Distributed Multiple Access with A General Link Layer Channel

Yanru Tang, Faeze Heydaryan, and Jie Luo

Electrical & Computer Engineering Department
Colorado State University, Fort Collins, CO 80523
Email: {yrtang, faezeh66, rockey}@colostate.edu

Abstract

This paper investigates the problem of distributed medium access control in a time slotted wireless multiple access network with an unknown finite number of homogeneous users. Assume that each user has a single transmission option. In each time slot, a user chooses either to idle or to transmit a packet. Under a general channel model, a distributed medium access control framework is proposed to adapt transmission probabilities of all users to a value that maximizes an arbitrarily chosen symmetric network utility. Probability target of each user in the proposed algorithm is calculated based upon a channel contention measure, which is defined as the success probability of a virtual packet. It is shown that the proposed algorithm falls into the classical stochastic approximation framework with guaranteed convergence when the contention measure can be directly obtained from the receiver. On the other hand, when the contention measure is not directly available, computer simulations show that a revised medium access control algorithm can still help the system to converge to the same designed equilibrium.

Keywords: wireless networking, medium access control, distributed systems

1. Introduction

In distributed communication when users access the channel opportunistically, packet collisions are often unavoidable. When communication optimization cannot be achieved fully at the physical layer, data link layer must get involved in communication adaptation. To support efficient data link layer adaptation, great efforts have been made to investigate the problem of distributed medium access control (MAC). Distributed MAC protocols can generally be categorized into non-adaptive ALOHA protocols [1], splitting algorithms [2][3] and back-off approaches [4][5]. ALOHA protocols are often used to investigate fundamental limits, such as the throughput and the stability regions, of a network [6][7]. In splitting algorithms such as the FCFS algorithm [2], users maintain a common virtual interval of their random identity values. Users partition the interval based upon a sequence of channel feedback messages and determine their transmission schedules accordingly. While splitting algorithms can often achieve a relatively high system throughput, their function depends on the assumptions of instant availability of noiseless channel feedback and correct reception of feedback sequences. Unfortunately, both of these conditions can be violated in a wireless environment. Theoretical analysis of a splitting algorithm can be extremely challenging, especially when wireless-related factors such as channel noise, feedback error, and transmission delay are taken into account. Back-off algorithms [8][4][5], on the other hand, have proven to enjoy more trackable analysis. In back-off algorithms such as the 802.11 DCF protocol, according to packet availability, each user transmits its packets with an associated probability parameter. A user should decrease its trans-

mission probability in response to a packet collision (or 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 stochastic approximation framework [8][4], with rigorously developed mathematical and statistical tools available for its convergence and performance analysis. It is well known that convergence proof of these algorithms often hold in the existence of measurement noise and feedback delay [9]. Practical back-off algorithms can also be analyzed using Markov models to characterize the impact of discrete probability updates [5].

In [4], a stochastic approximation model was proposed for distributed networking over a collision channel with an unknown finite number of users, each having a saturated message queue. By setting the transmission probability target of each user as a function of the channel idling probability, it was shown that the system can be designed to converge to a unique stable equilibrium. In the case of throughput maximization with homogeneous users, it was proposed that idling probability of the channel should be controlled toward the asymptotically optimal value of $1/e$. This is similar to the proposal of controlling the total traffic level toward 1, as discussed in [8] using a stochastic approximation framework for a system with an infinite number of users. Most of the existing analysis of the splitting and the back-off algorithms either assume a throughput optimization objective and/or a simple collision channel model. While significant research efforts have been made to revise collision resolution algorithms to incorporate wireless-related physical layer properties, such as capture effect [10] and multi-packet reception [11], not much progress has been reported since the 1980s on integrating these extensions with the insightful stochastic approximation-based frameworks, such as those introduced in [8][4].

Recently, an extended channel coding theory was developed in [12][13][14] for physical layer distributed communication that features opportunistic channel access and packet collision. The new coding theory enabled the derivation of fundamental limits of distributed communication systems. It also supported the derivation of a link layer channel model based upon the physical layer channel and coding parameters of the data packets. This motivated the investigation on the impact of a general link layer channel model to the design and optimization of collision resolution algorithms.

In this paper, we consider the problem of distributed utility optimization in a wireless multiple access network with an unknown finite number of homogeneous users. Assume that each user is backlogged with a saturated message queue. The link layer multiple access channel is generally modeled by two sets of channel parameters, detailed in the paper. Given the channel model, we propose a distributed MAC framework for each user to adapt its transmission probability according to a channel contention measure defined as the success probability of a virtual packet. We show that the proposed MAC algorithm falls into the classical stochastic approximation framework with guaranteed convergence, if the underlying ordinary differential equation (ODE) has a unique equilibrium and two key monotonicity conditions are satisfied. Without knowing the number of users, we show that one can develop the MAC algorithm to satisfy the required conditions and to place the unique equilibrium at a point that is not far from optimal with respect to a chosen utility. Our work extends the basic framework of [4] from a simple collision channel model to a general link-layer channel. Such extension is enabled by the following key ideas. First, as opposed to measuring contention level of the channel using a locally observable variable such as the channel idling probability [4], we measure channel contention level using the success probability of a carefully designed virtual packet. Coding parameters of the virtual packet affect the optimality of the MAC algorithm through a set of channel parameters that may need to be derived using the distributed channel coding theorems presented in [12][13][14]. Second, with the help of the channel contention measure and two key monotonicity properties, we show that each user can first estimate the unknown number of users, and then set its transmission probability target as a function of the estimated number of users. Compared with the approach of maintaining channel contention at a more or less fixed level for all values of number of users, as suggested in [4], the MAC algorithm to

2

be proposed can help the system to achieve a performance closer to optimal especially when the number of users is not large in value.

The rest of the paper is organized as follows. In Section 2, we present a stochastic approximation framework for a class of distributed MAC algorithms. While the framework and its convergence results are quite standard in the stochastic approximation literature, they characterize the key conditions for guaranteed convergence to a unique system equilibrium. Within the framework, the research objectives become to design the system to satisfy the convergence conditions, and to place the unique equilibrium at a point that maximizes a chosen network utility. We propose to measure contention level of the channel using the success probability of a carefully designed virtual packet, and require that users should derive a common transmission probability target as a function of the common channel contention measure. Such an approach guarantees that transmission probabilities of all users at any system equilibrium must be identical. In Section 3, under the assumption that channel contention measure can be directly fed back by the receiver to the transmitters, we propose a distributed MAC algorithm to adapt the transmission probabilities of the users to lead the actual channel contention level toward a predetermined theoretical value. Convergence of the proposed MAC algorithm is proven with the help of two key monotonicity properties. In Section 4, we consider the more practical case when each user only knows the conditional success probability of its own packets. A two step approach is proposed for each user to interpret the channel contention measure and to adapt its transmission probability accordingly. Simulation results are provided in Section 5 to demonstrate both the optimality and the convergence properties of the proposed MAC algorithms under various system settings.

To help reading the technical contents of the paper, we summarize the definitions of a list of key variables below.

Definitions of Key Variables

$\{C_{rj}\}$: real channel parameter set. C_{rj} is the conditional success probability of a real packet should it be transmitted in parallel with j other real packets.

$\{C_{vj}\}$: virtual channel parameter set. C_{vj} is the success probability of the virtual packet should it be transmitted in parallel with j real packets.

J_{ϵ_v} : $\arg\min_j C_{vj} > C_{v(j+1)} + \epsilon_v$.

K : actual number of users.

$\hat{K}$: estimated number of users.

$\boldsymbol{p}$: transmission probability vector of all users.

$\tilde{\boldsymbol{p}}$: transmission probability vector target computed using noisy measurements.

$\hat{\boldsymbol{p}}$: theoretical transmission probability vector target computed using noiseless measurements.

$\boldsymbol{p}^*$: transmission probability vector at an equilibrium.

