DISSERTATION

DISTRIBUTED WIRELESS NETWORKING WITH AN ENHANCED PHYSICAL-LINK LAYER INTERFACE

Submitted by

Yanru Tang

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2018

Doctoral Committee:

Advisor: Rockey Luo

Liuqing Yang
Ali Pezeshki
Haonan Wang

ABSTRACT

DISTRIBUTED WIRELESS NETWORKING WITH AN ENHANCED PHYSICAL-LINK LAYER INTERFACE

This thesis focuses on the cross-layer design of physical and data link layers to support efficient distributed wireless networking. At the physical layer, distributed coding theorems are proposed to prepare each transmitter with an ensemble of channel codes. In a time slot, a transmitter chooses a code to encode its messages and such a choice is not shared with other transmitters or with the receiver. The receiver guarantees either reliable message decoding or reliable collision report depending on whether a pre-determined reliability threshold can be met. Under the assumption that the codeword length can be taken to infinity, the distributed capacity of a discrete-time memoryless multiple access channel is derived and is shown to coincide with the classical Shannon capacity region of the same channel. An achievable error performance bound is also presented for the case when codeword length is finite. With the new coding theorems, link layer users can be equipped with multiple transmission options corresponding to the physical layer code ensemble. This enables link layer users to exploit advanced wireless capabilities such as rate and power adaptation, which is not supported in the current network architecture. To gain understandings on how link layer users should efficiently exploit these new capabilities, the corresponding link layer problem is investigated from two different perspectives.

Under the assumption that each user is provided with multiple transmission options, the link layer problem is first formulated using a game theoretic model where each user adapts its transmission scheme to maximize a utility function. The condition under which the medium access control game has a unique Nash equilibrium is obtained. Simulation results show that, when multiple transmission options are provided, users in a distributed network tend to converge to channel sharing schemes that are consistent with the well-known information theoretic understandings.

A stochastic approximation framework is adopted to further study the link layer problem for the case when each user has a single transmission option as well as the case when each user has multiple transmission options. Assume that each user is backlogged with a saturated message queue. With a generally-modeled channel, a distributed medium access control framework is proposed to adapt the transmission scheme of each user to maximize an arbitrarily chosen symmetric network utility. The proposed framework suggests that the receiver should measure the success probability of a carefully designed virtual packet or a set of virtual packets, and feed such information back to the transmitters. Given channel feedback from the receiver, each transmitter should obtain a user number estimate by comparing the measured success probability with the corresponding theoretical value, and then adapt its transmission scheme accordingly. Conditions under which the proposed algorithm should converge to a designed unique equilibrium are characterized. Simulation results are provided to demonstrate the optimality and the convergence properties of the proposed algorithm.

ACKNOWLEDGEMENTS

The work presented here could not have been accomplished without the help from various individuals. First and foremost, I would like to deeply thank my advisor, Dr. J. Rockey Luo, for his continuous support of this study and related research. Dr. Luo has selflessly dedicated countless time and imparted massive knowledge on my work and guided me on every aspect of my search. In addition, he has always inspired and encouraged me by his undeviating hardworking and passionate attitudes toward our research subjects. I feel extremely blessed to be supervised by him and work with him over the past 6 years.

I also would like to extend my gratitude to my committee members, Dr. Liuqing Yang, Dr. Ali Pezeshki, and Dr. Haonan Wang, for their valuable comments and insightful suggestion on my work. I sincerely thank the entire research group, particularly Ms. Faeze Heydaryan, Mr. Tiegang Zhao, Mr. Xiaoyin Qi, and Dr. Zheng Wang, for their precious help and assistance during my Ph.D program. I am also very grateful for my family and friends, especially my parents, my husband Jianbo Liu, and my beloved dog Summer, for their support and encouragement throughout this journey.

Disclaimer: Results presented in Chapters 2 and 4 of this thesis are based on joint contributions published in [1] [2] [3]. The distributed channel coding theorems presented in Chapter 2 are techinical improvements to existing theorems presented in [4] [5] [6]. While the author is a major contributor of these improvements, the results were indeed obtained from joint discussions with the coauthors Ms. Faeze Heydaryan and Dr. J. Rockey Luo. In Chapter 4, results of the single transmission option case were mainly obtained by the author. Ms. Faeze Heydaryan realized that the two monotonicy requirements can be relaxed into one monotinicity property, and this paved the way of extending the results to the multiple transmission option case. Overall, the technical results presented in Chapter 4 were obtained with a large number of technical detours each has its irreplaceable contribution that leads to the discovery of the final theorems presented in the thesis. While the author contributed independently on many of the core technical pieces, the final theorems should be regarded as joint works of the coauthors of [1] [2] [3] rather than exclusive contributions of the author of this thesis.

DEDICATION

I would like to dedicate this thesis to my family.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Classical channel coding theory assumes that users in a communication party should jointly optimize their channel codes and transmit encoded messages to the receiver over a long time duration to guarantee reliable message recovery [7] [8] [9]. Overhead of achieving the required user coordination is often ignored due to the fundamental assumption that coordinated message transmission should dominate the communication process. However, this core assumption is increasingly challenged by the growing popularity of distributed communication scenarios, where each user should be able to adjust its communication parameters such as rate and power without sharing it with other users including the targeted receiver. In distributed communication, users often have short and bursty messages that must be disseminated in a timely manner. Coordinating all users in such an environment could be infeasible or expensive in terms of overhead cost. Without full user coordination, reliable message transmissions cannot always be guaranteed, and therefore packet collisions are often unavoidable. When a packet collision happens, one needs to understand how users should efficiently adapt their transmission schemes in response to channel feedback. Despite the fact that a significant proportion of the messages in current wireless networks are transmitted using distributed communication protocols, a theoretical foundation that supports efficient distributed communication still does not exist.

Classical physical-link layer interface assumes that a link layer user can only determine whether a packet should be transmitted or not [10]. Other communication details should be handled at the physical layer. In distributed communication when physical layer does not have full capability of joint channel coding optimization, data link layer has to get involved into communication adaptations. A simple example is the collision resolution protocol such as the exponential backoff-based DCF protocol in IEEE 802.11 [11]. However, with each link layer user only having binary transmission/idling options, when a packet collision happens, the only option for users to control contention is to adapt their transmission probabilities. Advanced wireless capabilities such as rate, power and antenna beam adaptations all become irrelevant at the data link layer. This can lead to a quite significant efficiency reduction in the throughput performance of a wireless system.

For example, let us consider a multiple access system with K homogenous users and a single receiver. Assume a unit channel gain from each user to the receiver, and additive Gaussian noise with zero mean and variance N_0. Assume that each user has a transmission power of P. From classical channel coding theory [7], we know that, if each user encodes its own messages at a rate of $\frac{1}{2}\log_2\left(1+\frac{P}{N_0}\right)$ bits/symbol, then reliable message recovery is only possible if users transmit sequentially. Sum rate of the system is therefore upper bounded by the single user channel capacity of $C_1 = \frac{1}{2}\log_2\left(1+\frac{P}{N_0}\right)$ bits/symbol, irrespective of the user number K. Alternatively, if users transmit in parallel with an individual rate of $\frac{1}{2K}\log_2\left(1+\frac{KP}{N_0}\right)$, then sum rate of the system can approach the sum channel capacity of $C_K = \frac{1}{2}\log_2\left(1+\frac{KP}{N_0}\right)$ bits/symbol, which grows unboundedly in K. A similar conclusion applies to the same system with a distributed communication model as well. Assume that each user has bursty short messages and cannot afford the overhead of joint coding optimization. If messages of all users are encoded at a rate only slightly less than C_1[1], then sum rate of the system is upper bounded by C_1 bits/symbol. Alternatively, if messages arrive with a statistics such that on average $\tilde{K}$ users should have messages to transmit at any moment, from the perspective of throughput optimization, then it is generally beneficial for each user to encode its messages at a rate close to $\frac{1}{2\tilde{K}}\log_2\left(1+\frac{\tilde{K}P}{N_0}\right)$ to support parallel transmissions of up to $\tilde{K}$ users. However, because traffic statistics is unknown at the design stage of a protocol and may also vary in time, in the case of distributed communication, maintaining a high throughput efficiency requires users have reasonable flexibility of adapting their communication parameters, such as communication rate, at the data link layer. Unfortunately, such a capability is not supported by the physical-link layer interface in the current network architecture.

---

[1]Note that the rate needs to be smaller than C_1 in order to support reliable decoding with a finite codeword length [12].

1.2 An Enhanced Physical-Link Layer Interface

Distributed communication is often featured with short bursty messages and opportunistic channel access. The nature of distributed communication implies that communication parameters cannot be jointly and fully optimized at the physical layer. However, system traffic at the data link layer may still be more or less stationary. To improve communication efficiency, a data link layer should exploit advanced wireless capabilities to adapt its transmission scheme accordingly, and this needs to be done under the constraint of maintaining a layered (or modularized) network architecture.

To achieve such an objective, we propose an enhancement to the classical physical-link layer interface [5]. The enhanced interface should prepare each physical layer transmitter with an ensemble of channel codes as opposed to one code in a classical architecture. At the data link layer, each user can then be equipped with multiple transmission options corresponding to the available channel codes at the physical layer. Different transmission options may correspond to different communication settings such as different power, rate and antenna beam combinations. Therefore, link layer users can exploit advanced communication adaptation such as rate adaptation to further improve communication efficiency. To maintain the layered architecture, under the distributed communication model, we assume that a link layer protocol should inform the physical layer whether a message needs to be transmitted, and if so, which transmission option should be used. Such decisions are not controlled or optimized at the physical layer. According to the link layer protocol, a physical layer transmitter then chooses the corresponding code to encode its message and sends its codeword through the channel. The receiver should decode the messages only if a pre-determined error probability threshold can be met [10] [5]. Otherwise the receiver should report collision. At the data link layer, we assume that a user can only choose from the list of provided transmission options, as opposed to being able to adapt the communication parameters arbitrarily.

1.3 Contribution

While the interface enhancement appears to be minor, it involves key research questions whose answers cannot be found in the classical frameworks. At the physical layer, due to possible lack of user coordination, reliable message delivery cannot always be guaranteed. However, it is a fundamental requirement in the layered architecture that any message forwarded to the data link layer must be reliable [10]. Furthermore, because transmission decisions are made at the data link layer, i.e., they are not controlled by a physical layer protocol, any assumption of such a control, such as communication rate optimization, may not be valid in physical layer channel coding. With these constraints, one needs to understand whether the notion of fundamental limit still exists for a distributed communication system. It will be shown in Chapter 2 that not only the notion of channel capacity still exists for a distributed system, it also coincides with classical Shannon capacity region of the same channel without the convex hull operation. Meanwhile, at the data link layer when a user is equipped with multiple transmission options, one needs to understand how packet transmission schemes should be adapted in response to the events of transmission success and packet collision. In existing link layer protocols, when only a single transmission option (plus an idling option) is available, a common practice in response to packet collision is to reduce the packet transmission probability of each user [10] [11] [13]. From classical channel coding theory, we know that a more efficient approach for sum throughput optimization could be adapting the communication rate of each user [7]. However, while transmission options with different power and rate combinations may be available at the data link layer, there is no guarantee that the ideal option should be on the list. Furthermore, different link layer networks may also have different utility optimization objectives. Whether a general link layer distributed medium access control framework exists to optimize transmission schemes under these constraints is an important question that needs to be answered. In Chapter 3 and Chapter 4, a game theoretic model and a stochastic approximation framework are presented to investigate the corresponding link layer problem, respectively. The rest of the thesis is organized as follows.

Chapter 2 presents the distributed channel coding theorems, which establish the distributed channel capacity of a multiple access system when the codeword length can be taken to infinity. An upper bound for the worst case error event probability is also derived for the case of finite codeword length. The new coding theory provides theoretical support to enhance the classical physical-link layer interface in the sense that it enables the option of equipping each link layer user with multiple transmission options. When given multiple transmission options, each user has a transmission probability vector to describe its transmission scheme, with each entry of the vector denoting the probability of using different transmission options. In Chapter 3, under the assumption that users are backlogged with messages, we model the medium access control problem as a non-cooperative game where each user adapts its transmission probability vector to maximize an individual utility function. It is shown that existing understandings on stability and throughput of random access communication over collision and multi-packet reception channels can be exploited to design utility functions in the new system. Conditions under which the distributed medium access control game has a unique Nash equilibrium are obtained. Computer simulations show that, in a multiple access environment with a large number of users each being equipped with multiple transmission options, the game theoretic medium access control algorithm does favor low rate and parallel channel access options over high rate and exclusive channel access options. This is consistent with the well-known understandings in information theory. In Chapter 4, we present a stochastic approximation framework for a class of distributed MAC algorithms with guaranteed convergence to a unique system equilibrium. While the results are more or less standard in the stochastic approximation literature, they characterize the key conditions for convergence. Within the framework, the research problem becomes how one should design the system to place the unique equilibrium at the desired point that maximizes a chosen network utility and to make sure the conditions for convergence are satisfied. In Section 4.2 and 4.3, the case of single transmission option is discussed, under which the transmission scheme of each user is specified by a scalar transmission probability parameter. A distributed MAC algorithm is proposed to adapt the transmission probability of each user according to a channel contention measure defined as the success probability of a virtual packet. The MAC algorithm is then extended in Section 4.4 and 4.5 to the case when users have multiple transmission options. Simulation results are also included to demonstrate both the optimality and the convergence properties of the proposed MAC algorithms.

Chapter 2

Distributed Channel Coding Theorems

In a wireless network such as a Wi-Fi system, an increasing amount of messages are transmitted using distributed protocols, which are often featured with short bursty messages and opportunistic channel access. In distributed communication, users make their communication decisions individually and such decisions are not shared with other users or with the receiver. Due to lack of full user coordination, packet collisions happen occasionally. Such a communication model does not fall into the classical channel coding framework, which generally assumes joint coding optimization and long message transmission at the physical layer to achieve reliable message recovery. Therefore, fundamental limits of a distributed communication system cannot be understood without extending the classical channel coding tools.

Distributed channel coding theory, proposed in [4] [14] [5], assumes that each transmitter should be equipped with an ensemble of channel codes as opposed to one code assumed in classical channel coding theory. Code ensembles are shared off-line with the receiver, e.g., by specifying codebook generation algorithms in the physical layer protocol. Different codes can correspond to different communication settings such as different rate and power combinations. During online communication, possibly depending on a data link layer decision, each transmitter individually chooses a code to encode its messages. Without knowing the coding choices of the users, a receiver either decodes the messages of interest if a pre-determined decoding reliability requirement can be met, or reports collision otherwise. An achievable region is defined in [4] [5] as the set of code index vectors that support asymptotic reliable message recovery, and is shown to coincide with the Shannon information rate region in a sense explained in [4] [5]. Error performance bounds in the case of finite codeword length were obtained in [14] [5]. While fundamental understandings about distributed communication are greatly needed for packet-based wireless networks, coding theorems developed in [4] [14] [5] have not been attracting much attention in the research community so far.

In this Chapter, we will further extend the distributed channel coding theorems obtained in [4] [14] [5]. First, in [4] [5], achievable regions were defined not only as a function of the communication channel, but also as a function of the code ensembles selected by the users. We revise the definition to the one that only depends on the communication channel. Such a revision enables the definition of the distributed channel capacity, which is supported by the existing achievability proof and a new but quite straightforward converse proof. Second, error probability in a communication system is often dominated by a small number of error event types. In a distributed communication system, different error event types may or may not correspond to different code index vectors of the users. In [5, Theorem 3], the obtained achievable error performance bound contains a term that equals the probability of the worst case error event type multiplies the number of code index vectors outside the operation region. If the latter parameter takes a large value, the corresponding error performance bound can be very loose. We revise the definition to obtain a performance bound that essentially replaces the particular term with a summation of error probabilities each corresponding to one code index vector. The new error performance bound is tighter than the one obtained in [5] because the new bound is unlikely to scale in the number of code index vectors.

Throughout the Chapter, we only present results for channels with finite input and output alphabets. The results can be easily extended to channels with continuous input and output alphabets using the same approach for similar extensions in classical channel coding theory [8].

2.1 Distributed Multiple Access with Single User Decoding

Consider a multiple access system with K transmitters (users) and one receiver. Time is slotted with each time slot equaling the length of N channel symbols, and this is also the length of one codeword. Throughout Chapter 2, we assume that channel coding should be applied only within each time slot. Let the bold font variable represent a vector whose entries are the corresponding variables of all users. The discrete-time memoryless channel is modeled by a conditional distribution $P_{Y|\boldsymbol{X}}$, where $\boldsymbol{X} = [X_1, \dots, X_K] \in \boldsymbol{\mathcal{X}}$ is the channel input symbol vector with $\boldsymbol{\mathcal{X}}$ being the vector of finite input alphabets of all users, and $Y \in \mathcal{Y}$ is the channel output symbol with $\mathcal{Y}$

being the finite output alphabets. We assume that channel input alphabet of user k, denoted by $\mathcal{X}_k$, should be known at user k, for $k = 1, \ldots, K$, and the conditional distribution $P_{Y|\boldsymbol{X}}$ should be known at the receiver. Whether the conditional distribution $P_{Y|\boldsymbol{X}}$ is known to the transmitters or not doesn't affect the coding theorems to be proposed.

Each transmitter, say user k, is equipped with an ensemble of M channel codes, denoted by $\mathcal{G}_k^{(N)} = \{g_{k1}, \ldots, g_{kM}\}$. Let $\boldsymbol{\mathcal{G}}^{(N)}$ denote the vector of code ensembles of all users. Let $\boldsymbol{g} = [g_1, \ldots, g_K]$ be a code index vector. Define $\boldsymbol{g} \in \boldsymbol{\mathcal{G}}^{(N)}$ if $g_k \in \mathcal{G}_k^{(N)}$ for all $1 \leq k \leq K$. For each user k, code index $g_k \in \mathcal{G}_k^{(N)}$ represents a random block code described as follows. Let $\mathcal{L}_{g_k} = \left\{\mathcal{C}_{g_k\theta_k} : \theta_k \in \Theta_k^{(N)}\right\}$ be a library of codebooks, indexed by a set $\Theta_k^{(N)}$. Each codebook contains $\lfloor e^{Nr_{g_k}} \rfloor$ codewords of length N symbols, where r_{g_k} is a pre-determined parameter termed the "communication rate" (in nats/symbol) of code g_k. Let $[\mathcal{C}_{g_k\theta_k}(w_k)]_j$ denote the jth symbol of the codeword corresponding to message w_k in codebook $\mathcal{C}_{g_k\theta_k}$. At the beginning of each time slot, a codebook index θ_k is generated randomly according to a distribution $\gamma_k^{(N)}$. The distribution $\gamma_k^{(N)}$ and the codebooks $\mathcal{C}_{g_k\theta_k}$, $\forall g_k \in \mathcal{G}_k^{(N)}$, are chosen such that random variables $X_{g_k w_k j} : \theta_k \to [\mathcal{C}_{g_k\theta_k}(w_k)]_j$, $\forall j, w$ and $\forall g_k$, are i.i.d. according to a pre-determined input distribution $P_{g_k X_k}$. Assume that code library $\mathcal{L}_{g_k}$ and the value of θ_k are both known at the receiver. That is, the receiver knows the randomly generated codebook of g_k, and this is true for all codes and for all users. Note that this can be achieved by sharing the random codebook generation algorithms with the receiver. In the above description, we can see that a random block code g_k is characterized by its communication rate r_{g_k} and its input distribution $P_{g_k X_k}$. With an abuse of the notation, we regard $g_k = (r_{g_k}, P_{g_k X_k})$ as a variable representing a rate and distribution pair of user k, which is not a function of the codeword length N. Similarly, we regard $\boldsymbol{g} = (\boldsymbol{r_g}, \boldsymbol{P_{gX}})$ as a vector variable representing the rate and distribution pairs of all users. We will use "code space" to refer to the space of $\boldsymbol{g}$, which is also the space of rate vector and distribution vector pairs. We use $\boldsymbol{\mathcal{G}}$, i.e., without superscription (N), to represent a code ensemble in the code space where each $\boldsymbol{g} \in \boldsymbol{\mathcal{G}}$ represents a point in the code space.

At the beginning of each time slot, we assume that each user, say user k, arbitrarily chooses a code $g_k \in \mathcal{G}_k^{(N)}$, maps its message w_k to a codeword $X_{g_k}^{(N)}(w_k)$, and then sends the codeword through the channel. Here "arbitrary" refers to the assumption that the coding choice is made according to a data link layer protocol and is not controlled by, and even its statistical information may not be known to, the physical layer transmitter. Assume $(\boldsymbol{w}, \boldsymbol{g})$ is the actual message vector and code index vector chosen by the transmitters. Let $\boldsymbol{X}_{\boldsymbol{g}}^{(N)}(\boldsymbol{w})$ be the vector of codewords. Note that neither $\boldsymbol{g}$ nor $\boldsymbol{w}$ is known at the receiver.

We assume that the receiver is only interested in decoding the messages of user 1, but can choose to decode the messages of some other users if necessary. Because users choose their codes arbitrarily, reliable message decoding is not always possible. Upon receiving the channel output symbol sequence $Y^{(N)} = [Y_1, Y_2, \ldots, Y_N]$, the receiver either outputs an estimated message and code index pair $(\hat{w}_1, \hat{g}_1)$ for user 1, or reports collision for user 1. We assume that the receiver should choose an "operation region" $\boldsymbol{R}_1$ in the code space. Without knowing the actual message vector and code index vector pair $(\boldsymbol{w}, \boldsymbol{g})$, the receiver intends to decode the message of user 1 if $\boldsymbol{g} \in \boldsymbol{R}_1$, and intends to report collision for user 1 if $\boldsymbol{g} \notin \boldsymbol{R}_1$. Given the operation region $\boldsymbol{R}_1$ and conditioned on $\boldsymbol{g}$ being the actual code index vector, communication error probability as a function of $\boldsymbol{g}$ for codeword length N is defined as follows.

$$P_e^{(N)}(\boldsymbol{g}) = \begin{cases} \max_{\boldsymbol{w}} Pr\{(\hat{w}_1, \hat{g}_1) \neq (w_1, g_1) | (\boldsymbol{w}, \boldsymbol{g})\}, & \forall \boldsymbol{g} \in \boldsymbol{R}_1 \\ \max_{\boldsymbol{w}} 1 - Pr\left\{ \begin{array}{c} \text{"collision" or} \\ (\hat{w}_1, \hat{g}_1) = (w_1, g_1) \end{array} \middle| (\boldsymbol{w}, \boldsymbol{g}) \right\}, & \forall \boldsymbol{g} \notin \boldsymbol{R}_1 \end{cases} \quad (2.1)$$

Note that in the above error probability definition, for $\boldsymbol{g} \notin \boldsymbol{R}_1$, both correct message decoding and collision report are regarded as acceptable channel outcomes. In other words, collision report is not strictly enforced for $\boldsymbol{g} \notin \boldsymbol{R}_1$. A more general error probability definition will be discussed in Section 2.3.

Definition 1. *We say that an operation region $\boldsymbol{R}_1$ is asymptotically achievable for a multiple access channel $P_{Y|\boldsymbol{X}}$ for user 1, if for all finite M and all code ensemble vectors $\boldsymbol{\mathcal{G}}$ with each entry of* *code ensemble having a cardinality of M, decoding algorithms can be designed for the sequence of random code ensembles $\boldsymbol{\mathcal{G}}^{(N)}$ to achieve* $\lim_{N\to\infty} P_e^{(N)}(\boldsymbol{g}) = 0, \forall \boldsymbol{g} \in \boldsymbol{\mathcal{G}}$.

Compared with the achievable region definition given in [5, Section III], the achievable region defined in Definition 1 is only a function of the multiple access channel. It does not depend on the particular code ensembles $\boldsymbol{\mathcal{G}}$ chosen by the users. The following theorem is directly implied by the achievable region definition and the error probability definition given in (2.1).

Theorem 1. *For a discrete-time memoryless multiple access channel $P_{Y|\boldsymbol{X}}$ with finite input and output alphabets, if an operation region $\boldsymbol{R}_1$ is asymptotically achievable for user 1, then any subset $\tilde{\boldsymbol{R}}_1 \subseteq \boldsymbol{R}_1$ is also asymptotically achievable for user 1.*

The following theorem characterizes the maximum achievable region of multiple access channel $P_{Y|\boldsymbol{X}}$ for user 1.

Theorem 2. *For a discrete memoryless multiple access channel $P_{Y|\boldsymbol{X}}$ with finite input and output alphabets, the following region $\boldsymbol{C}_{d1}$ in the code space is asymptotically achievable for user 1.*

$$\boldsymbol{C}_{d1} = \left\{ \boldsymbol{g} \left| \begin{array}{l} \boldsymbol{g} = (\boldsymbol{r}_{\boldsymbol{g}}, \boldsymbol{P}_{\boldsymbol{g}\boldsymbol{X}}), \forall S \subseteq \{1, \dots, K\}, 1 \in S, \\ \exists \tilde{S} \subseteq S, 1 \in \tilde{S}, \textit{such that,} \\ \sum_{k\in\tilde{S}} r_{g_k} < I_{\boldsymbol{g}}(\boldsymbol{X}_{\tilde{S}}; Y | \boldsymbol{X}_{\bar{S}}) \end{array} \right. \right\}, \tag{2.2}$$

where $\bar{S}$ is the compliment set of S, $\boldsymbol{X}_{\bar{S}}$ is a vector of channel input symbols of users not in S, and $I_{\boldsymbol{g}}(\boldsymbol{X}_{\tilde{S}}; Y|\boldsymbol{X}_{\bar{S}})$ denotes the mutual information between $\boldsymbol{X}_{\tilde{S}}$ and Y given $\boldsymbol{X}_{\bar{S}}$ with respect to joint distribution $P_{\boldsymbol{X}Y} = P_{Y|\boldsymbol{X}} \prod_{k=1}^{K} P_{g_k X_k}$.

The achievable region $\boldsymbol{C}_{d1}$ is maximum in the sense that for any region $\boldsymbol{R}_1$ that is asymptotically achievable for user 1, we must have $\boldsymbol{R}_1 \subseteq \boldsymbol{C}_{d1}^c$, where $\boldsymbol{C}_{d1}^c$ is the closure of $\boldsymbol{C}_{d1}$.

The proof of Theorem 2 is given in [1, Appendix A].

Theorem 2 can be extended from decoding for a single user to decoding for a user subset.

Definition 2. *Let* $S_0 \subseteq \{1, \ldots, K\}$ *be a user subset. We say that an operation region* $\boldsymbol{R}_{S_0}$ *is asymptotically achievable for multiple access channel* $P_{Y|\boldsymbol{X}}$ *for user subset* S_0*, if* $\forall k \in S_0$, $\boldsymbol{R}_{S_0}$ *is asymptotically achievable for user* k*.*

Corollary 1. *For a discrete memoryless multiple access channel* $P_{Y|\boldsymbol{X}}$ *with finite input and output alphabets, let* C_{dk} *be the maximum achievable region for user* k*. The expression of* C_{dk} *can be obtained from (2.2) by replacing user index* 1 *with user index* k*. Let* $S_0 \subseteq \{1, \ldots, K\}$ *be a user subset. The maximum achievable region for user subset* S_0 *is given by*

$$\boldsymbol{C}_{dS_0} = \bigcap_{k \in S_0} \boldsymbol{C}_{dk} = \left\{ \boldsymbol{g} \left| \begin{array}{l} \boldsymbol{g} = (\boldsymbol{r}_{\boldsymbol{g}}, \boldsymbol{P}_{\boldsymbol{g}\boldsymbol{X}}), \forall S \subseteq \{1, \ldots, K\}, \\ S \cap S_0 \neq \phi, \exists \tilde{S}, S \cap S_0 \subseteq \tilde{S} \subseteq S, \\ \textit{such that, } \sum_{k \in \tilde{S}} r_{g_k} < I_{\boldsymbol{g}}(\boldsymbol{X}_{\tilde{S}}; Y | \boldsymbol{X}_{\bar{S}}) \end{array} \right. \right\}, \tag{2.3}$$

where ϕ *is the empty set.*

Corollary 1 can be obtained by following the proof of [4, Theorem 4].

