. From Figure 5.18, we can see that the τ-distance and hit probability follow the same rule, i.e., a smaller τ-distance corresponds to a larger hit probability, which is as expected.

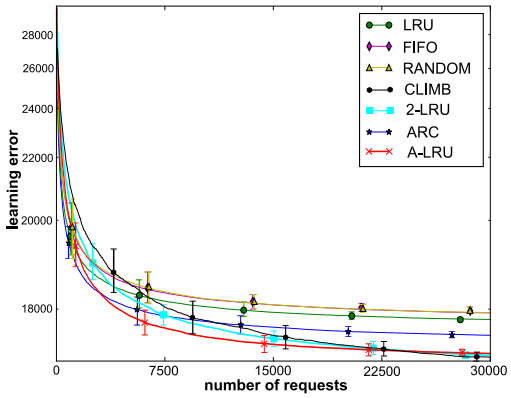

Figure 5.19: Learning error of various caching algorithms under the IRM arrival process.

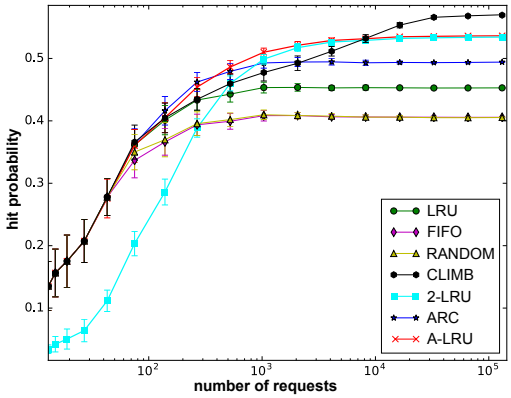

Figure 5.20: Hit probability of various caching algorithms under the IRM arrival process.

5.3.3 Learning Error

We use the *learning error* defined in Equation (4.23) of Section 4.7 in Section 4 to compare the performance of different caching algorithms, as illustrated in Figure 5.19. We use a small cache size for these simulations, since computing all permutations becomes prohibitively complex quickly. However, $(n = 20, m = 4)$ serves to illustrate the main insights. The learning error of various algorithms as a function of the number of requests received is shown in Figure 5.19, where the y-axis is shown with a logarithmic scale. Note also that the result in Figure 5.19 used a version of the A-LRU algorithm with a time-dependent selection of cache divisions that is discussed fully under the heading "Dynamic A-LRU" below. We see that LRU decreases fast initially and then levels off, whereas 2-LRU has a slower decay rate, but the eventual error is lower than that of LRU. This corresponds to faster mixing of LRU but a poorer eventual accuracy (τ-distance) as compared to 2-LRU. The ARC algorithm has a good performance initially, but it too levels off to an error larger than 2-LRU. The A-LRU algorithm with cache partitions $(1, 3)$ picks a combination of accurate

learning and fast mixing, and is able to attain a low learning error quickly.

The effects seen in Figure 5.19 are also visible in the evolution of hit probabilities shown in Figure 5.20, where the x-axis is on a logarithmic scale. Here, we choose $(n = 150, m = 30)$ in order to explore a range of cache partitions for A-LRU from $(0, 30)$–$(30, 0)$. We compare the upper envelope of achievable hit probability by A-LRU with various other caching algorithms. We find that for any given learning time (requests), there is a cache partition such that A-LRU will attain a higher hit probability after learning for that time. These effects become more pronounced as the partition space (cache size) available for A-LRU increases.

Dynamic A-LRU: Whereas in our description of A-LRU, we use a fixed partitioning parameter β, the algorithm (and an implementation of it) can easily consider time-varying β values. For the sake of argument, we consider a k levels A-LRU with a sequence of χs such that $\chi_1, \chi_2, \cdots, \chi_k$ becomes 0 as the number of requests go to infinity, satisfying (i) $\sum_t \chi_i(t) \to \infty$; (ii) $\sum_t \chi_i^2(t) < \infty$; and (iii) $\chi_i(t)/\chi_{i+1}(t) \to 0$. Here, χ_1 stands for the proportion of LRU to the rest, χ_2 stands for the proportion between 2-LRU and 3-LRU to the rest, etc. A typical choice of sequences will be $\chi_i(t) = m/(m + t^{\frac{i+1}{2i}}/c_i)$, where t counts the number of requests and $c_i > 0$ is a parameter to be varied. Under such setting, the βs in the previous definition of A-LRU satisfy that (i) at level $i \leq k-1$, it is $(1-\chi_1(t))(1-\chi_2(t))\cdots(1-\chi_{i-1}(t))\chi_i(t)$, and (ii) at level k, it is $(1-\chi_1(t))(1-\chi_2(t))\cdots(1-\chi_k(t))$.

In particular, we consider the 2-level A-LRU shown in Figure 5.17. Here, the βs are $\beta_1(t) = m/(m + t/c)$ and $\beta_2(t) = 1 - \beta_1(t)$, where we take a common constant c for simplicity. With such a sequence of βs, A-LRU will start at 1 (LRU) and (slowly) decrease to 0 (2-LRU). Under the setting $(n = 150, m = 30)$, we choose different values of the constant $c = 3, 10, 15$ for illustration, as shown in Figure 5.21. We observe that the resulting algorithm will learn fast initially, and then smoothly transition

to learning accurately. Finally, we note that the results for A-LRU presented in Figure 5.19 used $c = 600$.

If the popularity distribution changes with time (in the next section), we should only consider constant β algorithms. These two distinctions follow from stochastic approximation ideas where while decreasing step-size algorithms can converge to optimal solutions in stationary settings, constant step-size algorithms provide good tracking performance for non-stationary settings.

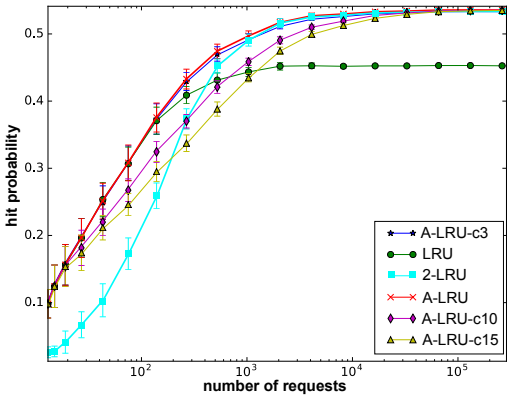

Figure 5.21: Hit probability for A-LRU with time-varying β under IRM arrival process.

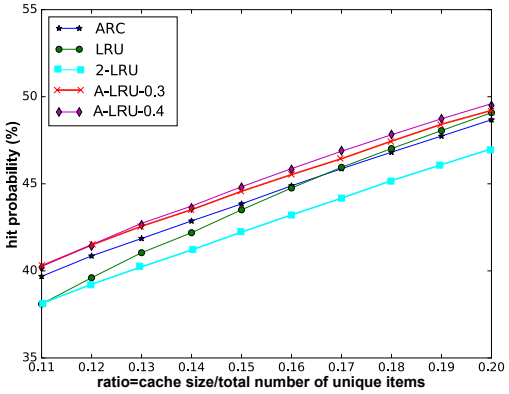

Figure 5.22: Hit probabilities under Markov-modulated arrivals with $\xi = 0.1$.

5.3.4 Markov Modulated Requests

While requests drawn from IRM allow us to study the accuracy of learning a fixed popularity distribution over time, a real arrival process would have a changing popularity distribution. Hence, we desire to construct an arrival process that continually changes the popularity distribution in order to understand how well different caching algorithms are able to track the arrival process.

A simple model that possesses the desired changeability property is a Markov modulated request process. Under this model, we have an underlying two-state Markov chain, in which each state corresponds to one popularity distribution. Requests are drawn from the distribution corresponding to the current state. Define a Markov Chain $\{B_h\}_{h\geq 0}$ with state space $\{0, 1\}$, each corresponding to one popularity distribution. We say $B_h = 0$ if the system is at state 0 and the popularity follows one Zipf-like distribution, and $B_h = 1$ if the system is at state 1 and follows another Zipf-like distribution. W.l.o.g., we consider two Zipf-like distributions over n unique items, one with increasing order of ranking, i.e., $p_i = A/i^\alpha$, the other with decreasing order of ranking, i.e., $p_j = A/(n - j + 1)^\alpha$, where $i, j \in \{1, \cdots, n\}$.

In our model, we assume that if the Markov chain is in some particular state, a fixed number of requests, r, will be drawn according to the distribution corresponding to that state. After that, a possible state transition can take place. For example, if $B_h = 0$, then with probability ξ, the system will stay in state 0 after r requests, otherwise, the system will switch to state 1. Similarly for $B_h = 1$.

Since the expected time in one state is $\frac{1}{1-\xi}$, the expected request rate is $\frac{r}{\frac{1}{1-\xi}} = r(1 - \xi)$. A larger ξ means the rate of change of popularity is low, i.e., an algorithm that has accurate learning is desirable. This situation corresponds to a higher weight on 2-LRU-like behavior, and we choose a smaller β for A-LRU. The extreme case is $\xi = 1$, which the system stays in one state and follows a fixed Zipf-like popularity distribution. Here, we choose $\beta = 0$, i.e., A-LRU is equivalent to 2-LRU. The complementary argument applies if ξ is small. Here, the popularity changes quickly, and hence fast mixing is desirable at the cost of losing accuracy. Thus, LRU-like behavior is desirable and we choose β large.

Figure 5.22 compares the hit probability of A-LRU with LRU, 2-LRU and ARC under Markov Modulated requests, where we take $n = 1000$ and $r = 1000$. We see

Table 5.1: Relation between ξ and β.

ξ	0.1	0.3	0.5	0.7	0.9
Optimal β	0.4	0.3	0.2	0.1	0

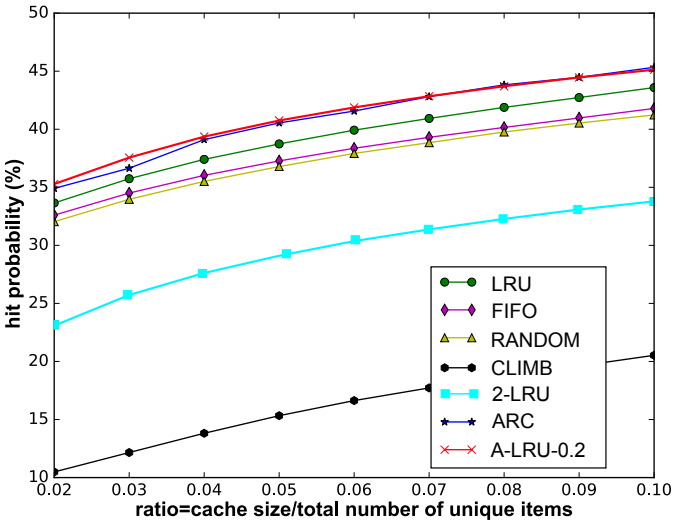

Figure 5.23: Hit probability vs. cache size, for various caching algorithms with two-week long YouTube trace [95].

that A-LRU is able to outperform all other algorithms for an appropriate choice of β. The relation between ξ and the optimal β for A-LRU is given in Table 5.1, which verifies the conclusion that β should decrease with increasing ξ.

5.3.5 Trace-based Simulations

The ideas presented thus far have been based on the hypothesis that the request distribution changes dynamically, and hence an optimal caching algorithm should track the changes at a time scale consistent with the time scale of change.

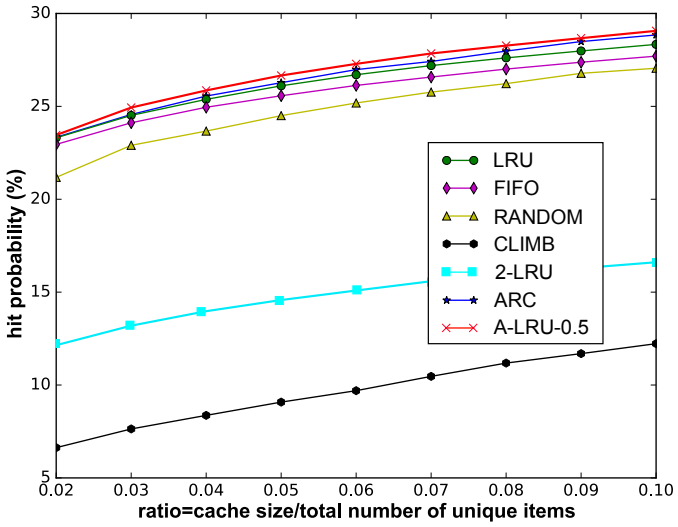

Figure 5.24: Hit probability vs. cache size, for various caching algorithms with one particular day YouTube trace [95].

5.3.5.1 YouTube Trace

We use the same data trace [95] that was used in Section 5.2.6. We find that $\alpha = 0.605$ is the best fit for a Zipf-like distribution. However, a detailed inspection shows that this trace exhibits significant non-stationarity, i.e., the popularity distribution is time-varying.

We compare the hit performance of different algorithms by varying cache size. Figure 5.23 depicts the hit probability as a function of the cache size when the total number of unique videos is $n = 303,331$, and we make the use of the ratio $m/n = 0.01, \cdots, 0.10$. For ease of visualization, we only depict A-LRU with the optimal β, which outperforms all the other caching algorithms.

We also conduct experiments on an one-day YouTube trace We randomly pick one day from the two-week traces, in which the total number of uniques videos is 3×10^5 and the Zipf-like distribution parameter $\alpha = 0.48$ (but popularity varies with

Table 5.2: SD network trace overview [13]

Date	# Obj	# Req	1-timers	α
02/18	854241	3571125	68.30%	0.817
02/19	993711	4121865	68.66%	0.815
02/20	871565	3593373	69.27%	0.814
02/21	811827	3416817	67.61%	0.821

time). Figure 5.24 depicts the hit probability as a function of the cache ratio.

5.3.5.2 ICN Traces

We run similar experiments using the traces from the IRCache project [13], with attention on data gathered from the SD Network Proxy (the most loaded proxy to which end-users can connect) in Feb. 2013. A detailed study shows that such traces capture regional traffic and exhibit significant non stationaries due to daily traffic fluctuations. We only considered the traces in the 4 hours peak traffic periods in order to measure the performance expected in the busy hour. The characteristics of the traces are shown in Table 5.2.

We consider a basic cache network hierarchy [13], in which there is a core cache that serves 4 edge caches, which are loaded by the real traces of the same four hours of four consecutive days, i.e., Feb. $18 + i$ trace loads the i-th edge cache, where $i = 0, \cdots, 3$. For simplicity, we assume that the size of edge caches are identical and equal to 1/10 of the core cache size. Figure 5.25 shows the overall cache hit probability versus the core cache size.

5.4 Conclusion

In this section, we first investigated the performance of a class of cache replacement algorithms in linear cache networks. We studied the stationary distributions of various replacement algorithms under the IRM model, and computed the τ-distance

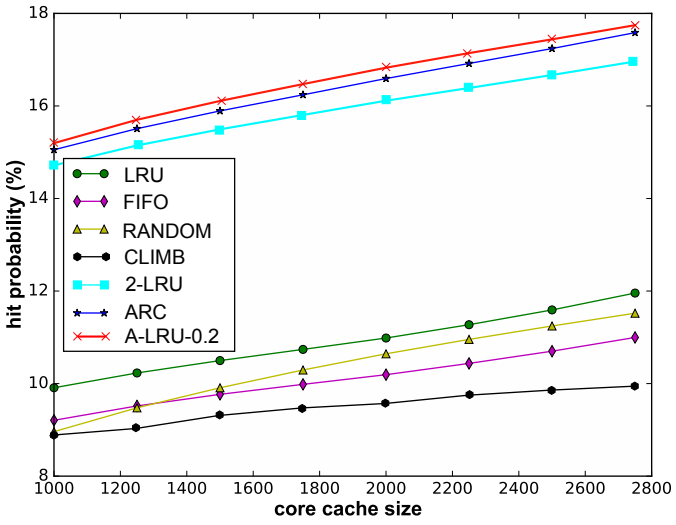

Figure 5.25: Hit probability vs cache size for various caching algorithms with SD network trace [13] for ICN.

between the stationary distributions and the best possible, which provided the insight that multiple caching levels with a cache partition that favors the higher levels promotes accuracy. We then analyzed the mixing time of each algorithm under a fixed popularity distribution to determine how long it would take to converge to the stationary distribution. We found that a larger number of caching levels and more resources allocated to higher level caches increases the mixing time, i.e, the two objectives of learning accuracy and speed are at odds with each other. We conducted a YouTube trace-based simulations to test the performance of these algorithms with real-word inputs. We observed the tradeoff between accuracy and speed in this realistic setting, showing that cache network selection should be done with a clear idea of the time constants in the system in order to obtain optimal performance.

