A Distributed Medium Access Control Algorithm with an Enhanced Physical-Link Layer Interface

Jie Luo, *Senior Member, IEEE*

***Abstract*— This paper presents a distributed medium access control (MAC) algorithm that supports an enhanced physical-link layer interface with multiple transmission options at each link layer user. The MAC algorithm is extended from the one given in [3], which was developed according to a stochastic-approximation-based optimization framework with a general utility function, a realistic link layer channel model, and a convergence guarantee. However, the MAC algorithm given in [3] only contains the component of "random access scheme". The systematic extension presented in this paper takes the algorithm closer to practice by adding the components of "fast adaptation algorithm" and "random scheduling approach". With a careful design to satisfy an independence assumption, such extension enabled the theoretical performance analysis of the MAC algorithm using the well known Markov model, as well as meaningful performance comparison with existing distributed MAC protocols such as the 802.11 distributed coordination function (DCF). Simulation results demonstrated the significant potential in throughput improvement of the distributed MAC algorithm compared with the 802.11 DCF, due to the support of flexible multi-packet reception.**

***Index Terms*— medium access control, distributed system, wireless network**

I. Introduction

With the growing demand of Machine-to-Machine communications and Internet of Things applications, wireless communication networks are evolving rapidly toward the next generation that features massive distributed devices with packet-based bursty messages. The increasing proportion of short messages and the difficulty of quickly coordinating wireless users challenged the classical channel coding wisdom at the physical layer that assumes the dominance of long message transmissions with full multi-user communication optimization [2]. Because communication adaptation often needs to be extended to the data link layer, the evolution also challenged the classical networking wisdom at the link layer that assumes binary transmitting/idling options at each user which significantly limited the exploitation of advanced wireless capabilities such as rate, power, and antenna adjustments [3]. It is becoming clear that the development of future wireless networks requires fundamental understandings of efficient distributed communication and networking without breaking the layered network architecture [3].

The author is with the Electrical and Computer Engineering Department, Colorado State University, Fort Collins, CO 80523. E-mail: rockey@colostate.edu.

This work was supported by the National Science Foundation under Grant ECCS-2128569. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Part of the results of this paper have been presented at the 2023 IEEE Wireless Communications and Networking Conference [1].

In [3][4][5][6], a new channel coding theory was proposed for the distributed communication model at the physical layer. The coding theory allows each physical layer transmitter to prepare an ensemble of channel codes corresponding to different communication settings. When message becomes available, a transmitter can choose an arbitrary code, possibly according to a link layer decision, to encode the message and to transmit the codeword symbols to the receiver. While code ensembles of the users are assumed to be known, actual coding choices are shared neither among the transmitters nor with the receiver. The receiver, on the other hand, should either decode the messages of interest or report collision, depending on whether a pre-determined error probability requirement can be met. Fundamental limit of the system was characterized using a distributed channel capacity region defined in the vector space of the coding choices of the transmitters [3]. The distributed capacity region was shown to coincide with the classical Shannon capacity region, in a sense explained in [3]. Error performance bounds in the case of finite codeword length were obtained in [5][6].

The new channel coding theory provided the basic physical layer support for an enhancement to the physical-link layer interface [3][6], which allows each link layer user to be equipped with multiple transmission options. These options correspond to different codes at the physical layer, possibly representing different communication settings such as different transmission power and rate combinations. The interface enhancement enables data link layer protocols to exploit advanced wireless communication adaptations through the navigation of different transmission options. To maintain a layered network architecture, a link layer user should be constrained to the provided options for transmission adaptation.

In [3][7], a distributed Medium Access Control (MAC) algorithm was proposed to support the enhanced physical-link layer interface at the data link layer. The MAC algorithm assumes homogeneous users with saturated message queues, and a general link layer channel model that can be derived from the physical layer channel and packet coding details [3][7]. Each user is associated with a vector of transmission probabilities corresponding to different transmission options. In each time slot, a user should randomly choose a transmission option to send a packet according to the probabilities specified in the transmission probability vector. A user also receives a measurement of the channel contention level from the receiver, and should apply incremental adaptation to the transmission probability vector accordingly. It was shown that the distributed MAC algorithm falls into the classical stochastic approximation framework [8][9][10], and trajectory of the transmission probability vectors of the users can be

approximated using an Ordinary Differential Equation (ODE) [3][7]. With a careful design of two key functions that guard the probability adaptation, the system ODE was shown to possess a unique equilibrium that is close to optimal with respect to a chosen network utility. Convergence to the designed equilibrium was proven in various senses depending on the step size choices of the probability adaption algorithm [3][7]. While the distributed MAC algorithm proposed in [3][7] is the first one that supports multiple transmission options at each link layer user, it is not yet a practical MAC protocol. This is due to the assumption of equal-sized short packets, as well as the incremental probability adaption that leads to slow convergence.

A practical distributed MAC protocol such as the 802.11 Distributed Coordination Function (DCF) [11] can be regarded as the integration of three key components, namely, a random access scheme, a fast adaptation algorithm, and a random scheduling approach. The "random access scheme" regulates how users with short packets should access the shared channel opportunistically. In 802.11 DCF, when a short packet becomes available at a user, the user can transmit the packet opportunistically only if the channel is first sensed to be idle. The conditional packet transmission probability of each user is guarded by an associated backoff window [11]. Depending on the success/failure status of each transmission, which is fed back by the receiver, the corresponding user should follow the "fast adaptation algorithm" to adjust the size of its backoff window accordingly. When a long message becomes available at a user, on the other hand, the "random scheduling approach" is invoked to schedule the transmission. In the example of 802.11 DCF, the user and the receiver should first exchange Request to Send (RTS) and Clear to Send (CTS) handshake messages using the random access scheme. Once the handshake is successful, other users hearing the handshake should remain idle and the channel should then be reserved exclusively for the long message transmission. According to the above description, it can be seen that the distributed MAC algorithm proposed in [3][7] only focused on the component of "random access scheme". While such simplification helped to obtain theoretical understandings about link layer communication adaptation and convergence proof with the enhanced physical-link layer interface, the components of "fast adaptation algorithm" and "random scheduling approach" must be added before the MAC algorithm can be adopted practically.