$p_{\max}$: upper bound to the transmission probability of a user.

q_v : actual channel contention measure.

q_v^* : theoretical channel contention measure.

x^* : the limit of Kp as $K \to \infty$. The value of x^* is obtained from the optimization of the asymptotic utility function.

2. Problem Formulation

Consider a wireless multiple access network with K homogeneous users (transmitters) and a common receiver. The value of K is known neither to the users nor to the receiver. Time is slotted such that each slot equals the length of one packet. Assume that each user has a saturated message queue. In each time slot, each individual user, say user k, makes its transmission/idling decision according to an associated transmission probability parameter, denoted by p_k. Transmission decision of a user is not shared with other users or with the receiver. We use transmission probability vector $\boldsymbol{p} = [p_1, \cdots, p_K]^T$ to denote the transmission probabilities of all users.

Let us add an integer-valued time index t and use $\boldsymbol{p}(t) = [p_1(t), \cdots, p_K(t)]^T$ to denote the transmission probability vector of the users at the beginning of time slot t. Assume that users intend to maximize a symmetric network utility, which we will discuss later. In each time slot, according to channel feedback obtained from the receiver, we assume that user k should derive a transmission probability target

3

$\tilde{p}_k(t)$. User k then updates its transmission probability by

$$p_k(t+1) = (1 - \alpha(t))p_k(t) + \alpha(t)\tilde{p}_k(t), \quad (1)$$

where $\alpha(t) \geq 0$ is the step size parameter of time slot t. Let $\tilde{\boldsymbol{p}}(t) = [\tilde{p}_1(t), \cdots, \tilde{p}_K(t)]^T$ be the vector of transmission probability targets of all users. Transmission probability vector $\boldsymbol{p}(t)$ is updated by

$$\boldsymbol{p}(t+1) = \boldsymbol{p}(t) + \alpha(t)(\tilde{\boldsymbol{p}}(t) - \boldsymbol{p}(t)). \quad (2)$$

Probability adaptation given in (2) falls into the stochastic approximation framework [9][15][16], and $\tilde{\boldsymbol{p}}(t)$ is often calculated based upon noisy estimates of certain system variables.

Let $\hat{\boldsymbol{p}}(t) = [\hat{p}_1(t), \cdots, \hat{p}_K(t)]^T$ be the vector of the theoretical transmission probability targets of all users, which is computed based upon presumed noiseless measurements and noiseless feedback in time slot t. Let $E_t[\tilde{\boldsymbol{p}}(t)]$ denote the conditional expectation of $\tilde{\boldsymbol{p}}(t)$ given system state at the beginning of time slot t. Let us write $E_t[\tilde{\boldsymbol{p}}(t)]$ as

$$E_t[\tilde{\boldsymbol{p}}(t)] = \hat{\boldsymbol{p}}(t) + \boldsymbol{g}(t) = \hat{\boldsymbol{p}}(\boldsymbol{p}(t)) + \boldsymbol{g}(\boldsymbol{p}(t)), \quad (3)$$

where $\boldsymbol{g}(t)$ is defined as the bias term in the probability vector target derivation. Given the communication channel, both $\hat{\boldsymbol{p}}(t)$ and $\boldsymbol{g}(t)$ are functions of $\boldsymbol{p}(t)$, which denotes the transmission probability vector in time slot t.

Next, we present two conditions that are typically required for the convergence of a stochastic approximation algorithm.

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

$$\|\boldsymbol{g}(\boldsymbol{p}(t))\| \leq K_m \beta(t). \quad (4)$$

We assume that $\beta(t)$ is controllable in the sense that, for any $\epsilon > 0$, one can design protocols to ensure $\beta(t) \leq \epsilon$ for a sufficiently large t.

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

$$\|\hat{\boldsymbol{p}}(\boldsymbol{p}_1) - \hat{\boldsymbol{p}}(\boldsymbol{p}_2)\| \leq K_l \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|, \quad \text{for all } \boldsymbol{p}_1, \boldsymbol{p}_2. \quad (5)$$

According to classical stochastic approximation theory [9][15][16], if Conditions 1 and 2 are met, and $\alpha(t)$, $\beta(t)$ are small enough, trajectory of probability vector $\boldsymbol{p}(t)$ under distributed adaptation given in (2) can be approximated by the following associated ODE

$$\frac{d\boldsymbol{p}(t)}{dt} = -\left[\boldsymbol{p}(t) - \hat{\boldsymbol{p}}(t)\right], \quad (6)$$

where, with an abuse of the 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 must satisfy $\boldsymbol{p}^* = \hat{\boldsymbol{p}}(\boldsymbol{p}^*)$.

Convergence of the distributed probability adaptation is stated in the following two theorems, which are quite standard for stochastic approximation algorithms.

Theorem 1. *Let Conditions 1 and 2 hold. Assume that the associated ODE given in (6) has a unique stable equilibrium at $\boldsymbol{p}^*$. If $\alpha(t)$ and $\beta(t)$ satisfy the following conditions*

$$\sum_{t=0}^{\infty} \alpha(t) = \infty, \sum_{t=0}^{\infty} \alpha(t)^2 < \infty, \sum_{t=0}^{\infty} \alpha(t)\beta(t) < \infty, \quad (7)$$

then under distributed probability adaptation given in (2), $\boldsymbol{p}(t)$ converges to $\boldsymbol{p}^*$ with probability one.

Theorem 1 is implied by [9, Theorems 4.3].

Theorem 2. *Let Conditions 1 and 2 hold. Assume that the associated ODE given in (6) has a unique stable equilibrium at $\boldsymbol{p}^*$. Under distributed probability adaptation given in (2), 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*

$$\exists T_0 \geq 0, \underline{\alpha} \leq \alpha(t) \leq \overline{\alpha}, \beta(t) \leq \sqrt{\overline{\alpha}}, \forall t \geq T_0, \quad (8)$$

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

$$\limsup_{t \to \infty} Pr\{\|\boldsymbol{p}(t) - \boldsymbol{p}^*\| \geq \epsilon\} < K_w \overline{\alpha}. \quad (9)$$

Theorem 2 can be obtained by following the proof of [16, Theorems 2.3] with minor revisions.

4

Note that, in the above discussion, we assumed the same $\alpha(t)$ and $\beta(t)$ for all users. We also assumed that feedback information should be obtained by all users in every time slot, and probability adaptations of all users should be synchronous. However, by following the literature of stochastic approximation theory [9], it is easy to show that these assumptions can be relaxed. So long as step size sequences and bounding sequences of all users satisfy the same constraints given in (7) and (8), and all users receive channel feedback frequently enough, then conclusions of Theorems 1 and 2 should remain valid.

With convergence of the system guaranteed by Theorems 1 and 2, key objectives of the system design are to develop the distributed MAC algorithm to satisfy Conditions 1 and 2, and to place the unique system equilibrium at a desired point that maximizes the chosen utility. Because users are homogeneous, due to symmetry, if a system equilibrium $\boldsymbol{p}^*$ is unique, it must take the form of $\boldsymbol{p}^* = p^*\mathbf{1}$, with $\mathbf{1}$ being the vector of all ones. That is, transmission probabilities of all users at the equilibrium must be identical. We choose to enforce such a property by requiring that all users should obtain the same transmission probability target in each time slot t, i.e., $\hat{\boldsymbol{p}}(t) = \hat{p}(t)\mathbf{1}$. The corresponding part of the system design is explained below.

Assume that there is a virtual packet being transmitted in each time slot. Virtual packets of different time slots are identical. A virtual packet is an assumed packet with coding parameters known both to the users and to the receiver, but it is not physically transmitted in the system, i.e., the packet is "virtual". We assume that, without knowing the transmission/idling status of the users, the receiver can detect in each time slot whether the virtual packet transmission should be regarded as successful or not. For example, suppose that the link layer channel is a collision channel, and a virtual packet has the same coding parameters of a real packet. Then, virtual packet reception in a time slot should be regarded as successful if and only if no real packet is transmitted. 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 packet are encoded using random block codes, given the physical layer channel, reception of each virtual packet corresponds to a detection task that judges whether or not the vector transmission status of all real users should belong to a specific region. Such detection tasks and their performance bounds have been extensively discussed in the distributed channel coding theory [12][13][14].