We then combined the ideas of our analysis to develop a new hybrid algorithm, A-LRU, that can be adapted to different non-stationary request processes and consequently has a higher hit probability than any of the standard algorithms that we

compared against under both synthetic and trace-based evaluation.

6. CONCLUSIONS

In this thesis, we explored equilibria in large scale networked systems from two areas: mean field games in large scale societal networks and dynamic adaptability of replacement algorithms in cache networks. More specifically, in each of the four preceding sections, we have analyzed and designed incentives and algorithms to achieve the desirable equilibria which benefits the system as a whole. However, several questions still remain. To conclude this thesis, we go back to each section and give a discussion on the possible directions for future research.

Mean Field Games in societal networks:

The two problems discussed in Section 2 and Section 3 indicate the value of the mean field game approach towards modeling and analysis of large scale societal networks. In both problems, the desire was to steer the system towards an equilibrium that benefits society. However, fundamental questions remain on both the convergence to and selection of the MFE.

- *Convergence to MFE*: In both the results presented, we used a fixed point approach to show the existence of an MFE. However, we have not characterized the convergence of the state distribution of agents to the mean field equilibrium. In our simulations, we presented an intuitive set of dynamics that appear to have the right properties for convergence. The dynamics took the form of providing the empirical distribution of state to each agent, which then takes a best response assuming that distribution would apply for all future time. The empirical distribution is updated and the cycle begins again. Such dynamics are simple, and appear to converge quickly to an MFE. We would like to show analytically that such dynamics would indeed possess convergence properties.

- *Selection of MFE*: In the problem of providing incentives for demand-response in the smart grid setting, we chose to provide coupons at different times of day to encourage customers to utilize energy at certain times of day. This selection was done heuristically, and we showed numerically that the resulting MFE is desirable form the perspective fo the LSE due to reduced hazard. However, the question arises as to how to steer MFE in a given direction, and the cost of doing so. In other words, the question is whether we can determine the difference in overall utility at MFE as a function of the structure of the incentives provided, hence characterizing the tradeoff between the cost of such incentives and the value of the MFE attained.

Dynamic adaptability of caching algorithms:

For the problems discussed in Section 4 and Section 5, we observe that allowing for multiple levels with sizes determined using a probability distribution, appropriately projected to yield integer allocations, A-LRU yields a suite of caching algorithms that can smoothly transition from LRU to LFU[1], which is ∞-LRU. We conjecture that given any finite number of requests t, within this suite of algorithms we can find at least one that will yield the lowest possible learning error at t over all possible caching algorithms; we also expect that it would be sufficient to consider a finite number of levels, possibly $O(\log(t))$. Furthermore, we believe that it would be possible to find a sequence of finite level algorithms going from LRU to LFU that has performance arbitrarily close to the best possible learning error (infimum over all possible algorithms). Establishing these conjectures will be our future goal.

[1] [70] the Least Frequently Used policy statically stores the most popular m items in the cache (assuming their popularity is known). LFU is known to be optimal under IRM.

REFERENCES

[1] Electric Reliability Council of Texas (ERCOT). Data Set Available at `http://www.ercot.com/`.

[2] Network Coding Utilities. Library Available at `http://arni.epfl.ch/software`.

[3] Pecan Street. Data Set Available at `https://dataport.pecanstreet.org/`.

[4] N. Abedini, S. Sampath, R. Bhattacharyya, S. Paul, and S. Shakkottai. Realtime Streaming with Guaranteed QoS over Wireless D2D Networks. In *Proc. of* ACM MOBIHOC 2013, Bangalore, India, July 2013.

[5] M. H. Albadi and E. F. El-Saadany. A Summary of Demand Response in Electricity Markets. *Electric Power Systems Research*, 78(11):1989–1996, 2008.

[6] H. Allcott and J. B. Kessler. The Welfare Effects of Nudges: A Case Study of Energy Use Social Comparisons. Technical report, National Bureau of Economic Research, 2015.

[7] C. Aperjis and R. Johari. A Peer-to-Peer System as an Exchange Economy. In *Proc. of GameNets*, Pisa, Italy, October 2006.

[8] S. Athey and I. Segal. An Efficient Dynamic Mechanism. *Econometrica*, 81(6):2463–2485, 2013.

[9] O. I. Aven, E. G. Coffman, and Y. A. Kogan. *Stochastic Analysis of Computer Storage*. Springer Science & Business Media, 1987.

[10] M. Benaïm and J.-Y. Le Boudec. A Class of Mean Field Interaction Models for Computer and Communication Systems. *Performamce Evaluation*, 65(11-12):823–838, November 2008.

[11] D. Bergemann and J. Välimäki. The Dynamic Pivot Mechanism. *Econometrica*, 78(2):771–789, 2010.

[12] D. S. Berger, P. Gland, S. Singla, and F. Ciucu. Exact Analysis of TTL Cache Networks. *Performance Evaluation*, 79:2–23, 2014.

[13] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi. Check Before Storing: What is the Performance Price of Content Integrity Verification in LRU Caching? *ACM SIGCOMM Computer Communication Review*, 43(3):59–67, 2013.

[14] P. Billingsley. *Convergence of Probability Measures*. John Wiley & Sons, 2013.

[15] E. Bitar. Coordinated Aggregation of Distributed Demand-Side Resources, 2015. `http://www.news.cornell.edu`.

[16] V. Borkar and R. Sundaresan. Asymptotics of the Invariant Measure in Mean Field Models with Jumps. *Stochastic Systems*, 2(2):322–380, 2012.

[17] D. S. Callaway. Tapping the Energy Storage Potential in Electric Loads to Deliver Load Following and Regulation with Application to Wind Energy. *Energy Conversion and Management*, 50(5):1389–1400, 2009.

[18] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *Proc. of ACM IMC*, San Diego, CA, October 2007.

[19] 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)*, 17(5):1357–1370, 2009.

[20] H. Che, Y. Tung, and Z. Wang. Hierarchical Web Caching Systems: Modeling, Design and Experimental Results. *Selected Areas in Communications, IEEE Journal on*, 20(7):1305–1314, 2002.

[21] J. Cheeger. A Lower Bound for the Smallest Eigenvalue of the Laplacian. *Problems in analysis*, 625:195–199, 1970.

[22] S. Chen and J.-S. Wang. Tax Evasion and Fraud Detection: A Theoretical Evaluation of Taiwan's Business Tax Policy for Internet Auctions. *Asian Social Science*, 6(12):23, 2010.

[23] F. Chung. Laplacians and the Cheeger Inequality for Directed Graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.

[24] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. *ACM SIGCOMM Computer Communication Review*, 32(4):347–356, 2002.

[25] E. G. Coffman and P. J. Denning. *Operating Systems Theory*. Prentice-Hall Englewood Cliffs, NJ, 1973.

[26] B. Cohen. Incentives Build Robustness in BitTorrent. In *Proc. of WEIS*, Berkeley, CA, June 2003.

[27] A. Dan and D. Towsley. An Approximate Analysis of the LRU and FIFO Buffer Replacement Schemes. In *Proc. of ACM Sigmetrics*, Boulder, CO, May 1990.

[28] S. Deb, M. Médard, and C. Choute. Algebraic Gossip: A Network Coding Approach to Optimal Multiple Rumor Mongering. *IEEE Trans. on Information Theory*, 52(6):2486–2507, 2006.

[29] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top K Lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[30] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley. Analysis of TTL-based Cache Networks. In *Proc. of IEEE VALUETOOLS*, Cargese, France, October 2012.

[31] C. Fricker, P. Robert, J. Roberts, and N. Sbihi. Impact of Traffic Mix on Caching Performance in a Content-Centric Network. In *Prof. of INFOCOM WKSHPS*, Orlando, FL, March 2012.

[32] S. Gao, E. Frejinger, and M. Ben-Akiva. Adaptive Route Choices in Risky Traffic Networks: A Prospect Theory Approach. *Transportation research part C: emerging technologies*, 18(5):727–740, 2010.

[33] N. Gast and B. Van Houdt. Transient and Steady-state Regime of a Family of List-based Cache Replacement Algorithms. In *Proc. of ACM SIGMETRICS*, Portland, OR, June 2015.

[34] N. Gast and B. Van Houdt. Asymptotically Exact TTL-approximations of the Cache Replacement Algorithms LRU(m) and h-LRU. *Preprint HAL Open Archive hal-01292269*, 2016.

[35] E. Gelenbe. A Unified Approach to the Evaluation of a Class of Replacement Algorithms. *IEEE Transactions on Computers*, 100(6):611–618, 1973.

[36] C. Graham and S. Méléard. Chaos Hypothesis for a System Interacting Through Shared Resources. *Probability Theory and Related Fields*, 100(2):157–174, 1994.

[37] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent. Aggregate Flexibility of Thermostatically Controlled Loads. *IEEE Transactions on Power Systems*, 30(1):189–198, 2015.

[38] M. Hao and L. Xie. Analysis of Coupon Incentive-Based Demand Response with Bounded Consumer Rationality. In *Proc. of NAPS*, Pullman, WA, September 2014.

[39] G. W. Harrison and E. E. Rutström. Expected Utility Theory and Prospect Theory: One Wedding and a Decent Funeral. *Experimental Economics*, 12(2):133–158, 2009.

[40] W. J. Hendricks. An Account of Self-Organizing Systems. *SIAM Journal on Computing*, 5(4):715–723, 1976.

[41] J. H. Hester and D. S. Hirschberg. Self-Organizing Linear Search. *ACM Computing Surveys (CSUR)*, 17(3):295–311, 1985.

[42] I-H. Hou, V. Borkar, and P. R. Kumar. A Theory of QoS for Wireless. In *Proc. of IEEE INFORM*, Rio de Janeiro, Brazil, April 2009.

[43] I-H. Hou, Y. Liu, and A. Sprintson. A Non-Monetary Protocol for Peer-to-Peer Content Distribution in Wireless Broadcast Networks with Network Coding. In *Proc. of WiOpt*, Tsukuba Science City, Japan, May 2013.

[44] M. Huang, R. P. Malhamé, and P. E. Caines. Large Population Stochastic Dynamic Games: Closed-Loop Mckean-Vlasov Systems and the Nash Certainty Equivalence Principle. *Communications in Information & Systems*, 6(3):221–252, 2006.

[45] D. R. Hunter. MM Algorithms for Generalized Bradley-Terry Models. *Annals of Statistics*, 32:384–406, 2004.

[46] K. Iyer, R. Johari, and M. Sundararajan. Mean Field Equilibria of Dynamic Auctions with Learning. *ACM SIGecom Exchanges*, 10(3):10–14, 2011.

[47] K. Iyer, R. Johari, and M. Sundararajan. Mean Field Equilibria of Dynamic Auctions with Learning. *Management Science*, 60(12):2949–2970, 2014.

[48] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi. Orchestrating Massively Distributed CDNs. In *Proc. of ACM CoNEXT*, Nice, France, December 2012.

[49] B. Jovanovic and R. W. Rosenthal. Anonymous Sequential Games. *Journal of Mathematical Economics*, 17(1):77–87, February 1988.

[50] D. Kahneman and A. Tversky. Prospect Theory: An Analysis of Decision under Risk. *Econometrica: Journal of the Econometric Society*, pages 263–291, 1979.

[51] D. Kahneman and A. Tversky. Choices, Values, and Frames. *American psychologist*, 39(4):341, 1984.

[52] V. Kavitha, E. Altman, R. El-Azouzi, and R. Sundaresan. Fair Scheduling in Cellular Systems in the Presence of Noncooperative Mobiles. *IEEE/ACM Transactions on Networking*, 22(2):580–594, April 2014.

[53] F. P. Kelly. *Reversibility and Stochastic Networks*. Cambridge University Press, 2011.

[54] F. P. Kelly and E. Yudovina. *Stochastic Networks*. Cambridge University Press, 2014.

[55] W. F. King-III. Analysis of Demanding Paging Algorithms. In *Proc. of IFIP Congress*, Ljubljana, Yugoslavia, August 1971.

[56] V. Krishna. *Auction Theory*. Academic Press, MA, U.S.A, 1997.

[57] R. Kumar and S. Vassilvitskii. Generalized Distances Between Rankings. In *Proc. of ACM WWW*, Raleigh, NC, April 2010.

[58] J. M. Lasry and P. L. Lions. Mean Field Games. *Japanese Journal of Mathematics*, 2(1):229–260, 2007.

[59] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Soc., 2009.

[60] J. Li, R. Bhattacharyya, S. Paul, S. Shakkottai, and V. Subramanian. Incentivizing Sharing in Realtime D2D Streaming Networks: A Mean Field Game Perspective. In *Proc. of IEEE INFOCOM*, Hong Kong, 2015.

[61] J. Li, R. Bhattacharyya, S. Paul, S. Shakkottai, and V. Subramanian. Incentivizing Sharing in Realtime D2D Streaming Networks: A Mean Field Game Perspective. *arXiv preprint arXiv:1604.02435*, 2016.

[62] J. Li, R. Bhattacharyya, S. Paul, S. Shakkottai, and V. Subramanian. Incentivizing Sharing in Realtime D2D Streaming Networks: A Mean Field Game Perspective. *IEEE/ACM Transactions on Networking*, to appear in 2016.

[63] J. Li, J. Wu, G. Dan, A. Arvidsson, and M. Kihl. Performance Analysis of Local Caching Replacement Policies for Internet Video Streaming Services. In *Proc. of SoftCOM*, Split, Croatia, September 2014.

[64] J. Li, B. Xia, X. Geng, M. Hao, S. Shakkottai, V. Subramanian, and X. Le. Energy Coupon: A Mean Field Game Perspective on Demand Response in Smart Grids. In *Proc. of ACM SIGMETRICS*, Portland, OR, June 2015.

[65] T. Li and N. B. Mandayam. Prospects in a Wireless Random Access Game. In *Proc. of CISS*, Princeton, NJ, March 2012.

[66] T. Lindvall. *Lectures on the Coupling Method*. Courier Corporation, 2002.

[67] P. Loiseau, G. A. Schwartz, J. Musacchio, S. Amin, and S. S. Sastry. Incentive Mechanisms for Internet Congestion Management: Fixed-Budget Rebate Versus Time-of-Day Pricing. *IEEE/ACM Transactions on Networking*, 22(2):647–661, April 2014.

[68] J. A. Lozano and E. Irurozki. Probabilistic Modeling on Rankings, 2012. available at `http://www.sc.ehu.es/ccwbayes/members/ekhine/tutorial_`

ranking/info.html.

[69] M. Manjrekar, V. Ramaswamy, and S. Shakkottai. A Mean Field Game Approach to Scheduling in Cellular Systems. In *Proc. of IEEE INFOCOM*, Toronto, Canada, April 2014.

[70] V. Martina, M. Garetto, and E. Leonardi. A Unified Approach to the Performance Analysis of Caching Systems. In *Proc. of IEEE INFOCOM*, Toronto, Canada, April 2014.