In [12], a Markov model was proposed to characterize the operations of a user with the 802.11 DCF protocol. Assume homogeneous users with saturated message queues. With the assumption that packets experience independent collisions when the system reaches stationary status, stationary distribution of the Markov chain and throughput performance of the system were derived theoretically. It was shown that the derived theoretical results match surprisingly well with the simulated performance of the system [12]. Since the publication of [12], the general approach has been widely adopted to carry out theoretical performance analysis and optimization of distributed MAC protocols [13].

In this paper, we extend the distributed MAC algorithm of [3][7] toward a practical distributed MAC protocol by adding to it the components of "fast adaptation algorithm" and "random scheduling approach". Assume that the system contains an unknown number of homogeneous users. We associate each user with a transmission probability vector, as well as an estimate of the number of users in the system. We develop fast adaptation algorithms by extending the exponential backoff algorithm of 802.11 DCF to adjust the estimated number of users, and then adopting the algorithm proposed in [3] to calculate the transmission probability vector as a function of the estimated number of users. Under the assumption of saturated message queues, theoretical performance analysis of the fast adaptation algorithm is carried out by extending the Markov model presented in [12]. However, unlike the collision channel case in 802.11 DCF, with multi-packet reception, channel contention experienced by different users can be highly correlated. To satisfy the assumption that transmission activities of the users should be mutually independent, as required in the theoretical analysis [12], we determine packet transmissions of each user using a random flag generated based on the estimated channel availability probability. With such a revision, computer simulations show that actual performance of the fast adaptation algorithm matches well with the theoretical results. In an example with a Gaussian multiple access channel, assuming equal-sized short packets and with appropriately designed transmission options, the proposed fast adaptation algorithm is shown to achieve a throughput that is consistently above three times the throughput of the 802.11 DCF. The throughput gain comes not only from the higher sum rate achieved by parallel multi-user transmission, compared with a single user transmission scheme, but also from a significantly reduced packet collision probability due to the support of multi-packet reception. When users have long messages, we develop the random scheduling approach by asking each user with a long message to first exchange RTS and CTS handshake messages with the receiver using the random access scheme, similar to the corresponding procedure in 802.11 DCF. Because the receiver may be able to receive multiple packets in parallel, it is possible that RTS/CTS handshakes of multiple users can be successful simultaneously. In this case, the random scheduling approach reserves the channel for parallel long message transmissions at the maximum sum rate. Simplified theoretical analysis and simulation results are given to characterize the throughput of the scheduled long message transmissions of the distributed MAC algorithm.

II. Review of the Random Access Scheme

In this section, we first review the distributed MAC algorithm proposed in [3] for the enhanced physical-link layer interface. Note that the MAC algorithm only focused on the component of "random access scheme".

Consider a distributed multiple access network with a memoryless channel and K homogeneous users (transmitters). Time is slotted, and the length of each time slot equals the transmission duration of one packet. We assume that the number of users K should be unknown to the users and also unknown to the receiver. This is a basic assumption

to most distributed MAC algorithms because otherwise the optimal transmission scheme can be calculated explicitly without the need of distributed adaptation. We assume that each user is backlogged with a saturated message queue. In the convergence analysis presented in [3], such an assumption was introduced to remove the correlation between transmission activities of different users. The assumption is also maintained in the performance analysis to be presented in Section III so that throughput of the system is not constrained by the limited supply of messages and therefore throughput comparison between different distributed MAC algorithms is meaningful.

We assume that each user is equipped with M transmission options plus an idling option. Different transmission options correspond to different coding choices at the physical layer [3]. For example, they could represent the choices of encoding different number of information bits in a packet. At the beginning of each time slot t, each user, say user k, $k = 1, \ldots, K$, should individually decide whether to idle or to send a packet with a randomly chosen transmission option. The corresponding probabilities are specified by an associated M-length probability vector, denoted by $\boldsymbol{p}_k$. We write $\boldsymbol{p}_k = p_k \boldsymbol{d}_k$, with $0 \leq p_k \leq 1$ being the probability that user k transmits a packet, and with vector $\boldsymbol{d}_k$ specifying the conditional probabilities for user k to choose each of the transmission options should it decide to transmit a packet. Entries of the $\boldsymbol{d}_k$ vector satisfy $0 \leq d_{km} \leq 1$ for $1 \leq m \leq M$, and $\sum_{m=1}^{M} d_{km} = 1$. We term p_k the "transmission probability" of user k, and term $\boldsymbol{d}_k$ the "transmission direction" vector of user k.