Let $q_v(t)$ denote the success probability of the virtual packet in time slot t. We assume that the receiver should estimate $q_v(t)$ and feed it back to all transmitters. We term $q_v(t)$ the "channel contention measure" because it is designed to serve as a measurement of the contention level of the link-layer multiple access channel. Note that, in the collision channel case when $q_v(t)$ equals the channel idling probability, feeding back $q_v(t)$ may not be necessary. So long as each user k knows the conditional success probability of its own packet, denoted by $q_k(t)$, idling probability of the channel can be estimated by $(1 - p_k(t))q_k(t)$. With a general link layer channel, however, estimating $q_v(t)$ may not always be possible if it is not directly fed back from the receiver. Upon receiving the estimate of $q_v(t)$, each user calculates its probability target as the same function of the $q_v(t)$ estimate. Denote the theoretical transmission probability target by $\hat{p}(q_v(t))$. The theoretical vector transmission probability target is given by $\hat{\boldsymbol{p}}(t) = \hat{p}(q_v(t))\mathbf{1}$. Because any equilibrium $\boldsymbol{p}^*$ of the ODE must satisfy $\boldsymbol{p}^* = \hat{\boldsymbol{p}}(\boldsymbol{p}^*)$, this guarantees that $\boldsymbol{p}^*$ must take the form of $\boldsymbol{p}^* = p^*\mathbf{1}$ with $p^* = \hat{p}(p^*)$, where $\hat{p}(p^*)$ is the theoretical transmission probability target computed under the assumption that all users should transmit with an identical probability of p^*.

In a practical system, the measurement of $q_v(t)$ is likely to be corrupted by noise. We assume that, if users keep their transmission probability vector $\boldsymbol{p}$ at a constant, and q_v is measured over an interval of Q time slots, then the measurement should converge to its true value with probability one as Q is taken to infinity. Other than this assumption, measurement noise is not involved in the discussion of the design objectives of meeting Conditions 1 and 2 and placing the unique system equilibrium at the desired point. Therefore, in the following section, we assume that $q_v(t)$ can be measured precisely and be fed back to the users. This leads to $\tilde{\boldsymbol{p}}(t) = \hat{\boldsymbol{p}}(t) = \hat{p}(t)\mathbf{1}$. We will

5

also skip the time index t to simplify the notations.

3. Distributed MAC With Receiver Feeding Back The Channel Contention Measure

In this section, we assume that success probability of the virtual packet can be measured at the receiver and be fed back to the transmitters. With a general link layer channel model, we will show that a distributed MAC algorithm can be developed to lead the transmission probabilities of all users to the same value that maximizes a chosen symmetric network utility.

We first introduce two sets of parameters to model the link layer channel. Define $\{C_{rj}\}$ for $j \geq 0$ as the "real channel parameter set". C_{rj} is the conditional success probability of a real packet should it be transmitted in parallel with j other real packets. Define $\{C_{vj}\}$ for $j \geq 0$ as the "virtual channel parameter set". C_{vj} is the success probability of the virtual packet should it be transmitted in parallel with j real packets. We assume that $C_{vj} \geq C_{v(j+1)} \geq 0$ should hold for all $j \geq 0$. This implies that an increased number of parallel real packet transmissions should not improve the chance of a virtual packet getting through the channel. Let $\epsilon_v \geq 0$ 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.,

$$J_{\epsilon_v} = \arg\min_j C_{vj} > C_{v(j+1)} + \epsilon_v. \qquad (10)$$

Because $\{C_{rj}\}$ and $\{C_{vj}\}$ can be theoretically derived from the physical layer channel and coding parameters of the real and the virtual packets, we assume that they should be known to the users and also to the receiver. Note that, while C_{rj} does not depend on the coding parameters of the virtual packet, coding design of the virtual packet does affect the value of $\{C_{vj}\}$.

We assume that the users intend to maximize a symmetric utility. Under the assumption that all users should transmit with the same probability, utility of the system is denoted by $U(K, p, \{C_{rj}\})$, which is defined as a function of the unknown number of users K, the common transmission probability p of

all users, and the real channel parameter set $\{C_{rj}\}$. For example, if users intend to maximize the sum throughput of the system, then $U(K, p, \{C_{rj}\})$ should be given by

$$\begin{aligned} &U(K, p, \{C_{rj}\}) \\ &= K \sum_{j=0}^{K-1} \binom{K-1}{j} p^{j+1}(1-p)^{K-1-j} C_{rj}. \end{aligned}$$
$$(11)$$

For many utility functions of interest, such as the sum throughput function given above, an asymptotically optimal solution should maintain the expected load of the channel at a constant [4][11]. Write $p = \frac{x}{K}$. We define x^* using the following asymptotic utility optimization

$$x^* = \arg\max_x \lim_{K \to \infty} U\left(K, \frac{x}{K}, \{C_{rj}\}\right). \qquad (12)$$

The calculation of x^* is only involved with the utility function and the real channel parameter set $\{C_{rj}\}$, and is irrelevant to the coding parameters of the virtual packet. We generally regard $p = \min\left\{1, \frac{x^*}{K}\right\}$ as an ideal solution to the utility optimization problem for all values of number of users. Note that, this is indeed the optimum solution for all K for sum throughput maximization over a collision channel [4][11].

Let $b \geq 1$ be a pre-determined design parameter whose value will be introduced later. Define $p_{\max}$ as

$$p_{\max} = \min\left\{1, \frac{x^*}{J_{\epsilon_v} + b}\right\}. \qquad (13)$$

We will show next that, without knowing the actual number of users K, it is possible to set the unique system equilibrium at $\boldsymbol{p}^* = p^*\mathbf{1} = \min\{p_{\max}, \frac{x^*}{K+b}\}\mathbf{1}$, which is not far from the assumed ideal solution of $\min\{1, \frac{x^*}{K}\}\mathbf{1}$.

We intend to design a distributed MAC algorithm to set the unique system equilibrium at $p^*\mathbf{1}$ by maintaining channel contention at an appropriate level. Note that, given the virtual channel parameter set $\{C_{vj}\}$, channel contention measure $q_v(\boldsymbol{p}, K)$ is a function of the unknown number of users K and the transmission probability vector $\boldsymbol{p}$. 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. If all users transmit with the same probability p, i.e., $\boldsymbol{p} = p\boldsymbol{1}$, $q_v(p\boldsymbol{1}, K)$ is given by

$$q_v(p\boldsymbol{1}, K) = \sum_{j=0}^{K} \binom{K}{j} p^j (1-p)^{K-j} C_{vj}, \qquad (14)$$

We assume that, upon receiving q_v from the receiver, each user should first obtain an estimated number of users, denoted by $\hat{K}$, and then set the corresponding transmission probability target at $\tilde{p} = \hat{p} = \min\{p_{\max}, \frac{x^*}{\hat{K}+b}\}$, where $x^* > 0$ is obtained from (12). We will show that, for any $x^* > 0$, one can always find an appropriate b and design a distributed MAC algorithm to ensure system convergence to the designed equilibrium of $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\boldsymbol{1}$. Note that, while the actual number of users K is always an integer, we do allow the estimated number of users $\hat{K}$ to take a non-integer value.

Convergence of the MAC algorithm to be proposed depends on two key monotonicity properties presented below. First, the following theorem shows that, given the number of users K, $q_v(p\boldsymbol{1}, K)$ is non-increasing in p.

Theorem 3. With $C_{vj} \geq C_{v(j+1)}$ for all $j \geq 0$, $q_v(p\boldsymbol{1}, K)$ given in (14) satisfies $\frac{\partial q_v(p\boldsymbol{1}, K)}{\partial p} \leq 0$. Furthermore, $\frac{\partial q_v(p\boldsymbol{1}, K)}{\partial p} < 0$ holds with strict inequality for $K > J_{\epsilon_v}$ and $p \in (0, 1)$.