[71] N. Megiddo and D. S. Modha. ARC: A Self-Tuning, Low Overhead Replacement Cache. In *Proc. of FAST*, San Francisco, CA, April 2003.

[72] D. Merugu, B. S. Prabhakar, and N. S. Rama. An Incentive Mechanism for Decongesting the Roads: A Pilot Program in Bangalore. In *Proc. of NetEcon*, Stanford, CA, July 2009.

[73] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer Science & Business Media, 2012.

[74] M. Mihail. Conductance and Convergence of Markov Chains-A Combinatorial Treatment of Expanders. In *Proc. of IEEE FOCS*, Research Triangle Park, NC, October 1989.

[75] R. R. Montenegro and P. Tetali. *Mathematical Aspects of Mixing Times in Markov Chains*. Now Publishers Inc, 2006.

[76] J. Naritomi. *Consumers as Tax Auditors*. working paper, International Development Department and Institute of Public Affairs, London School of Economics, 2013.

[77] M. Poco, C. Lopes, and A. Silva. *Perception of Tax Evasion and Tax Fraud in Portugal: A Sociological Study*. working paper, 2015.

[78] B. Prabhakar. Designing Large-Scale Nudge Engines. In *Proc. of ACM SIGMETRICS*, Pittsburgh, PA, June 2013.

[79] D. Prelec. The Probability Weighting Function. *Econometrica*, 66:497–527, 1998.

[80] M. L. Puterman. *Markov Ddecision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[81] T. Qin, X. Geng, and T. Liu. A New Probabilistic Model for Rank Aggregation. In *Proc. of NIPS*, Vancouver, Canada, December 2010.

[82] E. J. Rosensweig, J. Kurose, and D. Towsley. Approximate Models for General Cache Networks. In *Proc. of IEEE INFOCOM*, San Diego, CA, March 2010.

[83] G. A. Schwartz, H. Tembine, S. Amin, and S. S. Sastry. Electricity Demand Shaping via Randomized Rewards: A Mean Field Game Approach. In *Allerton Conference on Communication, Control, and Computing*, Allerton, IL, September 2012.

[84] A. Sinclair. Improved Bounds for Mixing Rates of Markov Chains and Multicommodity Flow. *Combinatorics, probability and Computing*, 1(04):351–370, 1992.

[85] D. Starobinski and D. Tse. Probabilistic Methods for Web Caching. *Performance evaluation*, 46(2):125–137, 2001.

[86] H. Thorisson. Coupling Methods in Probability Theory. *Scandinavian journal of statistics*, 22:159–182, 1995.

[87] A. Tversky and D. Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 211(4481):453–458, 1981.

[88] A. Tversky and D. Kahneman. Advances in Prospect Theory: Cumulative Representation of Uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[89] C. Villani. *Optimal Transport: Old and New*. Springer Science & Business Media, 2008.

[90] Y. Wang, W. Saad, N. B. Mandayam, and H. V. Poor. Integrating Energy Storage into the Smart Grid: A Prospect Theoretic Approach. In *Proc. of ICASSP*, Florence, Italy, May 2014.

[91] L. Xiao, N. B. Mandayam, and H. V. Poor. Prospect Theoretic Analysis of Energy Exchange Among Microgrids. *IEEE Transactions on Smart Grid*, 6(1):63–72, January 2015.

[92] J. Yu, M. H. Cheung, and J. Huang. Spectrum Investment with Uncertainty Based on Prospect Theory. In *Proc. of ICC*, Sydney, Australia, June 2014.

[93] T. Yu, Z. Zhou, D. Zhang, X. Wang, Y. Liu, and S. Lu. INDAPSON: An Incentive Data Plan Sharing System Based on Self-Organizing Network. In *Proc. of IEEE INFOCOM*, Toronto, Canada, April 2014.

[94] H. Zhong, L. Xie, and Q. Xia. Coupon Incentive-Based Demand Response: Theory and Case Study. *IEEE Transactions on Power Systems*, 28(2):1266–1276, May 2013.

[95] M. Zink, K. Suh, Y. Gu, and J. Kurose. Watch Global, Cache Local: YouTube Network Traffic at A Campus Network: Measurements and Implications. In *Electronic Imaging*, 2008.

APPENDIX A

PROOFS FROM SECTION 2

A.1 Properties of Allocation Scheme

A.1.1 Proof of Lemma 1

Given the B2D arrivals $(e_1[k], ..., e_M[k])$, we partition the set of devices $\{1, ..., M\}$ into sets $\mathcal{S}$ and $\mathcal{S}^c = \{1, ..., M\}\backslash\mathcal{S}$, based on whether $e_i[k]+T-N \geq 0$ or not. Those agents that satisfy this condition can potentially receive enough chunks during the D2D phase that they can decode the block, whereas the others cannot. Hence, all members of $\mathcal{S}^c$ can potentially transmit their chunks in the allocation solving (2.13). Let $T_1 = \min\{\sum_{i\in\mathcal{S}^c} e_i[k],\ \mathrm{T}\}$. So we can devote the first T_1 slots of the current frame to transmissions from the devices in $\mathcal{S}^c$.

Let the number of transmissions made by agent i in allocation $\mathbf{a}$ be denoted by $x_i[k]$. We can write down the constraints that any feasible allocation $\mathbf{a}$ must satisfy as

$$\begin{aligned} &0 \leq x_i[k] \leq e_i[k] \qquad \forall\ i \in \mathcal{S} \\ &\textstyle\sum_{i\in\mathcal{S}} x_i[k] = T - T_1 \end{aligned} \tag{A.1}$$

Observe that each agent can transmit $e_i[k] + T - N$ chunks without affecting the above constraints (*i.e.,* it does not change its chances of being able to decode the block, as there is enough time left for it to receive chunks that it requires). We call these as "extra" chunks. Suppose that all extra chunks have been transmitted by time $T_2 < T$, and no device has yet reached full rank. At this point, all agents in the system need the same number of chunks, and any agent that transmits a chunk will

not be able to receive enough chunks to decode the block. In other words, agents now have to "sacrifice" themselves one at a time, and transmit all their chunks. The question is, what is the order in which such sacrifices should take place?

Compare two agents i and j, with deficits $d_i > d_j$. Also, let $\chi \in \{0, 1\}$. Now, for either value of χ

$$d_i - (d_i - \chi)^+ \geq d_j - (d_j - \chi)^+.$$

Hence, since $c(.)$ is convex and monotone increasing,

$$\int_{(d_i-\chi)^+}^{d_i} c'(z)dz \geq \int_{(d_j-\chi)^+}^{d_j} c'(z)dz \geq 0 \tag{A.2}$$

$$\Rightarrow \quad c(d_i) - c((d_i - \chi)^+) \geq c(d_j) - c((d_j - \chi)^+) \geq 0. \tag{A.3}$$

Now, consider the following problem with $\chi_i, \chi_j \in \{0, 1\}$ under the constraint $\chi_i + \chi_j = 1$:

$$\min_{\chi_i,\chi_j} c(d_i - \chi_i) + c(d_j - \chi_j). \tag{A.4}$$

$$\Leftrightarrow \quad \max_{\chi_i,\chi_j} c(d_i) - c(d_i - \chi_i) + c(d_j) - c(d_j - \chi_j). \tag{A.5}$$

Then, from the above discussion, the solution is to set $\chi_i = 1$ and $\chi_j = 0$. Thus, comparing (A.4) and (2.13), the final stage of the allocation should be for agents to sacrifice themselves according to a min-deficit-first type policy. Algorithm 1 describes the final allocation rule.

A.2 Properties of Mechanisms

A.2.1 Proof of Theorem 1

The net-cost in frame k for agent i when reporting $\theta_i[k]$ versus $r_i[k]$ is given by

$$\begin{aligned}
&V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}_{s_{j_i[k]}}[k]), \theta_i[k]) - p^*(\theta_i[k], \hat{\boldsymbol{\theta}}_{-i}[k]) \\
&= \tilde{W}(i, (\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])) - H(\hat{\boldsymbol{\theta}}_{-i}[k]) \\
&\leq \tilde{W}(i, (r_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])) - H(\hat{\boldsymbol{\theta}}_{-i}[k]) \\
&= V(\mathbf{a}^*((r_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])), \theta_i[k]) - p^*(r_i[k], \hat{\boldsymbol{\theta}}_{-i}[k]), \qquad \text{(A.6)}
\end{aligned}$$

where θ_i is the true type and r_i is an arbitrary type; the equalities hold true due to the definition of value function and transfer; the last inequality follows by the optimality of allocation $\tilde{\mathbf{a}}(\theta_i, \hat{\boldsymbol{\theta}}_{-i})$ in cluster $s_{j_i[k]}$ maximizes the system utility from the perspective of agent i. Therefore, in every frame it is best for agent i to report truthfully and this holds irrespective of the reports of the other agents.

A.3 Nature of Transfers

A.3.1 Proof of Lemma 2

From (2.11), we have

$$\begin{aligned}
p^*(\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k]) &= V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]), \hat{\theta}_i[k]) + H(\hat{\boldsymbol{\theta}}_{-i}[k]) - \tilde{W}(i, (\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])) \\
&\quad + \tilde{W}_{-i}(i, (\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])) - \tilde{W}(i, (\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])) \\
&\overset{(a)}{\geq} V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]), \hat{\theta}_i[k]) - V(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \hat{\theta}_i[k]) \\
&\overset{(b)}{\geq} 0,
\end{aligned}$$

where (a) follows from the definition of allocation $\tilde{\mathbf{a}}$ and the inequality (b) is true by the monotonicity argument below.

We assume that under both systems (with the allocations $\mathbf{a}_{-i}$ and $\mathbf{a}^*$), the deficits are initialized with the same value. Also note that all the agents follow the same reporting strategy in frame k, and hence, $\chi(\mathbf{a}^*)$ and $\chi(\mathbf{a}_{-i})$ can be compared. Under allocation $\mathbf{a}_{-i}$, agent i never transmits and will pick up free chunks from other agents' transmissions. However, agent i may have to transmit under allocation $\mathbf{a}^*$. Thus, we have

$$\chi_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]), \theta_i[k]) \leq \chi_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \theta_i[k]), \tag{A.7}$$

as $e_i[k] + g_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k])) \leq e_i[k] + g_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]))$ is true for every k.

Using this we can compare the two deficits by considering the same allocation policy. For $k \geq 0$, we have

$$d_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k])) = \big(d_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k-1])) + \eta - \chi_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]), \theta_i[k])\big)^+, \tag{A.8}$$

$$d_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k])) = \big(d_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k-1])) + \eta - \chi_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \theta_i[k])\big)^+,$$

with $\chi_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]), \theta_i[k]) \leq \chi_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \theta_i[k])$ for all k, which implies that $d_i(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]))$ $\geq d_i(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]))$. Since the function $V(\cdot,\cdot)$ in (2.10) can be obtained by value iteration starting with $v(\cdot)$, then by the definition of value function $v(\cdot)$ and the monotonicity of holding cost function $c(\cdot)$ in d, we have $V(\cdot,\cdot)$ being an increasing function in d. Then it directly follows that

$$V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}[k]), \hat{\theta}_i[k]) \geq V(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \hat{\theta}_i[k]), \tag{A.9}$$

which completes our proof.

A.3.2 Proof of Lemma 3

The net-cost in frame k for agent i is given by

$$\begin{aligned} &V(\mathbf{a}^*(\hat{\boldsymbol{\theta}}_{s_{j_i[k]}}[k]), \theta_i[k]) - p^*(\theta_i[k], \hat{\boldsymbol{\theta}}_{-i}[k]) = V(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \theta_i[k]) \\ &- [\tilde{W}_{-i}(i, (\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k])) - \tilde{W}(i, (\hat{\theta}_i[k], \hat{\boldsymbol{\theta}}_{-i}[k]))] \\ &\leq V(\mathbf{a}_{-i}(\hat{\boldsymbol{\theta}}_{-i}[k]), \theta_i[k]), \end{aligned} \tag{A.10}$$

where we use the same logic as point (a) in (A.7).

A.4 Properties of the Optimal Value Function

A.4.1 Proof of Theorem 2

First, we consider statement 1). The proof follows by applying Theorem 6.10.4 in Puterman [80], and verifying the Assumptions 6.10.1, 6.10.2 and Propositions 6.10.1, 6.10.3.

Define the set of functions

$$\Phi = \left\{ w : (\mathbb{K}, \mathcal{T})^M \to \mathbb{R}^+ : \sup_{\boldsymbol{\theta} \in (\mathbb{K}, \mathcal{T})^M} \left| \frac{w(\boldsymbol{\theta})}{\alpha(\boldsymbol{\theta})} \right| < \infty \right\}, \tag{A.11}$$

where $\alpha(\boldsymbol{\theta}) = \max\{\sum_i^M v_i(a^*(\boldsymbol{\theta}), \theta_i), 1\}$. Note that Φ is a Banach space with α-norm,

$$||w||_\alpha = \sup_{\boldsymbol{\theta} \in (\mathbb{K}, \mathcal{T})^M} \left| \frac{w(\boldsymbol{\theta})}{\alpha(\boldsymbol{\theta})} \right| < \infty. \tag{A.12}$$

Also define the operation T_1 as

$$T_1 w(\boldsymbol{\theta}) = \sum_{i=1}^{M} v_i(\mathbf{a}^*(\boldsymbol{\theta}), \theta_i) + \delta \mathbb{E}\left\{ w(\boldsymbol{\Theta}) \right\}, \tag{A.13}$$

where $w \in \Phi$.

First, we need to show that for $\forall w \in \Phi$, $T_1 w \in \Phi$. From Equation (A.13) and the definition of value functions, we know the sum of all users' values are bounded, say $\sum_{i=1}^{M} v_i(\mathbf{a}^*(\boldsymbol{\theta}), \theta_i) \leq A$. Then we have

$$||T_1 w||_\alpha \leq A + \delta \mathbb{E}\left\{w(\boldsymbol{\Theta})\right\}, \tag{A.14}$$

where the rightside expression is bounded by the sum of A and some multiple of $||w||_\alpha$. Hence, $T_1 w \in \Phi$.

Next, we need to verify Assumptions 6.10.1 and 6.10.2 in Puterman [80]. Our theorem requires the verification of the following three conditions. Let $\boldsymbol{\Theta}[k]$ be the random variable denoting the current system state at frame k, where $\boldsymbol{\Theta}[k] = (\boldsymbol{d}[k-1], \boldsymbol{e}[k])$. Then we must show that $\forall \theta \in (\mathbb{K}, \mathcal{T})^M$, for some constants $0 < \gamma_1 < \infty$, $0 < \gamma_2 < \infty$ and $0 < \gamma_3 < 1$,

$$\sup_{a \in A} |\sum_{i}^{M} v_i(a^*(\boldsymbol{\theta}), \theta_i)| \leq \gamma_1 \alpha(\theta), \tag{A.15}$$

$$\mathbb{E}_{\boldsymbol{\theta}[1]}[w(\boldsymbol{\theta}[1])|\boldsymbol{\theta}[0] = \theta] \leq \gamma_2 \alpha(\theta), \quad \forall w \in \Phi \tag{A.16}$$

$$\beta^k \mathbb{E}_{\boldsymbol{\theta}[k]}[\alpha(\boldsymbol{\theta}[k])|\boldsymbol{\theta}[0] = \theta] \leq \gamma_3 \alpha(\theta), \quad \text{for some k} \tag{A.17}$$

(B.1) holds from the definition of $\alpha(\boldsymbol{\theta}) = \max\{\sum_i^M v_i(a^*(\boldsymbol{\theta}), \theta_i), 1\}$.