In each time slot, the receiver envisions the transmission of a carefully designed "virtual packet". Virtual packets assumed in different time slots are identical. A virtual packet 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". We assume that, without knowing the transmission/idling status of the users, the receiver should be able to detect whether the reception of a virtual packet is successful or not [6][3]. We use a simple example to illustrate how virtual packet reception works. Suppose that the link layer channel is a collision channel, and a virtual packet has the same coding parameters of a real packet. If a virtual packet is indeed transmitted, its reception should be successful if and only if no other real packet is transmitted in parallel. Therefore, in this example, virtual packet reception is equivalent to the detection of channel idling status. In other words, the receiver should regard virtual packet reception as successful if no physical transmission activity is detected, and should declare virtual packet reception failure otherwise. In a more general scenario, if all packets including the virtual packet are encoded using random block codes, reception of the virtual packet corresponds to a detection task that judges whether or not the vector of coding choices of all real users should belong to a specific region [6][3]. Such detection tasks and their performance bounds have been extensively investigated in the distributed channel coding literature [4][5][6][3]. Note that, in most cases, the outcome of virtual packet detection can be easily derived from the decoding outcome of real packets. This is because the detection region of the virtual packet is often contained inside the decoding region of the real packets [3]. On one hand, a collision report on the real packets should automatically imply reception failure of the virtual packet. On the other hand, when decoding of the real packets is successful and therefore the coding vector of the users is known, the receiver can easily tell whether or not the coding vector is located inside the detection region of the virtual packet. We assume that, the receiver should maintain an estimate of the success probability of the virtual packet, denoted by $q_v(t)$, and should feed it back to the users. $q_v(t)$ is termed the "channel contention measure" because it is used to measure the contention/availability level of the channel. A high value of $q_v(t)$ reflects a low channel contention level in time slot t.

We require that each user should maintain two key functions, $\boldsymbol{p}^*(\hat{K})$ and $q_v^*(\hat{K})$, both are functions of an estimated number of users $\hat{K}$ [3]. The set of functions should be the same across different users. The $\boldsymbol{p}^*(\hat{K})$ function, termed the "theoretical transmission probability vector" function, specifies the designed (or the targeted) transmission probability vector of the user should the number of users in the system equal $\hat{K}$. The $q_v^*(\hat{K})$ function, termed the "theoretical channel contention measure" function, represents the derived theoretical channel contention measure, if the number of users equals $\hat{K}$ and all users have the same transmission probability vector $\boldsymbol{p}^*(\hat{K})$. We require that $\boldsymbol{p}^*(\hat{K})$ and $q_v^*(\hat{K})$ functions should be specified for both integer and non-integer $\hat{K}$ values.

Define $K_{\min}$ as the maximum $\hat{K}$ that maximizes $q_v^*(\hat{K})$. With the two key functions, the distributed MAC algorithm operates as follows [3].

Distributed MAC Algorithm:

1) Each user initializes its transmission probability vector.
2) Let $Q > 0$ be a pre-determined integer. Over an interval of Q time slots, the receiver measures the success probability of the virtual packet, denoted by q_v, and feeds q_v back to all users.
3) Upon receiving q_v, each user derives an estimated number of users $\hat{K}$ by solving the following equation.

$$q_v^*(\hat{K}) = q_v, \qquad \text{s.t. } \hat{K} \geq K_{\min}. \tag{1}$$

If a $\hat{K}$ satisfying (1) cannot be found, a user should set $\hat{K} = K_{\min}$ if $q_v > q_v^*(K_{\min})$, or set $\hat{K} = \infty$ otherwise. Each user then sets the target transmission probability vector at $\hat{\boldsymbol{p}} = \boldsymbol{p}^*(\hat{K})$.
4) Each user, say user k, updates its transmission probability vector by

$$\boldsymbol{p}_k = (1 - \alpha)\boldsymbol{p}_k + \alpha \hat{\boldsymbol{p}}, \tag{2}$$

where $\alpha > 0$ is the step size parameter for user k.
5) The process is repeated from Step 2 till transmission probability vectors of all users converge.

The distributed MAC algorithm falls into the classical framework of stochastic approximations [8][9][10]. Given the actual number of users K, if all users have the same transmission probability vector $\boldsymbol{p}$, the actual channel contention measure $q_v(\boldsymbol{p}, K)$ can be written as a function of $\boldsymbol{p}$ and K. As shown in [3, Theorem 4.6], if the $\boldsymbol{p}^*(\hat{K})$ function is carefully designed such that $q_v^*(\hat{K})$ is continuous and

monotonically non-increasing in $\hat{K}$, and $q_v(\boldsymbol{p}^*(\hat{K}), K) = q_v^*(\hat{K})$ only happens when $\hat{K} = K$, then the distributed MAC algorithm should lead the transmission probability vectors of all users to converge to the unique equilibrium corresponding to $\hat{K} = K$. In addition, the design of $\boldsymbol{p}^*(\hat{K})$ should make sure that the system is close to optimal at its equilibrium in terms of maximizing a chosen utility function irrespective of the number of users in the system.

Example 1: Consider the simple case of a time-slotted random multiple access system with K homogeneous users over a collision channel. Each user only has a single transmission option. A packet can go through the channel successfully if no other packet is transmitted in parallel. We design the virtual packet to have the same coding details of a real packet. As explained above, virtual packet reception should be regarded as successful if and only if all users idle in the corresponding time slot. Assume that users intend to maximize the symmetric throughput of the system, which is defined as the minimum throughput achieved by a single user. If K is known, optimal transmission probability of each user can be derived as $p^*_{\text{opt}} = \arg\max_p p(1-p)^{K-1} = 1/K$. Following the design guideline presented in [7], however, we should choose the theoretical transmission probability function $p^*(\hat{K})$ as

$$p^*(\hat{K}) = \frac{1}{\hat{K} + 1.01}. \tag{3}$$

While being close to optimal, such a design leads to the following theoretical channel contention measure function

$$q_v^*(\hat{K}) = (1 - p^*(\hat{K}))^{\hat{K}} = \left(1 - \frac{1}{\hat{K} + 1.01}\right)^{\hat{K}}, \tag{4}$$

which is monotonically non-increasing in $\hat{K}$. Alternatively, if we choose $p^*(\hat{K}) = 1/\hat{K}$ or $p^*(\hat{K}) = \frac{1}{\hat{K}+0.5}$, which are closer to optimal, unfortunately, the resulting $q_v^*(\hat{K}) = (1 - p^*(\hat{K}))^{\hat{K}}$ functions will be monotonic in the wrong direction, which can consequently lead to convergence problems of the distributed MAC algorithm.