Note that, according to [5, Theorem 5], Theorem 2 and Corollary 1 still hold even if we strictly enforce collision report for $\boldsymbol{g} \notin \boldsymbol{R}_1$, by changing the error probability definition to the following.

$$P_e^{(N)}(\boldsymbol{g}) = \begin{cases} \max_{\boldsymbol{w}} Pr\{(\hat{w}_1, \hat{g}_1) \neq (w_1, g_1) | (\boldsymbol{w}, \boldsymbol{g})\}, \forall \boldsymbol{g} \in \boldsymbol{R}_1 \\ \max_{\boldsymbol{w}} 1 - Pr\left\{\text{“collision”} \middle| (\boldsymbol{w}, \boldsymbol{g})\right\}, \quad \forall \boldsymbol{g} \notin \boldsymbol{R}_1 \end{cases} \tag{2.4}$$

However, Theorem 1 does depend on error probability definition in (2.1), where we regard correct message decoding as an acceptable outcome for $\boldsymbol{g} \notin \boldsymbol{R}_1$. Because the receiver does not always decode the messages of users other than user 1, and the receiver may not be able to correctly detect the part of the code index vector $\boldsymbol{g}$ corresponding to the un-decoded users. Therefore, the receiver may not be able to tell whether the actual code index vector $\boldsymbol{g}$ satisfies $\boldsymbol{g} \in \boldsymbol{R}_1$ or not. With the error probability definition (2.1), correct detection of the full code index vector is not required. That is, so long as the receiver does not output an erroneous message for user 1, whether the receiver guarantees collision report for $\boldsymbol{g} \notin \boldsymbol{R}_1$ or not is not a concern to the system design.

Alternatively, suppose we only accept collision report for $\boldsymbol{g} \notin \boldsymbol{R}_1$ and change the the error probability definition to (2.4). The receiver will have to detect whether $\boldsymbol{g} \in \boldsymbol{R}_1$ or $\boldsymbol{g} \notin \boldsymbol{R}_1$. Depending on the feasibility of such a detection task, Theorem 1 may no longer hold. That is, even if a region $\boldsymbol{R}_1$ is asymptotically achievable for user 1, there may exist a subset $\tilde{\boldsymbol{R}}_1 \subseteq \boldsymbol{R}_1$ that is not asymptotically achievable for user 1. A simple example of such a situation is illustrated below.

Example 2.1: Consider a distributed multiple access system with two users. Let X_1, X_2 be the channel input symbols of the two users, and let Y be the channel output symbol, all having finite alphabets. Assume that input symbols of user 2 have no impact on the channel output. That is, the channel model satisfies $P(Y|X_1, X_2) = P(Y|X_1)$. With the error probability definition of (2.4), according to Theorem 2, the region $\boldsymbol{R}_1 = \left\{ \boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \middle| r_{g_1} < I_{g_1}(X_1; Y) \right\}$ is asymptotically achievable for user 1. However, a subset $\tilde{\boldsymbol{R}}_1 = \left\{ \boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \middle| r_{g_1} < I_{g_1}(X_1; Y), r_{g_2} < 0.5 \right\}$ with $\tilde{\boldsymbol{R}}_1 \subseteq \boldsymbol{R}_1$ is not asymptotically achievable for user 1. This is because the receiver has no capability of detecting the communication rate of user 2, and therefore cannot tell whether $r_{g_2} < 0.5$ is true or false (or equivalently, whether or not $\boldsymbol{g} \in \tilde{\boldsymbol{R}}_1$).

Let us come back to the error probability definition of (2.1). With the support of Theorem 2 and Corollary 1, we define $\boldsymbol{C}_{d1}$ as the "distributed capacity" for user 1, and $\boldsymbol{C}_{dS_0}$ as the "distributed capacity" for user subset S_0, of multiple access channel $P_{Y|\boldsymbol{X}}$. Interestingly, the distributed capacity can indeed be regarded as an extension to the classical Shannon capacity in the following sense.

Let $\boldsymbol{C}_d$ be the distributed capacity of the multiple access channel when the receiver is interested in decoding the messages of all users. According to Corollary 1, $\boldsymbol{C}_d$ is given by

$$\boldsymbol{C}_d = \left\{ \boldsymbol{g} \middle| \boldsymbol{g} = (\boldsymbol{r}_{\boldsymbol{g}}, \boldsymbol{P}_{\boldsymbol{g}\boldsymbol{X}}), \forall S \subseteq \{1, \dots, K\}, \sum_{k \in S} r_{g_k} < I_{\boldsymbol{g}}(\boldsymbol{X}_S; Y|\boldsymbol{X}_{\bar{S}}) \right\}. \quad (2.5)$$

It is well known that Shannon capacity of the multiple access channel [7], denoted by $\boldsymbol{C}$, is given by

$$\boldsymbol{C} = \text{convex hull}\left(\left\{\boldsymbol{r}\left|\exists \boldsymbol{P}_{\boldsymbol{X}}, \forall S \subseteq \{1,\ldots,K\},\ \sum_{k\in S} r_k \le I(\boldsymbol{X}_S; Y|\boldsymbol{X}_{\bar{S}})\right.\right\}\right), \quad (2.6)$$

where $I(\boldsymbol{X}_S; Y|\boldsymbol{X}_{\bar{S}})$ is calculated with respect to joint distribution $P_{\boldsymbol{X}Y} = P_{Y|\boldsymbol{X}} \prod_{k=1}^{K} P_{X_k}$. From (2.5) and (2.6), we can see that the two capacity terms satisfy

$$\boldsymbol{C}^c = \text{convex hull}\left(\{\boldsymbol{r}|\exists \boldsymbol{g} \in \boldsymbol{C}_d^c, \boldsymbol{r}_{\boldsymbol{g}} = \boldsymbol{r}\}\right). \quad (2.7)$$

However, the same capacity region has different meanings under different communication models. In coordinated communication, Shannon capacity region suggests that users should jointly choose a rate vector within the capacity region to guarantee reliable message delivery. In distributed communication, on the other hand, users choose their rates individually and the chosen rate vector could lie inside or outside the capacity region. If the rate vector happens to locate inside the capacity region, the receiver can detect it and decode the messages reliably. If the rate vector happens to locate outside the capacity region, the receiver can reliably detect it and report collision.

Similar to classical channel coding theory, Theorem 2 and Corollary 1 hold even if input and output alphabets of the channel are continuous. One can also pose a constraint in the code space to limit the coding choices of the users, and to define the constrained distributed channel capacity accordingly.

Example 2.2: Consider a K-user multiple access system over a discrete-time memoryless channel with additive Gaussian noise. The channel is modeled by

$$Y = \sum_{k=1}^{K} X_k + V, \quad (2.8)$$

where V is the Gaussian noise with zero mean and variance N_0. Assume that each user k can only choose random block codes with Gaussian input distribution of zero mean and variance P_k. With the input distributions being fixed, and if the receiver is only associated to user 1, then according to Theorem 2, the maximum achievable region for user 1 is given by

$$\boldsymbol{C}_{d1} = \left\{ \boldsymbol{r_g} \middle| \begin{array}{l} \forall S \subseteq \{1, \dots, K\}, 1 \in S, \exists \tilde{S} \subseteq S, 1 \in \tilde{S}, \\ \text{such that, } \sum_{k \in \tilde{S}} r_{g_k} < \frac{1}{2} \log \left(1 + \frac{\sum_{k \in \tilde{S}} P_k}{\sum_{k \in S \setminus \tilde{S}} P_k + N_0} \right) \end{array} \right\}. \tag{2.9}$$

Similarly, one can also use Theorem 2 and Corollary 1 to obtain the maximum achievable region for any other user and for any user group. If the receiver is interested in decoding messages of all users, closures of the constrained distributed channel capacity and the Shannon capacity both equal the following rate region.

$$\boldsymbol{C}_d^c = \boldsymbol{C}^c = \left\{ \boldsymbol{r_g} \middle| \forall S \subseteq \{1, \dots, K\}, \sum_{k \in S} r_{g_k} \leq \frac{1}{2} \log \left(1 + \frac{\sum_{k \in S} P_k}{N_0} \right) \right\}. \tag{2.10}$$

When $K = 2$ and $P_1 = P_2 = 5N_0$, the maximum achievable region for user 1 and the maximum achievable region for user 2 are illustrated respectively in Figure 2.1, and the capacity region is a pentagon illustrated in Figure 2.2. The rates in both figures are measured in nats/symbol. It can be seen that intersection of the two regions equals the constrained channel capacity, i.e. the pentagon region, as illustrated in Figure 2.2.

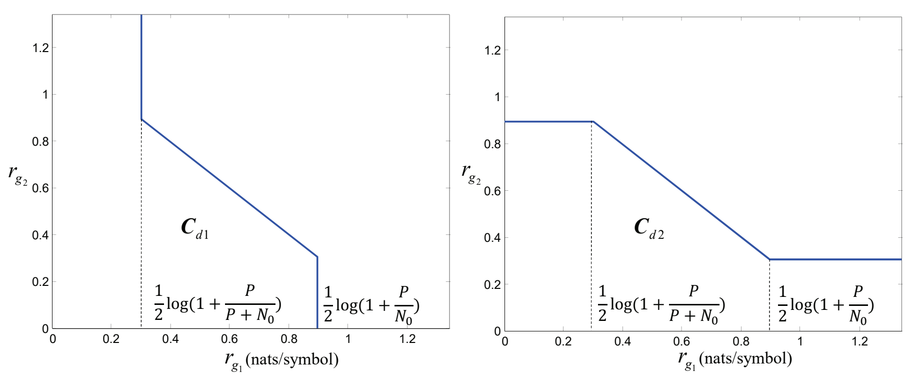

Figure 2.1: Maximum achievable regions for each individual user of a two user Gaussian channel.

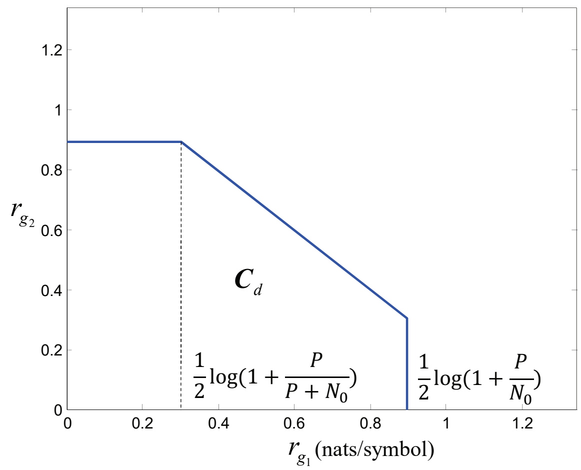

Figure 2.2: Distributed capacity of a two user Gaussian channel.

2.2 Interfering User and Compound Channel

In this section, we extend the coding theorems presented in Section 2.1 to the case when the system has an "interfering user". As explained in [5], an interfering user can be a remote user whose codebook is unknown to the receiver, and hence its messages are not decodable at the receiver. A "virtual" interfering user can also be used to model a compound channel whose realization affects the conditional channel distribution experienced by the users, but it is "virtual" in the sense of having no messages to be decoded at the receiver [5].

Assume that, in addition to the K regular users indexed by $\{1, \dots, K\}$, there is an interfering user indexed as user 0. Assume that the interfering user is equipped with M communication options, denoted by $\mathcal{G}_0 = \{g_{01}, \dots, g_{0M}\}$. For convenience, we still use $\mathcal{G}_0$ and $g_0 \in \mathcal{G}_0$ to represent a code ensemble and a code index of user 0, respectively. With the existence of the interfering user, the multiple access channel is now modeled by a conditional distribution $P_{Y|\boldsymbol{X}}(g_0)$, which is a function of the "coding" choice of the interfering user. Note that channel function $P_{Y|\boldsymbol{X}}(g_0)$ can be defined for a domain of g_0 that is beyond the ensemble $\mathcal{G}_0$. At the beginning of each time slot, assume that the interfering user should arbitrarily choose a "code" g_0, and this determines the multiple access channel $P_{Y|\boldsymbol{X}}(g_0)$ to be experienced by the regular users. The receiver knows the channel functions $P_{Y|\boldsymbol{X}}(g_0)$ for all $g_0 \in \mathcal{G}_0$, but does not know the value g_0 chosen by the interfering user. Let vectors $\boldsymbol{g}$ and $\boldsymbol{\mathcal{G}}$ now contain the entries of the regular users and the interfering user, while vectors $\boldsymbol{w}$ and $\boldsymbol{X}$ still only contain the entries of the regular users.

As in Section 2.1, assume that the receiver is only interested in decoding the messages of user 1. Let $(\boldsymbol{w}, \boldsymbol{g})$ be the actual message vector and code index vector pair, unknown to the receiver. The receiver should choose an operation region $\boldsymbol{R}_1$ in the space of $\boldsymbol{g}$. The receiver intends to decode the message of user 1 if $\boldsymbol{g} \in \boldsymbol{R}_1$, and intends to report collision for user 1 if $\boldsymbol{g} \notin \boldsymbol{R}_1$.

Theorem 3. *For a discrete-time memoryless multiple access channel $P_{Y|\boldsymbol{X}}(g_0)$ with finite input and output alphabets and with g_0 being the code index of an interfering user, conclusions of Theorems 1, 2, and Corollaries 1 still hold, if the following extensions are applied to the statements in the theorems, corollaries and in their proofs.*

1. Channel input vectors $\boldsymbol{X}$, rate vectors $\boldsymbol{r}_{\boldsymbol{g}}$, input distribution vectors $\boldsymbol{P}_{\boldsymbol{g}\boldsymbol{X}}$ should only contain entries corresponding to the regular users $1, \dots, K$.

2. Code index vectors $\boldsymbol{g} = (\boldsymbol{r}_{\boldsymbol{g}}, \boldsymbol{P}_{\boldsymbol{g}\boldsymbol{X}}, g_0)$ as well as code ensemble vector $\boldsymbol{\mathcal{G}}$ should contain one more entry corresponding to the code index of the interfering user.

3. Given code index vector $\boldsymbol{g}$, mutual information function $I_{\boldsymbol{g}}()$, entropy function $H_{\boldsymbol{g}}()$, and probability function $p_{\boldsymbol{g}}()$ should all be computed with respect to joint distribution given by

$$P_{\boldsymbol{X}Y} = P_{Y|\boldsymbol{X}}(g_0) \prod_{k=1}^{K} P_{g_k X_k}, \tag{2.11}$$

i.e., with a channel function of $P_{Y|\boldsymbol{X}}(g_0)$.

4. User subsets $S \subseteq \{1, \dots, K\}$ should only contain the regular users. The complement set $\bar{S}$ should be defined as $\bar{S} = \{1, \dots, K\} \setminus S$, i.e., excluding the interfering user.

5. The maximum number of possible code index vectors should be upper bounded by M^{K+1}.

With the above extensions, if error probability is defined in (2.1), then any subset of an achievable region should also be achievable. C_{d1} given in (2.2) is the maximum asymptotically achievable region for user 1, and $\boldsymbol{C}_{dS_0}$ given in (2.3) is the maximum asymptotically achievable region for user subset $S_0 \subseteq \{1, \dots, K\}$.

The proof of Theorem 3 is skipped.

Example 2.3: Consider a single user communication system over a discrete-time memoryless channel with an unknown channel gain and additive Gaussian noise. The channel is modeled by

$$Y = hX + V, \tag{2.12}$$

where $h \geq 0$ is the unknown channel gain and V is the Gaussian noise with zero mean and variance N_0.

Let us pose the constraint that input distributions of all coding options must be zero mean with variance P. We can formulate the problem by constructing a system with two users. User 1 is the regular user whose coding options $\mathcal{G}_1 = \{r_1, \dots, r_{M_1}\}$ represent an ensemble of Gaussian random block codes with the same input distribution but with different rates. User 0 is a interfering (or virtual) user whose communication options $\mathcal{G}_1 = \{h_1, \dots, h_{M_0}\}$ represent the ensemble of compound gains that can possibly be taken by the channel. Consequently, distributed capacity region of the system is simply a region in the space of rate and channel gain pairs given by

$$\boldsymbol{C}_{d1} = \left\{ (r_{g_1}, h) \middle| r_{g_1} < \frac{1}{2} \log \left(1 + \frac{h^2 P}{N_0} \right) \right\}, \tag{2.13}$$

where the rate is measured in nats/symbol. The capacity region is illustrated in Figure 2.3 for $P = 5N_0$.

2.3 Performance with A Finite Codeword Length

While performance bounds on tradeoffs among decoding error probability, communication rate, and codeword length have been extensively investigated in classical channel coding theory [12] [15] [16] [17] [18] [19] [20], the distributed communication model introduced in Section 2.1 and 2.2 brought several new challenges that must be carefully considered. First, because data packets in distributed communication are relatively short in length, validity of the obtained tradeoff bounds should not require a large codeword length. Second, each user in a distributed communica-

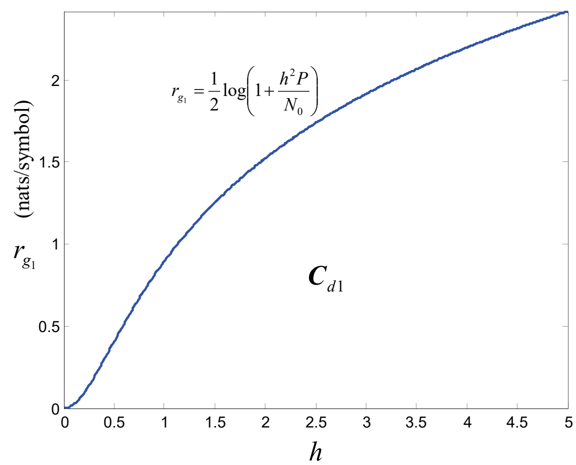

Figure 2.3: Distributed capacity region of a single user system over a Gaussian channel with an unknown channel gain.

tion system can choose its code from the ensemble arbitrarily. Different coding choices may lead to different types of error events such as decoding error and collision detection error. Therefore, when analyzing error probability performance of a distributed communication system, one may want to assign different weights to the probabilities of different error events. Faced with these challenges, in this section, we present the non-asymptotic analysis when the codeword length is finite and could be small in value. Throughout the section, codeword length N is assumed to be fixed at a constant.

As explained in [5], we will first need to consider an auxilliary decoder called the $(D, \boldsymbol{R}_D)$ decoder. Let $D \subseteq \{1, \dots, K\}$ be a subset of regular users with $1 \in D$. Assume that the receiver chooses an operation region $\boldsymbol{R}_D$ and an operation margin $\widehat{\boldsymbol{R}}_D$ both defined in the code space with $\boldsymbol{R}_D \cap \widehat{\boldsymbol{R}}_D = \phi$. A $(D, \boldsymbol{R}_D)$ decoder intends to decode the messages of all users in D by regarding messages from all other users as interference. Let $(\boldsymbol{w}, \boldsymbol{g})$ be the actual message vector and code index vector pair. For $\boldsymbol{g} \in \boldsymbol{R}_D$, the decoder intends to decode the messages of users in D. For $\boldsymbol{g} \in \widehat{\boldsymbol{R}}_D$, the decoder intends to either decode the messages or to report collision for users in D. For $\boldsymbol{g} \notin \boldsymbol{R}_D \cup \widehat{\boldsymbol{R}}_D$, the decoder intends to enforce collision report for users in D. Let $(\hat{\boldsymbol{w}}_D, \hat{\boldsymbol{g}}_D)$ be the estimated message vector and code index vector for users in D. Given $\boldsymbol{g}$, conditional error probability as a function of $\boldsymbol{g}$ is given by

$$P_e(\boldsymbol{g}) = \begin{cases} \max_{\boldsymbol{w}_D} Pr\{(\hat{\boldsymbol{w}}_D, \hat{\boldsymbol{g}}_D) \neq (\boldsymbol{w}_D, \boldsymbol{g}_D) | (\boldsymbol{w}_D, \boldsymbol{g})\}, & \forall \boldsymbol{g} \in \boldsymbol{R}_D \\ \max_{\boldsymbol{w}_D} 1 - Pr\left\{ \begin{array}{l} \text{“collision” or} \\ (\hat{\boldsymbol{w}}_D, \hat{\boldsymbol{g}}_D) = (\boldsymbol{w}_D, \boldsymbol{g}_D) \end{array} \middle| (\boldsymbol{w}_D, \boldsymbol{g}) \right\}, & \forall \boldsymbol{g} \in \widehat{\boldsymbol{R}}_D \\ \max_{\boldsymbol{w}_D} 1 - Pr\left\{\text{“collision”} | (\boldsymbol{w}_D, \boldsymbol{g})\right\}, & \forall \boldsymbol{g} \notin \boldsymbol{R}_D \cup \widehat{\boldsymbol{R}}_D \end{cases} \quad (2.14)$$

Let $\{\alpha_{\boldsymbol{g}}\}$ be a set of pre-determined weight parameters each being assigned to a code index vector $\boldsymbol{g} \in \boldsymbol{\mathcal{G}}$, such that

$$\left\{ \alpha_{\boldsymbol{g}} \middle| \alpha_{\boldsymbol{g}} \geq 0, \forall \boldsymbol{g} \in \boldsymbol{\mathcal{G}}, \sum_{\boldsymbol{g}} e^{-N\alpha_{\boldsymbol{g}}} = 1 \right\}. \quad (2.15)$$

We define the “generalized error performance” of the $(D, \boldsymbol{R}_D)$ decoder as

$$\text{GEP}_D = \sum_{\boldsymbol{g}} P_e(\boldsymbol{g}) e^{-N\alpha_{\boldsymbol{g}}}. \quad (2.16)$$

Let us use $P_{g_k}(X_k)$ to denote the probability of channel input symbol X_k under coding option g_k, and use $P(Y|\boldsymbol{X}_D, \boldsymbol{g}_{\bar{D}})$ to denote the conditional probability of channel output symbol Y given input symbol vector $\boldsymbol{X}_D$ for users in D, and code index vector $\boldsymbol{g}_{\bar{D}}$ for users not in D. The following theorem gives an achievable bound, improved from the corresponding bound presented in [5, Theorem 3], for the generalized error performance of the $(D, \boldsymbol{R}_D)$ decoder.

Theorem 4. *Consider the distributed multiple access system described above. There exists a decoding algorithm such that GEP$_D$ is upper bounded by*

$$\begin{aligned} GEP_D \leq \sum_{\boldsymbol{g} \in \boldsymbol{R}_D} \Bigg\{ \sum_{S \subset D} \Bigg[\sum_{\tilde{\boldsymbol{g}} \in \boldsymbol{R}_D, \tilde{\boldsymbol{g}}_S = \boldsymbol{g}_S} \exp(-NE_{mD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, S)) \\ +2 \sum_{\tilde{\boldsymbol{g}} \notin \boldsymbol{R}_D, \tilde{\boldsymbol{g}}_S = \boldsymbol{g}_S} \exp(-NE_{iD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, S)) \Bigg] +2 \sum_{\tilde{\boldsymbol{g}} \notin \boldsymbol{R}_D \cup \widehat{\boldsymbol{R}}_D, \tilde{\boldsymbol{g}}_D = \boldsymbol{g}_D} \exp(-NE_{iD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, D)) \Bigg\}, \end{aligned} \quad (2.17)$$

where $E_{mD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, S)$, $E_{iD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, S)$ for $S \subset D$ and $E_{iD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, D)$ in the above equation are given by

$$
\begin{aligned}
& E_{mD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, S) = \max_{0<\rho\leq 1} -\rho \sum_{k\in D\setminus S} r_{\tilde{g}_k} + \max_{0\leq s\leq 1} -\log \sum_{Y} \sum_{\boldsymbol{X}_S} \prod_{k\in S} P_{g_k}(X_k) \\
& \times \left(\sum_{\boldsymbol{X}_{D\setminus S}} \left[P(Y|\boldsymbol{X}_D, \boldsymbol{g}_{\bar{D}}) e^{-\alpha \boldsymbol{g}} \right]^{1-s} \prod_{k\in D\setminus S} P_{g_k}(X_k) \right) \\
& \times \left(\sum_{\boldsymbol{X}_{D\setminus S}} \left[P(Y|\boldsymbol{X}_D, \tilde{\boldsymbol{g}}_{\bar{D}}) e^{-\alpha \tilde{\boldsymbol{g}}} \right]^{\frac{s}{\rho}} \prod_{k\in D\setminus S} P_{\tilde{g}_k}(X_k) \right)^{\rho}, \\
& E_{iD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, S) = \max_{0<\rho\leq 1} -\rho \sum_{k\in D\setminus S} r_{g_k} + \max_{0\leq s\leq 1-\rho} -\log \sum_{Y} \sum_{\boldsymbol{X}_S} \prod_{k\in S} P_{g_k}(X_k) \times \\
& \left(\sum_{\boldsymbol{X}_{D\setminus S}} \left[P(Y|\boldsymbol{X}_D, \boldsymbol{g}_{\bar{D}}) e^{-\alpha \boldsymbol{g}} \right]^{\frac{s}{s+\rho}} \prod_{k\in D\setminus S} P_{g_k}(X_k) \right)^{s+\rho} \\
& \times \left(\sum_{\boldsymbol{X}_{D\setminus S}} P(Y|\boldsymbol{X}_D, \tilde{\boldsymbol{g}}_{\bar{D}}) e^{-\alpha \tilde{\boldsymbol{g}}} \prod_{k\in D\setminus S} P_{\tilde{g}_k}(X_k) \right)^{1-s}. \\
& E_{iD}(\boldsymbol{g}, \tilde{\boldsymbol{g}}, D) = \max_{0\leq s\leq 1} -\log \sum_{Y} \sum_{\boldsymbol{X}_D} \prod_{k\in D} P_{g_k}(X_k) \\
& \left[P(Y|\boldsymbol{X}_D, \boldsymbol{g}_{\bar{D}}) e^{-\alpha \boldsymbol{g}} \right]^{s} \left[P(Y|\boldsymbol{X}_D, \tilde{\boldsymbol{g}}_{\bar{D}}) e^{-\alpha \tilde{\boldsymbol{g}}} \right]^{1-s}.
\end{aligned} \tag{2.18}
$$

The proof of Theorem 4 is given in [1, Appendix B]. Compared with the bound presented in [5, Equation (7)], besides other minor improvements, the second and the third terms on the right hand side of (2.17) lead to a tighter bound because, if the summations are dominated by only a small number of terms, then the summations should not scale in the number of code index vectors satisfying $\tilde{\boldsymbol{g}} \notin \boldsymbol{R}_D$.