The proof of Theorem 3 is presented in Appendix A.

Next, we define the "theoretical channel contention measure", denoted by q_v^*, which represents the expected channel contention level at the system equilibrium if the estimated number of users is correct. Let $\hat{p} = \frac{x^*}{\hat{K}+b}$, and $N = \lfloor \hat{K} \rfloor$ be the largest integer below $\hat{K}$. We define theoretical channel contention measure as a continuous function $q_v^*(\hat{p})$, which can also be viewed as a function of $\hat{K}$, as follows

$$q_v^*(\hat{p}) = \frac{\hat{p} - p_{N+1}}{p_N - p_{N+1}} q_N(\hat{p}) + \frac{p_N - \hat{p}}{p_N - p_{N+1}} q_{N+1}(\hat{p}), \qquad (15)$$

where

$$p_N = \min\left\{p_{\max}, \frac{x^*}{N+b}\right\}$$
$$p_{N+1} = \min\left\{p_{\max}, \frac{x^*}{N+1+b}\right\}, \qquad (16)$$

and

$$q_N(p) = \sum_{j=0}^{N} \binom{N}{j} p^j (1-p)^{N-j} C_j$$
$$q_{N+1}(p) = \sum_{j=0}^{N+1} \binom{N+1}{j} p^j (1-p)^{N+1-j} C_j. \qquad (17)$$

If the number of users in the system indeed equals $K = \hat{K}$ with $\hat{K} \geq x^* - b$, then $q_v^*(\hat{p})$ defined in (15) equals the actual channel contention measure $q_v(\boldsymbol{p}^*, K)$ at the desired equilibrium $\boldsymbol{p}^* = \frac{x^*}{K+b}\boldsymbol{1} = \frac{x^*}{\hat{K}+b}\boldsymbol{1}$, i.e., when all users transmit with the same probability of $\hat{p} = \frac{x^*}{\hat{K}+b}$.

We intend to design the theoretical channel contention measure as a decreasing function in the estimated number of users $\hat{K}$. In other words, an increased number of users should lead to a more crowded channel. Equivalently, when being viewed as a function of $\hat{p}$, $q_v^*(\hat{p})$ is desired to be increasing in $\hat{p}$. Indeed, given an arbitrary $x^* > 0$, such a monotonicity property can be guaranteed with an appropriate choice of b.

Theorem 4. Let $x^* > 0$. If $b \geq \max\{1, x^* - \gamma_{\epsilon_v}\}$, with γ_{ϵ_v} being given by

$$\gamma_{\epsilon_v} = \min_{N, N \geq J_{\epsilon_v}, N \geq x^* - b}$$
$$\frac{\sum_{j=0}^{N} j \binom{N}{j} \left(\frac{p_{N+1}}{1-p_{N+1}}\right)^j (C_{vj} - C_{v(j+1)})}{\sum_{j=0}^{N} \binom{N}{j} \left(\frac{p_{N+1}}{1-p_{N+1}}\right)^j (C_{vj} - C_{v(j+1)})}, \qquad (18)$$

then $q_v^*(\hat{p})$ defined in (15) 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 4 is presented in Appendix B. We want to point out that, if ϵ_v is small enough to satisfy $C_{vj} = C_{v(j+1)}$ for all $j < J_{\epsilon_v}$, then we have $\gamma_{\epsilon_v} = J_{\epsilon_v}$. Otherwise, $\gamma_{\epsilon_v} \leq J_{\epsilon_v}$ is generally true.

We are now ready to propose the distributed MAC algorithm.

Distributed MAC algorithm:

1. Initialize the transmission probabilities of all users. Let the transmission probability of user k be denoted by p_k.
2. Over an interval of Q time slots, with $Q \geq 1$, 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

$$q_v^*(\hat{p}) = q_v. \tag{19}$$

   If a $\hat{p} \in [0, p_{\max}]$ satisfying (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. User k then updates its transmission probability by

$$p_k = (1 - \alpha)p_k + \alpha\hat{p}, \tag{20}$$

   where α is the step size parameter for user k.
5. The process is repeated from Step 2 till transmission probabilities of all users converge.

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

Theorem 5. *Given $x^* > 0$, let b be chosen to satisfy $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$. With the proposed MAC algorithm, the system has a unique equilibrium at $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\boldsymbol{1}$. Furthermore, given the number of users K, the probability target $\hat{p}(\boldsymbol{p})$ as a function of transmission probability vector $\boldsymbol{p}$ satisfies Conditions 1 and 2. Consequently, transmission probability vector $\boldsymbol{p}$ converges to $\boldsymbol{p}^*$ in the sense explained in Theorems 1 and 2.*

The proof of Theorem 5 is presented in Appendix C.

The above analysis indicates that, with an arbitrary virtual packet design and with the proposed MAC algorithm, the system should converge to the designed equilibrium so long as b is chosen to satisfy $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$. However, one should note that optimality of the algorithm does depend on the value of b, γ_{ϵ_v}, and J_{ϵ_v}, which are determined by the virtual channel parameter set $\{C_{vj}\}$, and therefore are dependent on the coding parameters of the virtual packet. For example, it is known that, to maximize the sum throughput of a distributed multiple access system over a collision channel, the optimal solution is to set the transmission probability of all users at $p = \frac{1}{K}$ with K being the number of users. This corresponds to $x^* = 1$ and $b = 0$ in our model. For a general system, assume that setting the transmission probabilities of all users at $p = \min\{1, \frac{x^*}{K}\}$ should be an ideal choice for maximizing the chosen utility. Because the proposed MAC algorithm sets the equilibrium at $\boldsymbol{p}^* = \min\{p_{\max}, \frac{x^*}{K+b}\}\boldsymbol{1}$, there are two optimality concerns. On one hand, when the number of users K is large, it is a general preference that one should design the virtual packet to allow a relatively small value of b. This implies that the values of γ_{ϵ_v} and J_{ϵ_v} should not be much smaller than x^*. On the other hand, when the actual number of users K is small, one should also try to get $p_{\max}$ close to 1. This implies that J_{ϵ_v} should also not be much larger than x^*. Combining both optimality concerns, a general guideline is to design coding parameters of the virtual packet such that J_{ϵ_v} and γ_{ϵ_v} should be slightly smaller than x^* and b should be close to 1.

4. Distributed MAC with Interpreted Channel Contention Measure

Classical MAC protocols often assume that a user should get feedback from the receiver on whether its own packets are successfully received or not [2]. 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. To simplify the discussion, we also assume that a virtual packet should have the same communication parameters as those of a real packet. In order

8

to apply the MAC algorithm proposed in Section 3, user k will need 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, under the assumption that a virtual packet should have the same coding parameters of a real packet, q_k equals the conditional success probability of the virtual packet given that user k idles. Consequently, user k can calculate the success probability of the virtual packet by

$$q_v = (1 - p_k)q_k + p_k d_k, \qquad (21)$$

where p_k is the transmission probability of user k, and d_k is the conditional success probability of the virtual packet given that user k transmits a packet[1]. 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 always be available at the transmitters unless additional feedback information is provided. When d_k is not available, we propose a two-step approach for each user to interpret d_k and hence the success probability of the virtual packet q_v, and then to update 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 estimated number of users $\breve{K}$, let $\breve{N} = \lfloor \breve{K} \rfloor$ denote the largest integer below $\breve{K}$. Let $\breve{p} = \min\{p_{\max}, \frac{x^*}{\breve{K}+b}\}$, $p_{\breve{N}} = \min\{p_{\max}, \frac{x^*}{\breve{N}+b}\}$ and $p_{\breve{N}+1} = \min\{p_{\max}, \frac{x^*}{\breve{N}+1+b}\}$, 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

$$q^*(\breve{p}) = \frac{\breve{p} - p_{\breve{N}+1}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}-1} \binom{\breve{N}-1}{j}$$
$$\times \breve{p}^j (1 - \breve{p})^{\breve{N}-1-j} C_{vj}$$