Detailed discussions on the design of $\boldsymbol{p}^*(\hat{K})$ and $q_v^*(\hat{K})$ functions can be found in [3].

III. Fast Adaptation Algorithms

The distributed MAC algorithm given in Section II only focused on the component of "random access scheme" due to the assumptions of equal-sized short packets and incremental probability adaption. In this section, we still keep the assumption of equal-sized short packets, but propose a "fast adaptation algorithm" to replace the incremental probability adaptation. The fast adaptation algorithm is developed by extending the exponential backoff approach of the 802.11 DCF protocol.

Note that, in 802.11 DCF, a collision avoidance mechanism is implemented to help reduce the probability of packet collision. More specifically, before transmitting any packet, a user needs to make sure that the channel has been idling for a short duration defined as the Distributed InterFrame Space (DIFS) [11][12]. Because DIFS is often much shorter in length than a packet, a quick collision detection can help to reduce the chance of a relatively long collision in packet transmission. Differs from the operation environment of 802.11 DCF, in this section, we maintain the assumption that time is slotted with the length and the transmission schedule of each packet being synchronized to one time slot. In this case, it is not difficult to see that collision avoidance, e.g., to make sure channel is available in the leading time slot before each packet transmission, should not reduce the probability of packet collision. Therefore, collision avoidance is not considered in any of the adaptation algorithms to be presented.

We will present the following three adaptation algorithms. First, to enable fair performance comparison, we will present a modified 802.11 DCF protocol that fits the time-slotted model with the collision avoidance mechanism being removed. Second, we will present the proposed fast adaptation algorithm. Third, we will also present a modified fast adaptation algorithm whose state adaptation is revised to mimic that of the 802.11 DCF protocol. We purposely present the three algorithms with similar description terms and organizations to enable their step-by-step comparison. The modified fast adaptation algorithm is presented to help illustrate the connection between the first two algorithms.

We assume that each user, say user k, should maintain an estimate of the number of users in the system, denoted by $\hat{K}_k$. $\hat{K}_k$ should be kept between pre-determined boundaries denoted by $K_{\min}$ and $K_{\max}$. The value of $K_{\max}$ should be set large enough such that the probability of the system having a number of users more than $K_{\max}$ is negligible. Furthermore, we assume that $K_{\max}$ should also be chosen to satisfy $K_{\max} = 2^c K_{\min}$ for a positive integer-valued c.

First, we present the modified 802.11 DCF protocol. The protocol assumes that each user should only have a single transmission option.

Modified 802.11 DCF:

1) Each user, say user k, initializes its estimated number of users at $\hat{K}_k = K_{\min}$.
2) User k sets its "backoff window," denoted by $W_k(\hat{K}_k)$, at $W_k(\hat{K}_k) = 2\hat{K}_k$. User k then initializes its random "backoff counter" uniformly between 0 and $W_k(\hat{K}_k)-1$.
3) In each time slot, if the "backoff counter" of user k equals 0, user k should transmit a packet.
4) The receiver feeds packet reception status back to the users.
5) At the end of each time slot, user k should take the following actions.
   a) Assume that the "backoff counter" of user k equals 0. If the packet transmitted by user k is received successfully, user k should update its backoff window by $\hat{K}_k = K_{\min}$. If the packet reception failed, user k should update its backoff window by $\hat{K}_k = \min\{K_{\max}, 2\hat{K}_k\}$. The process continues from Step 2.
   b) Assume that the "backoff counter" of user k is positive, user k should decrease its backoff counter by 1. The process continues from Step 3.

Next, we present the proposed fast adaptation algorithm.

Proposed Fast Adaptation Algorithm:

1) Each user, say user k, initializes its estimated number

of users at $\hat{K}_k = K_{\min}$.

2) Let $\boldsymbol{p}^*(\hat{K}_k) = p^*(\hat{K}_k)\boldsymbol{d}^*(\hat{K}_k)$ where $p^*(\hat{K}_k)$ is the transmission probability and $\boldsymbol{d}^*(\hat{K}_k)$ is the transmission direction vector of user k. Let $\lfloor 2/p^*(\hat{K}_k)\rfloor$ be the maximum integer no larger than $2/p^*(\hat{K}_k)$. User k first sets its "backoff window" randomly, denoted by $W_k(\hat{K}_k)$, at $W_k(\hat{K}_k) = \lfloor 2/p^*(\hat{K}_k)\rfloor - 1$ with probability $1 + \lfloor 2/p^*(\hat{K}_k)\rfloor - 2/p^*(\hat{K}_k)$ and at $W_k(\hat{K}_k) = \lfloor 2/p^*(\hat{K}_k)\rfloor$ with probability $2/p^*(\hat{K}_k) - \lfloor 2/p^*(\hat{K}_k)\rfloor$. User k then initialize its random "backoff counter" uniformly between 0 and $W_k(\hat{K}_k) - 1$.
3) In each time slot, if the "backoff counter" of user k equals 0, user k should transmit a packet by randomly choosing a transmission option according to the conditional probabilities specified in the transmission direction vector $\boldsymbol{d}^*(\hat{K}_k)$.
4) The receiver judges whether virtual packet reception in each time slot should be regarded as successful or not, and updates the users with an estimated virtual packet failure probability p.
5) At the end of each time slot, user k should take the following actions.
   a) Assume that the "backoff counter" of user k equals 0. User k should update its estimated number of users randomly by $\hat{K}_k = \max\{K_{\min}, \hat{K}_k/2\}$ with probability $1-p$, and by $\hat{K}_k = \min\{K_{\max}, 2\hat{K}_k\}$ with probability p. The process continues from Step 2.
   b) Assume that the "backoff counter" of user k is positive, user k should decrease its backoff counter by 1. The process continues from Step 3.