(B.2) holds true since

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\theta}[1]}[w(\boldsymbol{\theta}[1])|\boldsymbol{\theta}[0] = \theta] &\leq ||w||_\alpha \times \mathbb{E}_{\boldsymbol{\theta}[1]}[\alpha(\boldsymbol{\theta}[1])|\boldsymbol{\theta}[0] = \theta] \\ &\leq ||w||_\alpha \times \gamma_2' \alpha(\theta), \quad \text{for some large enough } \gamma_2' \\ &= \gamma_2 \times \alpha(\theta), \end{aligned} \tag{A.18}$$

as we know in our mean field model, $\boldsymbol{\theta}[1]$ are all drawn i.i.d. from the given distribution $[\otimes\rho^M, \otimes\zeta^M]$, with ρ pertaining to the deficit, and ζ pertaining to the B2D transmissions received by that agent, so the first inequality holds in (A.18).

Finally, we have (B.3) since,

$$\begin{aligned}\beta^k \mathbb{E}_{\boldsymbol{\theta}[k]}[\alpha(\boldsymbol{\theta}[k])|\boldsymbol{\theta}[0]=\theta] &= \beta^k \mathbb{E}_{\boldsymbol{\theta}[k]}[\sum_i^M v_i(a^*(\boldsymbol{\theta}[k]),\theta_i)|\boldsymbol{\theta}[0]=\theta] \\ &\leq \beta^j \times \gamma_3' \alpha(\boldsymbol{\theta}) \\ &= \gamma_3 \alpha(\boldsymbol{\theta}).\end{aligned} \tag{A.19}$$

The first equality holds from the definition of $\alpha(\theta)$, and the first inequality holds true is because in our mean field mode, $\boldsymbol{\theta}[j]$ are all drawn i.i.d. from the given distribution $[\otimes\rho^M, \otimes\zeta^M]$, with ρ pertaining to the deficit, and ζ pertaining to the B2D transmissions received by that agent, so it's identical for all k.

Since we have verified all the three conditions required by Theorem 6.10.4 in Puterman, Statement 1) holds true.

For statement 2), we can use the same argument as the above proof to show the existence of fix point. We omit the details here. The last part of Theorem 2 follows from the discussion before the statement of this theorem.

A.5 The Existence and Uniqueness of Stationary Surplus Distribution

A.5.1 Proof of Lemma 4

First, from (2.16), we note the Doeblin condition, namely,

$$\mathbb{P}(d_i[k] \in B | d_i[k-1] = d, e_i[k] = e, \mathbf{a}) \geq (1-\delta)\Psi(B), \tag{A.20}$$

where $0 < \delta < 1$ and Ψ is a probability measure. Then following the results in Chapter 12 of [73], the Markov chain with transition probabilities in (2.16) is positive Harris recurrent and has a unique stationary distribution.

Next, let $-\tau$ be the last time before 0 that regeneration happened. We have

$$\begin{aligned}\Pi_{\rho\times\zeta}(B) &= \sum_{k=0}^{\infty}\mathbb{P}(B,\tau=k)\\ &= \sum_{k=0}^{\infty}\mathbb{P}(B|\tau=k)\mathbb{P}(\tau=k).\end{aligned} \tag{A.21}$$

Since the regeneration happens independently of the deficit queue with inter regeneration times geometrically distributed with parameter $(1-\delta)$, it follows that $\mathbb{P}(\tau=k)=(1-\delta)\delta^k$. Hence

$$\begin{aligned}\Pi_{\rho\times\zeta}(B) &= \sum_{k=0}^{\infty}(1-\delta)\delta^k\mathbb{P}(D[0]\in B|\tau=k)\\ &= \sum_{k=0}^{\infty}(1-\delta)\delta^k\mathbb{E}(\mathbb{E}(1_{\{D[0]\in B\}}|\tau=k,D_{-k}=D,E)|\tau=k)\\ &= \sum_{k=0}^{\infty}(1-\delta)\delta^k\mathbb{E}(\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E)|\tau=k)\\ &= \sum_{k=0}^{\infty}(1-\delta)\delta^k\mathbb{E}_{\Psi}(\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E)),\end{aligned} \tag{A.22}$$

where the last equality holds since $D_{-k}\sim\Psi$ given $\tau=k$.

A.6 Existence of MFE

A.6.1 Proof of Lemma 5

We will establish the second property first. We're given a sequence $\{\sigma^n\}_{n\in\mathbb{N}}\subset\mathcal{C}$ that converges point-wise to σ; it obviously follows that $\sigma\in\mathcal{C}$ even with point-wise convergence so that we are, in fact, showing that $\mathcal{C}$ is closed in l_∞ too. Since

$\lim_{n\to\infty} b_n = 0$, given $\epsilon > 0$, there exists [1] N such that for all $n > N$, $b_n \leq \epsilon/2$ so that $\sup_{k\in\mathbb{N}} |\sigma_n^k| \leq b_n \leq \epsilon/2$ too. Since $\lim_{k\to\infty} \sigma_n^k = \sigma_n$ for all $n = 1, \ldots, N$, we can find N_n such that for all $k > N_n$, $|\sigma_n^k - \sigma_n| \leq \epsilon$. Therefore, for $k > \max(N, \max_{n=1,\ldots,N} N_n)$

$$|\sigma_n^k - \sigma_n| \leq \begin{cases} \epsilon, & n = 1, \ldots, N \\ |\sigma_n^k| + |\sigma_n| \leq \epsilon, & n > N \end{cases} \tag{A.23}$$

so that $\|\sigma^k - \sigma\| \leq \epsilon$.

Since we have already established that $\mathcal{C}$ is closed in l_∞, it is sufficient to prove that it is totally bounded as well. Here we first find N such that for all $n > N$, $b_n \leq \epsilon$ so that $\sup_{k\in\mathbb{N}} |\sigma_n^k| \leq b_n \leq \epsilon$ too. Then from the compactness of $\prod_{n=1}^N [-b_n, b_n] \in \mathbb{R}^N$, we can find a finite number of points $\{v^1, v^2, \ldots, v^L\} \subset \prod_{n=1}^N [-b_n, b_n]$ such that $\prod_{n=1}^N [-b_n, b_n]$ is covered by balls of radius ϵ around v^l, $l = 1, \ldots, L$. Now we construct $\{\hat{v}^1, \ldots, \hat{v}^L\} \in \mathcal{C}$ as follows for $l = 1, \ldots, L$

$$\hat{v}_n^l = \begin{cases} v_n^l, & \text{if } n \leq N \\ 0, & \text{otherwise.} \end{cases} \tag{A.24}$$

By our choice of N, $\{\hat{v}^1, \ldots, \hat{v}^L\}$ is a finite cover of $\mathcal{C}$ with balls of radius ϵ, proving that $\mathcal{C}$ is totally bounded too.

A.6.2 Proof of Lemma 6

The proof will involve three steps. The first is to establish that $\Pi_{\rho\times\zeta}$ is indeed a probability distribution, which is obvious. The second is to establish that $\Pi_{\rho\times\zeta} \in \mathcal{M}_1(\mathbb{K})$, which will be carried out using induction by analyzing the properties of the Markov transition kernel of the deficit process without any regenerations. Finally,

[1] Note the abuse of notation only in this section to use N to represent a positive integer.

using stochastic dominance we will show that $\Pi_{p\times\zeta} \in \mathcal{P}(F)$.

From earlier Lemma 4, we know that

$$\Pi_{\rho\times\zeta}(B) = \sum_{k=0}^{\infty}(1-\delta)\delta^k \mathbb{E}_{\Psi}(\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E)). \tag{A.25}$$

Therefore, for our proof we will show that $\mathbb{E}_{\Psi}(\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|D,E)) \in \mathcal{M}_1(\mathbb{K})$. Since

$$\mathbb{E}_{\Psi}(\Upsilon^{(k)}_{\rho\times\zeta}(B|D,E)) = \int \Upsilon^{(k)}_{\rho\times\zeta}(B|d,e)d\Psi(d)d\zeta(e), \tag{A.26}$$

and $\Psi \in \mathcal{M}_1(\mathbb{K})$ and $\zeta \in \mathcal{M}_1(\mathcal{T})$, it is sufficient to show that $\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d,e) \in \mathcal{M}_1(\mathbb{K})$ for every $(d,e) \in (\mathbb{K},\mathcal{T})$.

Since $\Upsilon^{(0)}_{\rho\times\zeta}(\cdot|d,e)$ is a point-mass at d, the initial condition is satisfied. We now make the induction assumption that $\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d,e) \in \mathcal{M}_1(\mathbb{K})$ and show that this implies that $\Upsilon^{(k+1)}_{\rho\times\zeta}(\cdot|d,e) \in \mathcal{M}_1(\mathbb{K})$. Since $\Upsilon^{(k+1)}_{\rho\times\zeta}(\cdot|d,e)$ is a probability measure, we only need to show that its support is $\mathbb{K}$. By the definition of the Markov transition kernel without regenerations, we have

$$\Upsilon^{(k+1)}_{\rho\times\zeta}(B|d,e) = \int \sum_{j=0}^{1} 1_{\{(d'+\eta-j)_+\in B\}} p_j(d',e') d\Upsilon^{(k)}(d'|d,e)d\zeta(e'), \tag{A.27}$$

for some measurable functions $\{p_j(d',e')\}_{j=0,1}$ that account for the states of the other users being chosen independently using distribution $\rho\times\zeta$ and the greedy optimal allocation function $\mathbf{a}^*(\cdot)$. The assertion that $\Upsilon^{(k+1)}_{\rho\times\zeta}(\cdot|d,e) \in \mathcal{M}_1(\mathbb{K})$ follows since $d' \in \mathbb{K}$ and the only possible updates are an increase of the deficit to $d'+\eta$ or a decrease to either 0 or $d'+\eta-1$ (depending on value of d').

The deficit process for any given user is stochastically dominated by the fictitious process where the user is never allowed to decode the contents of a frame during his

lifetime, this is irrespective of his state or the state of the other users. Denote this process by $\{\tilde{D}_k\}_{k\in\mathbb{N}}$; it is easily discerned that the process takes values in $\mathbb{K}$. The transition kernel for this process is given by

$$\mathbb{P}(\tilde{D}_{k+1} = d|\tilde{D}_k = d') = \delta 1_{\{d=d'+\eta\}} + (1-\delta)\Psi(d). \tag{A.28}$$

Using the same proof as in Lemma 4, the invariant distribution $\tilde{\Pi}$ of the $\{\tilde{D}_k\}_{k\in\mathbb{N}}$ process is given by

$$\tilde{\Pi}(d) = \sum_{k=0}^{\infty}(1-\delta)\delta^k \sum_{d'\in\mathbb{K}} \Psi(d')1_{\{d'+k\eta=d\}}. \tag{A.29}$$

By the stochastic ordering property, the proof follows by noting that

$$\begin{aligned}\mathbb{E}_{\Pi_{\rho\times\zeta}}[D] &\leq \mathbb{E}_{\tilde{\Pi}}[D] \\ &\leq \sum_{k=0}^{\infty}(1-\delta)\delta^k \sum_{d'\in\mathbb{K}} \Psi(d')(d'+k\eta) \\ &< F' + \frac{\delta\eta}{1-\delta}.\end{aligned} \tag{A.30}$$

A.6.3 Proof of Theorem 6

We will start by showing that Π^* is continuous in the topology of point-wise convergence. For this we will use the coupling from Theorem 5 to establish convergence in total variation norm of the Markov transition kernels of the deficit process without any regenerations. Then using Lemma 5 we can strengthen the topology to complete the proof of the first part. The fixed point result then follows from the Schauder fixed point theorem after noting both the convexity and compactness of $\mathcal{P}(F)$.

To establish the continuity of Π^* in the topology of point-wise convergence, we will start by proving that the Markov transition kernels without regeneration

$\{\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d,e)\}_{k=0}^{\infty}$ are continuous in the topology of point-wise convergence. Since $\Upsilon^{(0)}_{\rho\times\zeta}(\cdot|d,e)$ is a point-mass at d irrespective of $\rho \in \mathcal{P}(F)$, the continuity assertion holds. In fact, for all $n \geq 1$ and $d' \in \mathbb{K}$, $\Upsilon^{(0)}_{\rho_n\times\zeta}(d'|d,e) = \Upsilon^{(0)}_{\rho\times\zeta}(d'|d,e)$. Let $\{\rho_n\}_{n\in\mathbb{N}} \subset \mathcal{P}(F)$ be a sequence converging point-wise [2] to $\rho \in \mathcal{P}(F)$. We will show that $\Upsilon^{(k)}_{\rho_n\times\zeta}(\cdot|d,e)$ converges point-wise to $\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d,e)$ for all $k \in \mathbb{N}$. We will prove this by induction.

We will refer to any measures and random variables corresponding to ρ_n as coming from the n^{th} system and those corresponding to ρ as coming from the limiting system. We will prove the point-wise convergence of $\Upsilon^{(k)}_{\rho_n\times\zeta}(\cdot|d,e)$ converges point-wise to $\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d,e)$ for all $k \in \mathbb{N}$ using the metric given by the total variation norm. Following Lindvall [66], the total variation norm distance between two probability measures μ and ν on a countable measurable probability space Ω is given by

$$\begin{aligned} d_{TV}(\mu,\nu) &= \frac{1}{2}\sum_{\omega\in\Omega}\big|\mu(\omega)-\nu(\omega)\big| \\ &= \inf\{\mathbb{P}(X\neq Y) : \text{r.v.s } X,Y \text{s.t. } X\sim\mu \text{ and } Y\sim\nu\}, \end{aligned} \tag{A.31}$$

where the infimum is over all couplings or joint distributions such that the marginals are given by μ and ν, respectively; the second definition applies more generally while the first is restricted to countable spaces.

For ease of exposition we will denote by 1 the user whose deficit varies as per the Markov transition kernel $\Upsilon^{(k)}_{\bullet\times\zeta}(\cdot|d,e)$ and the remaining users in the cluster by indices $\{2,3,\ldots,M\}$. For the n^{th} system and in the limiting system, in every frame the B2D component of the state of every user (including 1) is chosen *i.i.d.* with distribution ζ. We will couple all the systems under consideration such that the B2D component of the state is exactly the same; denote the random vector by $\mathbf{E}$

[2] By Lemma 5, this convergence also holds in l_∞.

with components E_l for $l \in \{1, 2, \ldots, M\}$. For users $l \in \{2, 3, \ldots, M\}$ the deficit is chosen independently via distribution ρ_n in the n^{th} system and via distribution ρ in the limit system. Since ρ_n converges to ρ point-wise, using Theorem 5 we can find a coupling $\{\tilde{X}_n^l\}_{n\in\mathbb{N}}$, $\tilde{X}^l$ and an *a.s.* finite random integer $\tilde{N}_l$ for $l \in \{2, 3, \ldots, M\}$ such that for $n \geq \tilde{N}^l$, $\tilde{X}_n^l = \tilde{X}^l$.

Next by the induction hypothesis let $\Upsilon^{(k)}_{\rho_n\times\zeta}(\cdot|d, e)$ converge point-wise to $\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d, e)$ for some $k \in \mathbb{N}$, once again by Theorem 5, there exists a coupling $\{X_n\}_{n\in\mathbb{N}}$, X and an *a.s.* finite random variable $N_k \in \mathbb{N}$ such that $X_n \sim \Upsilon^{(k)}_{\rho_n\times\zeta}(\cdot|d, e)$ for all $n \in \mathbb{N}$, $X \sim \Upsilon^{(k)}_{\rho\times\zeta}(\cdot|d, e)$ and $X_n = X$ for all $n \geq N_k$.