---

¹Extensions 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 (21).

$$+ \frac{p_{\breve{N}} - \breve{p}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}} \binom{\breve{N}}{j} \breve{p}^j (1 - \breve{p})^{\breve{N}-j} C_{vj},$$

$$d^*(\breve{p}) = \frac{\breve{p} - p_{\breve{N}+1}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}-1} \binom{\breve{N}-1}{j}$$
$$\times \breve{p}^j (1 - \breve{p})^{\breve{N}-1-j} C_{v(j+1)}$$
$$+ \frac{p_{\breve{N}} - \breve{p}}{p_{\breve{N}} - p_{\breve{N}+1}} \sum_{j=0}^{\breve{N}} \binom{\breve{N}}{j} \breve{p}^j (1 - \breve{p})^{\breve{N}-j} C_{v(j+1)}.$$
$$(22)$$

In the case when $\breve{K}$ takes an integer value, $q^*(\breve{p})$ is the conditional success probability of the virtual packet under the assumptions that the system has $\breve{K}$ users, user k idles, and all other users have the same transmission probability of $\breve{p}$. Similarly, $d^*(\breve{p})$ represents the conditional success probability of the virtual packet under the assumptions that the system has $\breve{K}$ users, user k transmits a packet, and all other users have the same transmission probability of $\breve{p}$.

Next, we present the two-step approach that is suggested for each user to obtain its transmission probability target.

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

$$q^*(\breve{p}) = q_k. \qquad (23)$$

If a $\breve{p} \in [0, p_{\max}]$ satisfying (19) 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

$$q_v = (1 - p_k)q_k + p_k d^*(\breve{p}). \qquad (24)$$

An updated transmission probability target $\hat{p}$ for user k is then determined by solving equation (19). As before, if a $\hat{p} \in [0, p_{\max}]$ satisfying (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 when $\hat{p}$ is obtained by the two step approach, a convergence proof of the MAC algorithm

is no longer available. This is because the two step approach does not guarantee that transmission probability targets obtained by different users should be identical. Therefore, the assumption that any equilibrium $\boldsymbol{p}^*$ must take the form of $\boldsymbol{p}^* = p^*\boldsymbol{1}$ 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 $\check{p}$ obtained in (23) as its transmission probability target.

Theorem 6. *Let $x^* > 0$, and $b \geq \max\{1, x^* - \gamma_{\epsilon_v}\}$, where γ_{ϵ_v} is defined in (18). Suppose that each user, say user k, first obtains an intermediate transmission probability $\check{p}$ and then determines its transmission probability target $\hat{p}$ by following the two-step approach. Then $\check{p} \geq p_k$ implies $\hat{p} \geq p_k$, while $\check{p} \leq p_k$ implies $\hat{p} \leq p_k$.*

The proof of Theorem 6 is presented in Appendix D.

Theorem 6 suggests that each user can simplify the two step approach into Step 1 only and simply set the transmission probability target at $\hat{p} = \check{p}$. In cases when the two-step approach does lead the system to the designed equilibrium, the simplified one step approach should also lead the system to the same equilibrium.

5. Simulation Results

In this section, we use computer simulations to illustrate both the optimality and the convergence properties of the proposed MAC algorithms.

Example 1: (Optimality) The distributed MAC framework proposed in this paper shares certain proximity with the one proposed in [4], although [4] only considered the simple collision channel model. In the first example, we investigate the classical problem of symmetric sum throughput maximization over a collision channel.

Assume that the system has K users each having a saturated message queue. If K is known, the optimal solution that maximizes the sum throughput is to set the transmission probabilities of all users at $p_{\mathrm{opt}} = \frac{1}{K}$ [4]. It was suggested in [4] that, when K is unknown,

with the help of a proposed distributed MAC framework, each user should direct its transmission probability to converge to p_a, which is obtained by solving the following equation.

$$eP(\mathrm{idle}) - 1 - 0.5\sqrt{p_a} = 0, \quad P(\mathrm{idle}) = (1 - p_a)^K, \tag{25}$$

where $P(\mathrm{idle})$ is the idling probability of the channel that can be measured locally if each user knows the conditional success probability of its own packets.

Now consider the distributed MAC algorithm proposed in Section 3 of this paper. With the collision channel model, the real channel parameter set $\{C_{rj}\}$ is given by $C_{r0} = 1$ and $C_{rj} = 0$ for all $j > 0$. Because the utility function is chosen as the sum network throughput, we obtain from (12) that $x^* = 1$. Let us 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 $j \geq 0$. Choose $\epsilon_v = 0.01$, we get $\gamma_{\epsilon_v} = J_{\epsilon_v} = 0$. This supports the choice of $b = 1.01 > x^* - \gamma_{\epsilon_v}$. The unique equilibrium of the system is therefore set at $\boldsymbol{p}^* = p^*\boldsymbol{1} = \frac{1}{K+1.01}\boldsymbol{1}$.
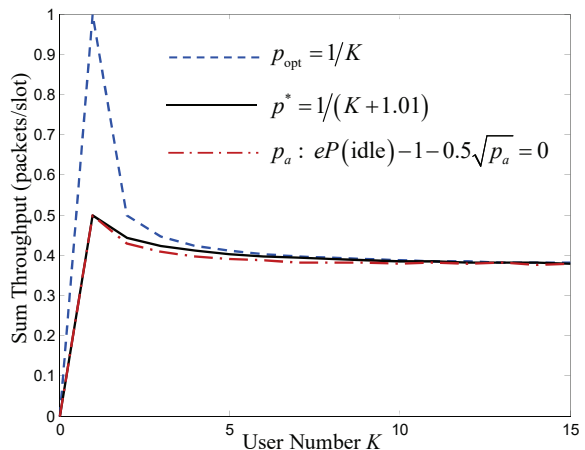

Figure 1: Sum throughput as functions of the number of users for distributed multiple access over a collision channel.

In Figure 1, we illustrate the achieved sum throughput in packets/slot as functions of the number of users under various system settings. The

dashed curve represents the optimum utility when transmission probabilities of all users are set at p_{opt}. Note that the optimum utility is not necessarily achievable because it requires the knowledge of the number of users K. The solid curve represents the achieved utility at the designed equilibrium of the proposed MAC algorithm, i.e., when transmission probabilities of all users are set at p^*. The dash-dotted curve is the achieved utility when transmission probabilities of all users are set at p_a, as suggested in [4]. It can be seen that, at the designed equilibrium, the proposed distributed MAC algorithm can achieve a throughput performance slightly higher than the approach suggested in [4].

Example 2: (Optimality) In this example, we consider a distributed multiple access network with K users and a simple fading channel. In each time slot, with a probability of 0.3, the channel can support no more than $M_1 = 4$ parallel real packet transmissions, and with a probability of 0.7, the channel can support no more than $M_2 = 6$ parallel real packet transmissions. Note that such a channel can appear if there is an interfering user that transmits a packet with probability 0.3 in each time slot. One packet from the interfering user is equivalent to the combination of two packets from a regular user. The real channel parameter set $\{C_{rj}\}$ in this case is given by $C_{rj} = 1$ 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 throughput weighted by a transmission energy cost of $E = 0.3$. With the number of users being K and all users transmitting with the same probability p, system utility $U(K, p, \{C_{rj}\})$ is given by

$$U(K, p, \{C_{rj}\}) = -EKp +$$
$$\sum_{j=0}^{K-1} K \binom{K-1}{j} p^{j+1}(1-p)^{K-1-j} C_{rj}. \quad (26)$$

Correspondingly, x^* can be obtained from (12) as $x^* = 3.29$. Assume that a virtual packet should have the same coding parameters as those of a real packet. The virtual channel parameter set $\{C_{vj}\}$ is therefore 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 $\gamma_{\epsilon_v} = J_{\epsilon_v} = 3$. Therefore, we can set $b = 1.01 > x^* - \gamma_{\epsilon_v}$.