Finally, we present the modified fast adaptation algorithm, which differs from the proposed fast adaptation algorithm only in Step 5a.

Modified Fast Adaptation Algorithm:

Step 5a) Assume that the "backoff counter" of user k equals 0. User k should update its estimated number of users randomly by $\hat{K}_k = K_{\min}$ with probability $1-p$, and by $\hat{K}_k = \min\{K_{\max}, 2\hat{K}_k\}$ with probability p. The process continues from Step 2.

Compared with the modified 802.11 DCF protocol, the following is a list of key extensions implemented in the proposed fast adaptation algorithm.

First, in the proposed fast adaptation algorithm, given $\hat{K}_k$, user k sets its backoff window randomly at $W_k(\hat{K}_k) = \lfloor 2/p^*(\hat{K}_k)\rfloor - 1$ or $W_k(\hat{K}_k) = \lfloor 2/p^*(\hat{K}_k)\rfloor$. We will show later in the theoretical analysis that the purpose of such a setting is to get the conditional transmission probability of the user given $\hat{K}_k$ to equal $p^*(\hat{K}_k)$. In comparison, backoff window in the modified 802.11 DCF protocol is set at $W_k(\hat{K}_k) = 2\hat{K}_k$. According to the analysis presented in [12], the resulting transmission probability of the user given $\hat{K}_k$ equals $\frac{1}{\hat{K}_k+0.5}$. As already explained in Example 1, if the same design of $p^*(\hat{K}) = \frac{1}{\hat{K}+0.5}$ is adopted in the stochastic-approximation-based distributed MAC algorithm reviewed in Section II, then the monotonicity property required in the convergence proof will not be satisfied.

Second, in the proposed fast adaptation algorithm, channel availability is sensed at the receiver using the success/failure status of the virtual packet. After each packet transmission, a user adjusts its estimated number of users randomly according to p, which can be regarded as an estimated channel unavailability probability. In comparison, in the modified 802.11 DCF protocol, channel availability is sensed using the success/failure status of a user's own packet transmission. Under the assumption of a collision channel, one can regard channel sensing in the modified 802.11 DCF protocol as checking the idling status of other users. Furthermore, in the modified 802.11 DCF protocol, after each packet transmission, a user adjusts its estimated number of users according to the success/failure status of the particular packet reception event, as opposed to an estimated probability. If such a design is extended to a channel with multi-packet reception, because multiple packets being transmitted in parallel experience the same success/failure outcome, correlation between state adjustments of different users can become significant. This will consequently violate the core assumption required in the theoretical performance analysis to be presented later, which states that transmission activities of the users should be independent [12].

Third, in the proposed fast adaptation algorithm, depending on channel availability (according to a random flag raised with probability $1-p$), a user should increase/decrease its estimated number of user by multipling/dividing its value by 2. Due to good convergence property of the underlying optimization framework, adaptation on increasing/decreasing the estimated number of users in the proposed fast adaptation algorithm is rather balanced. The algorithm can also achieve good throughput performance with a reasonably small $K_{\min}$ value. In comparison, in the modified 802.11 DCF protocol, if packet transmission of a user succeeded, the user should reset the size of its backoff window to $K_{\min}$. Due to such aggressive reduction, the estimated number of users in the modified 802.11 DCF protocol leans heavily toward low values close to $K_{\min}$. This leads to a relatively high transmission probability and consequently to a high packet collision probability when the system has a large number of users. To alleviate the collision problem, 802.11 DCF often needs to choose a high $K_{\min}$ value such as $K_{\min} \geq 16$.

Note that if we assume single transmission option for each user and a classical collision channel, then adaptation approach on the estimated number of users marks the key difference between the proposed fast adaptation algorithm and the modified 802.11 DCF protocol. More specifically, as we will see later in the simulation results of Example 1, if we modify Step 5a of the fast adaptation algorithm to become consistent with that of the 802.11 DCF protocol, then throughput performance of the modified fast adaptation algorithm becomes indistinguishable from that of the modified 802.11 DCF protocol.