With these definitions in place, further define the following

$$\begin{aligned} D_n^{k+1} = \Bigg(X_n + \eta - \chi_1 \Bigg(\mathbf{a}^* \Big(\big((X_n, E_1), (\tilde{X}_n^2, E_2), \ldots, (\tilde{X}_n^M, E_M) \big) \Big), \\ \Big(\big((X_n, E_1), (\tilde{X}_n^2, E_2), \ldots, (\tilde{X}_n^M, E_M) \big) \Big) \Bigg) \Bigg)_+ \end{aligned} \tag{A.32}$$

$$\begin{aligned} D^{k+1} = \Bigg(X + \eta - \chi_1 \Bigg(\mathbf{a}^* \Big(\big((X, E_1), (\tilde{X}^2, E_2), \ldots, (\tilde{X}^M, E_M) \big) \Big), \\ \Big(\big((X, E_1), (\tilde{X}^2, E_2), \ldots, (\tilde{X}^M, E_M) \big) \Big) \Bigg) \Bigg)_+, \end{aligned} \tag{A.33}$$

where we have taken care to explicitly spell out the states of all the users involved.

Then D_n^{k+1} is a random variable distributed as $\Upsilon^{(k+1)}_{\rho_n\times\zeta}(\cdot|d, e)$ and D^{k+1} is a random variable distributed as $\Upsilon^{(k+1)}_{\rho\times\zeta}(\cdot|d, e)$.

Furthermore, for $n \geq \hat{N} := \max(N_k, \tilde{N}_2, \ldots, \tilde{N}_M)$, we have $X_n = X$, $\tilde{X}_n^l = \tilde{X}^l$ for $l \in \{2, 3, \ldots, M\}$. The last statement then implies that $D_n^{k+1} = D^{k+1}$ for $n \geq \hat{N}$.

Therefore, it follows that

$$\{\omega : D_n^{k+1} \neq D^{k+1}\} \subset \{w : \hat{N} > n\}, \tag{A.34}$$

so that

$$\begin{aligned} d_{TV}\Big(\Upsilon^{(k+1)}_{\rho_n\times\zeta}(\cdot|d,e), \Upsilon^{(k+1)}_{\rho\times\zeta}(\cdot|d,e)\Big) &\leq \mathbb{P}\big(D_n^{k+1} \neq D^{k+1}\big) \\ &\leq \mathbb{P}(\hat{N} > n), \end{aligned} \tag{A.35}$$

which converges [3] to 0 as $n \to \infty$ by the *a.s.* finiteness of $\hat{N}$. From the definition of the metric $d_{TV}(\cdot,\cdot)$, it is follows that $\Upsilon^{(k+1)}_{\rho_n\times\zeta}(\cdot|d,e)$ converges to $\Upsilon^{(k+1)}_{\rho\times\zeta}(\cdot|d,e)$ in l_1, and so both in l_∞ and point-wise also.

Having established the basic convergence result, $\mathbb{E}_\Psi(\Upsilon^{(k)}_{\rho_n\times\zeta}(\cdot|D,E))$ converges point-wise to $\mathbb{E}_\Psi(\Upsilon^{(k)}_{\rho\times\zeta}(\cdot|D,E))$ for every $k \in \{0\} \cup \mathbb{N}$ by using the bounded convergence theorem since we are averaging probability distributions. Additionally, again using the bounded convergence theorem, $\Pi_{\rho_n\times\zeta}(\cdot)$ converges point-wise to $\Pi_{\rho\times\zeta}(\cdot)$.

A.6.4 Proof of Theorem 7

Suppose there exist two MFE, namely ρ_1 and ρ_2. Consider a generic agent 1. Agent 1 has a belief that the other agents in the same cluster will draw their states from ρ_1 or ρ_2 for deficits and ζ for B2D transmissions in an i.i.d. fashion. We assume that each agent has the same realization of B2D packets received under these two deficit distributions. Given this belief and our incentive compatible mechanism (that determines transfers as a fucntion of the belief), all the agents in this cluster will truthfully reveal their states, *i.e.*, the B2D term will be the same no matter whether the belief is ρ_1 or ρ_2. By Algorithm 1, this will result in the same deficit update for agent 1. Therefore, given the truth-telling mechanism and the unique policy, we

[3] Note that this yields a rate of convergence result as well.

achieve a unique MFE, i.e., $\rho_1 = \rho_2$.

APPENDIX B

PROOFS FROM SECTION 3

B.1 Properties of the Optimal Value Function

B.1.1 Proof of Lemma 7

We first show that $T_\rho f \in \Phi$ for $\forall f \in \Phi$. The proof then follows through a verification of the conditions of Theorem 6.10.4 in [80]. From the definition of T_ρ in (3.9), we have

$$|T_\rho f(x)| \leq |u(x)| + \max_{a(x)\in\mathcal{A}} \theta_{a(x)} + \beta \max(|f(x+w)|, |f(x-l)|).$$

From this it follows that

$$\begin{aligned}\sup_{x\in\mathbb{X}} \frac{|T_\rho f(x)|}{\Omega(x)} \leq \sup_{x\in\mathbb{X}} \frac{|u(x)|}{\Omega(x)} + \sup_{x\in\mathbb{X}} \frac{\max_{a(x)\in\mathcal{A}} \theta_{a(x)}}{\Omega(x)} \\ + \beta \max\left(\sup_{x\in\mathbb{X}} \frac{|f(x+w)|}{\Omega(x)}, \sup_{x\in\mathbb{X}} \frac{|f(x-l)|}{\Omega(x)}\right).\end{aligned}$$

Let x_+ be the unique positive surplus such that $u(x_+) = 1$ and x_- be the unique negative surplus such that $u(x_-) = -1$. Note that $\Omega(x)$ is non-decreasing for $x \geq x_-$ and non-increasing for $x \leq x_+$. To avoid cumbersome algebra we will assume $x_+ - w > 0$ and $x_- + l > 0$. Since $\Omega(x) \geq |u(x)| \geq 0$ and $\Omega(x) \geq 1$, the first two terms are bounded by 1 and $\max_{a(x)\in\mathcal{A}} \theta_{a(x)}$. For the last term we have

$$\sup_{x\in\mathbb{X}} \frac{|f(x+w)|}{\Omega(x)} \leq \|f\|_\Omega \sup_{x\in\mathbb{X}} \frac{\Omega(x+w)}{\Omega(x)}.$$

We have the following

$$\frac{\Omega(x+w)}{\Omega(x)} = \begin{cases} \frac{u(x+w)}{\max(u(x),1)} \leq \frac{u(x+w)}{u(x)}, & \text{if } x \geq x_+ \\ \frac{u(x+w)}{\max(u(x),1)} \leq \frac{u(x+w)}{u(x_+)}, & \text{if } x \in [x_+ - w, x_+] \\ 1, & \text{if } x \in [x_-, x_+ - w] \\ \frac{1}{|u(x)|} \leq 1, & \text{if } x \in [x_- - w, x_-] \\ \frac{u(x+w)}{u(x)} \leq 1, & \text{if } x \leq x_- - w. \end{cases}$$

For $x \geq x_+$, we know using monotonicity of $u(\cdot)$

$$\begin{aligned} \frac{u(x+w)}{u(x)} &= 1 + \frac{u(x+w) - u(w)}{w} \frac{w}{u(x)} \\ &\leq 1 + \frac{u(x+w) - u(w)}{w} w. \end{aligned}$$

Additionally, for $x \in [x_+ - w, x_+]$ we have

$$\begin{aligned} \frac{u(x+w)}{u(x_+)} &= 1 + \frac{u(x+w) - u(x_+)}{x+w-x_+}(x+w-x_+) \\ &\leq 1 + \frac{u(x+w) - u(x_+)}{x+w-x_+} w. \end{aligned}$$

For the analysis we assume that $u(\cdot)$ is Lipschitz such that $\sup_{x \in \mathbb{X}} u'(x) < +\infty$. Therefore, by the mean value theorem

$$\begin{aligned} &\frac{u(x+w) - u(x)}{w} = u'(\xi_1) \leq \sup_{x \geq x_+} u'(x), && \forall \xi_1, x \in [x_+, \infty) \\ &\frac{u(x+w) - u(x_+)}{x+w-x_+} = u'(\xi_2) \leq \sup_{x \in [x_+ - w, x_+]} u'(x), && \forall \xi_2, x \in [x_+ - w, x_+] \\ &\sup_{x \in \mathbb{X}} \frac{\Omega(x+w)}{\Omega(x)} \leq \|f\|_\Omega (1 + w \sup_{x \geq x_+ - w} u'(x)), && \forall x \in [x_+ - w, \infty). \end{aligned}$$

Similarly, we have

$$\sup_{x\in\mathbb{X}} \frac{|f(x-l)|}{\Omega(x)} \leq \|f\|_\Omega \sup_{x\in\mathbb{X}} \frac{\Omega(x-l)}{\Omega(x)}.$$

Now we have the following

$$\frac{\Omega(x-l)}{\Omega(x)} = \begin{cases} \frac{u(x-l)}{\min(u(x),-1)} \leq \frac{u(x-l)}{u(x)}, & \text{if } x \leq x_- \\ \frac{u(x-l)}{\max(u(x),-1)} \leq \frac{u(x-l)}{u(x_-)}, & \text{if } x \in [x_-, x_- + l] \\ 1, & \text{if } x \in [x_- + l, x_+] \\ \frac{1}{u(x)} \leq 1, & \text{if } x \in [x_+, x_+ + l] \\ \frac{u(x-l)}{u(x)} \leq 1, & \text{if } x \leq x_+ l \end{cases}$$

Using the same logic as before, we get

$$\sup_{x\in\mathbb{X}} \frac{\Omega(x-l)}{\Omega(x)} \leq \|f\|_\Omega (1 + l \sup_{x\in\mathbb{X}: x\leq x_-} u'(x)).$$

Since $u(\cdot)$ is Lipschitz, thus, there exists an $\alpha_0 \in (0, +\infty)$ such that $\|T_\rho f\|_\Omega \leq \alpha_0$.

Next, we need to verify the conditions of Theorem 6.10.4 in [80]. The lemma requires verification of the following three conditions. We set $x[k]$ to be the state variable denoting the surplus at time k. We need to show that $\forall x \in \mathbb{X}$, for some constants (independent of ρ) $\alpha_1 > 0$, $\alpha_2 > 0$ and $0 < \alpha_3 < 1$,

$$\sup_{a(x)\in\mathcal{A}} |u(x) - \theta_{a(x)}| \leq \alpha_1 \Omega(x), \tag{B.1}$$

$$\mathbb{E}_{x[1],a_0}[\Omega(x[1])|x[0] = x] \leq \alpha_2 \Omega(x), \quad \forall a_0 \in \mathcal{A}, \tag{B.2}$$

with the distribution of $x[1]$ chosen based on action a_0, and

$$\beta^J \mathbb{E}_{x[J],a_0,a_1,\ldots,a_{J-1}}[\Omega(x[J])|x[0]=x] \leq \alpha_3 \Omega(x), \tag{B.3}$$

for some $J > 0$ and all possible action sequences, i.e., $a_j \in \mathcal{A}$ for all $j = 0, 1, \ldots, J-1$ with the distribution of $x[J]$ chosen based on the action sequence $(a_0, a_1, \ldots, a_{J-1})$ chosen.

First consider (B.1). Since $\Omega(x) = \max(|u(x)|, 1)$, using the earlier analysis in Section 3.3, (B.1) is true with $\alpha_1 = 1 + \max_{a\in\mathcal{A}} \theta_a$. Now consider (B.2). We have

$$\begin{aligned}\mathbb{E}_{x[1],a_0}[\Omega(x[1])|x[0]=x] &= \mathbb{E}_{\rho}[\phi(p_{\rho,a}(x))\Omega(x+w) + \phi(1-p_{\rho,a}(x)), \Omega(x-l)] \\ &\leq \max(\Omega(x+w), \Omega(x-l)),\end{aligned}$$

which is bounded by $\alpha_2\Omega(x)$ using our analysis from before.

Finally, (B.3) holds true using the properties of $\Omega(\cdot)$, the bounds on the probability of winning and losing (from Section 3.3) and our analysis from earlier in the proof as follows:

$$\begin{aligned}&\beta^J \mathbb{E}_{x[J],a_0,a_1,\ldots,a_{J-1}}[\Omega(x[J])|x[0]=x] \\ &\leq \beta^J \max(\phi(\overline{p}_W), \phi(1-\underline{p}_W))^J \max(\Omega(x+Jw), \Omega(x-Jl)) \\ &\leq (\beta \max(\phi(\overline{p}_W), \phi(1-\underline{p}_W))^J \alpha_4(J)\Omega(x),\end{aligned}$$

for some affine $\alpha_4(J) > 0$ using our analysis from before. It now follows that take J large enough we obtain an $\alpha_3 < 1$ that is also independent of ρ. Note that we can get a simpler bound of

$$\beta^J \mathbb{E}_{x[J],a_0,a_1,\ldots,a_{J-1}}[\Omega(x[J])|x[0]=x] \leq \beta^J \alpha_4(J)\Omega(x),$$

using just the properties of $\Omega(\cdot)$. Again we can take J large enough to obtain a $\alpha_3 < 1$ that is independent of ρ. This bound is useful when there is an action for which the probability of winning or losing is 1. Since all the conditions of Theorem 6.10.4 of [80] are met, then the first result in the lemma holds true. The second then follows immediately from (3.8).

B.1.2 Proof of Lemma 8

For any given ρ, from Lemma 7 we know that there is a unique $V_\rho(\cdot)$. Furthermore, it is the unique fixed point of operator T_ρ where T_ρ^J is a contraction mapping with constant α_3 that is independent of ρ. From (3.9), it follows that T_ρ^J is a continuous in ρ: computing derivatives using the envelope theorem and the expressions from Section 3.3, it is easily established that T_ρ^J is, in fact, Lipschitz with constant $(M-1)^J$ when the uniform norm is used for ρ.

Let ρ_1 and ρ_2 be two population/action profiles such that $\|\rho_1 - \rho_2\| \leq \epsilon$ (the choice of norm is irrelevant as all are equivalent for finite dimensional Euclidean spaces). As T_ρ^J is continuous in ρ, there exists a $\delta > 0$ such that $\|T_{\rho_1}^J V_{\rho_2} - T_{\rho_2}^J V_{\rho_2}\|_\Omega \leq \delta$. However, since $T_{\rho_2}^J V_{\rho_2} = V_{\rho_2}$, we have shown that $\|T_{\rho_1}^J V_{\rho_2} - V_{\rho_2}\|_\Omega \leq \delta$. Applying $T_{\rho_1}^J$ n times and using the contraction property of $T_{\rho_1}^J$, we get

$$\|T_{\rho_1}^{(n+1)J} V_{\rho_2} - T_{\rho_1}^{nJ} V_{\rho_2}\|_\Omega \leq \alpha_3^n \delta.$$

The proof then follows since $\lim_{n\to\infty} \|T_{\rho_1}^{nJ} V_{\rho_2} - V_{\rho_1}\|_\Omega = 0$ so that

$$\|V_{\rho_1} - V_{\rho_2}\|_\Omega \leq \sum_{n=0}^{\infty} \|T_{\rho_1}^{(n+1)J} V_{\rho_2} - T_{\rho_1}^{nJ} V_{\rho_2}\|_\Omega \leq \frac{\delta}{1-\alpha_3}.$$

Furthermore, using the comment from above we can show that V_ρ is Lipschitz continuous in ρ.

B.2 The Existence and Uniqueness of Stationary Surplus Distribution

B.2.1 Proof of Lemma 9

First, from the transition kernel (3.11), we satisfy the Doeblin condition as

$$\mathbb{P}(x[k] \in B | x[k-1] = x) \geq (1-\beta)\Psi(B),$$

where $0 < \beta < 1$, and Ψ is a probability measure for the regeneration process. Then from results in [73, Chapter 12], we have a unique stationary surplus distribution.