In Figure 2, we illustrate three utilities all as functions of the number of users K. The solid curve represents the utility achieved by the proposed MAC algorithm at the designed equilibrium. The dashed curve represents the optimum utility under the assumption that number of users K is known, and this is not necessarily achievable without the knowledge of K. The dash-dotted curve represents the utility if we maintain the channel idling probability at its asymptotically optimal value of $\exp(-x^*)$, as suggested in [4][2]. This is equivalent to setting the transmission probabilities of all users at $1 - \exp\left(-\frac{x^*}{K}\right)$. It can be seen that, the proposed MAC algorithm can achieve a higher utility value compared with the approach suggested in [4]. Achieved utility of the proposed MAC algorithm is also reasonably close to optimal when the number of users K is not close to M.
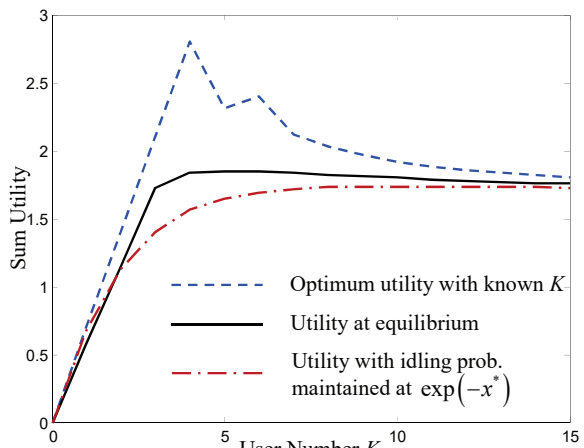

Figure 2: Sum utility as functions of the number of users for distributed multiple access over a simple fading channel.

Example 3: (Convergence with receiver feeding back the channel contention measure) Following Example 2, we have $x^* = 3.29$ and $b = 1.01$. Assume that the system has $K = 8$ users. We initialize the

---

[2]While [4] also suggested to maintain other variables at their asymptotically optimal values, these alternative approaches do not lead to a better performance in this example.

11

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 4 or 6 packets. Each user also randomly determines whether a packet should be transmitted according to its own transmission probability parameter. Whether the real packets and the virtual packet can go through the channel or not is then determined using the corresponding channel model. We use the following exponential moving average approach to measure q_v. q_v is initialized at $q_v = 1$. In each time slot, q_v is updated by $q_v = (1 - \frac{1}{300})q_v + \frac{1}{300}I_v$, where $I_v \in \{0, 1\}$ is an indicator of the success/failure status of the virtual packet in the current time slot, i.e., $I_v = 1$ indicates that the transmission of the virtual packet in this time slot should be regarded as successful and $I_v = 0$ otherwise. While this is different from the approach proposed in the distributed MAC algorithm, simulations show that an exponential averaging measurement of q_v can often lead the system to converge in a relatively smaller number of time slots. We assume that q_v is measured at the receiver and is then fed back in each time slot to all transmitters. The rest of probability adaptation proceeds according to the distributed MAC algorithm introduced in Section 3 with a constant step size of $\alpha = 0.05$.

Convergence behavior of the system utility is illustrated in Figure 3, where system utility is also measured using the same exponential moving average approach except that initial value of the utility is set at 0. The dash-dotted line represents the system utility if each user transmits with the desired probability $p^* = \frac{x^*}{K+b} = 0.365$. The dashed line represents the optimal system utility obtained by $\max_{0 \le p \le 1} U(K, p, \{C_{rj}\})$, where $U(K, p, \{C_{rj}\})$ is given in (26). In this case, system utility at the designed equilibrium is about 90% of the optimal value. In about 1000 interations, transmission probabilities of all users already become close to the equilibrium value.

Example 4: (Convergence with interpreted channel contention measure) In this example, we study the convergence property of the distributed MAC algorithm proposed in Section 4 when each user only knows the success/failure status of its own packets.
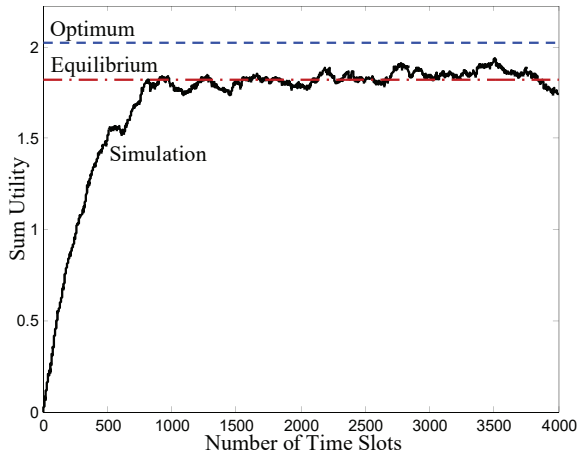

Figure 3: Convergence in system utility of a multiple access network with $K = 8$ users over a simple fading channel. Channel contention measure is fed back by the receiver.

Following Example 3, we assume that each user, say user k, should maintain a measurement of the conditional success probability of its own packets, denoted by q_k. q_k is initialized at $q_k = 1$. In each time slot, if user k transmits a packet, then 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 the packet transmitted by user k in the current time slot. We assume that the value of I_k should be fed back to user k from the receiver. If user k idles, on the other hand, the value of q_k should remain unchanged. With the measurement of q_k, user k then uses the simplified one step approach to derive its transmission probability target $\hat{p} = \check{p}$ by solving equation (23). Then, user k updates its transmission probability by $p_k = (1 - \alpha)p_k + \alpha\hat{p}$ with a constant step size of $\alpha = 0.05$.

In Figure 4, we illustrate the convergence behavior in sum utility of the system. As before, system utility is measured using the same exponential moving average approach with an initial value of 0. It can be seen that, when each user uses the one step approach to calculate its transmission probability target, the system can still converge to the same designed equilibrium.
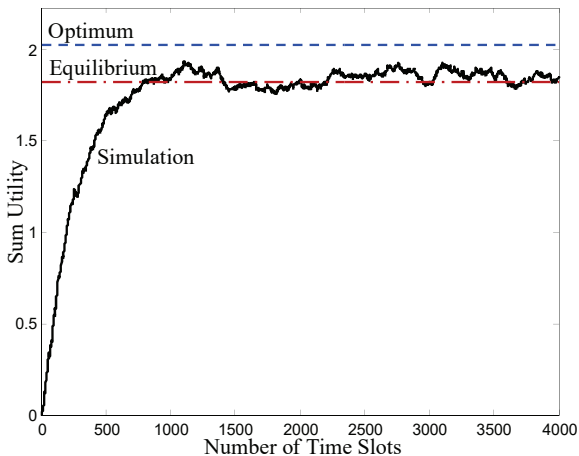
12

Figure 4: Convergence in system utility of a multiple access network with $K = 8$ users over a fading channel. Each user only knows the success/failure status of its own packets.

Example 5: (Convergence in a dynamic environment) In this example, we start with the system introduced in Example 4. We still assume that each user only knows the success/failure status of its own packets, and uses the simplified one step approach to calculate its transmission probability target. The system contains $K = 8$ users at the beginning. We say that the system starts with Stage 1. At the 3001th time slot, we assume that the system enters Stage 2 when 7 other users join the network. This leads to a total of $K = 15$ users. Each of the new users has its transmission probability initialized at zero and its packet conditional success probability initialized at one. Then at the 6001th time slot, we assume that the system enters Stage 3 when 5 users exit the network.