Under the assumption of saturated message queues, performance of the proposed fast adaptation algorithm can be analyzed by following the framework presented in [12]. Let us assume that, when the process of probability adaptations of the users become stationary, transmission activities of the users should be mutually independent. Consequently, virtual packet

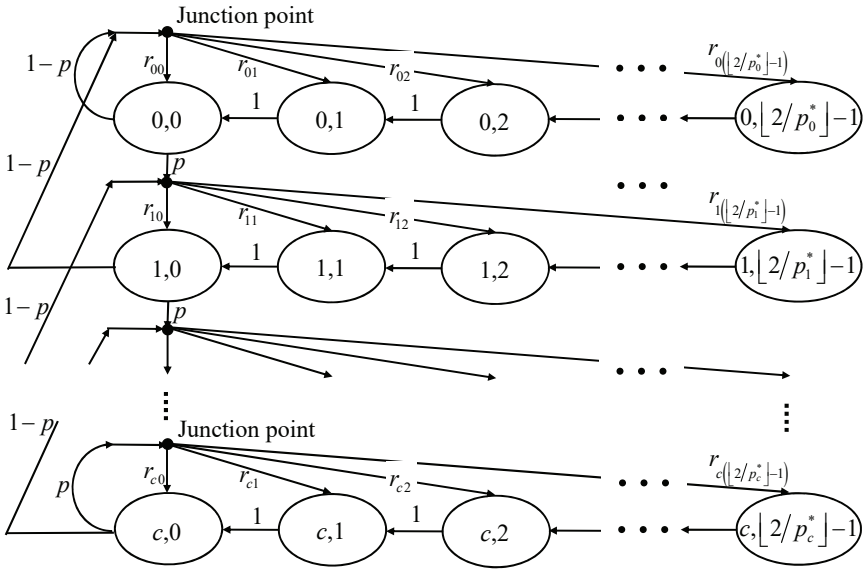

Fig. 1. Markov chain model of the proposed fast adaptation algorithm.

receptions in different time slots should be independent, each having a constant failure probability, denoted by p. From a single user's perspective, behavior of the user can be modeled using a Markov chain illustrated in Figure 1, where state transition of the Markov chain happens in every time slot[1].

The Markov chain has $c+1$ rows of states. In the ith row, $i = 0, 1, \ldots, c$, State (i, j) for $j = 0, 1, \ldots, \lfloor 2/p_i^* \rfloor - 1$ corresponds to an estimated number of users equaling $2^i K_{\min}$ and a "backoff counter" equaling j. In the figure, $p_i^* = p^*(2^i K_{\min})$, for $i = 0, \ldots, c$, is the theoretical transmission probability when the estimated number of users equal $2^i K_{\min}$. Also in the figure, r_{ij} for $i = 0, 1, \ldots, c$ and $j = 0, 1, \ldots, \lfloor 2/p_i^* \rfloor - 1$ represents the probability that, when the estimated number of users equals $2^i K_{\min}$, the user initialize its "backoff counter" at j. Because the backoff window can take two sizes randomly, the values of r_{ij} are given by

$$r_{ij} = \begin{cases} \frac{2/p_i^*}{\lfloor 2/p_i^* \rfloor} - 1 & \text{for } j = \lfloor 2/p_i^* \rfloor - 1 \\ \frac{2\lfloor 2/p_i^* \rfloor - 2/p_i^*}{\lfloor 2/p_i^* \rfloor (\lfloor 2/p_i^* \rfloor - 1)} & \text{for } j < \lfloor 2/p_i^* \rfloor - 1 \end{cases}. \quad (5)$$

Note that $\sum_{j=0}^{\lfloor 2/p_i^* \rfloor - 1} r_{ij} = 1$ for all $i = 0, 1, \ldots, c$.

Let us denote the stationary probability of State (i, j) by $b_{i,j}$. Define auxiliary variables $P_0, \ldots, P_c$ as the probabilities of the junction points marked in Figure 1. We have

$$\begin{aligned} P_0 &= (1-p)b_{0,0} + (1-p)b_{1,0}, \\ P_i &= pb_{i-1,0} + (1-p)b_{i+1,0}, \quad \text{for } i = 1, \ldots, c-1, \\ P_c &= pb_{c-1,0} + pb_{c,0}, \end{aligned} \quad (6)$$

and

$$\begin{aligned} b_{i,\lfloor 2/p_i^* \rfloor - 1} &= P_i r_{i(\lfloor 2/p_i^* \rfloor - 1)}, \\ b_{i,j} &= P_i r_{ij} + b_{i,j+1}, \\ &\qquad \text{for } i = 0, \ldots, c \text{ and } j < \lfloor 2/p_i^* \rfloor - 1. \end{aligned} \quad (7)$$

[1]This is different from the case discussed in [12].

Because $\sum_{j=0}^{\lfloor 2/p_i^* \rfloor - 1} r_{ij} = 1$ for all $i = 0, 1, \ldots, c$, (7) leads to

$$P_i = b_{i,0}, \qquad \text{for } i = 0, \ldots, c. \quad (8)$$

Consequently, we can see that

$$\begin{aligned} &pb_{0,0} = (1-p)b_{1,0}, \\ &b_{i,0} = pb_{i-1,0} + (1-p)b_{i+1,0}, \quad \text{for } i = 1, \ldots, c-1, \\ &(1-p)b_{c,0} = pb_{c-1,0}. \end{aligned} \quad (9)$$

This gives

$$b_{i,0} = \frac{p}{1-p} b_{i-1,0} = \left(\frac{p}{1-p}\right)^i b_{0,0}, \quad \text{for } i = 1, \ldots, c. \quad (10)$$

Combining (7), (8) and (17), we have

$$b_{i,j} = \left(\sum_{k=j}^{\lfloor 2/p_i^* \rfloor - 1} r_{ik} \right) \left(\frac{p}{1-p}\right)^i b_{0,0}, \quad \text{for } i = 0, \ldots, c \text{ and } j \le \lfloor 2/p_i^* \rfloor - 1. \quad (11)$$