Next, let $-\tau$ be the last time before 0 that the surplus has a regeneration. Then we have

$$\begin{aligned} \zeta_{\rho\times\sigma}(B) &= \sum_{k=0}^{\infty} \mathbb{P}(B, \tau = k) \\ &= \sum_{k=0}^{\infty} \mathbb{P}(B|\tau = k) \cdot \mathbb{P}(\tau = k). \end{aligned} \tag{B.4}$$

Since the regeneration process happens independently of the surplus with inter-regeneration times geometrically distributed with parameter $(1-\beta)$, then $\mathbb{P}(\tau = k) = (1-\beta)\beta^k$. Also given $\tau = k$, we have $X_{-k} \sim \Psi$. Therefore

$$\begin{aligned} \zeta_{\rho\times\sigma}(B) &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{P}(B|\tau = k) \\ &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}\left(\mathbb{E}\left(1_{x[0]\in B}|\tau = k, X_{-k} = X\right)|\tau = k\right) \\ &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}\left(\zeta^{(k)}_{\rho\times\sigma}(B|X)|\tau = k\right) \\ &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}_{\Psi}\left(\zeta^{(k)}_{\rho\times\sigma}(B|X)\right) \\ &= \sum_{k=0}^{\infty} (1-\beta)\beta^k \int \zeta^{(k)}_{\rho\times\sigma}(B|x) d\Psi(x). \end{aligned} \tag{B.5}$$

B.2.2 Existence of MFE

B.2.2.1 Proof of Lemma 10

Define the increasing and piecewise linear convex function

$$\begin{aligned} g_\rho(y) &= \max_{a\in\mathcal{A}} \phi(p_{\rho,a})y - \theta_a \\ &= \max_{\sigma\in\Delta(|\mathcal{A}|)} \sum_{a\in\mathcal{A}} \sigma_a\big(\phi(p_{\rho,a})y - \theta_a\big), \end{aligned} \tag{B.6}$$

where $\Delta(\mathcal{A})$ is the probability simplex on $A = |\mathcal{A}|$ elements. By the properties of the lottery and the weight function $\phi(\cdot)$, $\phi(p_{\rho,a})$ is continuous in ρ for all $a \in \mathcal{A}$. Using Berge's maximum theorem, we have

$$\arg\max_{\sigma\in\Delta(|\mathcal{A}|)} \sum_{a\in\mathcal{A}} \sigma_a \left(\phi(p_{\rho,a})y - \theta_a\right) \tag{B.7}$$

is upper semicontinuous in ρ.

Now let

$$\mathcal{A}(y) := \arg\max g(y) = \arg\max_{a\in\mathcal{A}} \phi(p_{\rho,a})y - \theta_a, \tag{B.8}$$

then set-valued function above is exactly $\Delta(|\mathcal{A}(y)|)$.

Hence, the optimal randomized policies at surplus x are a set-valued function $\Delta(|\mathcal{A}(y)|) = \Delta(|\mathcal{A}(V_\rho(x+w) - V_\rho(x-l))|)$, which is upper semicontinuous due to the Lipschitz continuity of $V_\rho(\cdot)$ in ρ and the u.s.c. of $\phi(p_{\rho,a})$ in ρ, i.e., for every state x, the action distribution $\sigma(x)$ is (pointwise) upper semicontinuous in ρ.

B.2.2.2 Proof of Lemma 11

The existence and uniqueness of $\zeta(x)$ for a given ρ and $\sigma(x)$, and the relationship between $\zeta(\cdot)$ and $\zeta^{(k)}(\cdot)$ are shown in Lemma 9. Now, we will prove the continuity of $\zeta_{\rho\times\sigma}$ in ρ and $\sigma(x)$ for every surplus $x \in \mathbb{X}$. For the assumed action distribution ρ on

the finite set $\mathcal{A}$, we consider the topology of pointwise convergence which is equivalent to the uniform convergence by the strong coupling results. For the randomized action distribution $\sigma(x)$ at each surplus x, we consider the topology with norm $||\sigma|| = \sum_{j=1}^{\infty} 2^{-j}\sigma_a(x_j)$.

First, we will show that the surplus distribution $\zeta^{(k)}_{\rho\times\sigma}$ is continuous in ρ and σ. By Portmanteau theorem, we only need to show that for any sequence $\rho_n \to \rho$ in uniform, $\sigma_n \to \sigma$ in pointwise, and any open set B ,we have $\liminf_{n\to\infty} \zeta^{(k)}_{\rho_n\times\sigma_n}(B|x) \geq \zeta^{(k)}_{\rho\times\sigma}(B|x)$.

Lemma 15 $\liminf_{n\to\infty} \zeta^{(k)}_{\rho_n\times\sigma_n}(B|x) \geq \zeta^{(k)}_{\rho\times\sigma}(B|x)$.

Proof of Lemma 15

We proceed the proof by induction on k. For $k = 0$, $\zeta^{(0)}_{\rho_n\times\sigma_n}(B|x) = 1_{(x\in B)}$ is a point-mass at x irrespective of $\rho_n \times \sigma_n$, and in fact, for any $n \in \mathbb{N}_+$, we have $\zeta^{(0)}_{\rho_n\times\sigma_n}(B|x) = \zeta^{(0)}_{\rho\times\sigma}(B|x)$. Let $\rho_n \to \rho$ uniform, and $\sigma_n(x) \to \sigma(x)$ pointwise for every surplus x. We will show that $\zeta^{(k)}_{\rho_n\times\sigma_n}(B|x)$ converges pointwise to $\zeta^{(k)}_{\rho\times\sigma}(B|x)$.

We will refer to the measure and random variables corresponding to $\rho_n \times \sigma_n$ for the n^{th} system and those corresponding to $\rho\times\sigma$ as coming from the limiting system. We will prove that $\zeta^{(k)}_{\rho_n\times\sigma_n}(B|x)$ converges to $\zeta^{(k)}_{\rho\times\sigma}(B|x)$ pointwise using the metrics given above.

Suppose that the hypothesis holds true for $k-1$ where $k > 1$, i.e., $\zeta^{(k-1)}_{\rho_n\times\sigma_n}(B|x)$ converges pointwise to $\zeta^{(k-1)}_{\rho\times\sigma}(B|x)$. To prove this lemma, we only need to show that the hypothesis holds for k. Let $\mathbb{P}_{\rho\times\sigma,x}(\cdot)$ be the one-step transition probability measure of the surplus dynamics conditioned on the initial state of the surplus being x, and there is no regeneration. Then we have $\mathbb{P}_{\rho_n\times\sigma_n,x}(x+w) = \sum_{a\in\sigma_n(x)} p_{\rho_n\times\sigma_n,a}$, $\mathbb{P}_{\rho_n\times\sigma_n,x}(x-l) = 1-\sum_{a\in\sigma_n(x)} p_{\rho_n\times\sigma_n,a}$ and $\mathbb{P}_{\rho\times\sigma,x}(x+w) = \sum_{a\in\sigma(x)} p_{\rho\times\sigma,a}$, $\mathbb{P}_{\rho\times\sigma,x}(x-l) = 1-\sum_{a\in\sigma(x)} p_{\rho\times\sigma,a}$. By the properties of the lottery, $p_{\rho\times\sigma,a}$ is continuous in $\rho\times\sigma$

for all $a \in \mathcal{A}$, thus we have $p_{\rho_n \times \sigma_n, a}$ converges to $p_{\rho \times \sigma, a}$ pointwise, i.e., $\mathbb{P}_{\rho_n \times \sigma_n, x}(\cdot)$ converges to $\mathbb{P}_{\rho \times \sigma, x}(\cdot)$ pointwise. By the Skorokhod representation theorem [14], there exist random variables X_n and X on common probability space and a random integer N such that $X_n \sim \mathbb{P}_{\rho_n \times \sigma_n, x}(\cdot)$ for all $n \in \mathbb{N}$, and $X \sim \mathbb{P}_{\rho \times \sigma, x}(\cdot)$, and $X_n = X$ for $n \geq N$.

Then we have,

$$\begin{aligned}
\liminf_{n\to\infty} \zeta^{(k)}_{\rho_n \times \sigma_n}(B|x) &= \liminf_{n\to\infty} \mathbb{E}\left(\zeta^{(k-1)}_{\rho_n \times \sigma_n}(B|X_n)\right) \\
&\geq \mathbb{E}\left(\liminf_{n\to\infty} \zeta^{(k-1)}_{\rho_n \times \sigma_n}(B|X_n)\right) \\
&\geq \mathbb{E}\left(\zeta^{(k-1)}_{\rho \times \sigma}(B|X)\right) \\
&= \zeta^{(k)}_{\rho \times \sigma}(B|x),
\end{aligned} \tag{B.9}$$

where the second and third inequality hold due to Fatou's lemma and the induction hypothesis. Hence, for a given ρ and randomized policies $\sigma(x)$, the unique stationary surplus distribution $\zeta^{(k)}_{\rho_n \times \sigma_n}(B|x)$ converges pointwise to $\zeta^{(k)}_{\rho \times \sigma}(B|x)$.

Now by Lemma 9 and Equation (3.12), we need to show that $\liminf_{n\to\infty} \zeta_{\rho_n \times \sigma_n}(B)$ $\geq \zeta_{\rho \times \sigma}(B)$. By Fatou's lemma, we have

$$\begin{aligned}
\liminf_{n\to\infty} \zeta_{\rho_n \times \sigma_n}(B) &= \liminf_{n\to\infty} \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}_\Psi\left(\zeta^{(k)}_{\rho_n \times \sigma_n}(B|X_n)\right) \\
&\geq \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}_\Psi\left(\liminf_{n\to\infty} \zeta^{(k)}_{\rho_n \times \sigma_n}(B|X_n)\right) \\
&\geq \sum_{k=0}^{\infty} (1-\beta)\beta^k \mathbb{E}_\Psi\left(\zeta^{(k)}_{\rho \times \sigma}(B|X)\right) \\
&= \zeta_{\rho \times \sigma}(B).
\end{aligned} \tag{B.10}$$

Thus, for a given ρ and the randomize policies $\sigma(x)$, the unique stationary surplus distribution $\zeta_{\rho_n \times \sigma_n}$ converges pointwise to $\zeta_{\rho \times \sigma}$. Then the stationary surplus distri-

bution $\zeta_{\rho\times\sigma}$ is continuous in ρ and $\sigma(x)$ for every surplus $x \in \mathbb{X}$.

B.2.2.3 Proof of Lemma 12

Given the stationary surplus distribution $\zeta(x)$ and the action distribution $\sigma(x)$ at every surplus x, those will introduce a population profile based on the actions chosen at each point x, denoted that action distribution as ρ, and we have $\rho_a = \sum_x \zeta(x) \cdot \sigma_a(x)$, where $x \in \mathbb{X}$, $a \in \mathcal{A}$, $\mathbb{X}$ is a countable set and $\mathcal{A}$ is a finite set.

To show that ρ is continuous in $\zeta(x)$ and $\sigma(x)$, we only need to show that for any sequence $\{\zeta_n\}_{n=1}^{\infty}$ converging to ζ in uniform norm, $\{\sigma_n(x)\}_{n=1}^{\infty}$ converging to $\sigma(x)$ pointwise, we have $\{\rho_n\}_{n=1}^{\infty}$ converges to ρ pointwise, which is equivalent to convergence in uniform for the topology on a finite set $\mathcal{A}$.

Since $\zeta_n \to \zeta$ uniformly, we have $\forall\epsilon_1 > 0$, $\exists N_1 \in \mathbb{N}$, so that $\forall n \geq N_1$, $\forall x \in \mathbb{X}$, $|\zeta_n(x) - \zeta(x)| < \epsilon_1$. Similarly, $\{\sigma_n(x)\}_{n=1}^{\infty}$ converges to $\sigma(x)$ pointwise, we have $\forall x \in \mathbb{X}$, and $\forall\epsilon_2 > 0$, $\exists N_2 \in \mathbb{N}$ so that $\forall n \geq N_2$, , $|\sigma_n(x) - \sigma(x)| < \epsilon_2$. Now consider $\forall\epsilon = \max(\epsilon_1, \epsilon_2)$, we can find an all but finite subset $\mathbb{X}_1$ of $\mathbb{X}$, such that $\sum_{x\in\mathbb{X}_1} \zeta(x) < \frac{\epsilon}{2}$. Let $N = \max(N_1, N_2)$, for $\forall x \in \mathbb{X} \setminus \mathbb{X}_1$, $\exists n > N$ large enough, such that $|\sigma_{n,a}(x) - \sigma_a(x)| < \frac{\epsilon}{2}$. Then $\forall x \in \mathbb{X}$, $\forall a \in \mathcal{A}$, we have

$$
\begin{aligned}
|\rho_{n,a} - \rho_a| &= \left|\sum_x \zeta_n(x)\sigma_{n,a}(x) - \sum_x \zeta(x)\sigma_a(x)\right| \\
&= \left|\sum_x \zeta_n(x)\sigma_{n,a}(x) - \sum_x \zeta_n(x)\sigma_a(x) + \sum_x \zeta_n(x)\sigma_a(x) - \sum_x \zeta(x)\sigma_a(x)\right| \\
&\leq \left|\sum_x \zeta_n(x)\sigma_{n,a}(x) - \sum_x \zeta_n(x)\sigma_a(x)\right| + \left|\sum_x \zeta_n(x)\sigma_a(x) - \sum_x \zeta(x)\sigma_a(x)\right| \\
&\leq \sum_x \zeta_n(x)|\sigma_{n,a}(x) - \sigma_a(x)| + \sum_x \sigma_a(x)|\zeta_n(x) - \zeta(x)|
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{x\in\mathbb{X}_1} \zeta_n(x)|\sigma_{n,a}(x) - \sigma_a(x)| + \sum_{x\in\mathbb{X}\setminus\mathbb{X}_1} \zeta_n(x)|\sigma_{n,a}(x) - \sigma_a(x)| \\
&+ \sum_x \sigma_a(x)|\zeta_n(x) - \zeta(x)| \\
&\overset{(a)}{\leq} \sum_{x\in\mathbb{X}_1} \zeta_n(x)\cdot 1 + \sum_{x\in\mathbb{X}\setminus\mathbb{X}_1} \zeta_n(x)\cdot\frac{\epsilon}{2} + \sum_x \sigma_a(x)\cdot\epsilon_1 \\
&\overset{(b)}{\leq} \frac{\epsilon}{2}\cdot 1 + 1\cdot\frac{\epsilon}{2} + \epsilon_1\cdot 1 \\
&\leq \epsilon\cdot 1 + \epsilon\cdot 1 \\
&= 2\epsilon,
\end{aligned} \tag{B.11}
$$

where (a) follows from the fact that $|\sigma_{n,a}(x) - \sigma_a(x)| < 1$ for $\forall x \in \mathbb{X}$, and $|\sigma_{n,a}(x) - \sigma_a(x)| < \frac{\epsilon}{2}$, for $x \in \mathbb{X}\setminus\mathbb{X}_1$ given $\epsilon > 0$ and n large enough, and the convergence of ζ_n. (b) follows from that $\sum_{x\in\mathbb{X}_1}\zeta(x) < \frac{\epsilon}{2}$ for $x \in \mathbb{X}_1$.

Therefore, $|\rho_{n,a} - \rho_a| < 2\epsilon$ for all $a \in \mathcal{A}$ and $\forall n \geq N$, hence $\rho_n \to \rho$ pointwise, which is equivalent to uniform convergence for the topology on finite set $\mathcal{A}$.