In Figure 5, we illustrate convergence behavior of the system in average transmission probability of the active users over the three stages. The corresponding optimal transmission probability (i.e., transmission probability that maximizes the symmetric utility) and the theoretical transmission probability at the designed equilibria are also illustrated in dashed lines and dash-dotted lines, respectively. While a theoretical convergence proof is not available in this case,

we can see that in a dynamic environment when users join/exit the system, the proposed MAC algorithm has a reasonably good capability to help active users tracking the designed equilibrium.
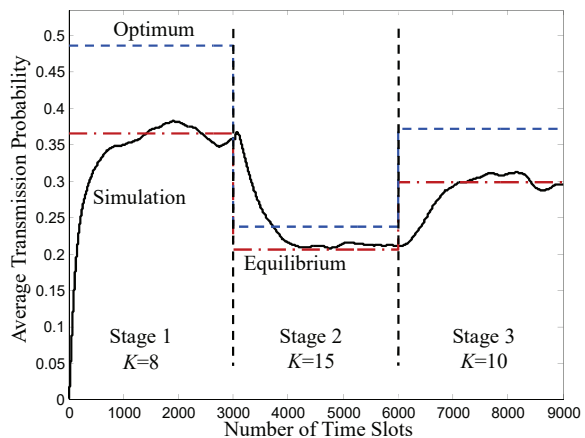

Figure 5: Convergence in average transmission probability of the active users of a multiple access network over three stages.

6. Conclusion

We investigated multiple access networking with an unknown finite number of homogeneous users. A distributed MAC algorithm is proposed to maximize an arbitrarily chosen symmetric network utility with a generally modeled link layer channel. We proposed to measure contention level of the channel using the success probability of a carefully designed virtual packet, and to adapt transmission probabilities of all users toward a direction that matches the actual channel contention measure with its theoretical value. Under the assumption that channel contention measure can be fed back by the receiver, we proved the convergence of the proposed MAC algorithm with the help of two key monotonicity properties. We also proposed a revised MAC algorithm for the case when each user only knows the success/failure status of its own packets. While a convergence proof of the revised MAC algorithm is not available, simulation results suggest that the revised MAC algorithm can indeed lead the system to the same designed equilibrium.

13

Appendix A. Proof of Theorem 3

Partial derivative of $q_v(p\mathbf{1}, K)$ with respect to p is given by

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

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

$$= - \sum_{j=0}^{K-1} K \binom{K-1}{j} p^j (1-p)^{K-1-j}$$

$$\times (C_{vj} - C_{v(j+1)})$$

$$\leq 0, \tag{A.1}$$

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

Appendix B. Proof of Theorem 4

Let us first consider the situation when $\frac{x^*}{N+b} \leq p_{\max}$. According to the definition of $q_v^*(\hat{p})$ given in (15), we have

$$\frac{dq_v^*(\hat{p})}{d\hat{p}} = \frac{q_N(\hat{p}) - q_{N+1}(\hat{p})}{p_N - p_{N+1}}$$

$$+ \frac{\hat{p} - p_{N+1}}{p_N - p_{N+1}} \frac{dq_N(\hat{p})}{d\hat{p}} + \frac{p_N - \hat{p}}{p_N - p_{N+1}} \frac{dq_{N+1}(\hat{p})}{d\hat{p}}. \tag{B.1}$$

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

$$\hat{p} - p_{N+1} = \frac{x^*}{\hat{K}+b} - \frac{x^*}{N+1+b} = \frac{\lambda}{N+1+b}\hat{p}, \tag{B.2}$$

and

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

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

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

$$= \hat{p} \sum_{j=0}^{N} \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j} C_{v(j+1)}$$

$$+ (1-\hat{p}) \sum_{j=0}^{N} \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j} C_{vj}, \tag{B.4}$$

we have

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

Furthermore, by taking derivatives of $q_N(\hat{p})$ and $q_{N+1}(\hat{p})$ with respect to $\hat{p}$, we get,

$$\frac{dq_N(\hat{p})}{d\hat{p}} = \sum_{j=0}^{N} (N-j) \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j-1}$$

$$\times (C_{v(j+1)} - C_{vj}), \tag{B.6}$$

and

$$\frac{dq_{N+1}(\hat{p})}{d\hat{p}} = \sum_{j=0}^{N} (N+1) \binom{N}{j} \hat{p}^j (1-\hat{p})^{N-j}$$

$$\times (C_{v(j+1)} - C_{vj}). \tag{B.7}$$

Substituting the above results into (B.1) yields

$$(p_N - p_{N+1}) \frac{dq_v^*(\hat{p})}{d\hat{p}}$$

$$= \sum_{j=0}^{N} \binom{N}{j} \hat{p}^{j+1} (1-\hat{p})^{N-j} \left(C_{vj} - C_{v(j+1)}\right)$$

$$- \frac{\lambda}{N+1+b} \sum_{j=0}^{N} (N-j) \binom{N}{j}$$

$$\times \hat{p}^{j+1} (1-\hat{p})^{N-j-1} \left(C_{vj} - C_{v(j+1)}\right)$$

$$- \frac{1-\lambda}{N+b} \sum_{j=0}^{N} (N+1) \binom{N}{j}$$

$$\times \hat{p}^{j+1} (1-\hat{p})^{N-j} \left(C_{vj} - C_{v(j+1)}\right)$$

$$= \sum_{j=0}^{N} \binom{N}{j} \hat{p}^{j+1} (1-\hat{p})^{N-j-1} (C_{vj} - C_{v(j+1)})$$

$$\times \left(\frac{\lambda((1-\hat{p})(N+1+b) - N + j)}{N+1+b}\right.$$

14

$$+\frac{(1-\lambda)(1-\hat{p})(b-1)}{N+b}\Bigg).\qquad\text{(B.8)}$$

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

$$\frac{\lambda((1-\hat{p})(N+1+b)-N+j)}{N+1+b} \geq \frac{\lambda(b-x^*+j)}{N+1+b}.\qquad\text{(B.9)}$$

Therefore, $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq 0$ if $b \geq 1$ and

$$\sum_{j=0}^{N}\binom{N}{j}\left(\frac{\hat{p}}{1-\hat{p}}\right)^j (C_{vj}-C_{v(j+1)})(b-x^*+j) \geq 0.\qquad\text{(B.10)}$$

(B.10) holds if $b \geq x^* - \gamma_{\epsilon_v}$ with γ_{ϵ_v} being defined in (18).

Furthermore, if we have both $b > 1$ and $b > x^* - \gamma_{\epsilon_v}$ holding with strict inequalities, and $N \geq J_{\epsilon_v}$, then $\frac{dq_v^*(\hat{p})}{d\hat{p}} > 0$ should also hold with strict inequality for $\hat{p} \in (0, p_{max})$.

Now consider the situation when $\frac{x^*}{N+b} \geq p_{max}$. It is easy to see that, when $\frac{x^*}{\hat{K}+b} \geq p_{max}$, we have $\frac{dq_v^*(\hat{p})}{d\hat{p}} = 0$. When $\frac{x^*}{\hat{K}+b} < p_{max}$ 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.1) and (B.2) still hold, but (B.3) should be replaced by

$$p_N - \hat{p} = p_{max} - \frac{x^*}{\hat{K}+b} \leq \frac{1-\lambda}{N+b}\hat{p}.\qquad\text{(B.11)}$$

Therefore, (B.8) becomes

$$(p_N - p_{N+1})\frac{dq_v^*(\hat{p})}{d\hat{p}}$$
$$\geq \sum_{j=0}^{N}\binom{N}{j}\hat{p}^{j+1}(1-\hat{p})^{N-j-1}\left(C_{vj}-C_{v(j+1)}\right)$$
$$\times\left(\frac{\lambda((1-\hat{p})(N+1+b)-N+j)}{N+1+b}\right.$$
$$\left.+\frac{(1-\lambda)(1-\hat{p})(b-1)}{N+b}\right).\qquad\text{(B.12)}$$

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

Appendix C. Proof of Theorem 5

First, according to Theorem 4, if b is chosen to satisfy $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$, $\frac{dq_v^*(\hat{p})}{d\hat{p}} > 0$ holds with strict inequality for $\hat{p} \in (0, p_{max})$. According to Theorem 3, $q_v(p\mathbf{1}, K)$ is non-increasing in $\hat{p}$ for any given number of users K. Therefore, if $K \geq J_{\epsilon_v}$, $q_v^*(\hat{p}) = q_v(\hat{p}\mathbf{1}, K)$ should have a unique solution at $\hat{p} = \frac{x^*}{K+b}$. If $K < J_{\epsilon_v}$ on the other hand, we must have $q_v(\hat{p}\mathbf{1}, K) > q_v^*(\hat{p})$ for all $\hat{p} \in [0, p_{max})$. Consequently, the proposed MAC algorithm should possess a unique equilibrium at $\boldsymbol{p}^* = p^*\mathbf{1} = \min\{p_{max}, \frac{x^*}{K+b}\}\mathbf{1}$.