Note that

$$\sum_{j=0}^{\lfloor 2/p_i^* \rfloor - 1} \sum_{k=j}^{\lfloor 2/p_i^* \rfloor - 1} r_{ik} = \frac{1}{p_i^*}, \quad \text{for } i = 0, \ldots, c, \quad (12)$$

which means that conditional transmission probability given $\hat{K}_k = 2^i K_{\min}$ equals

$$\frac{b_{i,0}}{\sum_{j=0}^{\lfloor 2/p_i^* \rfloor - 1} b_{i,j}} = p_i^*. \quad (13)$$

Substituting (12) into (11) yields

$$\sum_{j=0}^{\lfloor 2/p_i^* \rfloor - 1} b_{i,j} = \frac{1}{p_i^*} \left(\frac{p}{1-p}\right)^i b_{0,0}, \quad \text{for } i = 0, \ldots, c. \quad (14)$$

Because $\sum_{i=0}^{c}\sum_{j=0}^{\lfloor 2/p_i^*\rfloor-1} b_{i,j}=1$, $b_{0,0}$ can be obtained by

$$b_{0,0}=\frac{1}{\sum_{i=0}^{c}\frac{1}{p_i^*}\left(\frac{p}{1-p}\right)^i}. \tag{15}$$

Finally, given the number of users K, virtual packet failure probability p can be written as a function of $b_{i,0}$ for $i=0,\ldots,c$ using the link-layer channel model, which we will illustrate later in the examples to be presented. Consequently, p, stationary probabilities of the Markov chain, as well as the performance of the system, can be obtained.

For the modified fast adaptation algorithm, under the same independence assumption, behavior of a single user can be modeled using a similar Markov chain with the left side of the state transition diagram being modified as shown in Figure 2. Following an analysis similar to the one presented before, we

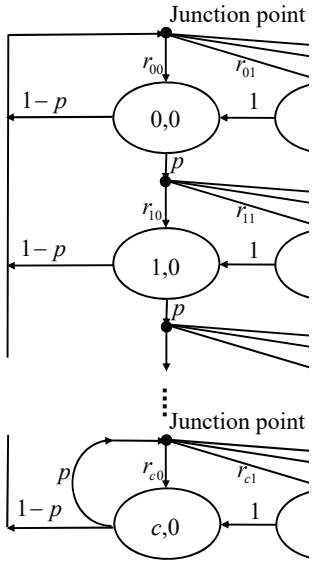

Fig. 2. Markov chain model of the modified fast adaptation algorithm.

have

$$\begin{aligned} b_{0,0}&=(1-p)\sum_{i=0}^{c} b_{i,0},\\ b_{i,0}&=pb_{i-1,0},\quad \text{for } i=1,\ldots,c-1,\\ b_{c,0}&=pb_{c-1,0}+pb_{c,0}. \end{aligned} \tag{16}$$

This gives

$$\begin{aligned} b_{i,0}&=p^i b_{0,0}\quad \text{for } i=1,\ldots,c-1,\\ b_{c,0}&=\frac{p^c}{1-p}b_{0,0}. \end{aligned} \tag{17}$$

Because $\sum_{j=0}^{\lfloor 2/p_i^*\rfloor-1} b_{i,j}=\frac{1}{p_i^*}b_{i,0}$ for $i=0,\ldots,c$, and $\sum_{i=0}^{c}\sum_{j=0}^{\lfloor 2/p_i^*\rfloor-1} b_{i,j}=1$, $b_{0,0}$ can be obtained by

$$b_{0,0}=\frac{1}{\sum_{i=0}^{c-1}\frac{1}{p_i^*}p^i+\frac{1}{p_i^*}\frac{p^c}{1-p}}. \tag{18}$$

Finally, given the number of users K, virtual packet failure probability p can be written as a function of $b_{i,0}$ for $i=0,\ldots,c$ using the link-layer channel model. Consequently, p, stationary probabilities of the Markov chain, as well as the performance of the system, can be obtained.

In the theoretical analysis of the proposed and the modified fast adaptation algorithms, we assumed that stationary transmission activities of the users should be mutually independent, and therefore virtual packet receptions in different time slots should be independent with a constant failure probability. In the following example, we first use a simple system to show that the theoretical performance obtained based on such an assumption is often close to the simulated performance of the corresponding algorithm.

Example 1 (continued): Following Example 1 presented in Section II. With the relatively simple channel model and virtual packet design, in the two fast adaptation algorithms, we can easily relate the stationary virtual packet reception probability to the stationary distribution of the Markov chain. For both algorithms, the stationary transmission probability of a user equals $\sum_{i=0}^{c} b_{i,0}$. Hence channel idling probability and sum throughput of the system are given by

$$\begin{aligned} 1-p&=\left(1-\sum_{i=0}^{c} b_{i,0}\right)^K,\\ \text{Throughput}&=K\left(\sum_{i=0}^{c} b_{i,0}\right)\left(1-\sum_{i=0}^{c} b_{i,0}\right)^{K-1} \end{aligned} \tag{19}$$

For the modified fast adaptation algorithm and the modified 802.11 DCF protocol, we choose $K_{\min}=16$ and $K_{\max}=512$, which implies $c=5$ because $K_{\max}=2^5K_{\min}$. Such a choice leads to the minimum and the maximum backoff window sizes equaling 32 and 1024 in the modified 802.11 DCF protocol, which are typical values specified in the standard [11]. For the proposed fast adaptation algorithm, we choose $K_{\min}=2$ and $K_{\max}=512$. Furthermore, in the proposed and the modified fast adaptation algorithms, we assume that the receiver measures the status of virtual packet reception in every time slot. The receiver initializes the estimated virtual packet failure probability at $p=0$. In each time slot, the receiver updates p by $p=\frac{19}{20}p+\frac{1}{20}I(\text{virtual packet failure})$, where $I\in\{0,1\}$ is the indicator of virtual packet failure in the particular time slot. The updated virtual packet failure probability p is then fed back to the users.