Let us now consider the case when the receiver is only interested in decoding the message of user 1 but can choose to decode the messages of other users if necessary. Assume that the receiver should choose an operation region $\boldsymbol{R}_1$ and an operation margin $\widehat{\boldsymbol{R}}_1$ in the code space with $\boldsymbol{R}_1 \cap \widehat{\boldsymbol{R}}_1 = \phi$. Let $\boldsymbol{g}$ be the actual code index vector. The receiver intends to decode the message of user 1 for $\boldsymbol{g} \in \boldsymbol{R}_1$, to either decode the messages of user 1 or to report collision for user 1 for $\boldsymbol{g} \in \widehat{\boldsymbol{R}}_1$, and to report collision for user 1 for $\boldsymbol{g} \notin \boldsymbol{R}_1 \cup \widehat{\boldsymbol{R}}_1$.

Let $(\hat{w}_1, \hat{g}_1)$ be the message and code index estimate of user 1. Let $(\boldsymbol{w}, \boldsymbol{g})$ be the actual message vector and code index vector pair, conditional error probability of the system as a function of $\boldsymbol{g}$ is defined as

$$P_e(\boldsymbol{g}) = \begin{cases} \max_{w_1} Pr\{(\hat{w}_1, \hat{g}_1) \neq (w_1, g_1)|(w_1, \boldsymbol{g})\}, & \forall \boldsymbol{g} \in \boldsymbol{R}_1 \\ \max_{w_1} 1 - Pr \left\{ \begin{array}{l} \text{“collision” or} \\ (\hat{w}_1, \hat{g}_1) = (w_1, g_1)|(w_1, \boldsymbol{g}) \end{array} \middle| (w_1, \boldsymbol{g}) \right\}, & \forall \boldsymbol{g} \in \widehat{\boldsymbol{R}}_1 \\ \max_{w_1} 1 - Pr \left\{\text{“collision”}|(w_1, \boldsymbol{g})\right\}, & \forall \boldsymbol{g} \notin \boldsymbol{R}_1 \cup \widehat{\boldsymbol{R}}_1 \end{cases} \quad (2.19)$$

Let $\{\alpha_{\boldsymbol{g}}\}$ be a set of pre-determined weight parameters each being assigned to a code index vector $\boldsymbol{g} \in \boldsymbol{\mathcal{G}}$ and satisfying constraint (2.15). We define the “generalized error performance” of the system as

$$\text{GEP} = \sum_{\boldsymbol{g}} P_e(\boldsymbol{g}) e^{-N\alpha_{\boldsymbol{g}}}. \quad (2.20)$$

According to [5, Theorem 4], an achievable bound on the generalized error performance of the system is given in the following theorem.

Theorem 5. *Consider the distributed multiple access system described above. Assume that the receiver is only interested in decoding the message of user* 1*. Let* $\boldsymbol{R}_1$ *be the operation region,* $\widehat{\boldsymbol{R}}_1$ *be the operation margin, and* $\{\alpha_{\boldsymbol{g}}\}$ *be the set of weight parameters. Let* σ *be a partition of the operation region* $\boldsymbol{R}_1$*, as described below*

$$\begin{array}{l} \boldsymbol{R}_1 = \displaystyle\bigcup_{D, D \subseteq \{1,\ldots,K\}, 1 \in D} \boldsymbol{R}_D, \qquad \boldsymbol{R}_{D'} \cap \boldsymbol{R}_D = \phi, \\ \forall D, D' \subseteq \{1, \ldots, K\}, D' \neq D, 1 \in D, D'. \end{array} \quad (2.21)$$

There exists a decoding algorithm such that the generalized error performance defined in (2.20) is upper bounded by

$$GEP \leq \min_{\sigma} \sum_{D, D \subseteq \{1,\ldots,K\}, 1 \in D} GEP_D, \quad (2.22)$$

where GEP_D represents the generalized error probability of the $(D, \boldsymbol{R}_D)$ decoder with receiver decoding the messages of all and only the users in D, with the operation region being $\boldsymbol{R}_D$ and the operation margin being $\widehat{\boldsymbol{R}}_D = \boldsymbol{R}_1 \cup \widehat{\boldsymbol{R}}_1 \setminus \boldsymbol{R}_D$.

The proof of Theorem 5 is provided in [1, Appendix B].

Note that Theorem 5 did not provide an explicit algorithm to calculate the partition that minimizes either $\sum_{D, D \subseteq \{1,\ldots,K\}, 1 \in D} \text{GEP}_D$ or its upper bound obtained from (2.17). To find the optimal partition, one may need to compute every single term on the right hand sides of (2.17), (2.18) and (2.22) for all code index vectors and for all user subsets. Complexity of such calculations is beyond the scope of this thesis.

Chapter 3

Medium Access Control Game

Classical medium access control (MAC) protocols assume that a link-layer user (transmitter) should choose either to idle in a time slot or to transmit a packet with pre-determined communication parameters. Under this assumption, when users in a distributed wireless network experience packet collisions, the only approach to control contention is to reduce and randomize their transmission activities [10]. While such a model and its derived contention control approaches are widely adopted in distributed MAC protocols such as the DCF protocol in 802.11, they do not permit exploitation at the link layer of the well known information theoretic result that parallel transmission with carefully controlled rates achieves the optimal sum throughput of a multiple access system.

The extended channel coding theorems introduced in Chapter 2 enhances the classical physical-link layer interface in the sense of giving a link layer user multiple transmission options corresponding to different communication settings such as different rates and power. It also enables the derivation and analysis of link layer channel model and medium access control performance using physical layer channel properties. Consequently, link layer users can now exploit advanced communication adaption approaches, such as rate adaptation, to improve channel sharing efficiency in a distributed networking. Understanding the impact of the enhanced physical-link layer interface on the strategy of link layer communication adaptation and contention control therefore becomes an important topic. Note that, while navigating through the provided transmission options enables the capability of advanced communication adaptation, due to the layering architecture (or more precisely, the modularity requirement), a link layer user is bounded with the provided transmission options and can only construct its transmission scheme within this constraint to optimize a chosen network utility.

For a wide range of distributed networks, game theoretic problem formulations and analyses have proven to provide new insights to reverse/forward engineering of existing MAC protocols for improved fairness and higher throughput, and for decoupling contention control from handling failed packets [21] [22]. The key idea of a game theoretic MAC algorithm is to control contention by distributively adapting transmission probabilities of the users to optimize their individual utilities, each being carefully chosen as a function of the transmission cost and the experienced contention level of the corresponding user. Utility function design often requires a good understanding on the expected contention level and the desired transmission probability, derived from performance objectives of the users. While contention control in a distributed wireless network has been rigorously investigated under the classical link layer model, extending the understandings to the case when each link layer user has multiple transmission options is a new research direction that deserves careful and extensive exploration. In Chapter 3 and Chapter 4, the problem of distributed MAC in a wireless network with and without an enhanced physical-link layer interface will be investigated from two different perspectives. In this Chapter, we model the distributed MAC as a non-cooperative game, where each user should adapt its transmission scheme to maximize an individual utility function according to the available channel feedback. The discussion of a stochastic approximation framework based MAC algorithm will be postponed to Chapter 4.

3.1 Game Theoretic Problem Formulation

Consider a wireless network with K users. Each user is equipped with $M+1$ transmission options corresponding to a set of $M+1$ channel coding options, denoted by $G_k = \{g_{k0}, g_{k1}, \cdots, g_{kM}\}$. Each element g_{km}, $m = 0, \cdots, M$, represents a particular transmission option of user k that includes the specifications of transmission power, communication rate, etc. We assume that the first element g_{k0} always corresponds to the “idling" option. Time is slotted with each slot equalling the length of a fixed number of channel symbols, and this is also the length of a codeword. In each time slot, user k chooses one of the transmission options and sends a packet with encoded message to its receiver. The choice of transmission option of a user is shared neither with other users nor with the receiver. Assume that the communication channel is memoryless and static. Depending on whether reliable message decoding is supported by the channel or not, a transmitted packet is either received successfully or experiencing a collision. In the latter case, collision report is fed back to the transmitter. Note that, the link layer model degrades to a classical one if each G_k only contains two elements corresponding to idling/transmission options, respectively.

We assume that users are backlogged with messages. In each time slot, user k randomly chooses a transmission option according to an M-length vector $\boldsymbol{p}_k = [p_{k1}, \cdots, p_{kM}]^T$ termed the "transmission probability vector" of user k. Here $p_{km} \geq 0$, $m = 1, \cdots, M$, denotes the probability that user k chooses option g_{km}, and $1 - \sum_{m=1}^{M} p_{km} \geq 0$ is the probability that user k idles. Let $\boldsymbol{P} = [\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \cdots, \boldsymbol{p}_K^T]^T$ denote the transmission probability vectors of all users, and $\boldsymbol{P}_{-k} = [\boldsymbol{p}_1^T, \cdots, \boldsymbol{p}_{k-1}^T, \boldsymbol{p}_{k+1}^T, \cdots, \boldsymbol{p}_K^T]^T$ the transmission probability vectors of all users but user k. Conditioned on user k transmitting with option g_{km}, let $0 \leq q_{km} \leq 1$ be the probability that the message (or packet) is successfully received, for $k = 1, \cdots, K$ and $m = 1, \cdots, M$. Define $\boldsymbol{q}_k = [q_{k1}, \cdots, q_{kM}]^T$ as the "conditional success probability vector" of user k. Clearly, given the communication channel, $\boldsymbol{q}_k$ is a function of $\boldsymbol{P}_{-k}$.

We model the medium access control as a non-cooperative game where users distributively adapt their transmission probability vectors to maximize individual utility functions. The utility function of user k is denoted by $U_k(\boldsymbol{p}_k, \boldsymbol{q}_k)$, which is a function of the transmission probability vector $\boldsymbol{p}_k$ and the conditional success probability vector $\boldsymbol{q}_k$. Given $\boldsymbol{P}_{-k}$ and consequently $\boldsymbol{q}_k$, the utility maximization problem of user k is represented by

$$\max_{\boldsymbol{p}_k} U_k(\boldsymbol{p}_k, \boldsymbol{q}_k), \quad \text{s.t. } \boldsymbol{p}_k \geq \boldsymbol{0}, \boldsymbol{p}_k^T \boldsymbol{1} \leq 1, \tag{3.1}$$

where $\boldsymbol{0}$ and $\boldsymbol{1}$ are vectors of all zeros and all ones, respectively.

We say $\boldsymbol{P}$ is a Nash equilibrium of the medium access control game if for all $k = 1, \cdots, K$, $\boldsymbol{p}_k$ maximizes $U_k(\boldsymbol{p}_k, \boldsymbol{q}_k)$ given $\boldsymbol{P}_{-k}$. The following theorem gives a sufficient condition for the existence of a Nash equilibrium.

Theorem 6. *The medium access control game admits at least one Nash equilibrium if, for all* $k = 1, \cdots, K$, *utility function* $U_k(\boldsymbol{p}_k, \boldsymbol{q}_k)$ *is concave in* $\boldsymbol{p}_k$.

Theorem 6 is implied by [23, Theorem 1].

Given $\boldsymbol{P}$, define $\boldsymbol{G}_{kl}(\boldsymbol{P})$ as the second order partial derivative of $U_k(\boldsymbol{p}_k, \boldsymbol{q}_k)$ with respect to $\boldsymbol{p}_k$ and $\boldsymbol{p}_l$,

$$\boldsymbol{G}_{kl}(\boldsymbol{P}) = \frac{\partial^2 U_k(\boldsymbol{p}_k, \boldsymbol{q}_k)}{\partial \boldsymbol{p}_k \partial \boldsymbol{p}_l}. \tag{3.2}$$

The following theorem gives a sufficient condition under which Nash equilibrium of the medium access control game is unique.

Theorem 7. *Assume that the medium access control game has at least one Nash equilibrium. Let* $\boldsymbol{P}^{(1)}$ *and* $\boldsymbol{P}^{(2)}$ *be two Nash equilibria. For any* $0 \leq \theta \leq 1$, *let* $\boldsymbol{P} = \theta \boldsymbol{P}^{(1)} + (1-\theta)\boldsymbol{P}^{(2)}$. *If* $\boldsymbol{P}^{(1)} \neq \boldsymbol{P}^{(2)}$ *implies*

$$\sum_{k=1}^{K} \sum_{l=1}^{K} (\boldsymbol{p}_k^{(1)} - \boldsymbol{p}_k^{(2)})^T \boldsymbol{G}_{kl}(\boldsymbol{P}) (\boldsymbol{p}_l^{(1)} - \boldsymbol{p}_l^{(2)}) < 0, \tag{3.3}$$

then Nash equilibrium of the medium access control game must be unique.

Theorem 7 is implied by [23, Theorems 2, 6].

3.2 Utility Design with A Classical Physical-Link Layer Interface

To help explaining the utility function design with a relatively simple notation, in this section, we will first consider wireless networks with the classical physical-link layer interface where each user only has binary transmission/idling options. We choose to skip subscripts of the variables if this causes no confusion.

Consider a multiple access system with a symmetric channel and homogeneous users. Each user only has two transmission options, $G = \{g_0, g_1\}$, where g_0 is the idling option. According to the achievable region result in Chapter 2, if multiple users transmit in parallel, the packets should be received successfully so long as the code index vector of the users lies inside an achievable region. Assume that packet transmissions should be successful in a time slot if and only if no more than N

users transmit in parallel, and the value of N is known to all users. Assume that $K \gg N \geq 1$ and users want to maximize the symmetric throughput. With binary transmission/idling options, the system described is a random multiple access system over a multi-packet reception channel [24] [25]. Optimal sum throughput of the system is approached when each user transmits at probability x^*/K where x^* is the solution of the following maximization problem [25].

$$x^* = \operatorname*{argmax}_x e^{-x} \sum_{i=1}^{N} \frac{x^i}{(i-1)!}. \tag{3.4}$$

When all users set their transmission probabilities at x^*/K, conditional success probability experienced by each user can be approximated by

$$q^* = e^{-x^*} \sum_{i=0}^{N-1} \frac{x^{*i}}{i!}. \tag{3.5}$$

Assume that a user estimates the total number of users to be $\tilde{K}$. According to the above understanding, in the medium access control game, we design the utility function of each user as follows.

$$U(p, q) = \frac{p}{x^*} t(q) - h \frac{p}{x^*} \log \frac{p}{ex^*/\tilde{K}}. \tag{3.6}$$

The utility function contains two parts. The first part $\frac{p}{x^*} t(q)$ is a linear function in p that intends to control the conditional success probability q above its desired value shown in (3.5). We require that $t(q)$, which is a function of q, should satisfy $\frac{dt(q)}{dq} \geq 0$. We say that function $t(q)$ is unbiased if $t(q^*) = 0$. Note that, with an unbiased $t(q)$ function, the $\frac{p}{x^*} t(q)$ term alone tells a user to increase p when $q < q^*$ and to decrease p when $q > q^*$[2]. The second part $h \frac{p}{x^*} \log \frac{p}{ex^*/\tilde{K}}$ in the utility function, with h being a scaling parameter, is a convex function in p that intends to keep the transmission probability p around its targeted value $x^*/\tilde{K}$. Note that $h \frac{p}{x^*} \log \frac{p}{ex^*/\tilde{K}}$ is minimized at $p = x^*/\tilde{K}$.

[2]For various reasons, users may prefer a biased $t(q)$ function over an unbiased one. Discussions on this issue are skipped.

According to Theorem 8, if parameter h is chosen appropriately, the non-cooperative medium access control game should have a unique Nash equilibrium. Furthermore, if $\tilde{K} = K \gg 1$, and function $t(q)$ is unbiased, then the Nash equilibrium is represented by $p = x^*/K$ for all users since

$$\left.\frac{\partial U(p,q)}{\partial p}\right|_{p=x^*/K, q=q^*} = \frac{t(q^*)}{x^*} - \frac{h}{x^*}\left[\log\frac{1}{e} + 1\right] = 0. \tag{3.7}$$

To understand the idea behind the utility function design of (3.6), we can think about the distributed channel sharing game as a social event. Let us regard transmission probability p and conditional success probability q as the "behavior" and the measured "feedback" of a user. To participate in the social event, each user chooses a behavior target $x^*/\tilde{K}$ and a feedback target q^*, which are calculated via utility maximization in an envisioned network. In the above discussion for example, the targets are computed via sum throughput optimization in a random multiple access network with homogeneous users. Once the targets are obtained, each user chooses a utility function that specifies how the user should try to keep his behavior around the behavior target, and how the user should respond to the social force if the measured feedback differs from the feedback target.

3.3 Utility Design with An Enhanced Physical-Link Layer Interface

Let us now consider the same system investigated in Section 3.2, but with each user having $M \geq 2$ transmission options. Assume that, each user, say user k, keeps an estimated total user number, denoted by $\tilde{K}_k$. For each user and each non-idling transmission option, say option g_{km}, user k chooses two parameters: a targeted conditional success probability q^*_{km} and a targeted transmission probability $x^*_{km}/\tilde{K}_k$, for $m = 1, \cdots, M$. Given these parameters, utility function of user k is designed as the summations of two parts each being the summation of M items corresponding to the M non-idling transmission options.

$$U_k(\boldsymbol{p}_k, \boldsymbol{q}_k) = \sum_{m=1}^{M} \frac{p_{km}}{x^*_{km}} d_{km} t_{km}(q_{km}) - h_k \sum_{m=1}^{M} \frac{p_{km}}{x^*_{km}} \log \frac{p_{km}}{s_{km} e x^*_{km}/\tilde{K}_k}. \tag{3.8}$$

The first part $\sum_{m=1}^{M} \frac{p_{km}}{x^*_{km}} d_{km} t_{km}(q_{km})$ is a linear function in $\boldsymbol{p}_k$ that intends to control the conditional success probability vector $\boldsymbol{q}_k$ above the desired values. We require that $\frac{dt_{km}(q)}{dq} \geq 0$. Different from the case of binary transmission/idling options, we introduce a "steering vector" $\boldsymbol{d}_k = [d_{k1}, d_{k2}, \cdots, d_{kM}]^T$, with $\boldsymbol{d}_k \geq \boldsymbol{0}, \boldsymbol{d}_k^T \boldsymbol{1} \leq 1$, to allow user k to assign different weights to terms corresponding to different transmission options. The second part $h_k \sum_{m=1}^{M} \frac{p_{km}}{x^*_{km}} \log \frac{p_{km}}{s_{km} e x^*_{km}/\tilde{K}_k}$ is a convex function in $\boldsymbol{p}_k$ that intends to keep the transmission probability vector $\boldsymbol{p}_k$ around a targeted value $\boldsymbol{p}_k^*$. We introduce another "steering vector" $\boldsymbol{s}_k = [s_{k1}, s_{k2}, \cdots, s_{kM}]^T$, with $\boldsymbol{s}_k \geq \boldsymbol{0}, \boldsymbol{s}_k^T \boldsymbol{1} \leq 1$, and construct the targeted transmission probability vector $\boldsymbol{p}_k^*$ as follows.

$$\boldsymbol{p}_k^* = [s_{k1} x^*_{k1}/\tilde{K}_k, s_{k2} x^*_{k2}/\tilde{K}_k, \cdots, s_{kM} x^*_{kM}/\tilde{K}_k]^T. \tag{3.9}$$

Without specifying how the q^*_{km} and $x^*_{km}/\tilde{K}_k$ parameters are determined, our key result is presented in the following theorem, which shows that if the scaling parameters h_k are chosen appropriately, then the non-cooperative medium access control game has a unique Nash equilibrium.

Theorem 8. *Given the steering vectors $\boldsymbol{d}_k$, $\boldsymbol{s}_k$, $k = 1, \cdots, K$, the medium access control game has a unique Nash equilibrium if the following inequality is satisfied for all $k = 1, \cdots, K$ and $m = 1, \cdots, M$*

$$\frac{\tilde{K}_k}{K} \frac{h_k}{x^{*2}_{km}} e^{-\frac{t^{(\max)}_{km}}{h_k}} \geq \max \left\{ \left(\frac{t'^{(\max)}_{km}}{x^*_{km}} \right)^2, 1 \right\}, \tag{3.10}$$

where $t^{(\max)}_{km} = \max_q t_{km}(q)$ and $t'^{(\max)}_{km} = \max_q \frac{dt_{km}(q)}{dq}$.

The proof of Theorem 8 is given in Appendix A.1.

To understand the utility function design and the significance of Theorem 8, we can again think about the distributed channel access game as a social event, and use $\boldsymbol{p}_k$, $\boldsymbol{q}_k$ to represent the "behavior" and the measured "feedback" of user k. We assume that user k should choose a targeted transmission probability $x^*_{km}/\tilde{K}_k$ and a targeted conditional success probability q^*_{km} for each of the non-idling transmission options. Since a user now has M non-idling options, the behavior target $\boldsymbol{p}_k^*$ is constructed using a steering vector $\boldsymbol{s}_k$ as $\boldsymbol{p}_k^* = [s_{k1}x_{k1}^*/\tilde{K}_k, s_{k2}x_{k2}^*/\tilde{K}_k, \cdots, s_{kM}x_{kM}^*/\tilde{K}_k]^T$. The term $h_k \sum_{m=1}^{M} \frac{p_{km}}{x_{km}^*} \log \frac{p_{km}}{s_{km}ex_{km}^*/\tilde{K}_k}$ in the utility function, which is referred as the "self-behavior preference" term, intends to keep the behavior of user k around the targeted behavior $\boldsymbol{p}_k^*$. On the other hand, we construct the feedback target $\boldsymbol{q}_k^*$ as $\boldsymbol{q}_k^* = [q_{k1}^*, q_{k2}^*, \cdots, q_{kM}^*]^T$. The term $\sum_{m=1}^{M} \frac{p_{km}}{x_{km}^*} d_{km} t_{km}(q_{km})$ in the utility function, which is referred as the "social-behavior preference" term, specifies how user k should adapt his behavior according to the social forces represented by the measured feedback. Here the steering vector $\boldsymbol{d}_k$ is introduced to allow user k to emphasize or ignore social forces corresponding to different transmission options. Online adaptations of the steering vectors $\boldsymbol{d}_k$ and $\boldsymbol{s}_k$ will be illustrated using an example in Section 3.4. Note that, if the the scaling parameters h_k, for $k = 1, \cdots, M$, are large enough, then the self-behavior preference term will dominate the utility function of each user. Consequently, users will keep their behaviors around their pre-determined targets, and this can easily lead to a unique Nash equilibrium for the distributed channel access game. Theorem 8 shows that, to achieve such an effect, so long as the estimated total number of users $\tilde{K}_k$ is not too far from the true value K, the values of h_k do not need to scale in the total number of users K or the total number of non-idling transmission options M.

3.4 Simulation Results

In this section, we use computer simulations to show that, once multiple transmission options are provided for each user, a distributed network often prefers low rate and parallel channel access options over high rate and exclusive channel access options. Such a property, being consistent with the well-known information theoretic understanding, can demonstrate the potential impact of the enhanced physical-link layer interface on the design of medium access control algorithms.

Example 3.1: Consider a multiple access system where $K = 100$ users stay on a circle centered around the receiver. The multiple access channel is memoryless with additive Gaussian noise of zero mean and variance N_0. The channel gains from all the users to the receiver are assumed to be unit-valued. Each user has three non-idling transmission options, denoted by g_1, g_2, and g_3. The three non-idling options all correspond to random Gaussian block channel codes at the physical layer with the same transmission power P, but with different rates $r_1 = \frac{1}{14}\log\left(1+\frac{7P}{N_0}\right)$, $r_2 = \frac{1}{6}\log\left(1+\frac{3P}{N_0}\right)$ and $r_3 = \frac{1}{2}\log\left(1+\frac{P}{N_0}\right)$, respectively. We set $P/N_0 = 10$. Assume that the total number of users is known, i.e., $\tilde{K} = 100$. For every transmission option g_i, $i = 1, 2, 3$, targeted conditional success probability q_i^* and transmission probability $x_i^*/100$ are chosen to maximize the sum throughput of a classical system with each user having binary transmission options of $\{g_0, g_i\}$. In other words, (q_1^*, x_1^*), (q_2^*, x_2^*), (q_3^*, x_3^*) are determined using (3.5) and (3.4) by setting N at 7, 3, 1, respectively. We choose $t_i(q_i) = x_i^* d_i r_i q_i$, which is a biased function since $t_i(q_i^*) \neq 0$. We also set the scaling parameter h at the minimum value satisfying (3.10).

We initialize the transmission probability vectors of all users at $\boldsymbol{p} = [1/4, 1/4, 1/4]^T$, and their steering vectors at $\boldsymbol{d} = \boldsymbol{s} = [1/3, 1/3, 1/3]^T$. During the distributed channel sharing game, each user first uses 200 time slots to measure the conditional success probability vector $\boldsymbol{q}$. If during this time interval a user does not have a sufficient number of transmission attempts using a particular option g_i, then q_i is set to a small but none-zero value. After measuring the conditional success probability vector $\boldsymbol{q}$, each user then updates its transmission probability vector $\boldsymbol{p}$ in the gradient direction that maximizes the utility function. Steering vector $\boldsymbol{s}$ is updated in the gradient direction that minimizes the term $\sum_{m=1}^{M} \frac{p_m}{x_m^*} \log \frac{p_m}{s_m e x_m^*/\tilde{K}}$. Steering vector $\boldsymbol{d}$ is updated to increase the weights of feedback terms with larger values of $t_i(q_i)$. The procedure iterates till transmission probability vectors of all users converge. Figure 3.1 illustrates the sum throughput of the system in bits/symbol in each iteration (one iteration takes 200 time slots). The three dashed-red lines respectively correspond to the targeted sum throughput of the systems where each user only has binary transmission options of $\{g_0, g_1\}$, $\{g_0, g_2\}$, $\{g_0, g_3\}$. In this example, transmission probability vectors of all users converge quickly to $\boldsymbol{p} = [0.0507, 0, 0]^T$. In other words, users will only use the low rate option to share the multiple access channel. Note that the resulting sum throughput is slightly higher than the top dashed-red line because the targeted probability x_1^*/K is only approximately optimal for a finite K.

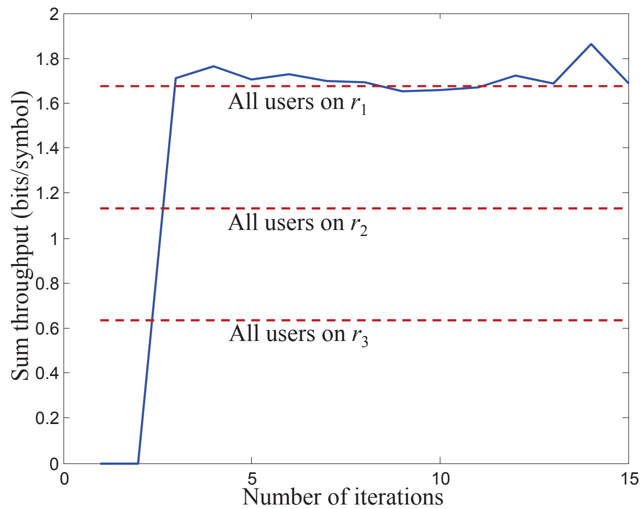

Figure 3.1: Convergence of the sum throughput in bits/symbol. $P/N_0 = 10$. One iteration takes 200 time slots.