B.3 Characteristics of the Best Response Policy

B.3.1 Proof of Lemma 13

First, we consider $x \in \mathbb{X}$ and $x \geq 0$. We have

$$
\begin{aligned}
&u(x) - \theta_{a_2(x)} + \beta[p_{\rho,a_2}(x)V_\rho(x+w) + (1-p_{\rho,a_2}(x))V_\rho(x-l)] \\
&\gtrless u(x) - \theta_{a_1(x)} + \beta[p_{\rho,a_1}(x)V_\rho(x+w) + (1-p_{\rho,a_1}(x))V_\rho(x-l)] \\
&\Leftrightarrow \theta_{a_1(x)} - \theta_{a_2(x)} \\
&\gtrless \beta[(p_{\rho,a_1}(x) - p_{\rho,a_2}(x))V_\rho(x+w) + ((1-p_{\rho,a_1}(x)) - (1-p_{\rho,a_2}(x)))V_\rho(x-l)] \\
&\Leftrightarrow \theta_{a_1(x)} - \theta_{a_2(x)} \gtrless \beta(p_{\rho,a_1}(x) - p_{\rho,a_2}(x))[V_\rho(x+w) - V_\rho(x-l)].
\end{aligned} \tag{B.12}
$$

As we assumed $\theta_{a_1(x)} > \theta_{a_2(x)}$, it follows that $p_{\rho,a_1}(x) > p_{\rho,a_2}(x)$. Also, since $w+l > 0$ and $V_\rho(x)$ is increasing in x, so both sides of the above inequality are non-negative. Since $V_\rho(x)$ is submodular when $x \geq -l$, the RHS is a decreasing function of x. Let $x^*_{a_1,a_2} \in \mathbb{X}$ be the smallest value such that LHS $\geq$ RHS, then for all $x > x^*_{a_1,a_2}$ action $a_2(x)$ is preferred to action $a_1(x)$, for all $x < x^*_{a_1,a_2}$ action $a_1(x)$ is preferred to action $a_2(x)$, and finally, if at $x^*_{a_1,a_2}$ LHS=RHS, then at $x^*_{a_1,a_2}$ the agent is indifferent between the two actions, and if instead LHS > RHS, then action $a_2(x)$ is preferred to action $a_1(x)$. We call $x^*_{a_1,a_2}$ the threshold value of surplus for actions $a_1(x)$ and $a_2(x)$.

Similarly, for $x \in \mathbb{X}$ and $x \leq 0$, $V_\rho(x)$ is supermodular when $x \leq w$, which implies the existence of a threshold policy.

B.3.2 Proof of Lemma 14

First, let $f \in \Phi$, suppose that f is an increasing and submodular function. First we prove that $T_\rho f$ is increasing and submodular too. Let $a^*(x)$ be an optimal action in the definition of $T_\rho f(x)$ when the surplus is x, i.e., one of the maximizers from (3.9). Let $x_1 > x_2$, then

$$
\begin{aligned}
&T_\rho f(x_1) - T_\rho f(x_2) = u(x_1) - u(x_2) - \theta_{a^*(x_1)} + \theta_{a^*(x_2)} + \beta\big[p_{\rho,a^*(x_1)}(x_1) f(x_1 + w) + \\
&(1 - p_{\rho,a^*(x_1)}(x_1)) f(x_1 - l) - p_{\rho,a^*(x_2)}(x_2) f(x_2 + w) - (1 - p_{\rho,a^*(x_2)}(x_2)) f(x_2 - l)\big] \\
&\geq u(x_1) - u(x_2) - \theta_{a^*(x_2)} + \theta_{a^*(x_2)} + \beta\big[p_{\rho,a^*(x_2)}(x_2) f(x_1 + w) \\
&+ (1 - p_{\rho,a^*(x_2)}(x_2)) f(x_1 - l) - p_{\rho,a^*(x_2)}(x_2) f(x_2 + w) - (1 - p_{\rho,a^*(x_2)}(x_2)) f(x_2 - l)\big] \\
&= u(x_1) - u(x_2) + \beta\big[p_{\rho,a^*(x_2)}(x_2)(f(x_1 + w + a) - f(x_2 + w) \\
&+ (1 - p_{\rho,a^*(x_2)}(x_2))(f(x_1 - l) - f(x_2 - l))\big] \geq 0.
\end{aligned}
$$

The first inequality holds because $a^*(x_2)$ need not be an optimal action when the surplus is x_1.

Again, let $x_1 > x_2$ and let $x > 0$. Since $u(\cdot)$ is a concave function, it follows that it is submodular, i.e.,

$$u(x_1 + x) - u(x_1) \leq u(x_2 + x) - u(x_2) \Leftrightarrow u(x_1 + x) + u(x_2) \leq u(x_2 + x) + u(x_1).$$

Assuming that $f \in \Phi$ is submodular, we will now show that $T_\rho f$ is also submodular. Consider

$$\begin{aligned} T_\rho f(x_1 + x) + T_\rho f(x_2) &= u(x_1 + x) + u(x_2) - \theta_{a^*(x_1+x)} - \theta_{a^*(x_2)} \\ &+ \beta\big[p_{\rho,a^*(x_1+x)}(x_1 + x) f(x_1 + x + w) + p_{\rho,a^*(x_2)}(x_2) f(x_2 + w) \\ &+ (1 - p_{\rho,a^*(x_1+x)}(x_1 + x)) f(x_1 + x - l) + (1 - p_{\rho,a^*(x_2)}(x_2)) f(x_2 - l)\big]. \end{aligned}$$

We assume without loss of generality that $p_{\rho,a^*(x_1+x)}(x_1 + x) \geq p_{\rho,a^*(x_2)}(x_2)$ and let δ be the difference; if $p_{\rho,a^*(x_1+x)}(x_1 + x) \leq p_{\rho,a^*(x_2)}(x_2)$, then a similar proof establishes the result. Using this we have the RHS (denoted by d) being

$$\begin{aligned} d &= u(x_1 + x) + u(x_2) - \theta_{a^*(x_1+x)} - \theta_{a^*(x_2)} \\ &+ \beta\big[p_{\rho,a^*(x_2)}(x_2)(f(x_1 + x + w) + f(x_2 + w)) \\ &+ (1 - p_{\rho,a^*(x_1+x)}(x_1 + x))(f(x_1 + x - l) + f(x_2 - l)) \\ &+ \delta(f(x_1 + x + w) + f(x_2 - l))\big]. \end{aligned}$$

By submodularity of $f(\cdot)$ we have

$$\begin{aligned} f(x_1 + x + w) + f(x_2 + w) &\leq f(x_2 + x + w) + f(x_1 + w), \\ f(x_1 + x - l) + f(x_2 - l) &\leq f(x_2 + x - l) + f(x_1 - l), \\ f(x_1 + x + w) + f(x_2 - l) &\leq f(x_2 + x + w) + f(x_1 - l). \end{aligned}$$

With these and using the submodularity of $u(\cdot)$ we get

$$\begin{aligned} & d \leq u(x_2 + x) + u(x_1) - \theta_{a^*(x_1+x)} - \theta_{a^*(x_2)} \\ & + \beta\big[p_{\rho,a^*(x_2)}(x_2)(f(x_2 + x + w) + f(x_1 + w)) \\ & + (1 - p_{\rho,a^*(x_1+x)}(x_1 + x))(f(x_2 + x - l) + f(x_1 - l)) \\ & + \delta(f(x_2 + x + w) + f(x_1 - l))\big] \\ & = u(x_2 + x) - \theta_{a^*(x_1+x)} + \beta[p_{\rho,a^*(x_2)}(x_2) f(x_2 + x + w) \\ & + (1 - p_{\rho,a^*(x_2)}(x_2)) f(x_2 + x - l)] + u(x_1) - \theta_{a^*(x_2)} \\ & + \beta[p_{\rho,a^*(x_2)}(x_2) f(x_1 + w) + (1 - p_{\rho,a^*(x_1+x)}(x_1 + x)) f(x_1 - l)] \\ & \leq T_\rho f(x_2 + x) + T_\rho f(x_1), \end{aligned}$$

where the last inequality holds as using the optimal actions $(a^*(x_2+x), a^*(x_1))$ yields a higher value as opposed to the sub-optimal actions $(a^*(x_1 + x), a^*(x_2))$ when the surplus is $x_2 + x$ and x_1.

Since both the monotonicity and submodularity properties are preserved when taking pointwise limits, choosing $f(\cdot) \equiv 0$ (or $u(\cdot)$) to start the value iteration proves that the value function $V_\rho(\cdot)$ is increasing and submodular.

Similarly, if $f \in \Phi$ is an increasing and supermodular function, following the same argument, we can prove that the value function $V_\rho(\cdot)$ is increasing and supermodular.

B.4 Numerical Study: Reward, Saving and Profit

Here we present two mappings of actions to coupons that are different from that shown in Tables 3.3 and 3.4, and conduct the experiment with $l = 1, 3, 5$. We find that our results are robust to the mapping of actions to coupons. For example, when we set $l = 1$, most savings can be achieved by giving a \$40 reward and the break-even point is about \$80 as observed earlier. The total rewards are bounded by \$80 in all cases.

B.4.1 Case 1

Our coupon choices are shown in Table B.1, where x_1 and x_6 are energy usage in the corresponding periods (measured in kWh) and the day-ahead price is of one day randomly drawn from the three months.

Table B.1: Day-ahead price and energy coupons

Index	Period	Day-ahead Price/MWh	Coupons/kWh
1	$2-3$ PM	\$47	90 if $x_1 > 2.464$; 1.8 otherwise
2	$3-4$ PM	\$55	5.4
3	$4-5$ PM	\$78	1.8
4	$5-6$ PM	\$99.6	0
5	$6-7$ PM	\$66.5	3.6
6	$7-8$ PM	\$49.5	36 if $x_6 > 2.24$; 1.8 otherwise

We identified 6 actions as before, and computed the number of coupons received under the new coupon awarding policy shown in Table B.1. These values are shown in Table B.2.

Figure B.1 indicates results quite similar to those presented earlier in Figure 3.11. The breakeven point and maximum profit are much the same.

Table B.2: Actions, costs and energy coupons

Index	Action Vector	Cost	Coupons
0	$(22.5, 22.5, 22.5, 22.5, 22.5)$	0	37.4
1	$(21.5, 21.5, 22.25, 23.5, 23.75, 21.25)$	3.68	560
2	$(21.5, 21.5, 22.25, 24, 23.5, 21.75)$	3.51	559
3	$(21.5, 21.5, 22.25, 24, 23.5, 22)$	3.50	553
4	$(21.5, 21.5, 22.25, 23.5, 23.25, 22.25)$	3.146	547
5	$(21.5, 21.5, 22.25, 24, 23, 22.5)$	2.68	471

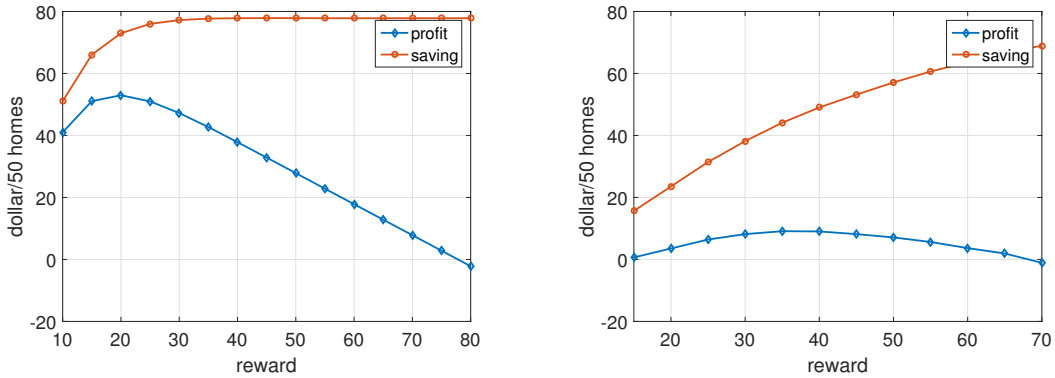

Figure B.1: The relation between customer reward, LSE savings and LSE profit. Left: $l = 1$. Right: $l = 5$.

B.4.2 Case 2

Our coupon choices are shown in Table B.3, where x_1 and x_6 are energy usage in the corresponding periods (measured in kWh) and the day-ahead price is of one day randomly drawn from the three months.

We identified 6 actions as before, and compute the number of coupons received under the awarding policy shown in Table B.3. These values are shown in Table B.4.

Figure B.2 indicates results quite similar to those presented earlier in Figure 3.11. The breakeven point and maximum profit are much the same.

Table B.3: Day-ahead price and energy coupons

Index	Period	Day-ahead Price/MWh	Coupons/kWh
1	$2-3$ PM	\$47	144 if $x_1 > 2.464$; 1.8 otherwise
2	$3-4$ PM	\$55	5.4
3	$4-5$ PM	\$78	1.8
4	$5-6$ PM	\$99.6	0
5	$6-7$ PM	\$66.5	3.6
6	$7-8$ PM	\$49.5	72 if $x_6 > 2.24$; 1.8 otherwise

Table B.4: Actions, costs and energy coupons

Index	Action Vector	Cost	Coupons
0	$(22.5, 22.5, 22.5, 22.5, 22.5)$	0	37.4
1	$(21.5, 21.5, 22.25, 23.5, 23.75, 21.25)$	3.68	947
2	$(21.5, 21.5, 22.25, 24, 23.5, 21.75)$	3.51	944
3	$(21.5, 21.5, 22.25, 24, 23.5, 22)$	3.50	933
4	$(21.5, 21.5, 22.25, 23.5, 23.25, 22.25)$	3.146	916
5	$(21.5, 21.5, 22.25, 24, 23, 22.5)$	2.68	760

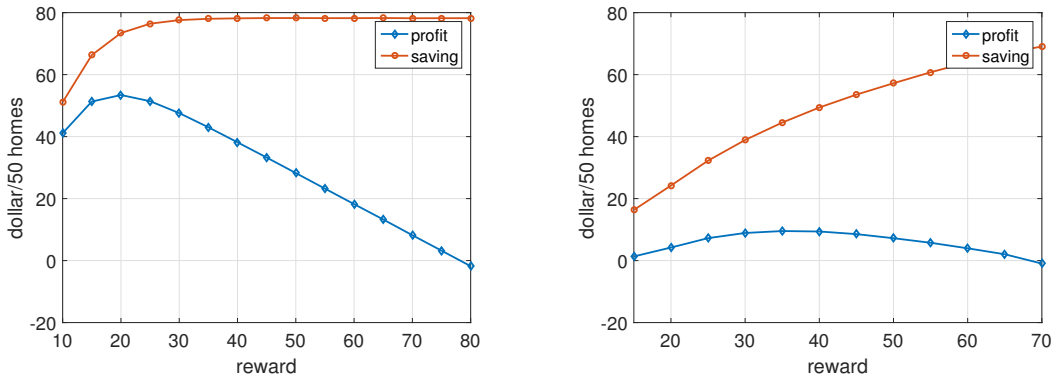

Figure B.2: The relation between customer reward, LSE savings and LSE profit. Left: $l = 1$. Right: $l = 5$.

APPENDIX C

PROOFS FROM SECTION 4

C.1 Steady State Distribution

C.1.1 Proof of Theorem 9

We provide the proof for the steady state probability $\pi^*_{\mathrm{LRU}}(\boldsymbol{x})$ by using a probabilistic argument, following [40]. We consider the current state $\boldsymbol{x}$, and attempt to reconstruct the past history by looking backwards in time. In order to achieve the current state $\boldsymbol{x} = (x_1, \cdots, x_m)$, the past history of requests, listed from the most remote to the most recent, must be ordered as follows:

$(2m)$: A last request for item x_m is made;

$(2m-1)$: Requests for item $(x_1, x_2, \cdots, x_{m-1})$ are made;

$(2m-2)$: A last request for item x_{m-1} is made;

$(2m-3)$: Requests for item $(x_1, x_2, \cdots, x_{m-2})$ are made;

$(2m-4)$: A last request for item x_{m-2} is made;

$(2m-5)$: Requests for item $(x_1, x_2, \cdots, x_{m-3})$ are made;

...

(2): A last request for item x_2 is made;

(1): At least one request for item x_1 is made.