Second, consider an arbitrary $\hat{p} < p_{max}$, which implies that the corresponding $\hat{K}$ should satisfy $\hat{K} > J_{\epsilon_v}$. According to (B.8) and (B.9), we have

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

It can be seen that the right hand side of (C.1) has a positive limit as $\hat{p}$ is taken to zero. Therefore, we can find two positive constants $\epsilon_0, \epsilon_1 > 0$, such that $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq \epsilon_0 > 0$ for all $\hat{p} \leq \epsilon_1$. In the meantime, when $\epsilon_1 \leq \hat{p} < p_{max}$, because $b > \max\{1, x^* - \gamma_{\epsilon_v}\}$, we can find another positive constant $\epsilon_2 > 0$, such that the right hand side of (C.1) is larger than or equal to ϵ_2. Consequently, there exists a positive constant $\epsilon = \min\{\epsilon_0, \epsilon_2\}$, such that $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq \epsilon > 0$ for all $\hat{p} < p_{max}$.

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

$$\hat{p} = q_v^{*-1}(q_v) = q_v^{*-1}(q_v(\boldsymbol{p}, K)).\qquad\text{(C.2)}$$

Because $\frac{dq_v^*(\hat{p})}{d\hat{p}} \geq \epsilon > 0$ for all $\hat{p} < p_{max}$, there must exist a constant $K_{l_1} > 0$ such that

$$\|\hat{p}_1 - \hat{p}_2\| \leq K_{l_1}\|q_{v1} - q_{v2}\|,\qquad\text{(C.3)}$$

for all $\hat{p}_1 = q_v^{*-1}(q_{v1})$ and $\hat{p}_2 = q_v^{*-1}(q_{v2})$. Furthermore, because $q_v = q_v(\boldsymbol{p}, K)$ is Lipschitz continuous

in $\boldsymbol{p}$, for all $q_{v1} = q_v(\boldsymbol{p}_1, K)$ and $q_{v2} = q_v(\boldsymbol{p}_2, K)$, there exists a constant $K_{l_2} > 0$ to satisfy

$$\|q_{v1} - q_{v2}\| \leq K_{l_2} \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|. \qquad (C.4)$$

From (C.3) and (C.4), for all $\hat{p}_1 = q_v^{*-1}(q_v(\boldsymbol{p}_1, K))$ and $\hat{p}_2 = q_v^{*-1}(q_v(\boldsymbol{p}_2, K))$, we have

$$\|\hat{p}_1 - \hat{p}_2\| \leq K_{l_1} K_{l_2} \|\boldsymbol{p}_1 - \boldsymbol{p}_2\|. \qquad (C.5)$$

This implies that the probability target function given in (C.2) satisfies the Lipschitz condition given in 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 distributed MAC algorithm. If users maintain their transmission probabilities during the Q time slots, by assumption, an increased value of Q can reduce the potential measurement (or estimation) bias in the system arbitrarily close to zero. Therefore the bias condition given in Condition 1 is also satisfied. Consequently, convergence of the proposed distributed MAC algorithm is supported by Theorems 1 and 2.

Appendix D. Proof of Theorem 6

According to the two-step approach, q_v is interpreted by $q_v = (1 - p_k)q_k + p_k d^*(\breve{p})$. When $\breve{p} \geq p_k$, we should either have $q_k = q^*(\breve{p})$ when $\breve{p} < p_{\max}$, or $q_k \geq q^*(\breve{p})$ when $\breve{p} = p_{\max}$. Therefore

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

where the last inequality is due to the fact that $q^*(\breve{p}) - d^*(\breve{p}) \geq 0$ should always hold.

Because $b \geq \max\{1, x^* - \gamma_{\epsilon_v}\}$, according to Theorem 4, $q_v^*(\hat{p})$ is non-decreasing in $\hat{p}$. Therefore, if $q_v > q_v^*(p_{\max})$, we have $\hat{p} = p_{\max} \geq p_k$. Otherwise, we have

$$q_v^*(\hat{p}) = q_v \geq q_v^*(\breve{p}) \geq q_v^*(p_k). \qquad (D.2)$$

This also implies that we $\hat{p} \geq p_k$.

Similarly, when $\breve{p} \leq p_k$, it can be shown that the two-step approach will yield $\hat{p} \leq p_k$.

Acknowledgement

References

[1] N. Abramson, The ALOHA System - Another Alternative for Computer Communication, Proc. AFIPS Conf., Vol. 37, pp. 281-285, Houston, TX, Nov. 1970.

[2] D. Bertsekas and R. Gallager, Data Networks, 2nd Ed., Prentice Hall, NJ, 1992.

[3] X. Qin and R. Berry, Opportunistic Splitting Algorithms for Wireless Networks, IEEE INFOCOM, Hong Kong, China, Mar. 2004.

[4] B. Hajek, Stochastic Approximation Methods for Decentralized Control of Multiaccess Communications, IEEE Trans. on Inform. Theory, Vol. IT-31, pp. 176-184, Mar. 1985.

[5] G. Bianchi, Performance Analysis of the IEEE 802.11 Distributed Coordination Function, IEEE J. on Sel. Areas in Commun., Vol. 18, pp. 535-547, Mar. 2000.

[6] R. Rao and A. Ephremides, On The Stability of Interacting Queues in A Multiple-access System, IEEE Trans. on Inform. Theory, Vol. 34, pp. 918–930, Sep. 1988.

[7] J. Luo and A. Ephremides, On the Throughput, Capacity and Stability Regions of Random Multiple Access, IEEE Trans. on Inform. Theory, Vol. 52, pp. 2593–2607, Jun. 2006.

[8] B. Hajek and T. Loon, Decentralized Dynamic Control of a Multiaccess Broadcast Channel, IEEE Trans. on Auto. Cont., Vol. AC-27, pp. 559-569, Jun. 1982.

[9] H. Kushner and G. Yin, Stochastic Approximation Algorithms and Applications, New York: Springer-Verlag, 1997.

[10] C. Lau and C. Leung, Capture Models for Model Packet Radio Networks, IEEE Trans. on Commun., Vol. 40, pp. 917-925, May 1992.

[11] S. Ghez, S. Verdú, and S. Schwartz, Stability Properties of Slotted ALOHA with Multi-packet Reception Capability, IEEE Trans. on Auto. Contr., Vol. 33, pp. 640-649, Jul. 1988.

[12] J. Luo and A. Ephremides, A New Approach to Random Access: Reliable Communication and Reliable Collision Detection, IEEE Trans. on Inform. Theory, Vol. 58, pp. 989-1002, Feb. 2012.

[13] Z. Wang and J. Luo, Error Performance of Channel Coding in Random Access Communication, IEEE Trans. on Inform. Theory, Vol. 58, pp. 3961-3974, Jun. 2012.

[14] J. Luo, A Generalized Channel Coding Theory for Distributed Communication, IEEE Trans. on Commun., Vol. 63, pp. 1043-1056, Apr. 2015.

[15] S. Karlin and H. Taylor, A First Course in Stochastic Processes, 2nd Ed., Academic, San Diego, CA, 1975.

[16] V. Borkar, and S. Meyn, The O.D.E Method for Convergence of Stochastic Approximation and Reinforcement Learning, New York: Springer-Verlag, 1997.