In this example, we update steering vector $\boldsymbol{s}$ to minimize the “self-behavior preference" term. Let us define the following region of the transmission probability vector,

$$R_{\boldsymbol{p}} = \left\{\boldsymbol{p} | \exists \tilde{\boldsymbol{s}}, \tilde{\boldsymbol{s}} \geq \boldsymbol{0}, \tilde{\boldsymbol{s}}^T \boldsymbol{1} = 1, \text{such that } \boldsymbol{p} = [\tilde{s}_1 x_1^*/\tilde{K}, \tilde{s}_2 x_2^*/\tilde{K}, \cdots, \tilde{s}_M x_M^*/\tilde{K}]^T \right\}. \tag{3.11}$$

Note that, if we take the adaptation of $\boldsymbol{s}$ into consideration, the self-behavior preference term $\min_{\boldsymbol{s}, \boldsymbol{s} \geq \boldsymbol{0}, \boldsymbol{s}^T \boldsymbol{1} \leq 1} \sum_{m=1}^{M} \frac{p_m}{x_m^*} \log \frac{p_m}{s_m e x_m^*/\tilde{K}}$ achieves the same minimum value of $-\tilde{K}$ at any $\boldsymbol{p} \in R_{\boldsymbol{p}}$. Therefore, with the help of the steering vector adaptation, the self-behavior preference term only intends to keep the behavior $\boldsymbol{p}$ of a user around region $R_{\boldsymbol{p}}$. It however does not provide any preference on which transmission option should be more favorable to the user. On the other hand, the “social-behavior preference" term intends to help a user to find the best transmission option based on feedback received from the system. If we fix steering vector $\boldsymbol{d}$ at $\boldsymbol{d} = [1/3, 1/3, 1/3]^T$, then each user will assign positive probabilities to all entries of the transmission probability vector. Adaptation of steering vector $\boldsymbol{d}$ helps a user to favor the transmission option with the best feedback, which maximizes $t_i(q_i)$.

Chapter 4

Utility Optimization in A Distributed Multiple Access System

In this Chapter, we will discuss the support of the interface enhancement at the data link layer from a different perspective. The enhanced interface equips each link layer user with multiple transmission options as opposed to binary transmission/idling options in a classical interface, and therefore enables advanced wireless capabilities such as rate and power adaptation. As in Chapter 3, we still seek the answer to the question that, for data link layer users in a distributed network, whether there exists a general framework to efficiently exploit an arbitrary and often limited set of provided transmission options to optimize a chosen network utility.

Distributed adaptive medium access control (MAC) protocols can be categorized into splitting algorithms [26] [27] [28] [29] [30] [31] [32] and back-off approaches [11] [13] [33] [34] [35]. In splitting algorithms such as the FCFS algorithm [26], under the assumption that noiseless channel feedback is instantly available, users maintain a common virtual interval of their random identity values. The interval is partitioned and ordered, which determines the transmission schedule of the users, according to a sequence of channel feedback messages. While splitting algorithms can often achieve a relatively high system throughput, their function depends on the assumptions of instant availability of channel feedback and correct reception of feedback sequences. Both of the two conditions, unfortunately, can be violated in a wireless environment. Theoretical analysis of a splitting algorithm, taking into account the wireless-related factors such as channel fading, measurement noise, feedback error and transmission delay, can be extremely challenging. Analysis of the back-off algorithms, on the other hand, has proven to be more trackable [11] [13] [36]. In back-off algorithms such as the 802.11 DCF protocol [11], conditioned on packet availability, each user should transmit with a particular probability. In most cases [13] [35], a user should decrease its transmission probability in response to packet collision (or a transmission failure) event, and increase its transmission probability in response to a transmission success event. Distributed probability adaptation in a back-off algorithm often falls into the framework of stochastic approximation [13] [36], whose theoretical analysis enjoys a rigorous set of mathematical and statistical tools developed in the literature [37] [38] [39] [40] [41]. Practical back-off algorithms can also be analyzed using Markov models [11]. Most of the existing analyses of the splitting and the back-off algorithms either assume a throughput optimization objective and/or a simple collision channel model. While there has been no analytical framework that can deal with the optimization of an arbitrary network utility with a general channel model, the interesting topic of how collision resolution algorithms should be revised to work with wireless-related physical layer properties, such as capture effect and multipacket reception, has attracted significant research efforts in the literature [24] [25] [35] [42] [43] [44].

In this Chapter, a distributed MAC framework abstracted from the back-off algorithms will be introduced. In order to maintain a relatively simple and trackable investigation, we focus on distributed link-layer multiple access networking with an unknown number of homogeneous users, and also assume that all users should have saturated message queues. Motivations of such a focus are explained as follows. First, the assumption of saturated message queues is introduced to avoid the complication of random message arrivals. While bursty message arrival is rather an important character of distributed network systems [10] [45], it is known to create coupling between transmission activities of the users [46] [47], and such coupling often leads to open research problems in throughput and stability analysis [48] [49] [50] [51] of systems with a relatively small number of users [52] [53]. Results obtained under the assumption of saturated message queues can often serve as achievable bounds to the corresponding results for systems with random message arrivals [36] [51]. Second, because each user only interacts with the receiver, the assumption of multiple access networking with homogeneous users mainly represents the communication environment envisioned by each link layer user[3]. In other words, without further knowledge about the

[3]Note that the assumption of user symmetry is also reflected in many existing channel models such as the collision channel model [10] and the multipacket reception channel model [24] [25].

actual networking environment, a link layer protocol should be designed to help a user to get a fair share of the multiple access channel under the assumption of user homogeneity. Early research investigation aims at achieving such a design objective in the assumed networking environment. Understanding the behavior of the link layer algorithm in a general networking environment is a future research task that is beyond the scope of this thesis. Finally, because users in a distributed network often access the channel opportunistically, it is difficult to know how many users are actually active [36]. Suppose that the homogeneous users in a distributed multiple access network should be able to calculate their optimal transmission schemes if the user number is known, but it is desired to develop distributed algorithms to lead the system to a desired operation point without the knowledge of the actual user number[4]. Note that, rather than developing a practical MAC protocol, the primary objective is to obtain useful insights about distributed medium access control through the analysis of systems with/without the enhanced physical-link layer interface.

4.1 A Stochastic Approximation Framework

Consider a time-slotted distributed multiple access network with a memoryless channel and K homogeneous users. The length of each time slot equals the transmission duration of one packet. The user number K is assumed to be known neither to the users nor to the receiver. Each user, say user k, is equipped with M transmission options plus an idling option, denoted by $\mathcal{G}_k = \{g_{k0}, g_{k1}, \ldots, g_{kM}\}$ with g_{k0} being the idling option. These options correspond to the code ensemble $\mathcal{G}_k$ prepared by the physical layer transmitter of user k, as explained in Chapter 2. We assume that all users are backlogged with saturated message queues. At the beginning of each time slot t, according to an associated probability vector, each user either idles or randomly chooses a transmission option to send its message. Transmission decisions of the users are made individually in the sense that the decisions are not shared among the users or with the receiver. The M-length probability vector associated with user k in time slot t can be written as $\boldsymbol{p}_k(t) = p_k(t)\boldsymbol{d}_k(t)$, where

[4] In back-off algorithms, the necessity of probability adaptation generally implies the assumption that the number of active users is unknown to the system.

$p_k(t)$ is termed the "transmission probability" of user k, and $\boldsymbol{d}_k(t)$, termed the "transmission direction" vector of user k, is an M-length probability vector whose entries $d_{km}(t)$, for $1 \le m \le M$, satisfy $d_{km}(t) \ge 0$ and $\sum_{m=1}^{M} d_{km}(t) = 1$.

At the end of each time slot t, based upon available channel feedback, each user, say user k, calculates a target probability vector $\tilde{\boldsymbol{p}}_k(t) = \tilde{p}_k(t)\tilde{\boldsymbol{d}}_k(t)$. User k then updates its transmission probability vector by

$$\boldsymbol{p}_k(t+1) = (1-\alpha(t))\boldsymbol{p}_k(t) + \alpha(t)\tilde{\boldsymbol{p}}_k(t) = \boldsymbol{p}_k(t) + \alpha(t)(\tilde{\boldsymbol{p}}_k(t) - \boldsymbol{p}_k(t)), \tag{4.1}$$

where $\alpha(t) > 0$ is a step size parameter of time slot t. Let $\boldsymbol{P}(t) = [\boldsymbol{p}_1^T(t), \boldsymbol{p}_2^T(t), \ldots, \boldsymbol{p}_K^T(t)]^T$ be a vector of length MK that consists of the transmission probability vectors of all users in time slot t. Let $\tilde{\boldsymbol{P}}(t) = [\tilde{\boldsymbol{p}}_1^T(t), \tilde{\boldsymbol{p}}_2^T(t), \ldots, \tilde{\boldsymbol{p}}_K^T(t)]^T$ be the corresponding target vector. $\boldsymbol{P}(t)$ is updated by

$$\boldsymbol{P}(t+1) = \boldsymbol{P}(t) + \alpha(t)(\tilde{\boldsymbol{P}}(t) - \boldsymbol{P}(t)). \tag{4.2}$$

Note that (4.2) falls into the framework of stochastic approximation algorithms [37] [38] [39], where the actual target transmission probability vector $\tilde{\boldsymbol{P}}(t)$ is often calculated based upon noisy estimates of certain system variables.

Define $\hat{\boldsymbol{P}}(t) = [\hat{\boldsymbol{p}}_1^T(t), \hat{\boldsymbol{p}}_2^T(t), \ldots, \hat{\boldsymbol{p}}_K^T(t)]^T$ as the "theoretical value" of $\tilde{\boldsymbol{P}}(t)$ when there is no measurement noise and no feedback error in time slot t, with $\hat{\boldsymbol{p}}_k(t)$ being the corresponding theoretical value of $\tilde{\boldsymbol{p}}_k(t)$, for $1 \le k \le K$. Let $E_t[\tilde{\boldsymbol{P}}(t)]$ be the expectation of $\tilde{\boldsymbol{P}}(t)$ conditioned on system state at the beginning of time slot t. Write $E_t[\tilde{\boldsymbol{P}}(t)]$ as follows

$$E_t[\tilde{\boldsymbol{P}}(t)] = \hat{\boldsymbol{P}}(t) + \boldsymbol{G}(t) = \hat{\boldsymbol{P}}(\boldsymbol{P}(t)) + \boldsymbol{G}(\boldsymbol{P}(t)), \tag{4.3}$$

where $\boldsymbol{G}(t) = E_t[\tilde{\boldsymbol{P}}(t)] - \hat{\boldsymbol{P}}(t)$ is defined as the bias term in the target probability vector calculation. Given the communication channel, both $\hat{\boldsymbol{P}}(t)$ and $\boldsymbol{G}(t)$ are functions of $\boldsymbol{P}(t)$, which consists of transmission probability vectors of all users in time slot t.

Next, two conditions are presented, which are typically required for the convergence of a stochastic approximation algorithm [37] [38] [39].

Condition 1. *(Mean and Bias) There exists a constant* $K_m > 0$ *and a bounding sequence* $0 \leq \beta(t) \leq 1$*, such that*

$$\|\boldsymbol{G}(\boldsymbol{P}(t))\| \leq K_m \beta(t). \tag{4.4}$$

Furthermore, we assume that $\beta(t)$ *should be controllable in the sense that one can design protocols to ensure* $\beta(t) \leq \epsilon$ *for any chosen* $\epsilon > 0$ *and for large enough* t*.*

Condition 2. *(Lipschitz Continuity) There exists a constant* $K_l > 0$*, such that*

$$\|\hat{\boldsymbol{P}}(\boldsymbol{P}_a) - \hat{\boldsymbol{P}}(\boldsymbol{P}_b)\| \leq K_l \|\boldsymbol{P}_a - \boldsymbol{P}_b\|, \qquad \textit{for all } \boldsymbol{P}_a, \boldsymbol{P}_b. \tag{4.5}$$

Under these conditions, according to stochastic approximation theory [39] [40] [41], if the step size sequence $\alpha(t)$ and the bounding sequence $\beta(t)$ are small enough, trajectory of the transmission probability vector $\boldsymbol{P}(t)$ under distributed adaptation given in (4.2) can be approximated by the following associated ordinary differential equation (ODE),

$$\frac{d\boldsymbol{P}(t)}{dt} = -[\boldsymbol{P}(t) - \hat{\boldsymbol{P}}(t)], \tag{4.6}$$

where, with an abuse of notation, we also used t to denote the continuous time variable. Because all entries of $\boldsymbol{P}(t)$ and $\hat{\boldsymbol{P}}(t)$ stay in the range of $[0, 1]$, any equilibrium $\boldsymbol{P}$ of the associated ODE given in (4.6) must satisfy

$$\boldsymbol{P} = \hat{\boldsymbol{P}}(\boldsymbol{P}). \tag{4.7}$$

Suppose that the solution to (4.7), which is also the equilibrium of (4.6), is unique at $\boldsymbol{P}^* = [\boldsymbol{p}_1^{*T}, \ldots, \boldsymbol{p}_K^{*T}]^T$. According to stochastic approximation theory, if the step size sequence $\alpha(t)$ and the bounding sequence $\beta(t)$ are small in value, convergence results are stated as follows.

Theorem 9. *For distributed transmission probability vector adaptation given in (4.2), assume that the associated ODE given in (4.6) has a unique stable equilibrium at* $\boldsymbol{P}^*$*. Suppose that* $\alpha(t)$ *and*

$\beta(t)$ satisfy the following conditions

$$\sum_{t=0}^{\infty} \alpha(t) = \infty, \sum_{t=0}^{\infty} \alpha(t)^2 < \infty, \sum_{t=0}^{\infty} \alpha(t)\beta(t) < \infty. \tag{4.8}$$

Under Conditions 1 and 2, $\boldsymbol{P}(t)$ converges to $\boldsymbol{P}^*$ with probability one.

Theorem 9 is implied by [40, Theorem 4.3].

Theorem 10. *For distributed transmission probability vector adaptation given in (4.2), assume that the associated ODE given in (4.6) has a unique stable equilibrium at $\boldsymbol{P}^*$. Let Conditions 1 and 2 be true. Then for any $\epsilon > 0$, there exists a constant $K_w > 0$, such that, for any $0 < \underline{\alpha} < \overline{\alpha} < 1$ satisfying the following constraint*

$$\exists T_0 \geq 0, \underline{\alpha} \leq \alpha(t) \leq \overline{\alpha}, \beta(t) \leq \sqrt{\overline{\alpha}}, \forall t \geq T_0, \tag{4.9}$$

$\boldsymbol{P}(t)$ converges to $\boldsymbol{P}^*$ in the following sense

$$\limsup_{t \to \infty} Pr\left\{ \|\boldsymbol{P}(t) - \boldsymbol{P}^*\| \geq \epsilon \right\} < K_w \overline{\alpha}. \tag{4.10}$$

Theorem 10 can be obtained by following the proof of [41, Theorem 2.3] with minor revisions.

Note that, for simplicity, the above discussion assumes the same step size sequence $\alpha(t)$ and the same bounding sequence $\beta(t)$ for all users. It is also assumed that all users should update their transmission probability vectors (synchronously) in each time slot. However, by following the literature of stochastic approximation theory [39] [40], it is easy to show that different users can use different step sizes and bounding sequences, and can also adapt their probability vectors asynchronously. So long as the step sizes and bounding sequences of all users satisfy the same constraints given in (4.8) and (4.9), and the users update their probability vectors frequently enough, then convergence results stated in Theorems 9 and 10 remain valid.

With convergence of the probability vectors guaranteed by Theorems 9 and 10, the key objective of the system design is to develop distributed MAC algorithms to satisfy Conditions 1 and 2 and to place the unique equilibrium of the associated ODE at the desired point. Unfortunately, achieving such an objective is not always easy especially when the enhanced physical-link layer interface is considered. Because users are homogeneous, due to symmetry, if an equilibrium of the system is unique, transmission probability vectors of the users at the equilibrium must be identical. Such a property is enforced by guaranteeing that all users should obtain the same target transmission probability vector in each time slot. The corresponding part of the system design is introduced below.

In each time slot, assume that there is a set of V virtual packets being transmitted through the channel. The virtual packet set remains the same over different time slots. Each virtual packet in the set is an assumed packet whose coding parameters are known to the users and to the receiver, but it is not physically transmitted in the system, i.e., the packet is "virtual". Assume that, without knowing the transmission status of the users, the receiver can detect whether the reception of each virtual packet should be regarded as successful or not, and therefore can estimate the success probability of each virtual packet. For example, suppose that the link layer channel is a collision channel, and a virtual packet has the same coding parameters as those of a real packet. Then, the virtual packet reception should be regarded as successful if and only if no real packet is transmitted in the given time slot. Success probability of the virtual packet in this case equals the idling probability of the collision channel. For another example, if all packets including the virtual packets are encoded using random block codes, given the physical layer channel model, reception of each virtual packet corresponds to a detection task of determining whether or not the code index vector of the real users should belong to a specific operation region. Such detection tasks and their performance bounds have been discussed in Chapter 2.

Let $\boldsymbol{q}_v(t)$ be a V-length vector whose entry $q_{vi}(t)$, for $1 \leq i \leq V$, is the success probability of the ith virtual packet in time slot t. We assume that the receiver should measure and feed the estimated $\boldsymbol{q}_v(t)$ back to all users (transmitters). Upon receiving the estimated $\boldsymbol{q}_v(t)$, each user should calculate the M-length target transmission probability vector as the same function of the $\boldsymbol{q}_v(t)$ estimate. Denote the theoretical target probability vector by $\hat{\boldsymbol{p}}(\boldsymbol{q}_v(t))$. The target transmission probability vectors of all users are given by $\hat{\boldsymbol{P}}(t) = \mathbf{1} \otimes \hat{\boldsymbol{p}}(\boldsymbol{q}_v(t))$, where $\mathbf{1}$ denotes a K-length vector of all 1's and $\otimes$ represents the Kronecker product. Consequently, according to (4.6), if $\boldsymbol{P}^*$ is an equilibrium of the system, we must have $\boldsymbol{P}^* = \mathbf{1} \otimes \boldsymbol{p}^*$. Because $\boldsymbol{q}_v$ is a function of the transmission probability vectors, we must have $\boldsymbol{P}^* = \mathbf{1} \otimes \boldsymbol{p}^* = \mathbf{1} \otimes \hat{\boldsymbol{p}}(\boldsymbol{p}^*)$, where $\hat{\boldsymbol{p}}(\boldsymbol{p}^*)$ denotes the theoretical target probability vector of the users given that all users have the same transmission probability vector $\boldsymbol{p}^*$.

Note that the introduction of virtual packets and the assumption of feeding back $\boldsymbol{q}_v(t)$ to the transmitters are rather rare both in the literature of MAC algorithms and in practical MAC protocols. The key purpose of such a system design is to feed back a measure that is common to all users. This enables users to calculate the same target transmission probability vector and consequently guarantees that transmission probability vectors of all users at any system equilibrium must be identical. As what will be shown in the following sections, such a property can significantly simplify the design and analysis of the distributed MAC algorithm. If a user only knows the success/failure status of its own packets on the other hand, as commonly assumed in existing MAC algorithms, then guaranteeing identical transmission probability vector at the equilibrium under our problem formulation can become a challenging task.

In a practical system, the measurement of $\boldsymbol{q}_v(t)$ is likely to experience measurement noise and feedback error. Assume that, if users keep $\boldsymbol{P}$ at a constant vector for a duration of Q time slots, and $\boldsymbol{q}_v$ is measured over these time slots, then the measurement should converge to its true value with probability one as Q is taken to infinity. Other than this assumption, system noise is not involved in the discussion of the design objectives, i.e., to meet Conditions 1 and 2 and to place the unique equilibrium of the associated ODE at the desired point. Therefore, in the following sections, the assumption that $\boldsymbol{q}_v(t)$ can be measured precisely at the receiver and be fed back to the users will be adopted. This leads to $\tilde{\boldsymbol{P}}(t) = \hat{\boldsymbol{P}}(t) = \mathbf{1} \otimes \hat{\boldsymbol{p}}(t)$. To simplify the notation, time index t will be skipped in the rest of the discussions.

4.2 Single Transmission Option with Actual Channel Contention Measure

Let us first consider the simple case of classical physical-link layer interface, where each user only has a single transmission option plus an idling option. Each user, say user k, should maintain a scalar transmission probability parameter p_k to specify the probability at which user k transmits a packet in a time slot. Transmission probabilities of all users are listed in a K-length vector $\boldsymbol{p}$. In this section, it will be shown that, with a general channel model and without knowing the user number K, a distributed MAC algorithm can be designed to lead the system to converge to a unique equilibrium that is not far from optimal with respect to a chosen symmetric network utility.

Given the physical layer channel and the provided transmission options, the link layer multiple access channel is modeled using two sets of channel parameters. Define $\{C_{rj}\}$ for $j \geq 0$ as the "real channel parameter set", where C_{rj} is the conditional success probability of a real packet should it be transmitted in parallel with j other real packets. Assume that there is a single virtual packet being transmitted in each time slot. Virtual packets transmitted in different time slots are identical. Given coding parameters of the virtual packet, let $\{C_{vj}\}$ for $j \geq 0$ be the "virtual channel parameter set", where C_{vj} is the success probability of the virtual packet should it be transmitted in parallel with j real packets. Assume $C_{vj} \geq C_{v(j+1)}$ for all $j \geq 0$, which means that, if the number of parallel real packet transmissions increases, the virtual packet should have a non-increasing chance of getting through the channel. Let $\epsilon_v > 0$ be a pre-determined small constant. Define J_{ϵ_v} as the minimum integer such that $C_{vJ_{\epsilon_v}}$ is strictly larger than $C_{v(J_{\epsilon_v}+1)} + \epsilon_v$, i.e.,

$$J_{\epsilon_v} = \arg\min_j C_{vj} > C_{v(j+1)} + \epsilon_v. \tag{4.11}$$

Assume that both the real and the virtual channel parameter sets should be known at the users and at the receiver. Note that, while $\{C_{rj}\}$ has nothing to do with the virtual packet, $\{C_{vj}\}$ is dependent on the coding parameters of the virtual packet.

We assume that users intend to maximize a symmetric network utility, denoted by $U(K, p, \{C_{rj}\})$. Under the assumption that all users should transmit with the same probability, i.e., $\boldsymbol{p} = p\mathbf{1}$, system utility is a function of the unknown user number K, the common transmission probability p, and the real channel parameter set $\{C_{rj}\}$. For example, if we choose sum throughput of the system as the utility function, $U(K, p, \{C_{rj}\})$ should be given by

$$U(K, p, \{C_{rj}\}) = K \sum_{j=0}^{K-1} \binom{K-1}{j} p^{j+1}(1-p)^{K-1-j} C_{rj}. \tag{4.12}$$

For most of the utility functions of interest, such as the sum throughput function given above, an asymptotically optimal solution should roughly keep the expected load of the channel at a constant [24] [25]. Therefore, if p_K^* is the optimum transmission probability for user number K, we should have $\lim_{K\to\infty} Kp_K^* = x^*$, where $x^* > 0$ is obtained from the following asymptotic utility optimization.

$$x^* = \arg\max_x \lim_{K\to\infty} U\left(K, \frac{x}{K}, \{C_{rj}\}\right). \tag{4.13}$$

Note that virtual packet is not involved in the calculation of x^*.

Without knowing the actual user number K, we will show next that it is possible to set the system equilibrium at $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$, where $b \geq 1$ is a pre-determined design parameter, and $p_{\max}$ is defined as

$$p_{\max} = \min\left\{1, \frac{x^*}{J_{\epsilon_v} + b}\right\}. \tag{4.14}$$

It will be shown later that, when the optimum transmission probability satisfies $\lim_{K\to\infty} Kp_K^* = x^*$, setting the equilibrium at $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$ is often not far from the optimal even when the user number is small.

We intend to design a distributed MAC algorithm to maximize $U(K, p, \{C_{rj}\})$ by maintaining channel contention at a desired level. Let q_v denote the success probability of the virtual packet, measured at the receiver. Term q_v the "channel contention measure" because it is a measurement of the contention level of the system. Note that $q_v(\boldsymbol{p}, K)$ is a function of user number K and transmission probabilities of all users listed in the vector $\boldsymbol{p}$. Because $q_v(\boldsymbol{p}, K)$ equals the summation of a finite number of polynomial terms, $q_v(\boldsymbol{p}, K)$ should be Lipschitz continuous in $\boldsymbol{p}$ for any finite K. When the transmission probabilities of all users are equal, i.e., $\boldsymbol{p} = p\mathbf{1}$, success probability of the virtual packet can be written as

$$q_v(p, K) = \sum_{j=0}^{K} \binom{K}{j} p^j (1-p)^{K-j} C_{vj}, \tag{4.15}$$

where $\{C_{vj}\}$ is the set of virtual channel parameters. Assume that, upon obtaining q_v from the receiver, each user should first obtain a user number estimate, denoted by $\hat{K}$, and then set the corresponding transmission probability target at $\tilde{p} = \hat{p} = \min\left\{p_{\max}, \frac{x^*}{\hat{K}+b}\right\}$, where $x^* > 0$ is obtained from (4.13). It will be shown that, for any $x^* > 0$, without knowing K, one can always choose an appropriate b and design a distributed MAC algorithm to ensure system convergence to the desired equilibrium of $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$.

Convergence of the MAC algorithm to be introduced depends on two key monotonicity properties presented below. First, the following theorem shows that, given user number K, $q_v(p, K)$ is non-increasing in p.

Theorem 11. *Under the assumption that $C_{vj} \geq C_{v(j+1)}$ for all $j \geq 0$, $q_v(p, K)$ given in (4.15) satisfies $\frac{\partial q_v(p,K)}{\partial p} \leq 0$. Furthermore, $\frac{\partial q_v(p,K)}{\partial p} < 0$ holds with strict inequality for $K > J_{\epsilon_v}$ and $p \in (0, 1)$.*

The proof of Theorem 11 is given in Appendix B.1.

Given that $\hat{p} = \frac{x^*}{\hat{K}+b}$. Let $N = \lfloor \hat{K} \rfloor$ be the largest integer below $\hat{K}$. Define a continuous function $q_v^*(\hat{p})$, which can also be viewed as a function of $\hat{K}$, as follows

$$q_v^*(\hat{p}) = \frac{\hat{p} - p_{N+1}}{p_N - p_{N+1}} q_N(\hat{p}) + \frac{p_N - \hat{p}}{p_N - p_{N+1}} q_{N+1}(\hat{p}), \tag{4.16}$$

where $p_N = \min\left\{p_{\max}, \frac{x^*}{N+b}\right\}$, $p_{N+1} = \min\left\{p_{\max}, \frac{x^*}{N+1+b}\right\}$, and

$$q_N(p) = \sum_{j=0}^{N} \binom{N}{j} p^j (1-p)^{N-j} C_{vj},$$

$$q_{N+1}(p) = \sum_{j=0}^{N+1} \binom{N+1}{j} p^j (1-p)^{N+1-j} C_{vj}. \tag{4.17}$$

Term $q_v^*(\hat{p})$ the "theoretical channel contention measure" because it serves as a reference to the theoretical contention level of the system in the following sense. If user number of the system indeed equals $K = \hat{K}$ with $\hat{K} \geq J_{\epsilon_v}$, then $q_v^*(\hat{p})$ defined in (4.16) equals the actual channel contention measure at the desired equilibrium $\boldsymbol{p}^* = \frac{x^*}{K+b}\mathbf{1} = \frac{x^*}{\hat{K}+b}\mathbf{1}$ when all users transmit with the same probability $\hat{p} = \frac{x^*}{\hat{K}+b}$.

The following theorem gives the second monotonicity property, which shows that, given an arbitrary $x^* > 0$, with an appropriate choice of b, $q_v^*(\hat{p})$ is non-decreasing in $\hat{p}$.