The probability that step $2j$ $(2 \leq j \leq m)$ occurs with probability p_{x_j}, and the probability steps $2j$ and $2j-1$ together is given by

$$p_{x_j} \sum_{k=0}^{\infty} \left(\sum_{l=1}^{j-1} p_{x_l} \right)^k = \frac{p_{x_j}}{1 - \sum_{l=1}^{j-1} p_{x_l}}. \tag{C.1}$$

Note that the last two steps occur with probability $p_{x_2} \sum_{k=1} p_{x_1}^k = p_{x_2} \cdot \frac{p_{x_1}}{1-p_{x_1}}$. Combing them together, we get the steady state probability of state $\boldsymbol{x}$ as

$$\pi^*_{\text{LRU}}(\boldsymbol{x}) = \frac{\Pi_{i=1}^m p_{x_i}}{(1-p_{x_1})(1-p_{x_1}-p_{x_2})\cdots(1-p_{x_1}-\cdots-p_{x_{m-1}})}, \tag{C.2}$$

which gives us the desirable result.

C.2 Characteristics of Mixing Time

C.2.1 Proof of Theorem 11

Since the transition matrix $\mathbb{P}^{\text{LRU}}$ of LRU is non-reversible, we consider the constructive reversible Markov chain with transition matrix $\frac{\mathbb{P}^{\text{LRU}}+\mathbb{P}^{\text{LRU},*}}{2}$, where $\mathbb{P}^{\text{LRU},*}$ is the time reversal chain of $\mathbb{P}^{\text{LRU}}$, and $\pi^*_{\mathbb{P}^{\text{LRU}}} = \pi^*_{\frac{\mathbb{P}^{\text{LRU}}+\mathbb{P}^{\text{LRU},*}}{2}}$.

In order to show that the Markov chain associated with LRU is rapid mixing, we only need to show that the conductance is greater than some polynomial in the size of state space. Furthermore, from the relations between conductance and congestion in (4.22), we only need to show that the congestion is polynomial in the size of state space.

Based on the definition of congestion in (4.21), we will focus on achieving the maximum over the RHS. We first consider the term $\frac{1}{\pi(u)P_{uv}}$. By the policy of LRU and the steady state probability given in (4.2), we only need to consider the minimal steady state probabilities with the minimum transition probability. It is obvious that the state that achieves minimal steady state probability should be $\boldsymbol{c} = (n, n-1, \cdots, n-m+1)$, in other words, the least m popular items are stored in the cache in a decreasing order from the most recently used position to the least recently used

position, hence we have

$$\pi_{\min}^{\mathrm{LRU}} = \frac{p_n p_{n-1} \cdots p_{n-m+1}}{(1-p_n)\cdots(1-p_n-\cdots-p_{n-m+1})}. \tag{C.3}$$

For the Zipf-like distribution, the normalization factor is given asymptotically by $A \approx (1-\alpha)/n^{1-\alpha}$, where n is the total number of unique items in the system. Since $p_i = A/i^\alpha$, $(1-p_n)(1-p_n-p_{n-1})\cdots(1-p_n-p_{n-m+1}) \to 1$ as n becomes large, then we have $\pi_{\min}^{\mathrm{LRU}} \simeq p_n p_{n-1} \cdots p_{n-m+1} = \frac{A}{n^\alpha}\frac{A}{(n-1)^\alpha}\cdots\frac{A}{(n-m+1)^\alpha}$. For the minimal transition probability, we can consider $P_{uv} = \arg\min_i p_i = \theta(1/n)$. Hence, we have

$$\frac{1}{\pi_{\min}^{\mathrm{LRU}} P_{uv}} = \frac{[n(n-1)\cdots(n-m+1)]^\alpha \cdot n}{A^m} = O(n^{m+1}) \tag{C.4}$$

Next we consider the summation part in (4.21), which can be upper bounded by

$$\sum_{i,j\in\Omega,\gamma_{ij}\,\mathrm{uses}(u,v)} \pi(i)\pi(j) \le \Gamma(\pi_{\max}^{\mathrm{LRU}})^2, \tag{C.5}$$

where Γ is the number of states and $\pi_{\max}^{\mathrm{LRU}}$ is maximal steady state probabilities. From the steady state probability of LRU given in (4.2), it is clear that the state that achieves the maximal steady state probability is $\boldsymbol{c} = (1, 2, \cdots, m)$. Hence, $\pi_{\max}^{\mathrm{LRU}} = \frac{p_1\cdots p_m}{(1-p_1)\cdots(1-p_1-\cdots-p_{m-1})} = O(\frac{1}{n^{2(1-\alpha)m}})$. The number of states $\Gamma = O(n^m)$. Therefore, we have

$$\sum_{i,j\in\Omega,\gamma_{ij}\,\mathrm{uses}(u,v)} \pi(i)\pi(j) = O(n^{-2(1-\alpha)m}) \cdot O(n^m) = O(n^{(2\alpha-1)m}). \tag{C.6}$$

Therefore, the congestion of LRU is upper bounded by

$$\rho^{\mathrm{LRU}} = O(n^{m+1}) \cdot O(n^{(2\alpha-1)m}) = O(n^{2\alpha m+1}), \tag{C.7}$$

which is polynomial in the size of the state space, i.e., the Markov chain associated with LRU is rapidly mixing.

C.2.2 Proof of Theorem 12

By Cheeger inequality (4.19) and the relation of congestion and conductance (4.22), we have

$$t_{\text{rel}} \leq \frac{2}{\Phi^2} \ln \frac{1}{\pi_{\min}} \leq 8\rho^2 \ln \frac{1}{\pi_{\min}}. \tag{C.8}$$

By Theorem 11, we have $\rho^{\text{LRU}} = O(n^{2\alpha m+1})$, and $\pi_{\min}^{\text{LRU}} = \Theta(n^{-m})$, then we obtain

$$t_{\text{mix}}^{\text{LRU}}(\epsilon) = O(n^{4\alpha m+2} \ln n). \tag{C.9}$$

C.2.3 Proof of Theorem 13

In order to show that the Markov chain associated with RANDOM is rapid mixing, we only need to show that the conductance is greater than some polynomial in the size of state space. Furthermore, from the relations between conductance and congestion in (4.22), we only need to show that the congestion is polynomial in the size of state space.

Based on the definition of congestion in (4.21), we will focus on achieving the maximum over the RHS. We first consider the term $\frac{1}{\pi(u)P_{uv}}$, by the eviction and insertion policies of RANDOM and the steady state probability given in (4.2), we can take state u as the minimum state with $\pi(u) = \frac{p_n p_{n-1} \cdots p_{n-m+1}}{\sum_{\boldsymbol{c} \in \Lambda'_{n,m}} \prod p_{c_i}} \triangleq \frac{p_n p_{n-1} \cdots p_{n-m+1}}{\Upsilon}$, where $\Upsilon = \sum_{\boldsymbol{c} \in \Lambda'_{n,m}} \prod p_{c_i}$, and consider state v as one state such that the $(n-m)$-th popular item is requested when we are in state u, hence we can achieve a minimum transition probability $P_{uv} = p_{n-m}/m$. Then, we have

$$\frac{1}{\pi(u)P_{uv}} = \frac{[n(n-1)\cdots(n-m+1)]^{\alpha}\Upsilon}{A^m} \cdot \frac{(n-m)^{\alpha} m}{A}$$

$$
\begin{aligned}
&= \frac{m[n(n-1)\cdots(n-m)]^{\alpha}\Upsilon}{A^{m+1}} \\
&\overset{(a)}{\leq} \frac{m[n(n-1)\cdots(n-m)]^{\alpha}}{A^{m+1}} \\
&\cdot \frac{n(n-1)\cdots(n-m+1)}{m!} \cdot \frac{A^m}{[1\cdot 2\cdots m]^{\alpha}} \\
&\overset{(b)}{\leq} \frac{n^{m(\alpha+1)+1}}{(m!)^{\alpha+1}} = O(n^{(\alpha+1)m+1}),
\end{aligned} \tag{C.10}
$$

where (a) follows from that Υ is upper bounded by the total number of states $\frac{n(n-1)\cdots(n-m+1)}{m!}$ multiply the largest steady state probability $\frac{A^m}{[1\cdot 2\cdots m]^{\alpha}}$, and (b) follows from that $A \approx (1-\alpha)/n^{1-\alpha}$.

The summation part in (4.21) can be upper bounded by

$$
\begin{aligned}
&\sum_{i,j\in\Omega,\gamma_{ij}\,\text{uses}(u,v)} \pi(i)\pi(j) \leq \left(\frac{(p_1p_2\cdots p_m)^2}{\Upsilon}\right)\cdot\Gamma \overset{(c)}{\leq} \frac{(p_1p_2\cdots p_m)^2}{\frac{n(n-1)\cdots(n-m+1)}{m!}p_n\cdots p_{n-m+1}} \\
&= \frac{\left(\frac{A^m}{(m!)^{\alpha}}\right)^2}{\frac{n(n-1)\cdots(n-m+1)}{m!}\frac{A^m}{[n(n-1)\cdots(n-m+1)]^{\alpha}}} \\
&= A^m[n(n-1)\cdots(n-m+1)]^{\alpha-1}(m!)^{1-2\alpha} \\
&\overset{(d)}{=} O(n^{(2\alpha-1)m}),
\end{aligned} \tag{C.11}
$$

where (c) and (d) follow the same arguments as (a) and (b), respectively. Therefore, the congestion of RANDOM is upper bounded as

$$
\rho^{\text{RANDOM}} = O(n^{(\alpha+1)m+1})\cdot O(n^{(2\alpha-1)m}) = O(n^{3\alpha m+1}). \tag{C.12}
$$

which is polynomial in the size of state space $\binom{n}{m}\cdot m!$, then from (4.22), we know that the mixing process of Markov chain associated with RANDOM is rapidly mixing. A similar argument holds true for FIFO, therefore, both RANDOM and FIFO are rapidly mixing.

C.2.4 Proof of Theorem 14

From Theorem 13, we have $\rho^{\text{RANDOM}} = O(n^{3\alpha m+1})$, and $\pi_{\min}^{\text{RANDOM}} = \Theta(n^{-m})$. Given the relation between mixing time and congestion in (C.8), we obtain

$$t_{\text{mix}}^{\text{RANDOM}}(\epsilon) = O(n^{6\alpha m+2} \ln n). \tag{C.13}$$

C.2.5 Proofs of Theorem 15 and Theorem 16

We follow the same arguments as we did in the proof of Theorem 11 and Theorem 13 to show that CLIMB is rapidly mixing, we omit the details here and only give a closed form of the upper bound on the congestions.

$$\rho^{\text{CLIMB}} = O(n^{\frac{3\alpha m(m+1)}{2}+1}). \tag{C.14}$$

Similarly, by the relation between mixing time and congestion defined in (C.8), we have

$$t_{\text{mix}}^{\text{CLIMB}}(\epsilon) = O(n^{3\alpha m(m+1)+2} \ln n). \tag{C.15}$$

APPENDIX D

PROOFS FROM SECTION 5

D.1 Characteristics of Mixing Time

D.1.1 Proof of Theorem 18:

Here, we show that the Markov chain associate with RANDOM($\boldsymbol{m}$) is rapid mixing. In order to achieve this, we need to show that its conductance is great than some polynomial in the size of state space. Furthermore, from Equation (4.22), we only need show that its congestion is polynomial in the size of state space.

From the definition of congestion in Equation (4.21), we aim to achieve the supremum over the right hand side (RHS). First, we consider the term $\frac{1}{\pi(u)P_{uv}}$. Given the eviction and insertion rules of RANDOM($\boldsymbol{m}$) and stationary distribution given in Equation (5.1), we can take the state u that achieve the minimal stationary probability, and consider state v as one state such that the last spot (the spot in cache 1) is different from v, hence, the minimal transition probability $P_{uv} = p_{n-m}/m_1$. Then we have

$$
\begin{aligned}
&\frac{1}{\pi(u)P_{uv}} = G(\boldsymbol{m}) \left(\frac{[n(n-1)\cdots(n-m_h+1)]^{\alpha}}{A^{m_h}} \right)^{h} \\
&\cdot \left(\frac{[(n-m_h)\cdots(n-m_h-m_{h-1}+1)]^{\alpha}}{A^{m_{h-1}}} \right)^{h-1} \cdots \\
&\cdot \left(\frac{[(n-\sum_{i=2}^{h} m_i)\cdots(n-m+1)]^{\alpha}}{A^{m_1}} \right)^{1} \\
&= \frac{G(\boldsymbol{m})m_1}{p_{n-m}A^{m_1+2m_2+\cdots+hm_h}} \cdot \big(n(n-1)\cdots(n-m_h+1)\big)^{\alpha h} \\
&\cdot \left((n-m_h)\cdots(n-m_h-m_{h-1}+1)\right)^{\alpha(h-1)} \cdots
\end{aligned}
$$

$$\cdot \left((n - \sum_{i=2}^{h} m_i) \cdots (n - m + 1) \right)^{\alpha}$$
$$= O(n^{(1+\alpha)(m_1+2m_2+\cdots+hm_h)+1}). \tag{D.1}$$

Next, we give an upper bound on the summation part in (4.21)

$$\sum_{i,j \in \Lambda, \gamma_{ij} \text{uses}(u,v)} \pi(i)\pi(j) \overset{(a)}{\leq} [\pi(\boldsymbol{x}^*)]^2 \cdot \Upsilon$$
$$= \frac{1}{G(\boldsymbol{m})} (p_1 \cdots p_{m_1})^2 \cdot \left([p_{m_1+1} \cdots p_{m_1+m_2}]^2 \right)^2 \cdot \cdots \left(\left[p_{\sum_{j=1}^{h-1} m_j + 1} \cdots p_m \right]^h \right)^2 \cdot \Upsilon$$
$$= \frac{A^{2(m_1+2m_2+\cdots+hm_h)} \cdot \Upsilon}{G(\boldsymbol{m})} \cdot \frac{1}{([1 \cdots m_1]^{\alpha})^2}$$
$$\cdot \frac{1}{([(m_1+1) \cdots (m_1+m_2)]^{2\alpha})^2} \cdots \frac{1}{\left([(\sum_{j=1}^{h-1} m_j + 1) \cdot m]^{h\alpha} \right)^2}, \tag{D.2}$$
$$= O(n^{(2\alpha-1)(m_1+2m_2+\cdots+hm_h)}), \tag{D.3}$$

where (c) follow that $\pi(\boldsymbol{x}^*)$ achieves the largest stationary probability and Υ is the number of total states, i.e., $\Upsilon = O(n^m)$. Furthermore, for the normalization factor A, we have $A = O(1/n^{1-\alpha})$. Therefore, plugging (D.1) and (D.2) into Equation (4.21), we have that the congestion of RANDOM is upper bounded as

$$\rho = O(n^{3\alpha(m_1+2m_2+\cdots+hm_h)+1}), \tag{D.4}$$

which is polynomial in the size of state space, then from (4.22), we know that the mixing process of Markov chain associated with RANDOM($\boldsymbol{m}$) is rapidly mixing. A similar argument holds true for FIFO($\boldsymbol{m}$). Therefore, RANDOM($\boldsymbol{m}$) and FIFO($\boldsymbol{m}$) are rapidly mixing.

D.1.2 Proof of Theorem 5.2:

By Cheeger inequality (4.19) and the relation of congestion and conductance (4.22), we have

$$t_{\text{rel}} \leq \frac{2}{\Phi^2} \ln \frac{1}{\pi_{\min}} \leq 8\rho^2 \ln \frac{1}{\pi_{\min}}. \tag{D.5}$$

By Theorem 18, we have

$$t_{\text{mix}}^{\text{RANDOM}(\boldsymbol{m})}(\epsilon) = O(n^{6\alpha(m_1+2m_2+\cdots+hm_h)+2} \ln n), \tag{D.6}$$

where $m = m_1 + m_2 + \cdots + m_h$.