Theorem 12. *Let $x^* > 0$. Let $b \geq \max\{1, x^* - \gamma_{\epsilon_v}\}$, with γ_{ϵ_v} being defined as*

$$\gamma_{\epsilon_v} = \min_{N, N \geq J_{\epsilon_v}, N \geq x^* - b} \frac{\sum_{j=0}^{N} j \binom{N}{j} \left(\frac{p_{N+1}}{1-p_{N+1}}\right)^j (C_{vj} - C_{v(j+1)})}{\sum_{j=0}^{N} \binom{N}{j} \left(\frac{p_{N+1}}{1-p_{N+1}}\right)^j (C_{vj} - C_{v(j+1)})}. \tag{4.18}$$

Then $q_v^*(\hat{p})$ defined in (4.16) is non-decreasing in $\hat{p}$. Furthermore, if $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$ holds with strict inequality, then $q_v^*(\hat{p})$ is strictly increasing in $\hat{p}$ for $\hat{p} \in (0, p_{\max})$.

The proof of Theorem 12 is given in Appendix B.2.

Note that, if ϵ_v is small enough to satisfy $C_{vj} = C_{v(j+1)}$ for all $j < J_{\epsilon_v}$, then it should hold that $\gamma_{\epsilon_v} = J_{\epsilon_v}$. Otherwise, $\gamma_{\epsilon_v} \leq J_{\epsilon_v}$ is generally true.

With the two key monotonicity properties, the distributed MAC algorithm is presented in the following.

Distributed MAC algorithm:

1) Initialize the transmission probabilities of all users. Let the transmission probability of user k be denoted by p_k.

2) Let $Q \geq 1$ be a pre-determined integer. Over an interval of Q time slots, the receiver measures the success probability of a virtual packet, denoted by q_v, and feeds q_v back to all transmitters.

3) Upon receiving q_v, each user (transmitter) derives a transmission probability target $\hat{p}$ by solving the following equation

$$q_v^*(\hat{p}) = q_v. \tag{4.19}$$

If a $\hat{p} \in [0, p_{\max}]$ satisfying (4.19) cannot be found, each user sets $\hat{p}$ at $\hat{p} = p_{\max}$ when $q_v > q_v^*(p_{\max})$, or at $\hat{p} = 0$ when $q_v < q_v^*(0)$.

4) Each user, say user k, then updates its transmission probability by

$$p_k = (1 - \alpha)p_k + \alpha\hat{p}, \tag{4.20}$$

where $\alpha > 0$ is the step size parameter.

5) The process is repeated from Step 2 till probabilities of all users converge.

Convergence of the proposed MAC algorithm is stated in the following theorem.

Theorem 13. *Consider the K-user distributed multiple access network presented in this section. Given $x^* > 0$ and $\epsilon_v > 0$. Suppose that b is chosen to satisfy $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$ where γ_{ϵ_v} is defined in (4.18). With the proposed MAC algorithm, associated ODE of the system given in (4.6) has a unique equilibrium at $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$. Furthermore, probability target $\hat{p}(\boldsymbol{p})$ as a function of the transmission probability vector $\mathbf{p}$ satisfies Conditions 1 and 2. Consequently, the distributed probability adaptation converges to the equilibrium $\boldsymbol{p}^*$ in the sense specified in Theorems 9 and 10.*

The proof of Theorem 13 is given in Appendix B.3.

In the above analysis, there is no design constraint on the coding parameters of the virtual packet. Convergence of the distributed MAC algorithm is guaranteed so long as parameter b is chosen to satisfy $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$, where $\gamma_{\epsilon_v} = J_{\epsilon_v}$ if ϵ_v is small enough. However, one should note that optimality of the MAC algorithm can be affected by the value of b and J_{ϵ_v}. Both b and J_{ϵ_v} are determined by the virtual channel parameter set $\{C_{vj}\}$ which is dependent on the virtual packet design. Assume that setting the transmission probabilities of all users at $p = \min\left\{1, \frac{x^*}{K}\right\}$ is an ideal choice for optimizing the chosen utility, which is indeed the case for sum throughput optimization over a collision channel [24] [36]. Because the proposed MAC algorithm sets the system equilibrium at $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$, there are two optimality concerns. On one hand, for a large user number K, it is a general preference that one should design the virtual packet to allow a relatively small value of b, which implies that γ_{ϵ_v} and J_{ϵ_v} should not be much smaller than x^*. On the other hand, for a small user number K, one should also design the virtual packet to support a J_{ϵ_v} value not much larger than x^*, so that $p_{\max} = \min\{1, \frac{x^*}{J_{\epsilon_v}+b}\}$ can be as close to 1 as possible. Considering both optimality concerns, a general guideline is to design coding parameters of the virtual packets such that J_{ϵ_v} (and γ_{ϵ_v}) should be slightly smaller than x^*, and b should be close to 1.

Example 4.1: Consider distributed multiple access networking over a multi-packet reception channel. Assume that all packets should be received successfully if the number of users transmitting in parallel is no more than $\hat{M} = 5$. Otherwise, the receiver should report collision to all users. The real channel parameter set $\{C_{rj}\}$ in this case is given by $C_{rj} = 1$ for $j < 5$ and $C_{rj} = 0$ for $j \geq 5$. Assume that users intend to optimize the symmetric throughput of the system. Consequently, if the user number equals K and all users transmit with an identical probability of p, system utility $U(K, p, \{C_{rj}\})$ is given by

$$U(K, p, \{C_{rj}\}) = \sum_{j=0}^{\min\{K-1,4\}} K\binom{K-1}{j} p^{j+1}(1-p)^{K-1-j}. \quad (4.21)$$

Let $U_{\text{opt}}(K)$ be the optimal sum throughput of the system under the assumption that K is known.

$$U_{\text{opt}}(K) = \max_p U(K, p, \{C_{rj}\}). \quad (4.22)$$

From the asymptotic utility optimization given in (4.13), we obtain $x^* = 3.64$. According to the design guideline presented above, we assume that a virtual packet should be equivalent to the combination of 2 real packets. Consequently, the virtual channel parameter set $\{C_{vj}\}$ is given by $C_{vj} = 1$ for $j < 4$ and $C_{vj} = 0$ for $j \geq 4$. Choose $\epsilon_v = 0.01$, which implies $\gamma_{\epsilon_v} = J_{\epsilon_v} = 3$, and hence we can set $b = 1.01 > \max\{1, x^* - \gamma_{\epsilon_v}\}$. We use $U^*(K)$ to denote the sum throughput of the system when transmission probabilities of all users are set at $p = \min\{p_{\max}, \frac{x^*}{K+b}\}$, where $p_{\max} = \min\{1, \frac{x^*}{J_{\epsilon_v}+b}\}$.

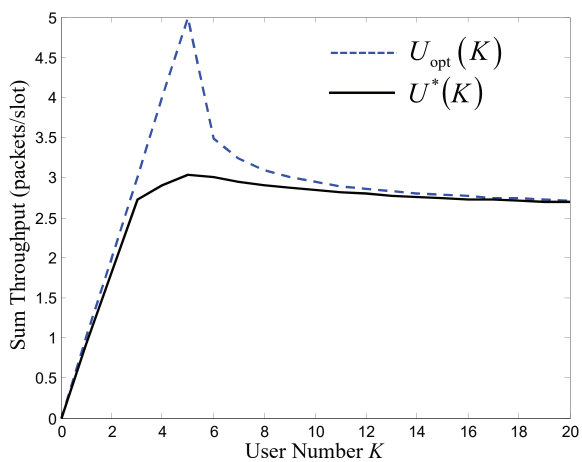

Figure 4.1: Sum throughput of the system as functions of the user number.

Figure 4.1 illustrates the two utility values, $U_{\text{opt}}(K)$ and $U^*(K)$, as functions of user number K. It can be seen that $U^*(K)$ is reasonably close to $U_{\text{opt}}(K)$ when user number K is not close to $\hat{M}$. Note that $U_{\text{opt}}(K)$ is not necessarily achievable since user number K is unknown to the system.

Example 4.2: In this example, we consider distributed multiple access networking over a simple fading channel. Assume that the system has $K = 8$ users and one receiver. In each time slot, with a probability of 0.3, the channel can support no more than $\hat{M}_1 = 4$ parallel real packet transmissions, and with a probability of 0.7, the channel can support no more than $\hat{M}_2 = 6$ parallel real packet transmissions[5]. The real channel parameter set $\{C_{rj}\}$ in this case is given by $C_{rj} = 1$

[5]Such a channel can appear if there is an interfering user that transmits a packet with a probability of 0.3 in each time slot. One packet from the interfering user is equivalent to the combination of two packets from a regular user.

for $j < 4$, $C_{rj} = 0.7$ for $4 \leq j < 6$, and $C_{rj} = 0$ for $j \geq 6$. Assume that users intend to optimize the symmetric system throughput weighted by a transmission energy cost of $E = 0.3$. If user number equals K and all users transmit with a probability of p, system utility $U(K, p, \{C_{rj}\})$ is given by

$$U(K, p, \{C_{rj}\}) = \sum_{j=0}^{K-1} K \binom{K-1}{j} p^{j+1}(1-p)^{K-1-j} C_{rj} - EKp. \tag{4.23}$$

Consequently, x^* can be obtained from the asymptotic utility optimization (4.13) as $x^* = 3.29$.

Assume that a virtual packet should have the same coding parameters as those of a real packet. Consequently, the virtual channel parameter set $\{C_{vj}\}$ is identical to the real channel parameter set, i.e., $C_{vj} = C_{rj}$ for all $j \geq 0$. With $\epsilon_v = 0.01$, we have $\gamma_{\epsilon_v} = J_{\epsilon_v} = 3$ and hence we can set $b = 1.01 > \max\{1, x^* - \gamma_{\epsilon_v}\}$.

We initialize the transmission probabilities of all users at 0. In each time slot, a channel state flag is randomly generated to indicate whether the channel can support the parallel transmissions of no more than $\hat{M}_1 = 4$ packets or $\hat{M}_2 = 6$ packets. Each user also randomly determines whether a packet should be transmitted according to its own transmission probability parameter. Consequently, whether a real packet and the virtual packet can go through the channel successfully or not is determined using the corresponding channel model. We use the following exponential moving average approach to measure q_v[6], which is the success probability of the virtual packet. q_v is initialized at $q_v = 1$. In each time slot, if the virtual packet can be received successfully, an indicator variable I_v is set at $I_v = 1$. If the virtual packet reception fails, we set $I_v = 0$. Success probability of the virtual packet is then updated by $q_v = (1 - \frac{1}{300})q_v + \frac{1}{300}I_v$. The rest of the probability updates proceeds according to the distributed MAC algorithm introduced previously with a constant step size of $\alpha = 0.05$. Convergence behavior in system utility is illustrated in Figure 4.2, where system utility is measured using the same exponential moving average approach as the measurement of q_v except that initial value of the utility is set at 0. Two reference values

[6]While this approach is different from the one proposed in the distributed MAC algorithm, simulations show that an exponential averaging measurement of q_v can often lead the system to converge to the designed equilibrium in a relatively smaller number of time slots.

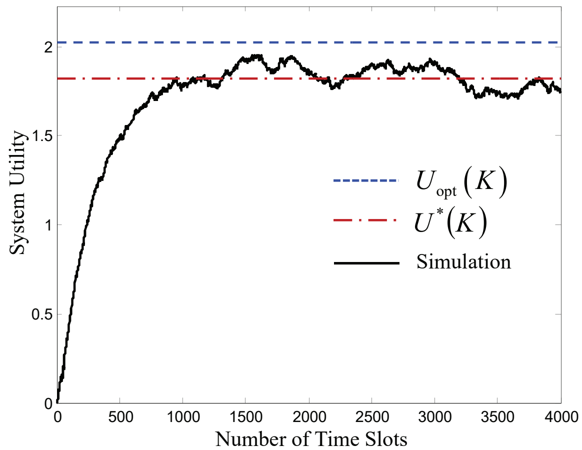

Figure 4.2: Convergence in sum utility of a $K = 8$ user multiple access network over a simple fading channel.

are shown in the figure. $U_{\text{opt}}(K)$ is the optimal utility as defined in (4.22), while $U^*(K)$ is the theoretical utility at the designed equilibrium.

4.3 Single Transmission Option with Interpreted Channel Contention Measure

Classical MAC protocols often assume that a user should get feedback from the receiver on whether its own packet is successfully received or not. This enables each user, say user k, to measure the conditional success probability of its own packet transmissions, denoted by q_k. In this section, we consider the case when q_k is the only feedback available to user k. We also assume that a virtual packet should have the same communication parameters as those of a real packet. In order to apply the MAC algorithm proposed in section 4.2, user k needs to interpret the success probability of the virtual packet based on the measurement of q_k. Because transmission activities of the users are mutually independent, q_k equals the success probability of the virtual packet conditioned on that user k decides to idle. Consequently, user k can calculate the success probability of the virtual packet according to

$$q_v = (1 - p_k)q_k + p_k d_k, \tag{4.24}$$

where p_k is the transmission probability of user k, and d_k is the success probability of the virtual packet given that user k transmits a packet[7]. Note that d_k can be easily calculated in special cases. For example, under a collision channel model, we have $d_k = 0$. In this case, $q_v = (1 - p_k)q_k$ is the actual success probability of the virtual packet. However, for a general channel, d_k may not be available at the transmitters unless additional feedback information is provided.

When d_k is not available, we propose a two-step approach for user k to interpret d_k and hence the success probability of the virtual packet q_v. Based on the interpreted q_v, each user then updates its transmission probability accordingly.

To explain the detail of the two-step approach, we need to define two auxiliary functions. More specifically, for an arbitrary user number estimate $\breve{K}$, let $\breve{N} = \lfloor \breve{K} \rfloor$ denote the largest integer no more than $\breve{K}$. Let $\breve{p} = \frac{x^*}{\breve{K}+b}$, $p_{\breve{N}} = \frac{x^*}{\breve{N}+b}$ and $p_{\breve{N}+1} = \frac{x^*}{\breve{N}+1+b}$, where b is a constant satisfying $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$. We define auxiliary functions $q^*(\breve{p})$ and $d^*(\breve{p})$ as follows

$$
\begin{aligned}
q^*(\breve{p}) = & \frac{\breve{p} - p_{\breve{N}+1}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}-1} \binom{\breve{N}-1}{j} \breve{p}^j (1-\breve{p})^{\breve{N}-1-j} C_{vj} \\
& + \frac{p_{\breve{N}} - \breve{p}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}} \binom{\breve{N}}{j} \breve{p}^j (1-\breve{p})^{\breve{N}-j} C_{vj}, \\
d^*(\breve{p}) = & \frac{\breve{p} - p_{\breve{N}+1}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}-1} \binom{\breve{N}-1}{j} \breve{p}^j (1-\breve{p})^{\breve{N}-1-j} C_{v(j+1)} \\
& + \frac{p_{\breve{N}} - \breve{p}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}} \binom{\breve{N}}{j} \breve{p}^j (1-\breve{p})^{\breve{N}-j} C_{v(j+1)}.
\end{aligned}
\tag{4.25}
$$

The two-step approach is described below.

Step 1: Over an interval of $Q \geq 1$ time slots, each user, say user k, measures its own conditional success probability q_k. User k then obtains an intermediate transmission probability $\breve{p}$ by solving the following equation

$$
q^*(\breve{p}) = q_k. \tag{4.26}
$$

[7]Extensions can be made to the case when a virtual packet is equivalent to the combination of R real packets by decomposing q_k in a similar way as shown in (4.24).

If a $\breve{p} \in [0, p_{\max}]$ satisfying (4.26) cannot be found, user k sets $\breve{p}$ at $\breve{p} = p_{\max}$ when $q_k > q^*(p_{\max})$, or at $\breve{p} = 0$ when $q_k < q^*(0)$.

Step 2: In the second step, user k interprets channel contention measure q_v as

$$q_v = (1 - p_k)q_k + p_k d^*(\breve{p}). \tag{4.27}$$

An updated transmission probability target $\hat{p}$ for user k is then determined by solving equation (4.19). As before, if a $\hat{p} \in [0, p_{\max}]$ satisfying (4.19) cannot be found, user k sets $\hat{p}$ at $\hat{p} = p_{\max}$ when $q_v > q_v^*(p_{\max})$, or at $\hat{p} = 0$ when $q_v < q_v^*(0)$.

Note that with $\hat{p}$ being obtained by the two-step approach, a convergence proof of the MAC algorithm is no longer available. Because one cannot guarantee that the transmission probability target $\hat{p}$ derived by different users should always be identical, and therefore the assumption of all users taking the same transmission probability at any system equilibrium is no longer valid. Nevertheless, in the following theorem, we show that the two-step approach is equivalent to a simplified one-step approach where user k directly uses $\breve{p}$ obtained in (4.26) as its transmission probability target.

Theorem 14. *Suppose for each user, say user k, first obtains an intermediate transmission probability $\breve{p}$ and then determines its transmission probability target $\hat{p}$ by following the two-step approach. Then $\breve{p} \geq p_k$ implies $\hat{p} \geq p_k$, while $\breve{p} \leq p_k$ implies $\hat{p} \leq p_k$.*

The proof of Theorem 14 is given in Appendix B.4.

Note that $q^*(\breve{p})$ is non-decreasing in $\breve{p}$ if $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$, whose proof follows that of Theorem 12. Theorem 14 implies that the two-step approach and the simplified one-step approach are equivalent in the sense of giving the same directional information on whether the transmission probability should be increased or decreased. In the case when the two-step approach does lead the system to the desired equilibrium, Theorem 14 suggests a simple probability adaptation principle. That is, user k can compare its own conditional success probability q_k with $q^*(p_k)$ to determine whether p_k should be increased or decreased.

Example 4.3: In this example, we study the convergence property of the algorithm proposed in Section 4.3 by assuming that each user only gets feedback from the receiver on whether its own packet is successfully received or not. Due to the equivalence of the two-step and one-step approach shown in Theorem 14, one-step approach will be used to update each user's transmission probability.

Consider a distributed multiple access network with $K = 7$ users over a multi-packet reception channel. The channel can support the parallel transmissions of up to $M = 4$ users. The real channel parameter set $\{C_{rj}\}$ is then given by $C_{rj} = 1$ for $j \leq 3$, $C_{rj} = 0$ otherwise. Assume that users intend to optimize the symmetric system throughput, and x^* is calculated from

$$x^* = \operatorname*{argmax}_x xe^{-x} \sum_{j=0}^{M-1} \frac{x^j}{j!} \approx 2.95. \tag{4.28}$$

Assume that a virtual packet should have the same coding parameters as those of a real packet. Consequently, the virtual channel parameter set $\{C_{vj}\}$ is identical to the real channel parameter set, i.e., $C_{vj} = C_{rj}$ for all $j \geq 0$. This implies that $\gamma_{\epsilon_v} = J_{\epsilon_v} = 3$ with $\epsilon_v = 0.01$ and we can therefore choose $b = 1.01$ to guarantee convergence.

Similarly, we initialize the transmission probabilities of all users at 0. In a time slot, each user randomly determines either to transmit a packet or to idle according to its own transmission probability parameter. In this example, we assume that each user only has access to the success/failure status of its own packet transmissions. This enables each user, say user k, to measure the conditional success probability of its own packet transmissions, denoted by q_k. We use the similar exponential moving average approach for user k to measure q_k, and q_k is initialized at $q_k = 1$ for all k. In each time slot, q_k is updated by $q_k = (1 - \frac{1}{300})q_k + \frac{1}{300}I_k$, where $I_k \in \{0, 1\}$ is an indicator of the success/failure status of packet transmission of user k in the current time slot. User k then obtains the transmission probability target $\breve{p}$ from (4.26), and sets $\hat{p} = \breve{p}$. The rest of the probability updates proceeds according to the distributed MAC algorithm introduced before with a constant step size of $\alpha(t) = 0.05$. Convergence of the sum utility of the system is illustrated in Figure 4.3, where system utility is also measured using the same exponential moving average approach with initial value of the utility being set at 0. The dash-dotted line represents the sum throughput of the system if all users transmit with the desired probability $p^* = \frac{x^*}{K+b} = 0.37$. In this example, sum system throughput at the desired equilibrium is about 5% below the optimum value, which is illustrated by the dotted line. After 4000 time slots, the vector of transmission probabilities of all users is given by $[0.37, 0.37, 0.36, 0.35, 0.36, 0.36, 0.36]^T$.

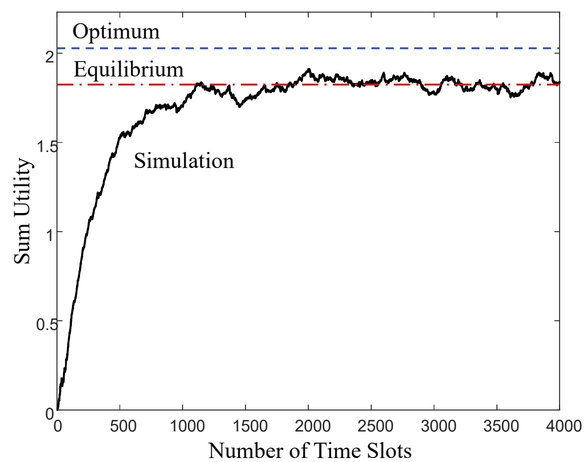

Figure 4.3: Convergence in sum system throughput of a $K = 7$ users multiple access network over a multi-packet reception channel.

4.4 Multiple Transmission Options, Single Virtual Packet

In this section, we consider the case when each user is equipped with $M \geq 2$ transmission options plus an idling option. Each user, say user k, should maintain an M-length transmission probability vector $\boldsymbol{p}_k = p_k \boldsymbol{d}_k$, where p_k is the transmission probability and $\boldsymbol{d}_k$ is the transmission direction vector of user k. Transmission probability vectors of all users are listed in an MK-length vector $\boldsymbol{P} = [\boldsymbol{p}_1^T, \ldots, \boldsymbol{p}_K^T]^T$. As in the previous section, with a general channel model and without knowing the user number K, the objective is to design a distributed MAC algorithm to lead the system to a unique equilibrium that maximizes a chosen symmetric network utility.

Given the physical layer channel and the transmission options, the link layer multiple access channel is specified using two sets of channel parameter functions. Assume that all users have the same transmission direction vector $\boldsymbol{d}$. Define $\{C_{rij}(\boldsymbol{d})\}$ for $1 \le i \le M$ and $j \ge 0$ as the "real channel parameter function set", where $C_{rij}(\boldsymbol{d})$ is the conditional success probability of the ith real packet, should it be transmitted in parallel with other j real packets. Because each packet can be generated from a randomly chosen transmission option, $C_{rij}(\boldsymbol{d})$ is a function of $\boldsymbol{d}$. In this section, ssume that there is a single virtual packet being transmitted in each time slot. Virtual packets being transmitted in different time slots are identical. Given coding parameters of the virtual packet, under the assumption that all users should have the same transmission probability vector $\boldsymbol{p} = p\boldsymbol{d}$, define $\{C_{vj}(\boldsymbol{d})\}$ as the "virtual channel parameter function set", where $C_{vj}(\boldsymbol{d})$ is the success probability of the virtual packet should it be transmitted in parallel with j real packets. Assume that $C_{vj}(\boldsymbol{d}) \ge C_{v(j+1)}(\boldsymbol{d})$ should hold for all $j \ge 0$ and for any $\boldsymbol{d}$. That is, under the same transmission direction vector $\boldsymbol{d}$, if the number of parallel real packet transmissions increases, the chance of a virtual packet getting through the channel should not increase. Let $\epsilon_v > 0$ be a pre-determined small constant. We define $J_{\epsilon_v}(\boldsymbol{d})$ as the minimum integer such that $C_{vJ_{\epsilon_v}}(\boldsymbol{d})$ is ϵ_v larger than $C_{v(J_{\epsilon_v}+1)}(\boldsymbol{d})$, i.e.,

$$J_{\epsilon_v}(\boldsymbol{d}) = \arg\min_j C_{vj}(\boldsymbol{d}) > C_{v(j+1)}(\boldsymbol{d}) + \epsilon_v. \tag{4.29}$$

Both the real and the virtual channel parameter function sets are assumed to be known at the transmitters and at the receiver.

Assume that users intend to maximize a symmetric network utility $U(K, p\boldsymbol{d}, \{C_{rij}(\boldsymbol{d})\})$. Under the assumption that all users should have the same transmission probability vector $\boldsymbol{p}$, system utility is a function of the unknown user number K, the common transmission probability vector $\boldsymbol{p} = p\boldsymbol{d}$, and the real channel parameter function set $\{C_{rij}(\boldsymbol{d})\}$. For example, if the ith real packet has a communication rate of r_i (in units/time slot), and we choose sum throughput of the system as the utility function, then $U(K, \boldsymbol{p}, \{C_{rij}(\boldsymbol{d})\})$ should be given by

$$U(K, \boldsymbol{p}, \{C_{rij}(\boldsymbol{d})\}) = K \sum_{i=1}^{M} d_i r_i \sum_{j=0}^{K-1} \binom{K-1}{j} p^{j+1}(1-p)^{K-1-j} C_{rij}(\boldsymbol{d}). \tag{4.30}$$

We intend to design a distributed MAC algorithm to maximize $U(K, \boldsymbol{p}, \{C_{rij}(\boldsymbol{d})\})$ by maintaining channel contention at a desired level. Let q_v denote the success probability of the virtual packet. As before, we term q_v the "channel contention measure" because it is used to measure the contention level of the system. $q_v(\boldsymbol{P}, K)$ is a function of the user number K and the MK-length transmission probability vector $\boldsymbol{P}$ consisting of transmission probability vectors of all users. Because $q_v(\boldsymbol{P}, K)$ equals the summation of a finite number of polynomial terms, it should be Lipschitz continuous in $\boldsymbol{P}$ for any finite K. When all users have the same transmission probability vector $\boldsymbol{p} = p\boldsymbol{d}$, we also write q_v as

$$q_v(\boldsymbol{p}, K) = \sum_{j=0}^{K} \binom{K}{j} p^j (1-p)^{K-j} C_{vj}(\boldsymbol{d}). \tag{4.31}$$

Upon obtaining q_v from the receiver, we assume that each user should first derive a user number estimate $\hat{K}$ by comparing q_v with a "theoretical channel contention measure" $q_v^*(\hat{K})$, which is defined as a continuous function of $\hat{K}$. A user should then set its transmission probability vector target $\hat{\boldsymbol{p}}$ according to a designed theoretical vector parameter function $\boldsymbol{p}(\hat{K})$. To understand key properties that the $\boldsymbol{p}(\hat{K})$ function should possess, let us first take a look at the following example.

Example 4.4: Consider a distributed multiple access network with K homogeneous users. Assume that each user has two transmission options plus an idling option. The two transmission options are labeled as the "high rate" option and the "low rate" option, respectively. If users transmit with the low rate option only, then the channel can support the parallel transmissions of no more than 12 packets. Assume that a high rate packet is equivalent to the combination of 4 low rate packets. Therefore, if n_l low rate packets and n_h high rate packets are transmitted in parallel, the packets can be received successfully if and only if $n_l + 4n_h \leq 12$. Assume that users intend to optimize the sum throughput of the network, and transmission probability vectors of the users should be identical at the equilibrium. When all users have the same probability vector

$\boldsymbol{p} = [p_h, p_l]^T$, system utility, denoted by $U(K, \boldsymbol{p})$ as a function of K and $\boldsymbol{p}$, is given by

$$U(K,\boldsymbol{p}) = \sum_{\substack{n_h \geq 0, n_l \geq 0,\\ n_h + n_l \leq K-1,\\ 4(n_h+1)+n_l \leq 12}} 4K\binom{K-1}{n_h, n_l} p_h^{n_h+1} p_l^{n_l} (1-p_h-p_l)^{K-1-n_h-n_l} + \sum_{\substack{n_h \geq 0, n_l \geq 0,\\ n_h + n_l \leq K-1,\\ 4n_h+n_l+1 \leq 12}} K\binom{K-1}{n_h, n_l} p_h^{n_h} p_l^{n_l+1} (1-p_h-p_l)^{K-1-n_h-n_l}. \tag{4.32}$$

Given user number K, let $\boldsymbol{p}^* = \arg\max_{\boldsymbol{p}} U(K, \boldsymbol{p})$ be the optimal transmission probability vector. p_h^* and p_l^* as functions of user number K are illustrated in Figure 4.4. We can see that, if we write

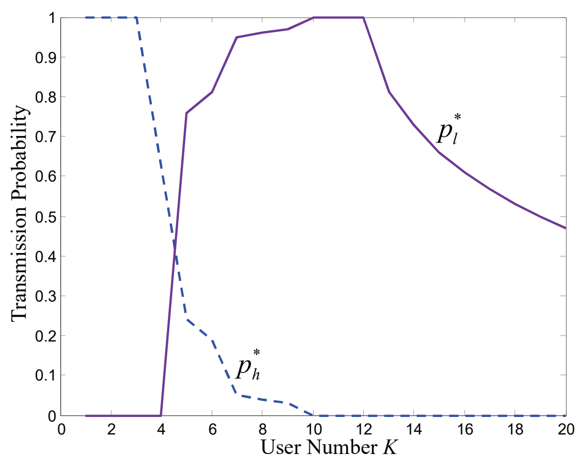

Figure 4.4: Optimal transmission probabilities of a K-user multiple access system with each user having two transmission options.

$\boldsymbol{p}^* = p^* \boldsymbol{d}^*$, then we have $\boldsymbol{d}^* = [1, 0]^T$ for $K \leq 4$, and $\boldsymbol{d}^* = [0, 1]^T$ for $K \geq 10$. $\boldsymbol{d}^*$ transits from $[1, 0]^T$ to $[0, 1]^T$ in the region of $4 \leq K \leq 10$.

According to the above observation, we assume that the vector parameter function $\boldsymbol{p}(\hat{K})$ to be designed should possess the following property termed the "Head and Tail Condition".

Condition 3. *(Head and Tail) Let $\epsilon_v > 0$ be a pre-determined constant. Let J_{ϵ_v} be defined in (4.29). There exist two integer-valued constants $0 < \underline{K} \le \overline{K}$, such that,*

1) $\boldsymbol{d}(\hat{K}) = \boldsymbol{d}(\underline{K})$, for all $\hat{K} \le \underline{K}$, $\underline{K} \ge J_{\epsilon_v}(\boldsymbol{d}(\underline{K}))$.

2) $\boldsymbol{d}(\hat{K}) = \boldsymbol{d}(\overline{K})$, for all $\hat{K} \ge \overline{K}$, $\overline{K} > J_{\epsilon_v}(\boldsymbol{d}(\overline{K}))$.

Condition 3 indicates that, when $\hat{K} \le \underline{K}$ is small enough or when $\hat{K} \ge \overline{K}$ is large enough, $\boldsymbol{d}(\hat{K})$ should stop changing in $\hat{K}$. In these two regimes, the system with multiple transmission options becomes equivalent to a system with a single transmission option. The virtual channel parameter set of the equivalent system is given by $\{C_{vj}\} = \{C_{vj}(\boldsymbol{d})\}$. Calculation of the real channel parameter set of the equivalent system, on the other hand, depends on the chosen utility function. If the utility function is the sum throughput given in (4.30) for example, the equivalent real channel parameter set $\{C_{rj}\}$ should be obtained by $C_{rj} = \sum_{i=1}^{M} d_i r_i C_{rij}(\boldsymbol{d})$, for $j \ge 0$. We assume that core parameter functions of the MAC algorithm, i.e., the theoretical channel contention measure $q_v^*(\hat{K})$ and the probability target function $p(\hat{K})$, should be designed according to the guideline given in Section 4.2 for $\hat{K} \le \underline{K}$ and $\hat{K} \ge \overline{K}$. The corresponding details are not repeated in this section.

Let us temporarily assume that the vector parameter function $\boldsymbol{p}(\hat{K})$ has been determined completely, not just for $\hat{K} \le \underline{K}$ and $\hat{K} \ge \overline{K}$, but also for $\underline{K} < \hat{K} < \overline{K}$. To present the distributed MAC algorithm, we need to define the theoretical channel contention measure $q_v^*(\hat{K})$ as follows. Let $N = \lfloor \hat{K} \rfloor$ be the largest integer below $\hat{K}$. For $\hat{K} \le \underline{K}$ and $\hat{K} \ge \overline{K}$, $q_v^*(\hat{K})$ is defined by

$$q_v^*(\hat{K}) = \frac{p(\hat{K}) - p(N+1)}{p(N) - p(N+1)} q_v(\boldsymbol{p}(\hat{K}), N) + \frac{p(N) - p(\hat{K})}{p(N) - p(N+1)} q_v(\boldsymbol{p}(\hat{K}), N+1), \quad (4.33)$$

which is consistent with (4.16). For $\underline{K} \le \hat{K} \le \overline{K}$, $q_v^*(\hat{K})$ is defined by

$$q_v^*(\hat{K}) = (N + 1 - \hat{K}) q_v(\boldsymbol{p}(\hat{K}), N) + (\hat{K} - N) q_v(\boldsymbol{p}(\hat{K}), N+1). \quad (4.34)$$

In other words, if $\hat{K}$ is integer-valued, $q_v^*(\hat{K}) = q_v(\boldsymbol{p}(\hat{K}), \hat{K})$ equals the channel contention measure when all users have the same transmission probability vector $\boldsymbol{p}(\hat{K})$ and the user number equals $K = \hat{K}$. If $\hat{K}$ is not integer-valued, on the other hand, $q_v^*(\hat{K})$ is a linear interpolation between $q_v(\boldsymbol{p}(\hat{K}), \lfloor \hat{K} \rfloor)$ and $q_v(\boldsymbol{p}(\hat{K}), \lfloor \hat{K} \rfloor + 1)$. Note that the interpolation approach used for $\hat{K} \leq \underline{K}$ and $\hat{K} \geq \overline{K}$ is different from the one used for $\underline{K} \leq \hat{K} \leq \overline{K}$.

Next, we present the distributed MAC algorithm below.

Distributed MAC algorithm:

1) Initialize the transmission probability vectors of all users. Let the transmission probability vector of user k be denoted by $\boldsymbol{p}_k$.

2) Let $Q \geq 1$ be a pre-determined integer. Over an interval of Q time slots, the receiver measures (or estimates) the success probability of the virtual packet, denoted by q_v, and feeds q_v back to all transmitters.

3) Upon receiving q_v, each user (transmitter) derives a user number estimate $\hat{K}$ by solving the following equation

$$q_v^*(\hat{K}) = q_v. \tag{4.35}$$

If a $\hat{K}$ satisfying (4.35) cannot be found, user k sets $\hat{K} = J_{\epsilon_v}(\boldsymbol{d}(\underline{K}))$ if $q_v > q_v^*(J_{\epsilon_v}(\boldsymbol{d}(\underline{K})))$, or sets $\hat{K} = \infty$ otherwise.

4) Each user, say user k, then updates its transmission probability vector by

$$\boldsymbol{p}_k = (1 - \alpha)\boldsymbol{p}_k + \alpha \boldsymbol{p}(\hat{K}), \tag{4.36}$$

where $\alpha > 0$ is the step size parameter.

5) The process is repeated from Step 2 till transmission probability vectors of all users converge.

We intend to design the distributed MAC algorithm with the following convergence property. Note that user number K is assumed to be unknown to all users and to the receiver. If

$K \geq J_{\epsilon_v}(\boldsymbol{d}(\underline{K}))$, we intend to have $\hat{K} = K$ at the equilibrium, while if $K < J_{\epsilon_v}(\boldsymbol{d}(\underline{K}))$, we intend to have $\hat{K} = J_{\epsilon_v}(\boldsymbol{d}(\underline{K}))$ at the equilibrium. In order to ensure convergence of the proposed MAC algorithm, we require that the vector parameter function $\boldsymbol{p}(\hat{K})$ and the corresponding theoretical channel contention measure $q_v^*(\hat{K})$ should satisfy the following "Monotonicty and Gradient Condition" for $\underline{K} \leq \hat{K} \leq \overline{K}$.

Condition 4. *(Monotonicity and Gradient) For $\underline{K} \leq \hat{K} \leq \overline{K}$,*

1) *$\boldsymbol{p}(\hat{K}) = p(\hat{K})\boldsymbol{d}(\hat{K})$ should be Lipschitz continuous in $\hat{K}$, i.e., there exists a constant $K_g > 0$, such that for all $\hat{K}_a, \hat{K}_b \in [\underline{K}, \overline{K}]$, we have*

$$\|\boldsymbol{p}(\hat{K}_a) - \boldsymbol{p}(\hat{K}_b)\| \leq K_g|\hat{K}_a - \hat{K}_b|. \tag{4.37}$$

2) *$q_v^*(\hat{K})$ should be continuous and be strictly decreasing in $\hat{K}$. There exists a positive constant $\epsilon_q > 0$, such that for all $\hat{K}_a, \hat{K}_b \in [\underline{K}, \overline{K}]$, we have*

$$|q_v^*(\hat{K}_a) - q_v^*(\hat{K}_b)| \geq \epsilon_q|\hat{K}_a - \hat{K}_b|. \tag{4.38}$$

3) *There exists a constant $\epsilon_v > 0$, such that $\hat{K} > J_{\epsilon_v}(\boldsymbol{d}(\hat{K}))$ should be satisfied for all $\hat{K} \in [\underline{K}, \overline{K}]$.*

4) *There exist constants $0 < \underline{p} < \overline{p} < 1$, such that $\underline{p} \leq p(\hat{K}) \leq \overline{p}$ should be satisfied for all $\hat{K} \in [\underline{K}, \overline{K}]$.*

As a special case, it can be verified that, in terms of designing function $\boldsymbol{p}(\hat{K}) = p(\hat{K})\boldsymbol{d}(\hat{K})$, if one fixes $\boldsymbol{d}(\hat{K}) = \boldsymbol{d}(\underline{K}) = \boldsymbol{d}(\overline{K})$ and sets $p(\hat{K})$ according to the guideline given in Section 4.2, the resulting $\boldsymbol{p}(\hat{K})$ and $q_v^*(\hat{K})$ functions do satisfy the Monotonicity and Gradient Condition for $\underline{K} \leq \hat{K} \leq \overline{K}$.

Convergence of the distributed MAC algorithm is stated in the following theorem.

Theorem 15. *Consider a multiple access system with K users adopting the proposed distributed MAC algorithm to update their transmission probability vectors. Under Condition 3, let $\mathbf{p}(\hat{K})$ and $q_v^*(\hat{K})$ be designed for $\hat{K} \leq \underline{K}$ and $\hat{K} \geq \overline{K}$ according to the guideline given in Section 4.2. Let $\boldsymbol{p}(\hat{K})$ and $q_v^*(\hat{K})$ be designed to satisfy Condition 4 for $\underline{K} \leq \hat{K} \leq \overline{K}$. Then the associated ODE of the system given in (4.6) has a unique equilibrium at $\boldsymbol{P}^* = \mathbf{1} \otimes \boldsymbol{p}(K)$. The probability vector target $\hat{\mathbf{p}}(\boldsymbol{P})$ as a function of the transmission probability vector $\boldsymbol{P}$ satisfies Conditions 1 and 2. Consequently, the distributed probability vector adaptation converges to the unique equilibrium $\boldsymbol{P}^*$ in the sense explained in Theorems 9 and 10.*

Theorem 15 is implied by Theorem 17.

Note that, in the Monotonicity and Gradient Condition 4, while we still require $q_v^*(\hat{K})$ be strictly decreasing in $\hat{K}$, being different from the single transmission option case, we no longer require $q_v(\boldsymbol{p}(\hat{K}), K)$ be strictly increasing in $\hat{K}$ for a given K. Also being different from the single transmission option case where the $\boldsymbol{p}(\hat{K})$ function is completely specified in a closed form, Condition 4 does not explain how $\boldsymbol{p}(\hat{K})$ should be designed to satisfy the conditions.

Next, we will show that, so long as one can manually design $\boldsymbol{p}(\hat{K})$ for a set of chosen points with integer-valued $\hat{K}$ to satisfy a set of "Pinpoints Condition", then there is a simple approach to complete the $\boldsymbol{p}(\hat{K})$ function for $\underline{K} \leq \hat{K} \leq \overline{K}$ to satisfy Condition 4.

Condition 5. *(Pinpoints) Let $\underline{K} = \hat{K}_0 < \hat{K}_1 < \cdots < \hat{K}_L = \overline{K}$ be a collection of integer-valued points. For $i = 1, \ldots, L$, and $0 \leq \lambda < 1$, define*

$$\begin{aligned}
\hat{K}_{i\lambda} &= (1-\lambda)\hat{K}_{i-1} + \lambda \hat{K}_i \\
\boldsymbol{d}_{i\lambda} &= (1-\lambda)\boldsymbol{d}(\hat{K}_{i-1}) + \lambda \boldsymbol{d}(\hat{K}_i) \\
q^*_{vi\lambda} &= (1-\lambda)q_v^*(\hat{K}_{i-1}) + \lambda q_v^*(\hat{K}_i). \qquad (4.39)
\end{aligned}$$

We have the following conditions.

1) *There exists a positive constant $\epsilon_q > 0$, such that, for all $i = 1, \ldots, L$, $q_v^*(\hat{K}_{i-1}) - q_v^*(\hat{K}_i) \geq \epsilon_q$.*

2) There exists a constant $\epsilon_v > 0$, such that for all $i = 1, \ldots, L$ and $0 \leq \lambda < 1$, $\hat{K}_{i\lambda} > J_{\epsilon_v}(\boldsymbol{d}_{i\lambda})$, where $J_{\epsilon_v}(\boldsymbol{d}_{i\lambda})$ is defined in (4.29).

3) There exist $0 < \underline{p} < \overline{p} < 1$, such that $\underline{p} \leq p(\hat{K}_i) \leq \overline{p}$ should be satisfied for all $i = 1, \ldots, L$.

3) Extend the definition of $q_v(\boldsymbol{p}, \hat{K})$ to non-integer-valued $\hat{K}$ as

$$q_v(\boldsymbol{p}, \hat{K}) = (\lfloor \hat{K} \rfloor + 1 - \hat{K}) q_v(\boldsymbol{p}, \lfloor \hat{K} \rfloor) + (\hat{K} - \lfloor \hat{K} \rfloor) q_v(\boldsymbol{p}, \lfloor \hat{K} \rfloor + 1). \quad (4.40)$$

The following inequality should be satisfied for all $i = 1, \ldots, L$ and for all $0 \leq \lambda < 1$.

$$q_v\left(\overline{p}\boldsymbol{d}_{i\lambda}, \hat{K}_{i\lambda}\right) \leq q^*_{vi\lambda} \leq q_v\left(\underline{p}\boldsymbol{d}_{i\lambda}, \hat{K}_{i\lambda}\right). \quad (4.41)$$

The next "Interpolation Approach" shows that, so long as $\boldsymbol{p}(\hat{K})$ is designed for the pinpoints, it is easy to complete the whole $\boldsymbol{p}(\hat{K})$ function for $\underline{K} \leq \hat{K} \leq \overline{K}$.

Interpolation Approach: Assume that $\boldsymbol{p}(\hat{K})$ is designed for a given set of pinpoints $\{\hat{K}_i\}$, $i = 0, \ldots, L$, with $\hat{K}_0 = \underline{K} < \hat{K}_1, < \cdots < \hat{K}_L = \overline{K}$, to satisfy Condition 5. For $i = 1, \ldots, L$ and $0 \leq \lambda < 1$, let $\hat{K}_{i\lambda}$, $\boldsymbol{d}_{i\lambda}$ and $q^*_{vi\lambda}$ be defined in (4.39). Let $q_v(\boldsymbol{p}, \hat{K})$ be defined in (4.40). We choose $p(\hat{K}_{i\lambda})$ to satisfy

$$q_v(p(\hat{K}_{i\lambda})\boldsymbol{d}(\hat{K}_{i\lambda}), \hat{K}_{i\lambda}) = q^*_{vi\lambda}. \quad (4.42)$$

Consequently, $\boldsymbol{p}(\hat{K}_{i\lambda})$ is designed as $\boldsymbol{p}(\hat{K}_{i\lambda}) = p(\hat{K}_{i\lambda})\boldsymbol{d}_{i\lambda}$.

Note that according to (4.41), a solution of $\underline{p} \leq p(\hat{K}_{i\lambda}) \leq \overline{p}$ satisfying (4.42) must exist. Effectiveness of the Interpolation Approach is stated in the following theorem.

Theorem 16. *Assume that $\boldsymbol{p}(\hat{K})$ is designed for a set of $L+1$ pinpoints $\{\hat{K}_i\}$, $i = 0, \ldots, L$, with $\hat{K}_0 = \underline{K} < \hat{K}_1, < \cdots < \hat{K}_L = \overline{K}$, to satisfy Condition 5. After completing the function using the Interpolation Approach, we have $\boldsymbol{p}(\hat{K})$ and $q^*_v(\hat{K})$ functions satisfy the Monotonicity and Gradient Condition 4 for $\underline{K} \leq \hat{K} \leq \overline{K}$.*

Theorem 16 is implied by Theorem 18.

Note that, in the single transmission option case discussed in Section 4.2, $p(\hat{K})$ is specified in a closed form with a small number of design parameters. Monotonicity property of $q_v^*(\hat{K})$ is proven theoretically. With multiple transmission options, however, such a direct-design approach faces a key challenge. Due to generality of the system model, when $\boldsymbol{d}(\hat{K})$ changes in $\hat{K}$ and consequently affects the channel parameters, it is often difficult to theoretically characterize its impact on the $q_v^*(\hat{K})$ function. Alternatively, we switch to a search-assisted approach to first manually design $\boldsymbol{p}(\hat{K})$ for a set of pinpoints to satisfy Condition 5, and then to use the Interpolation Approach to complete the $\boldsymbol{p}(\hat{K})$ function. Note that the Interpolation Approach only ensures convergence of the proposed MAC algorithm. It pays no attention to the optimality, in terms of the utility value, of the design outcome. Therefore, one often needs to carefully adjust the design of the pinpoints to direct the $\boldsymbol{p}(\hat{K})$ function toward a near optimal solution.

Example 4.5: Let us use the system introduced in Example 4.4 to illustrate the design procedure of the $\boldsymbol{p}(\hat{K})$ function. First, we consider the "Head" and the "Tail" regimes when $\hat{K}$ is either small or large in value. We will add subscript "H" to parameters of the "Head" regime, and add subscript "T" to parameters of the "Tail" regime. Without specifying the values of $\underline{K}$ and $\overline{K}$, we first determine the optimal transmission directions $\boldsymbol{d}_H = [1, 0]^T$ and $\boldsymbol{d}_T = [0, 1]^T$ for the "Head" regime and "Tail" regime, respectively. In other words, users should only use the high rate option in the "Head" regime and only use the low rate option in the "Tail" regime. In the "Head" regime, the channel can support the parallel transmissions of no more than 3 high rate packets. The real channel parameter set for the equivalent single option system is given by $\{C_{rj}\}_H$ with $C_{rj} = 1$ for $j \leq 2$ and $C_{rj} = 0$ otherwise. By following the single option system design guideline, we get $x_H^* = \arg\max_x (x + x^2 + \frac{x^3}{2})e^{-x} = 2.27$. We design the virtual packet to be equivalent to a real high rate packet. Consequently, virtual channel parameter set for the equivalent single option system is given by $\{C_{vj}\}_H = \{C_{rj}\}_H$. With $\epsilon_v = 0.01$, we get $\gamma_{\epsilon_v H} = J_{\epsilon_v H} = 2$, and $b_H = 1.01$. In the "Tail" regime, on the other hand, the channel can support the parallel transmissions of no more than 12 low rate packets. The real channel parameter set for the equivalent system is given by $\{C_{rj}\}_T$ with $C_{rj} = 1$ for $j \leq 11$ and $C_{rj} = 0$ otherwise. This yields $x_T^* = \arg\max_x \sum_{i=0}^{11} \frac{x^{i+1}}{i!} e^{-x} = 8.82$.

Because we already chose the virtual packet to be equivalent to a high rate real packet, virtual channel parameter set in this case is given by $\{C_{vj}\}_T$ with $C_{vj} = 1$ for $j \leq 8$ and $C_{vj} = 0$ otherwise. Therefore, with $\epsilon_v = 0.01$, we have $\gamma_{\epsilon_v T} = J_{\epsilon_v T} = 8$, and luckily, this supports $b_T = 1.01$.

To determine the values of $\underline{K}$ and $\overline{K}$, we compare the following two schemes. In the first "high rate option only" scheme, we fix $\boldsymbol{d}(\hat{K})$ at $[1, 0]^T$ for all $\hat{K}$, and set $p(\hat{K}) = \min\left\{p_{\max H}, \frac{x_H^*}{\hat{K}+b_H}\right\}$, where $p_{\max H} = \frac{x_H^*}{J_{\epsilon_v H}+b_H}$. In the second "low rate option only" scheme, we fix $\boldsymbol{d}(\hat{K})$ at $[0, 1]^T$ for all $\hat{K}$, and set $p(\hat{K}) = \min\left\{p_{\max T}, \frac{x_T^*}{\hat{K}+b_T}\right\}$, where $p_{\max T} = \frac{x_T^*}{J_{\epsilon_v T}+b_T}$. By comparing the utility values and the theoretical channel contention measures of the two schemes, we choose $\underline{K} = 4$ and $\overline{K} = 10$.

Now consider the design conditions for $\underline{K} \leq \hat{K} \leq \overline{K}$. For transmission direction $\boldsymbol{d}$ with $d_1 > 0$, we generally have $J_{\epsilon_v} = 2$. Therefore, so long as $\boldsymbol{d}(\hat{K})$ does not transit too quickly to $[0, 1]^T$, the condition of $\hat{K} > J_{\epsilon_v}(\boldsymbol{d}(\hat{K}))$ should hold. Consequently, only two other key conditions need to be satisfied. The first condition is that $q_v^*(\hat{K})$ of the selected pinpoints must be strictly decreasing in $\hat{K}$. The second condition is that $p(\hat{K})$ found in the Interpolation Approach should be bounded away from 0 and 1. From the optimal scheme, we can see that $\boldsymbol{d}(\hat{K})$ should transit toward $[0, 1]^T$ faster than a linearly transition from $\underline{K}$ to $\overline{K}$.

With these considerations, we choose the following 4 pinpoints. At the edge of the "Head" and the "Tail" regimes, we have $\hat{K}_0 = \underline{K} = 4$ with $\boldsymbol{p}(4) = \frac{x_H^*}{\underline{K}+b_H}[1, 0]^T$ and $\hat{K}_3 = \overline{K} = 10$ with $\boldsymbol{p}(10) = \frac{x_T^*}{\overline{K}+b_T}[0, 1]^T$. For the other two pinpoints, $\hat{K}_1 = 5$ and $\hat{K}_2 = 6$, we set their transmission directions at the corresponding optimal transmission direction vectors, i.e., direction vectors extracted from the optimal $\boldsymbol{p}$ vectors that maximize the sum throughput at $K = 5$ and $K = 6$, respectively. Transmission probabilities of these two pinpoints are chosen such that the resulting $q_v^*(\hat{K})$ equals $\frac{\overline{K}-\hat{K}}{\overline{K}-\underline{K}} q_v^*(\underline{K}) + \frac{\hat{K}-\underline{K}}{\overline{K}-\underline{K}} q_v^*(\overline{K})$. The purpose of including $\hat{K}_1 = 5$ and $\hat{K}_2 = 6$ in the pinpoint set is to force $\boldsymbol{d}(\hat{K})$ to transit quickly toward $[0, 1]^T$. The rest of the $\boldsymbol{p}(\hat{K})$ function is completed using the Interpolation Approach. Theoretical channel contention measure $q_v^*(\hat{K})$ of the designed system is illustrated in Figure 4.5 as a function of the user number.

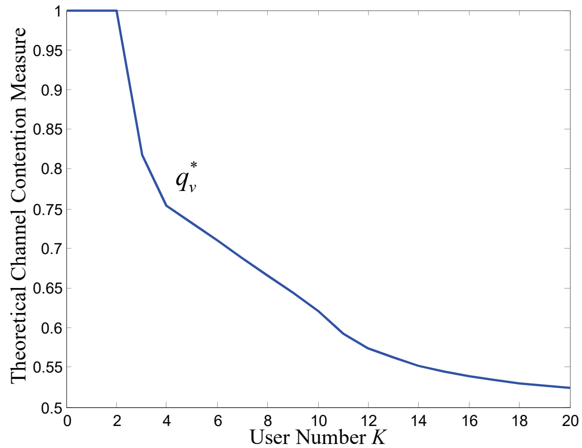

Figure 4.5: Theoretical channel contention measure q_v^* as a function of the user number.

In the following figure (Figure 4.6), we illustrate the theoretical sum system throughput as a function of user number K for the following four different scenarios: optimum $\boldsymbol{p}(K)$, designed $\boldsymbol{p}(K)$, $\boldsymbol{p}(K)$ from the high rate option only scheme, and $\boldsymbol{p}(K)$ from the low rate option only scheme. Assume that the high rate option only scheme should be reasonably good in the "Head" regime while the low rate option only scheme should be reasonably good in the "Tail" only regime. It can be seen that the designed $\boldsymbol{p}(\hat{K})$ function can help to bridge the two simple schemes and to efficiently exploit the benefit of the two transmission options. Note that the optimal utility illustrated in Figure 4.6 is not necessarily achievable without the knowledge of user number K.

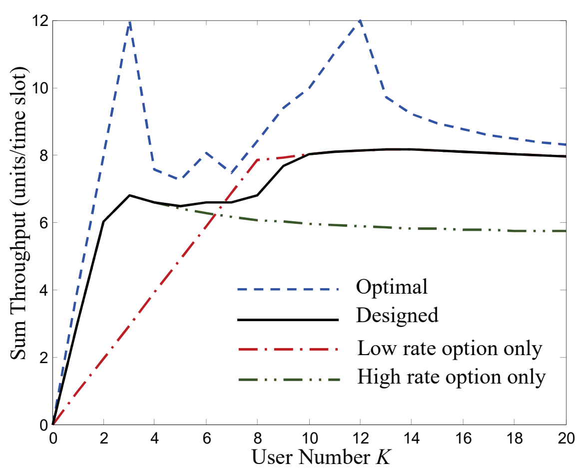

Figure 4.6: Sum throughput of the system as functions of the user number.

Example 4.6: Following Example 4.4, we assume that the system has 8 users initially. Each user has a transmission probability vector $\boldsymbol{p} = [p_h, p_l]^T$ to describe its transmission strategy, with p_h and p_l denoting the probability of transmitting with high rate option and low rate option, respectively. Transmission probability vectors of the users are initialized at $[0, 0]^T$. In each time slot, according to its own transmission probability vector, each user randomly determines whether to transmit a packet or not, and if the answer is positive, which option should be used. The receiver uses the following exponential moving average approach to measure q_v. q_v is initialized at $q_v = 1$. In each time slot, an indicator variable $I_v \in \{0, 1\}$ is used to represent the success/failure status of the virtual packet reception. q_v is then updated as $q_v = (1 - \frac{1}{300})q_v + \frac{1}{300}I_v$, and is fed back to the transmitters at the end of each time slot. Each user then adapts its transmission probability vector according to the proposed MAC algorithm with a constant step size of $\alpha = 0.05$.

We assume that the system experiences three stages. At the beginning of Stage one, the system has 8 users. The system enters Stage two at the 3001th time slot, when 6 more users enter the system with their transmission probability vectors initialized at $[0, 0]^T$. Then at the 6001th time slot, the system enters Stage three when 8 users exit the system. Convergence behavior in sum throughput of the system is illustrated in Figure 4.7, together with the corresponding optimal throughput and the theoretical throughput at the equilibrium being provided as references. In Fig-

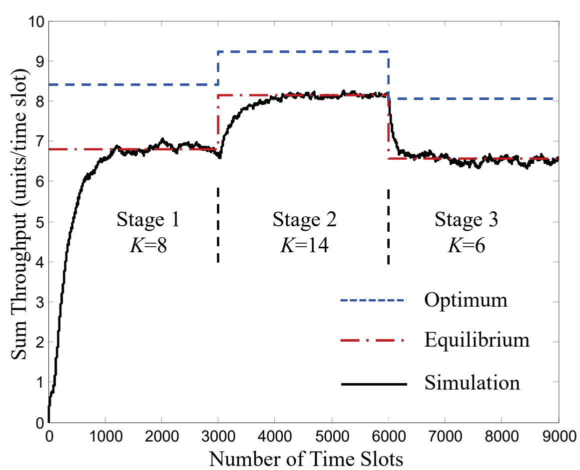

Figure 4.7: Convergence in sum throughput of the system. User number changes from 8 to 14 and then to 6 in three stages.

ure 4.8, we also illustrates entries of the transmission probability vector target calculated by the users together with the corresponding theoretical values being provided as references. Note that the simulated probability values presented in the figure are calculated using the same exponential averaging approach explained above.

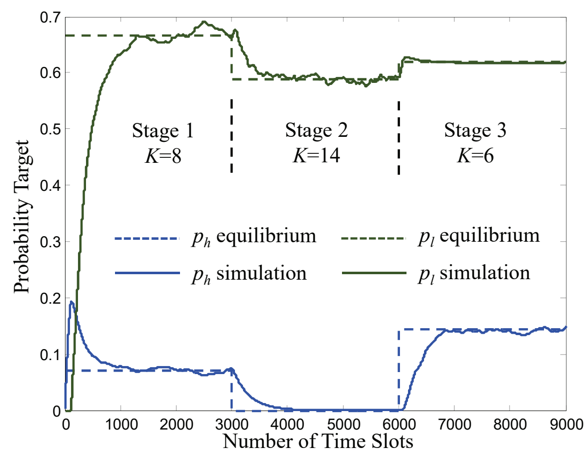

Figure 4.8: Entries of the transmission probability vector taget and their corresponding theoretical values.

Figures 4.7 and 4.8 demonstrated that, with the proposed MAC algorithm and the designed $\boldsymbol{p}(\hat{K})$, $q_v^*(\hat{K})$ functions, users have the capability to quickly adapt to the changes of stages and to adjust their transmission probability vectors to the new equilibrium.

According to the Head and Tail Condition 3, the system degrades to an equivalent single option system when $K \le \underline{K}$ or $K \ge \overline{K}$. In Example 4.4, while $\boldsymbol{d}(\underline{K}) \neq \boldsymbol{d}(\overline{K})$, we found a virtual packet design that supports $b_H = 1.01$ in the "Head" regime and $b_T = 1.01$ in the "Tail" regime. One may think that such a lucky result should not always happen for a general system. Surprisingly, according to our observations, in most of the problems of interest, even if one may not be able to get the ideal result of $b_H = b_T \approx 1$, a single virtual packet can often be designed to support close to ideal performance in both the "Head" and the "Tail" regimes.

4.5 Multiple Transmission Options, Multiple Virtual Packets

Following the system model introduced in Section 4.4, in this section, we assume that there is a set of V virtual packets being transmitted in each time slot. We present such a model extension not only because it enables extra flexibility in system design, but also because obtaining the corresponding technical results is nontrivial.

We assume that the virtual packet set being transmitted in different time slots should be identical. The link layer multiple access channel is still specified using two sets of channel parameter functions. Definition of the "real channel parameter function set" $\{C_{rij}(\boldsymbol{d})\}$ remains the same as that in Section 4.4. Given coding parameters of the virtual packets, under the assumption that all users should have the same transmission direction vector $\boldsymbol{d}$, we define $\{C_{vij}(\boldsymbol{d})\}$ as the "virtual channel parameter function set", where $C_{vij}(\boldsymbol{d})$ is the success probability of the ith virtual packet should it be transmitted in parallel with j real packets. We assume that, given any direction vector $\boldsymbol{d}$, $C_{vij}(\boldsymbol{d}) \geq C_{vi(j+1)}(\boldsymbol{d})$ should hold for all $1 \leq i \leq V$ and for all $j \geq 0$. Both the real and the virtual channel parameter function sets are assumed to be known at the transmitters and at the receiver.

Let $\boldsymbol{q}_v$ be the vector of success probabilities of the set of V virtual packets. We term $\boldsymbol{q}_v(\boldsymbol{P}, K)$ the "channel contention measure vector", which is a function of the user number K and the MK-length transmission probability vector $\boldsymbol{P}$. Because $\boldsymbol{q}_v(\boldsymbol{P}, K)$ equals the summation of a finite number of polynomial terms, it should be Lipschitz continuous in $\boldsymbol{P}$ for any finite K. When all users have the same transmission probability vector $\boldsymbol{p} = p\boldsymbol{d}$, success probability of the ith virtual packet, denoted by q_{vi} for $1 \leq i \leq V$, is also written as

$$q_{vi}(\boldsymbol{p}, K) = \sum_{j=0}^{K} \binom{K}{j} p^j (1-p)^{K-j} C_{vij}(\boldsymbol{d}). \tag{4.43}$$

Let us introduce a new design parameter $\boldsymbol{w}$, termed the "observation vector". $\boldsymbol{w}$ is a V-length vector whose entries satisfy $w_i \geq 0$ for all $1 \leq i \leq V$ and $\sum_{i=1}^{V} w_i = 1$. Upon receiving $\boldsymbol{q}_v$ from the receiver, users calculate the "channel contention measure" q_v as $q_v = \boldsymbol{w}^T \boldsymbol{q}_v$. Note that, given observation vector $\boldsymbol{w}$ and the common transmission direction vector $\boldsymbol{d}$, the system is equivalent to one with a single transmission option. Calculation of the equivalent real channel parameter set $\{C_{rj}\}$, which is dependent on the chosen system utility $U(K, p\boldsymbol{d}, \{C_{rij}(\boldsymbol{d})\})$, remains the same as that explained in Section 4.4. The equivalent virtual channel parameter set $\{C_{vj}\}$ is given by $C_{vj} = \sum_{i=1}^{V} w_i C_{vij}(\boldsymbol{d})$. Let $\epsilon_v > 0$ be a pre-determined small constant. We define $J_{\epsilon_v}(\boldsymbol{w}, \boldsymbol{d})$ as the minimum integer such that $C_{vJ_{\epsilon_v}}$ is ϵ_v larger than $C_{v(J_{\epsilon_v}+1)}$, i.e.,

$$J_{\epsilon_v}(\boldsymbol{w}, \boldsymbol{d}) = \arg\min_{j} \sum_{i=1}^{V} w_i C_{vij}(\boldsymbol{d}) > \sum_{i=1}^{V} w_i C_{vi(j+1)}(\boldsymbol{d}) + \epsilon_v. \tag{4.44}$$

We intend to design two vector parameter functions $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{p}(\hat{K})$, both functions of the user number estimate $\hat{K}$. As will be explained later, upon receiving $\boldsymbol{q}_v$, a user will use $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{q}_v$ to jointly determine a user number estimate $\hat{K}$, and then to set the transmission probability vector target at $\hat{\boldsymbol{p}} = \boldsymbol{p}(\hat{K})$. As in Section 4.4, we assume that the vector parameter functions $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{p}(\hat{K})$ to be designed should satisfy the following "Head and Tail Condition".

Condition 6. *(Head and Tail) Let $\epsilon_v > 0$ be a pre-determined constant. Let J_{ϵ_v} be defined in (4.44). There exist two integer-valued constants $0 < \underline{K} \leq \overline{K}$, such that,*

1) $\boldsymbol{d}(\hat{K}) = \boldsymbol{d}(\underline{K})$ and $\boldsymbol{w}(\hat{K}) = \boldsymbol{w}(\underline{K})$, for all $\hat{K} \leq \underline{K}$ with $\underline{K} \geq J_{\epsilon_v}(\boldsymbol{w}(\underline{K}), \boldsymbol{d}(\underline{K}))$.

2) $\boldsymbol{d}(\hat{K}) = \boldsymbol{d}(\overline{K})$ and $\boldsymbol{w}(\hat{K}) = \boldsymbol{w}(\overline{K})$ for all $\hat{K} \geq \overline{K}$ with $\overline{K} > J_{\epsilon_v}(\boldsymbol{w}(\overline{K}), \boldsymbol{d}(\overline{K}))$.

Condition 6 indicates that, when $\hat{K} \leq \underline{K}$ or $\hat{K} \geq \overline{K}$, $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{d}(\hat{K})$ should stop changing in $\hat{K}$. As explained in Section 4.4, in these two regimes, the system with multiple transmission options becomes equivalent to one with a single transmission option. We assume that core parameter functions of the MAC algorithm, i.e., the theoretical channel contention measure $q_v^*(\hat{K})$ and the probability target function $p(\hat{K})$, should be designed according to the guideline given in Section 4.2.

Let us temporarily assume that the vector parameter functions $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{p}(\hat{K})$ have been completely determined for all $\hat{K}$ values. To present the distributed MAC algorithm, we need to define the theoretical channel contention measure $q_v^*(\hat{K})$ as follows. Let $N = \lfloor \hat{K} \rfloor$ be the largest integer below $\hat{K}$. For $\hat{K} \le \underline{K}$ and $\hat{K} \ge \overline{K}$, $q_v^*(\hat{K})$ is defined by

$$q_v^*(\hat{K}) = \frac{p(\hat{K}) - p(N+1)}{p(N) - p(N+1)} \boldsymbol{w}(\hat{K})^T \boldsymbol{q}_v(\boldsymbol{p}(\hat{K}), N) + \frac{p(N) - p(\hat{K})}{p(N) - p(N+1)} \boldsymbol{w}(\hat{K})^T \boldsymbol{q}_v(\boldsymbol{p}(\hat{K}), N+1), \tag{4.45}$$

which is consistent with (4.16). For $\underline{K} \le \hat{K} \le \overline{K}$, $q_v^*(\hat{K})$ is defined by

$$q_v^*(\hat{K}) = (N + 1 - \hat{K}) \boldsymbol{w}(\hat{K})^T \boldsymbol{q}_v(\boldsymbol{p}(\hat{K}), N) + (\hat{K} - N) \boldsymbol{w}(\hat{K})^T \boldsymbol{q}_v(\boldsymbol{p}(\hat{K}), N+1). \tag{4.46}$$

Next, we present the distributed MAC algorithm below.

Distributed MAC algorithm:

1) Initialize the transmission probability vectors of all users. Let the transmission probability vector of user k be denoted by $\boldsymbol{p}_k$.

2) Let $Q \ge 1$ be a pre-determined integer. Over an interval of Q time slots, the receiver measures (or estimates) the success probabilities of all virtual packets, denoted by $\boldsymbol{q}_v$, and feeds $\boldsymbol{q}_v$ back to all transmitters.

3) Upon receiving $\boldsymbol{q}_v$, each user (transmitter) derives a user number estimate $\hat{K}$ by solving the following equation

$$q_v^*(\hat{K}) = \boldsymbol{w}(\hat{K})^T \boldsymbol{q}_v. \tag{4.47}$$

   If a $\hat{K}$ satisfying (4.47) cannot be found, then each user should set $\hat{K} = J_{\epsilon_v}(\boldsymbol{w}(\underline{K}), \boldsymbol{d}(\underline{K}))$ if $\boldsymbol{w}(\underline{K})^T \boldsymbol{q}_v > q_v^*(J_{\epsilon_v}(\boldsymbol{w}(\underline{K}), \boldsymbol{d}(\underline{K})))$, or set $\hat{K} = \infty$ otherwise.

4) Each user, say user k, then updates its transmission probability vector by

$$\boldsymbol{p}_k = (1 - \alpha) \boldsymbol{p}_k + \alpha \boldsymbol{p}(\hat{K}), \tag{4.48}$$

where $\alpha > 0$ is the step size parameter.

5) The process is repeated from Step 2 till transmission probability vectors of all users converge.

We intend to design the distributed MAC algorithm to possess a unique equilibrium at $\boldsymbol{P}^* = \mathbf{1} \otimes \boldsymbol{p}(\hat{K})$ with $\hat{K} = \max\{K, J_{\epsilon_v}(\boldsymbol{w}(\underline{K}), \boldsymbol{d}(\underline{K}))\}$. In order to ensure convergence, we require the virtual packet design, the vector parameter function $\boldsymbol{w}(\hat{K})$ and the $q_v^*(\hat{K})$ function should satisfy the following "Majorization Condition".

Condition 7. *(Majorization)*

1) *Channel contention measure vector $\boldsymbol{q}_v$ should satisfy $q_{vi} \le q_{vj}$ for all $i < j$. This condition can be met, for example, by assuming that all virtual packets should be encoded using random block codes with the same input distribution, but with rate parameters satisfying $r_{vi} \ge r_{vj}$ for all $1 \le i < j \le V$.*

2) *Observation vector function $\boldsymbol{w}(\hat{K})$ should be Lipschitz continuous in $\hat{K}$. There exists a constant $\epsilon_w > 0$, such that $\boldsymbol{w}(\hat{K})$ and $q_v^*(\hat{K})$ should satisfy the following majorization constraint*

$$\begin{aligned} &\sum_{i=j}^{V} w_i(\hat{K}_1) - \sum_{i=j}^{V} w_i(\hat{K}_2) \\ &\le (1-\epsilon_w)[q_v^*(\hat{K}_1) - q_v^*(\hat{K}_2)], \qquad \forall j \le V, \textit{ and } \forall \hat{K}_1 \le \hat{K}_2. \end{aligned} \tag{4.49}$$

Note that, because we generally require $q_v^*(\hat{K})$ be monotonically decreasing in $\hat{K}$, (4.49) can be replaced by the following stronger condition

$$\sum_{i=j}^{V} w_i(\hat{K}_1) \le \sum_{i=j}^{V} w_i(\hat{K}_2), \qquad \forall j \le V, \text{ and } \forall \hat{K}_1 \le \hat{K}_2, \tag{4.50}$$

which does not involve the evaluation of $q_v^*(\hat{K})$.

Furthermore, we also require that the vector parameter functions $\boldsymbol{p}(\hat{K})$, $\boldsymbol{w}(\hat{K})$, and the corresponding theoretical channel contention measure $q_v^*(\hat{K})$ should satisfy the following "Monotonicty and Gradient Condition" for $\underline{K} \le \hat{K} \le \overline{K}$.

Condition 8. *(Monotonicity and Gradient) For* $\underline{K} \leq \hat{K} \leq \overline{K}$*,*

1) $\boldsymbol{p}(\hat{K}) = p(\hat{K})\boldsymbol{d}(\hat{K})$ *should be Lipschitz continuous in* $\hat{K}$*, i.e., there exists a constant* $K_g > 0$ *to satisfy (4.37).*

2) $q_v^*(\hat{K})$ *should be continuous and be strictly decreasing in* $\hat{K}$*. There exists a positive constant* $\epsilon_q > 0$ *to satisfy (4.38).*

3) *There exists a constant* $\epsilon_v > 0$*, such that* $\hat{K} > J_{\epsilon_v}(\boldsymbol{w}(\hat{K}), \boldsymbol{d}(\hat{K}))$ *should be satisfied for all* $\hat{K} \in [\underline{K}, \overline{K}]$.

4) *There exist constants* $0 < \underline{p} < \overline{p} < 1$*, such that* $\underline{p} \leq p(\hat{K}) \leq \overline{p}$ *should be satisfied for all* $\hat{K} \in [\underline{K}, \overline{K}]$.

Convergence of the distributed MAC algorithm is stated in the following theorem.

Theorem 17. *Consider a multiple access system with* K *users adopting the proposed distributed MAC algorithm to update their transmission probability vectors. Under Condition 6, let* $p(\hat{K})$ *and* $q_v^*(\hat{K})$ *be designed for* $\hat{K} \leq \underline{K}$ *and* $\hat{K} \geq \overline{K}$ *according to the guideline given in Section 4.2. Let virtual packets,* $\boldsymbol{w}(\hat{K})$*,* $\boldsymbol{p}(\hat{K})$*, and* $q_v^*(\hat{K})$ *be designed to satisfy Conditions 7 and 8. Then, the associated ODE of the system given in (4.6) has a unique equilibrium at* $\boldsymbol{P}^* = \mathbf{1} \otimes \boldsymbol{p}(K)$*. The target probability vector* $\hat{\mathbf{p}}(\boldsymbol{P})$ *as a function of* $\boldsymbol{P}$ *satisfies Conditions 1 and 2. Consequently, the distributed probability vector adaptation converges to the equilibrium* $\boldsymbol{P}^*$ *in the sense specified in Theorems 9 and 10.*

The proof of Theorem 17 is given in [1, Appendix C]..

Next, we will show that, so long as one can manually design $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{p}(\hat{K})$ for a set of chosen points with integer-valued $\hat{K}$ to satisfy the following "Pinpoints Condition", then $\boldsymbol{p}(\hat{K})$ can be completed using the "Interpolation Approach" to satisfy Condition 8.

Condition 9. *(Pinpoints) Let* $\underline{K} = \hat{K}_0 < \hat{K}_1 < \cdots < \hat{K}_L = \overline{K}$ *be a set of integer-valued points. For* $i = 1, \ldots, L$*, and* $0 \le \lambda < 1$*, define*

$$
\begin{aligned}
\hat{K}_{i\lambda} &= (1-\lambda)\hat{K}_{i-1} + \lambda \hat{K}_i \\
\boldsymbol{w}_{i\lambda} &= (1-\lambda)\boldsymbol{w}(\hat{K}_{i-1}) + \lambda \boldsymbol{w}(\hat{K}_i) \\
\boldsymbol{d}_{i\lambda} &= (1-\lambda)\boldsymbol{d}(\hat{K}_{i-1}) + \lambda \boldsymbol{d}(\hat{K}_i) \\
q^*_{vi\lambda} &= (1-\lambda)q^*_v(\hat{K}_{i-1}) + \lambda q^*_v(\hat{K}_i).
\end{aligned} \tag{4.51}
$$

We have the following conditions.

1) *There exists a positive constant* $\epsilon_q > 0$*, such that, for all* $i = 1, \ldots, L$*,* $q^*_v(\hat{K}_{i-1}) - q^*_v(\hat{K}_i) \ge \epsilon_q$*.*

2) *There exists a constant* $\epsilon_v > 0$*, such that for all* $i = 1, \ldots, L$ *and* $0 \le \lambda < 1$*,* $\hat{K}_{i\lambda} > J_{\epsilon_v}(\boldsymbol{w}_{i\lambda}, \boldsymbol{d}_{i\lambda})$*, where* $J_{\epsilon_v}(\boldsymbol{w}_{i\lambda}, \boldsymbol{d}_{i\lambda})$ *is defined in (4.44).*

3) *There exist* $0 < \underline{p} < \overline{p} < 1$*, such that* $\underline{p} \le p(\hat{K}_i) \le \overline{p}$ *should be satisfied for all* $i = 1, \ldots, L$*.*

3) *Extend the definition of* $\boldsymbol{q}_v(\boldsymbol{p}, \hat{K})$ *to non-integer-valued* $\hat{K}$ *as*

$$
\boldsymbol{q}_v(\boldsymbol{p}, \hat{K}) = (\lfloor \hat{K} \rfloor + 1 - \hat{K})\boldsymbol{q}_v(\boldsymbol{p}, \lfloor \hat{K} \rfloor) + (\hat{K} - \lfloor \hat{K} \rfloor)\boldsymbol{q}_v(\boldsymbol{p}, \lfloor \hat{K} \rfloor + 1). \tag{4.52}
$$

The following inequality should be satisfied for all $i = 1, \ldots, L$ *and for all* $0 \le \lambda < 1$*.*

$$
\boldsymbol{w}^T_{i\lambda}\boldsymbol{q}_v\left(\overline{p}\boldsymbol{d}_{i\lambda}, \hat{K}_{i\lambda}\right) \le q^*_{vi\lambda} \le \boldsymbol{w}^T_{i\lambda}\boldsymbol{q}_v\left(\underline{p}\boldsymbol{d}_{i\lambda}, \hat{K}_{i\lambda}\right). \tag{4.53}
$$

Interpolation Approach: Assume that $\boldsymbol{p}(\hat{K})$ is designed for a given set of pinpoints $\{\hat{K}_i\}$, $i = 0, \ldots, L$, with $\hat{K}_0 = \underline{K} < \hat{K}_1, < \cdots < \hat{K}_L = \overline{K}$, to satisfy Conditions 7 and 9. For $i = 1, \ldots, L$ and $0 \le \lambda < 1$, let $\hat{K}_{i\lambda}$, $\boldsymbol{w}_{i\lambda}$, $\boldsymbol{d}_{i\lambda}$ and $q^*_{vi\lambda}$ be defined in (4.51). Let $\boldsymbol{q}_v(\boldsymbol{p}, \hat{K})$ be defined in (4.52). We choose $p(\hat{K}_{i\lambda})$ to satisfy

$$\boldsymbol{w}_{i\lambda}^T \boldsymbol{q}_v(p(\hat{K}_{i\lambda})\boldsymbol{d}(\hat{K}_{i\lambda}), \hat{K}_{i\lambda}) = q_{vi\lambda}^*. \tag{4.54}$$

Consequently, $\boldsymbol{p}(\hat{K}_{i\lambda})$ is designed as $\boldsymbol{p}(\hat{K}_{i\lambda}) = p(\hat{K}_{i\lambda})\boldsymbol{d}_{i\lambda}$.

Note that the existence of a solution with $\underline{p} \leq p(\hat{K}_{i\lambda}) \leq \overline{p}$ to (4.54) is guaranteed by (4.53). The following theorem shows that, combined with the Interpolation Approach, the Majorization Condition 7 and the Pinpoints Condition 9 imply the Monotonicity and Gradient Condition 8.

Theorem 18. *Assume that $\boldsymbol{w}(\hat{K})$ and $\boldsymbol{p}(\hat{K})$ are designed for a set of $L+1$ pinpoints $\{\hat{K}_i\}$, for $i = 0, \ldots, L$, with $\hat{K}_0 = \underline{K} < \hat{K}_1, < \cdots < \hat{K}_L = \overline{K}$. Let Conditions 7 and 9 be met for the pinpoints. After completing the function using the Interpolation Approach, $\boldsymbol{w}(\hat{K})$, $\boldsymbol{p}(\hat{K})$, and $q_v^*(\hat{K})$ functions satisfy both the Majorization Condition 7 and the Monotonicity and Gradient Condition 8 for $\underline{K} \leq \hat{K} \leq \overline{K}$.*

The proof of Theorem 18 is given in [1, Appendix C].

Let us consider the case when all virtual packets are encoded using random block codes with the same input distribution but with their rate parameters satisfying $r_{v1} > r_{v2} > \cdots > r_{vV}$. The Majorization Condition enables the system to shift observation weights, as $\hat{K}$ increases, either toward the low rate virtual packets (by using the simplified condition given in (4.50)) or toward the high rate virtual packets (by using (4.49)). Such flexibility can help to move the system equilibrium closer to its optimal value. Nevertheless, according to our observations, for most of the cases of interest, performance gain obtained by varying the observation vector in $\hat{K}$ is often minor compared with a carefully optimized system design either using a single virtual packet or using multiple virtual packets but with a constant observation vector.

Chapter 5

Conclusion

This thesis focuses on the cross-layer design of physical layer and data link layer to enhance the interface between these two layers while preserving the system modularity. At the physical layer, distributed coding theorems for a multiple access environment were proposed in Chapter 2. The coding theorems equipped each physical layer transmitter with an ensemble of channel codes, each corresponding to a specific communication setting. According to a data link layer protocol and when message is available, a transmitter should choose a coding option to encode its message and then send the codeword through the multiple access channel. Such a choice is not shared with other users or with the receiver. Since users are not coordinated, reliable message transmissions cannot always be supported by the channel. The receiver, on the other hand, guarantees either reliable decoding or reliable collision report depending on whether a pre-determined reliability threshold can be met. Under the assumption of infinite codeword length, the capacity of such a communication model was established, which coincides with the classical Shannon capacity region of the same channel. In the case of finite codeword length, error exponents were derived to characterize how fast the upper bound for the worst case error probability should decrease exponentially in the codeword length.

The new coding theorems provided a physical layer theoretical foundation to support an enhancement to the classical physical-link layer interface. Compared with the classical interface, which gives a link layer user binary transmission/idling options and only allows transmission probability adaptation in response to a packet transmission success/failure event, the enhanced interface essentially equipped each link layer user with multiple transmission options corresponding to the available coding options at the physical layer. Different transmission options correspond to different communication settings, such as different rates and power. Consequently, link layer users can exploit advanced wireless capabilities such as rate and power adaptation. To preserve the layered architecture, we assume link layer users can only construct their transmission schemes constraint to the provided transmission options as opposed to adapting the communication parameters arbitrarily. Link layer problems were then investigated to understand how users should adapt their transmission schemes efficiently in response to channel feedback.

In Chapter 3, the link layer problem was formulated as a non-cooperative game where each user adapts its transmission probability vector to maximize a carefully designed utility function. The condition under which the medium access control game should have a unique equilibrium was derived. Simulation results showed that from the perspective of throughput optimization, when provided with multiple transmission options, users in a multiple access system tend to use the low rate option to share the channel. This is consistent with the well known information theoretical result that parallel transmission achieves higher sum system throughput when the rates of the users are carefully chosen.

In Chapter 4, we proposed a distributed MAC framework that is capable of serving a general channel model as well as a wide range of network utilities. It has been shown that the distributed probability vector update of the users falls into the classical stochastic approximation framework with guaranteed convergence when the success probability of a virtual packet can be fed back to all users. Under the proposed MAC algorithm, rate adaptation was supported as opposed to simply adapting transmission probability in a classical system. Simulation results showed that the proposed MAC algorithm is able to lead all users' transmission probability vectors to converge to a point that is near-optimal with respect to an arbitrarily chosen network utility.

Bibliography

[1] Y. Tang, F. Heydaryan, and J. Luo. Distributed coding in a multiple access environment. *Foundations and Trends in Networking*, 12:260–412, June 2018.

[2] Y. Tang, F. Heydaryan, and J. Luo. Distributed multiple access with a general link layer channel. *Ad Hoc Networks*, 85:120–130, Mar. 2019.

[3] F. Heydaryan, Y. Tang, and J. Luo. Distributed multiple access with multiple transmission options at the link layer. Submitted to IEEE Transactions on Information Theory.

[4] J. Luo and A. Ephremides. A new approach to random access: Reliable communication and reliable collision detection. *IEEE Transactions on Information Theory*, 58:989–1002, February 2012.

[5] J. Luo. A generalized channel coding theory for distributed communication. *IEEE Transactions on Communications*, 63:1043–1056, April 2015.

[6] Z. Wang and J. Luo. Achievable error exponent of channel coding in random access communication. In *IEEE ISIT*, Austin, TX, June 2010.

[7] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Interscience, 2nd edition, 2005.

[8] R. Gallager. *Information Theory and Reliable Communication*. Wiley, 1968.

[9] I. Csiszar and J. Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, 1981.

[10] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 2nd edition, 1992.

[11] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18:535–547, March 2000.

[12] R. Gallager. A simple derivation of the coding theorem and some applications. *IEEE Transactions on Information Theory*, 11:3–18, January 1965.

[13] B. Hajek and T. Loon. Decentralized dynamic control of a multiaccess broadcast channel. *IEEE Transactions on Automatic Control*, 27:559–569, June 1982.

[14] Z. Wang and J. Luo. Error performance of channel coding in random access communication. *IEEE Transactions on Information Theory*, 58:3961–3974, June 2012.

[15] A. Feinstein. Error bounds in noisy channels without memory. *IEEE Transactions on Information Theory*, 1:13–14, September 1955.

[16] C. Shannon, R. Gallager, and E. Berlekamp. Lower bounds to error probability for coding on discrete memoryless channels. i and ii. *Information and Control*, 10:65–103, 522–552, 1967.

[17] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–269, April 1967.

[18] G. Forney. *Concatenated Codes*. MIT Press, Cambridge, MA, 1966.

[19] E. Blokh and V. Zyablov. *Linear Concatenated Codes*. Nauka, Moscow, 1982 (In Russian).

[20] Y. Polyanskiy, V. Poor, and S. Verdú. Channel coding rate in the finite blocklength regime. *IEEE Transactions on Information Theory*, 56:2307–2359, May 2010.

[21] L. Chen, S. Low, and J. Doyle. Random access game and medium access control design. *IEEE Trans. on Networking*, 18:1303–1316, Aug. 2010.

[22] Sarikaya, T. Alpean, and O. Ercetin. Dynamic pricing and queue stability in wireless random access games. *IEEE Trans. on Sel. T. in Sig. Proc.*, 6:40–150, Apr. 2012.

[23] J. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33:520–534, Jul. 1965.

[24] S. Ghez, S. Verdú, and S. Schwartz. Stability properties of slotted aloha with multipacket reception capability. *IEEE Transactions on Automatic Control*, 33:640–649, July 1988.

[25] S. Ghez, S. Verdú, and S. Schwartz. Optimal decentralized control in the random access multipacket channel. *IEEE Transactions on Automatic Control*, 34:1153–1163, November 1989.

[26] R. Gallager. Conflict resolution in random access broadcast networks. In *AFOSR Workshop Communication Theory Applications*, pages 74–76, Provincetown, MA, September 1978.

[27] J. Capetanakis. Tree algorithms for packets broadcast channel. *IEEE Transactions on Information Theory*, IT-25:505–515, September 1979.

[28] R. Garces and J. Garcia-Luna-Aceves. Collision avoidance and resolution multiple access for multichannel wireless networks. In *IEEE INFOCOM*, pages 595–602, Tel Aviv, Israel, March 2000.

[29] M. Sidi and I. Cidon. Splitting protocols in presence of capture. *IEEE Transactions on Information Theory*, 31:295–301, March 1985.

[30] X. Qin and R. Berry. Opportunistic splitting algorithms for wireless networks. In *IEEE INFOCOM*, pages 1662–1672, Hong Kong, China, March 2004.

[31] Y. Yu and G. Giannakis. High-throughput random acccess using successive interference cancellation in a tree algorithm. *IEEE Transactions on Information Theory*, 53:4628–4639, December 2007.

[32] R. Gau. Tree/stack splitting with remainder for distributed wireless medium acccess control with multipacket reception. *IEEE Transactions on Wireless Communications*, 10:3909–3923, November 2011.

[33] B. Tsybakov and V. Mikhailov. Random multiple access of packets: Part and try algorithm. *Problems of Information Transmission*, 16:65–79, October 1980.

[34] F. Kelly and I. MacPhee. The number of packets transmitted by collision detect random access schemes. *Annals of Probability*, 15:1557–1568, April 1987.

[35] G. Celik, G. Zussman, W. Khan, and E. Modiano. Mac for networks with multipacket reception capability and spatially distributed nodes. *IEEE Transactions on Mobile Computing*, 9:226–240, February 2010.

[36] B. Hajek. Stochastic approximation methods for decentralized control of multiaccess communications. *IEEE Transactions on Information Theory*, IT-31:176–184, March 1985.

[37] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, March 1951.

[38] J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23:462–466, March 1952.

[39] S. Karlin and H. Taylor. *A First Course in Stochastic Processes*. Academic, 2 edition, 1975.

[40] H. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer Verlag, 1997.

[41] V. Borkar and S. Meyn. The o.d.e method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38:447–469, January 2000.

[42] J. Metzner. On improving utilization in aloha networks. *IEEE Transactions on Communications*, COM-24:447–448, April 1976.

[43] I. Cidon, H. Kodesh, and M. Sidi. Erasure, capture and random power level section in multiaccess systems. *IEEE Transactions on Communications*, 36:263–271, March 1988.

[44] C. Lau and C. Leung. Capture models for model packet radio networks. *IEEE Transactions on Communications*, 40:917–925, May 1992.

[45] A. Ephremides and B. Hajek. Information theory and communication networks: An unconsummated union. *IEEE Transactions on Information Theory*, 44:2416–2434, October 1998.

[46] B. Tsybakov and V. Mikhailov. Ergodicity of a slotted aloha system. *Problemy Peredachi Informatsii*, 15:73–87, April 1979.

[47] R. Rao and A. Ephremides. On the stability of interacting queues in a multiple-access system. *IEEE Transactions on Information Theory*, 34:918–930, September 1988.

[48] V. Anantharam. The stability region of the finite-user slotted aloha protocol. *IEEE Transactions on Information Theory*, 37:535–540, May 1991.

[49] W. Szpankowski. Stability conditions for some distributed systems: Buffered random access systems. *Advances in Applied Probability*, 26:498–515, February 1994.

[50] W. Luo. Stability of n interacting queues in random-access systems. *IEEE Transactions on Information Theory*, 45:1579–1587, May 1999.

[51] J. Luo and A. Ephremides. On the throughput, capacity and stability regions of random multiple access. *IEEE Transactions on Information Theory*, 52:2593–2607, June 2006.

[52] F. Baccelli and B. Blaszczyszyn. Stochastic geometry and wireless networks. *Foundations and Trends in Networking*, 3:249–449, 2009.

[53] M. Haenggi. *Stochastic Geometry for Wireless Networks*. Cambridge Univ. Press, 2012.

Appendix A

Proofs of Theorems in Chapter 3

A.1 Proof of Theorem 8

According to Theorem 6, the medium access control game has at least one Nash equilibrium. We will use Theorem 7 to prove that the Nash equilibrium must be unique.

Assume that $\boldsymbol{P}^{(1)}$ and $\boldsymbol{P}^{(2)}$ are two different equilibria of the medium access control game. Let $0 \leq \theta \leq 1$. Define $\boldsymbol{P} = \theta \boldsymbol{P}^{(1)} + (1-\theta)\boldsymbol{P}^{(2)}$. Because $\boldsymbol{P}^{(1)}$ is a Nash equilibrium, the following inequality holds for all $k = 1, \cdots, K$ and $m = 1, \cdots, M$,

$$d_{km} t_{km}(q_{km}^{(1)}) - h_k \left(\log \frac{p_{km}^{(1)}}{s_{km} x_{km}^* / \tilde{K}_k} \right) \geq 0, \tag{A.1}$$

where $q_{km}^{(1)}$ is the conditional success probability corresponding to equilibrium $\boldsymbol{P}^{(1)}$. Because $d_{km} \leq 1$, we get from (A.1) that $p_{km}^{(1)} \leq \frac{s_{km} x_{km}^*}{\tilde{K}_k} e^{\frac{t_{km}^{(\max)}}{h_k}}$. Since $\boldsymbol{P}^{(2)}$ is also a Nash equilibrium and therefore has to satisfy a similar inequality, from $\boldsymbol{P} = \theta \boldsymbol{P}^{(1)} + (1-\theta)\boldsymbol{P}^{(2)}$, we get

$$p_{km} \leq \frac{s_{km} x_{km}^*}{\tilde{K}_k} e^{\frac{t_{km}^{(\max)}}{h_k}}. \tag{A.2}$$

Let $\boldsymbol{G}_{kl}(\boldsymbol{P})$ be defined in (3.2). We first obtain the following inequality according to (A.2).

$$\begin{aligned}
&\sum_{k=1}^{K} [\boldsymbol{p}_k^{(1)} - \boldsymbol{p}_k^{(2)}]^T \boldsymbol{G}_{kk}(\boldsymbol{P}) [\boldsymbol{p}_k^{(1)} - \boldsymbol{p}_k^{(2)}] = -\sum_{k=1}^{K} \sum_{m=1}^{M} (p_{km}^{(1)} - p_{km}^{(2)})^2 \frac{h_k}{x_{km}^* p_{km}} \\
&\leq -\sum_{k=1}^{K} \sum_{m=1}^{M} (p_{km}^{(1)} - p_{km}^{(2)})^2 \frac{\tilde{K}_k h_k}{s_{km} x_{km}^{*2}} e^{-\frac{t_{km}^{(\max)}}{h_k}} \\
&\leq -\sum_{k=1}^{K} \left(\sum_{m=1}^{M} |p_{km}^{(1)} - p_{km}^{(2)}| \sqrt{\frac{\tilde{K}_k h_k}{x_{km}^{*2}} e^{-\frac{t_{km}^{(\max)}}{h_k}}} \right)^2 \\
&\leq -\left(\sum_{k=1}^{K} \sum_{m=1}^{M} |p_{km}^{(1)} - p_{km}^{(2)}| \sqrt{\frac{\tilde{K}_k}{K} \frac{h_k}{x_{km}^{*2}} e^{-\frac{t_{km}^{(\max)}}{h_k}}} \right)^2
\end{aligned} \tag{A.3}$$

Next, we show that, for any $k, l = 1, \cdots, K$, $l \neq k$, and for any $m, n = 1, \cdots, M$, we have $-1 \leq \frac{\partial q_{km}}{\partial p_{ln}} \leq 0$. Define $\xi_{(k)}(\boldsymbol{P}|g_{km}, g_{ln})$ as the probability that the packet from user k is received successfully conditioned on that user k chooses transmission option g_{km} and user l chooses transmission option g_{ln}. Define $\xi_{(k)}(\boldsymbol{P}|g_{km}, g_{l0})$ as the probability that the packet from user k is received successfully conditioned on that user k chooses transmission option g_{km} and user l idles. Because idling causes no more interference than transmitting a packet, we have $\xi_{(k)}(\boldsymbol{P}|g_{km}, g_{l0}) \geq \xi_{(k)}(\boldsymbol{P}|g_{km}, g_{ln})$. Note that

$$\begin{aligned} \frac{\partial q_{km}}{\partial p_{ln}} &= \frac{\partial \left[p_{ln}\xi_{(k)}(\boldsymbol{P}|g_{km}, g_{ln})\right]}{\partial p_{ln}} + \frac{\partial \left[\left(1 - \sum_{i=1}^{M} p_{li}\right)\xi_{(k)}(\boldsymbol{P}|g_{km}, g_{l0})\right]}{\partial p_{ln}} \\ &= \xi_{(k)}(\boldsymbol{P}|g_{km}, g_{ln}) - \xi_{(k)}(\boldsymbol{P}|g_{km}, g_{l0}) \in [-1, 0]. \end{aligned} \tag{A.4}$$

From (A.4), we get

$$\begin{aligned} &\sum_{k=1}^{K}\sum_{l=1, l\neq k}^{K} [\boldsymbol{p}_k^{(1)} - \boldsymbol{p}_k^{(2)}]^T \boldsymbol{G}_{kl}(\boldsymbol{P})[\boldsymbol{p}_l^{(1)} - \boldsymbol{p}_l^{(2)}] \\ &\leq \sum_{k=1}^{K}\left[\sum_{m=1}^{M}\frac{d_{km}}{x_{km}^*}{t'}_{km}^{(\max)}|p_{km}^{(1)} - p_{km}^{(2)}|\right] \times \left[\sum_{l=1, l\neq k}^{K}\sum_{m=1}^{M}|p_{lm}^{(1)} - p_{lm}^{(2)}|\right] \\ &< \left[\sum_{k=1}^{K}\sum_{m=1}^{M}\max\left\{\frac{{t'}_{km}^{(\max)}}{x_{km}^*}, 1\right\}|p_{km}^{(1)} - p_{km}^{(2)}|\right]^2. \end{aligned} \tag{A.5}$$

Combining (A.3), (A.5) and assumption (3.10), we obtain

$$\sum_{k=1}^{K}\sum_{l=1}^{K}(\boldsymbol{p}_k^{(1)} - \boldsymbol{p}_k^{(2)})^T \boldsymbol{G}_{kl}(\boldsymbol{P})(\boldsymbol{p}_l^{(1)} - \boldsymbol{p}_l^{(2)}) < 0. \tag{A.6}$$

According to Theorem 7, the Nash equilibrium must be unique.

Appendix B

Proofs of Theorems in Chapter 4

B.1 Proof of Theorem 11

The partial derivative of $q_v(p, K)$ with respect to p is given by

$$\begin{aligned}\frac{\partial q_v(p,K)}{\partial p} &= \sum_{j=0}^{K} \binom{K}{j} j p^{j-1}(1-p)^{K-j} C_{vj} - \sum_{j=0}^{K} \binom{K}{j} p^{j}(K-j)(1-p)^{K-j-1} C_{vj} \\ &= -\sum_{j=0}^{K-1} K \binom{K-1}{j} p^{j}(1-p)^{K-1-j}(C_{vj} - C_{v(j+1)}) \\ &\leq 0, \end{aligned} \tag{B.1}$$

where the last inequality is due to the assumption that $C_{vj} \geq C_{v(j+1)}$ for all $j \geq 0$. Note that (B.1) holds with strict inequality if $K > J_{\epsilon_v}$ and $p(1-p) \neq 0$, where $J_{\epsilon_v} = \arg\min_j C_{vj} > C_{v(j+1)} + \epsilon_v$ for some $\epsilon_v > 0$.

B.2 Proof of Theorem 12

Let us first consider the situation when $\frac{x^*}{N+b} \leq p_{\max}$.

According to the definition of $q_v^*(\hat{p})$ in (4.16), we have

$$\frac{dq_v^*(\hat{p})}{d\hat{p}} = \frac{q_N(\hat{p}) - q_{N+1}(\hat{p})}{p_N - p_{N+1}} + \frac{\hat{p} - p_{N+1}}{p_N - p_{N+1}} \frac{dq_N(\hat{p})}{d\hat{p}} + \frac{p_N - \hat{p}}{p_N - p_{N+1}} \frac{dq_{N+1}(\hat{p})}{d\hat{p}}. \tag{B.2}$$

Write $\hat{K} = N + 1 - \lambda$ with $\lambda \in (0, 1]$. We have

$$\hat{p} - p_{N+1} = \frac{x^*}{\hat{K}+b} - \frac{x^*}{N+1+b} = \frac{\lambda}{N+1+b}\hat{p}, \tag{B.3}$$

and

$$p_N - \hat{p} = \frac{x^*}{N+b} - \frac{x^*}{\hat{K}+b} = \frac{1-\lambda}{N+b}\hat{p}. \tag{B.4}$$

Meanwhile, because function $q_{N+1}(\hat{p})$ can be decomposed as

$$
\begin{aligned}
q_{N+1}(\hat{p}) &= \sum_{j=0}^{N+1} \binom{N+1}{j} \hat{p}^j (1-\hat{p})^{N+1-j} C_{vj} \\
&= \hat{p} \sum_{j=0}^{N} \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j} C_{v(j+1)} + (1-\hat{p}) \sum_{j=0}^{N} \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j} C_{vj},
\end{aligned} \tag{B.5}
$$

we have

$$
q_N - q_{N+1} = \hat{p} \sum_{j=0}^{N} \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j} (C_{vj} - C_{v(j+1)}). \tag{B.6}
$$

Furthermore, the derivatives of $q_N(\hat{p})$ and $q_{N+1}(\hat{p})$ are given by

$$
\frac{dq_N(\hat{p})}{d\hat{p}} = -\sum_{j=0}^{N} (N-j) \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j-1} (C_{vj} - C_{v(j+1)}), \tag{B.7}
$$

and

$$
\frac{dq_{N+1}(\hat{p})}{d\hat{p}} = -\sum_{j=0}^{N} (N+1) \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j} (C_{vj} - C_{v(j+1)}). \tag{B.8}
$$

Substituting the above results into (B.2), we get

$$
\begin{aligned}
(p_N - p_{N+1})\frac{dq_v^*(\hat{p})}{\hat{p}} &= \hat{p}\sum_{j=0}^{N}\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j}(C_{vj}-C_{v(j+1)}) \\
&-\frac{\lambda}{N+1+b}\hat{p}\sum_{j=0}^{N}\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j-1}(N-j)(C_{vj}-C_{v(j+1)}) \\
&-\frac{1-\lambda}{N+b}\hat{p}\sum_{j=0}^{N}(N+1)\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j}(C_{vj}-C_{v(j+1)}) \\
&= \hat{p}\sum_{j=0}^{N}\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j-1}(C_{vj}-C_{v(j+1)}) \\
&\times\left(1-\hat{p}-\frac{\lambda(N-j)}{N+1+b}-\frac{(1-\lambda)(1-\hat{p})(N+1)}{N+b}\right) \\
&= \hat{p}\sum_{j=0}^{N}\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j-1}(C_{vj}-C_{v(j+1)}) \\
&\times\left(\frac{\lambda((1-\hat{p})(N+1+b)-N+j)}{N+1+b}+\frac{(1-\lambda)(1-\hat{p})(b-1)}{N+b}\right).
\end{aligned}
\tag{B.9}
$$

Note that, for all $j \geq 0$, we have

$$
\begin{aligned}
\frac{\lambda((1-\hat{p})(N+1+b)-N+j)}{N+1+b} &\geq \frac{\lambda((1-p_N)(N+1+b)-N+j)}{N+1+b} \\
&\geq \frac{\lambda(b-x^*+j)}{N+1+b}.
\end{aligned}
\tag{B.10}
$$

Therefore, $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq 0$ if $b \geq 1$ and the following inequality is satisfied.

$$
\sum_{j=0}^{N}\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j-1}(C_{vj}-C_{v(j+1)})(b-x^*+j) \geq 0. \tag{B.11}
$$

It is easy to see that (B.11) holds if $b \geq x^* - \gamma_{\epsilon_v}$, with γ_{ϵ_v} being defined in (4.18).

Furthermore, if we have both $b > 1$ and $b > x^* - J_{\epsilon_v}$ hold with strict inequality, and $C_{vj} > C_{v(j+1)}$ for at least one $j \leq N$, then $\frac{dq_v^*(\hat{p})}{d\hat{p}} > 0$ should also hold with strict inequality for $\hat{p} \in (0, p_{\max})$.

Next, consider the case when $\frac{x^*}{N+b} \geq p_{\max}$. It is easy to see that $\frac{dq_v^*(\hat{p})}{d\hat{p}} = 0$ if $\frac{x^*}{\hat{K}+b} \geq p_{\max}$. If $\frac{x^*}{\hat{K}+b} < p_{\max}$ but $\frac{x^*}{N+b} \geq p_{\max}$ on the other hand, we can write $\hat{K} = N + 1 - \lambda$ with $0 < \lambda \leq N + 1 + b - \frac{x^*}{p_{\max}}$. Consequently, (B.2) and (B.3) still hold. But (B.4) should be changed to

$$p_N - \hat{p} = p_{\max} - \frac{x^*}{\hat{K}+b} \leq \frac{1-\lambda}{N+b}\hat{p}. \tag{B.12}$$

As a result, (B.9) becomes

$$(p_N - p_{N+1})\frac{dq_v^*(\hat{p}}{\hat{p}} \geq \hat{p}\sum_{j=0}^{N}\binom{N}{j}\hat{p}^j(1-\hat{p})^{N-j-1}(C_{vj} - C_{v(j+1)}) \times \left(\frac{\lambda((1-\hat{p})(N+1+b) - N + j)}{N+1+b} + \frac{(1-\lambda)(1-\hat{p})(b-1)}{N+b}\right). \tag{B.13}$$

By following the rest of the derivations, it can be seen that conclusion of the theorem still holds.

B.3 Proof of Theorem 13

First, because $b > \max\{1, x^* - J_{\epsilon_v}\}$ holds with strict inequality, the theoretical channel contention measure $q_v^*(\hat{p})$ is strictly increasing in $\hat{p}$ for $\hat{p} \in (0, p_{\max})$. Given user number K, $q_v(\hat{p}, K)$ is non-increasing in $\hat{p}$. Therefore, if $K \geq J_{\epsilon_v}$, then $\hat{p} = p^* = \frac{x^*}{K+b}$ is the only solution to $q_v(\hat{p}, K) = q_v^*(\hat{p})$. When $K < J_{\epsilon_v}$ on the other hand, we have $q_v(\hat{p}, K) > q_v^*(\hat{p})$ for all $\hat{p} \in [0, p_{\max})$. This implies that $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$ is the only equilibrium of the system.

Second, we show that there exists a constant $\epsilon > 0$, such that $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq \epsilon > 0$ for all $\hat{p} < p_{\max}$. Note that $\hat{p} < p_{\max}$ implies $\hat{K} > J_{\epsilon_v}$. From (B.9) and (B.10), we get

$$\frac{dq_v^*(\hat{p})}{\hat{p}} \geq \frac{\hat{p}}{p_N - p_{N+1}}\binom{N}{J_{\epsilon_v}}\hat{p}^{J_{\epsilon_v}}(1-\hat{p})^{N-J_{\epsilon_v}-1}(C_{vJ_{\epsilon_v}} - C_{v(J_{\epsilon_v}+1)}) \times \left(\frac{\lambda(b - x^* + J_{\epsilon_v})}{N+1+b} + \frac{(1-\lambda)(1-\hat{p})(b-1)}{N+b}\right). \tag{B.14}$$

Because the right hand side of (B.14) has a positive limit when $\hat{p} \to 0$, we can find two small positive constants $\epsilon_0, \epsilon_1 > 0$, such that $\frac{dq_v^*(\hat{p})}{\hat{p}} \geq \epsilon_0$ for all $\hat{p} \leq \epsilon_1$. On the other hand, when

$\epsilon_1 \leq \hat{p} < p_{\max}$, because $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$ holds with strict inequality, we can find a small positive constant $\epsilon_2 > 0$, such that the right hand side of (B.14) is no less than ϵ_2. Therefore, by choosing $\epsilon = \min\{\epsilon_0, \epsilon_2\}$, we have

$$\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq \epsilon > 0, \qquad \text{for all } \hat{p} < p_{\max}. \tag{B.15}$$

Third, let $q_v^{*-1}(.)$ be the inverse function of $q_v^*(p)$. For any given transmission probability vector $\boldsymbol{p}$, transmission probability target $\hat{p}$ is obtained by

$$\hat{p} = q_v^{*-1}(q_v) = q_v^{*-1}(q_v(\boldsymbol{p}, K)). \tag{B.16}$$

Because $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq \epsilon > 0$, we can find a constant $K_{l1} > 0$ such that

$$|\hat{p}_1 - \hat{p}_2| \leq K_{l1}|q_{v1} - q_{v2}|, \tag{B.17}$$

for all $\hat{p}_1 = q_v^{*-1}(q_{v1})$ and $\hat{p}_2 = q_v^{*-1}(q_{v2})$. In the meantime, since $q_v = q_v(\boldsymbol{p}, K)$ is Lipschitz continuous in $\boldsymbol{p}$ for any given K, there must exist a constant $K_{l2} > 0$ to satisfy

$$|q_{v1} - q_{v2}| \leq K_{l2}\|\boldsymbol{p}_1 - \boldsymbol{p}_2\|, \tag{B.18}$$

for all $q_{v1} = q_v(\boldsymbol{p}_1, K)$ and $q_{v2} = q_v(\boldsymbol{p}_2, K)$. Consequently, by combining (B.17) and (B.18), we have

$$|\hat{p}_1 - \hat{p}_2| \leq K_{l1}K_{l2}\|\boldsymbol{p}_1 - \boldsymbol{p}_2\|, \tag{B.19}$$

for all $\hat{p}_1 = q_v^{*-1}(q_v(\boldsymbol{p}_1, K))$ and $\hat{p}_2 = q_v^{*-1}(q_v(\boldsymbol{p}_2, K))$. This implies that the probability target function given in (B.16) satisfies the Lipschitz Condition 2.

Finally, when the system is noisy, the receiver can choose to measure q_v over an extended number of time slots, namely increasing the value of Q introduced in Step 2 of the proposed MAC algorithm. If users maintain their transmission probabilities during the Q times slots, it is often the case that the potential measurement bias in the system can be reduced arbitrarily close to zero with a large enough Q. Therefore, the Mean and Bias Condition 1 is also satisfied.

Consequently, convergence of the distributed probability adaptation is supported by Theorems 9 and 10.

B.4 Proof of Theorem 14

According to the two-step approach, q_v is interpreted by $q_v = (1 - p_k)q_k + p_k d^*(\breve{p})$. When $\breve{p} \geq p_k$ and $\breve{p}$ is the solution to $q_k = q^*(\breve{p})$, we have

$$\begin{aligned} q_v &= (1 - p_k)q^*(\breve{p}) + p_k d^*(\breve{p}) = q^*(\breve{p}) - p_k(q^*(\breve{p}) - d^*(\breve{p})) \\ &\geq q^*(\breve{p}) - \breve{p}(q^*(\breve{p}) - d^*(\breve{p})) = q_v^*(\breve{p}), \end{aligned} \tag{B.20}$$

where the inequality is due to the fact that $q^*(\breve{p}) - d^*(\breve{p}) \geq 0$ for all $j \geq 0$. By the monotonicity of $q_v^*(.)$, when $\breve{p} \geq p_k$, we have

$$q_v^*(\hat{p}) = q_v \geq q_v^*(\breve{p}) \geq q_v^*(p_k). \tag{B.21}$$

This implies that we must have $\hat{p} \geq p_k$ when $\breve{p} \geq p_k$. Similarly, when $\breve{p} \leq p_k$ and $\breve{p}$ is the solution to $q_k = q^*(\breve{p})$, the two-step approach will yield $\hat{p} \leq p_k$.

In the case when $q_k < q^*(0)$, we have $\breve{p} = 0$. Hence the interpreted q_v satisfies

$$q_v = (1 - p_k)q_k + p_k d^*(0) < (1 - p_k)q^*(0) + p_k d^*(0) \leq q^*(0) = q_v^*(0). \tag{B.22}$$

This implies that we have $\hat{p} = 0$ when $\breve{p} = 0$. Similarly, we have $\hat{p} = p_{\max}$ when $\breve{p} = p_{\max}$.