In Figure 3, we illustrated the theoretical and the simulated throughput of the two fast adaptation algorithms as well as the simulated throughput of the modified 802.11 DCF protocol, all as functions of the number of users. The figure contains two groups of performance curves. Let us take the column corresponding to 50 users as a reference. The top group of curves contains a solid and two dashed curves respectively representing the theoretical and the simulated throughput of the modified fast adaptation algorithm, and the simulated throughput of the modified 802.11 DCF protocol. The lower group of curves contains a solid curve and a dashed curve representing the theoretical and the simulated throughput of the proposed fast adaptation algorithm.

We can see from the figure that simulated throughput of the two fast adaptation algorithms matches well with their corresponding theoretical values. Throughput performance of

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2023.3297407

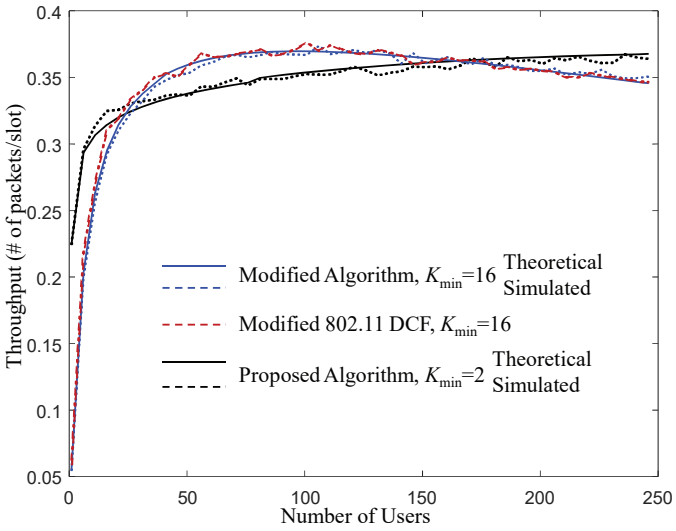

Fig. 3. Throughput as a function of the number of users.

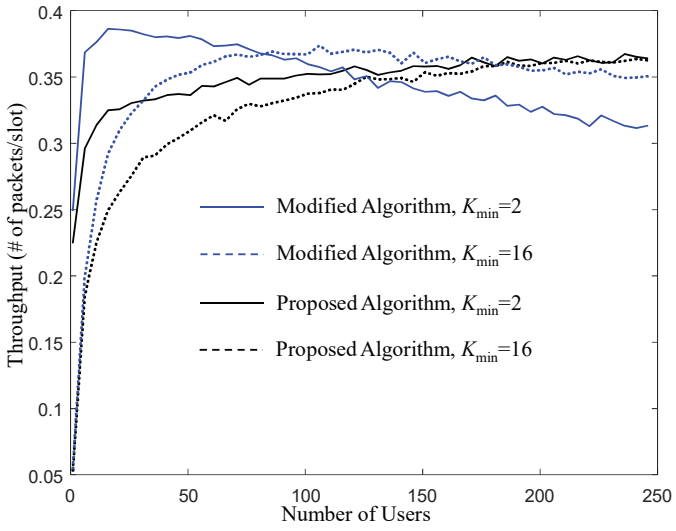

Fig. 4. Throughput performance with different $K_{\min}$ values.

the modified fast adaptation algorithm is indistinguishable from that of the modified 802.11 DCF protocol. This is because, with the collision channel model, the designed conditional transmission probability of the modified fast adaptation algorithm ($\frac{1}{\hat{K}_k+1.01}$) is close to that of the 802.11 DCF protocol ($\frac{1}{\hat{K}_k+0.5}$). The minor difference on channel sensing of the two adaptation algorithms, i.e., idling of all users vs idling of other users, also does not lead to a visible impact on their throughput performance. In both the modified fast adaptation algorithm and the modified 802.11 DCF protocol, a user resets its estimated number of users to $K_{\min}$ when the channel is sensed as available. Consequently, stationary transmission probabilities of a user in these two protocols lean heavily toward the maximum value. While this allows the protocol to achieve a high throughput with a small number of users, when the system has a large number of users, throughput of the two algorithms decreases in the number of users due to excessive collision. Such a problem can be alleviated by adopting a relatively high $K_{\min}$ value, which forces each user to cap its transmission probability at a relatively low value even when channel is available. However, lowering the maximum transmission probability also leads to low throughput performance when the number of users is less than $K_{\min}$. In comparison, throughput of the proposed fast adaptation algorithm increases monotonically in the number of users, and can therefore benefit from a low $K_{\min}$ value. We want to point out that, by changing adaptation schemes on the estimated number of users, and by adjusting the value of $K_{\min}$, one can come up with a wide range of fast adaptation algorithms between the two extreme cases represented by the proposed and the modified algorithms. Optimal adaptation algorithm design is system dependent and is also dependent on the targeted range of the number of users.

To illustrate the impact of different $K_{\min}$ values, we present in Figure 4 the throughput performances of the proposed and the modified fast adaptation algorithms with $K_{\min} = 2$ and $K_{\min} = 16$. Note that we did not include the curves of the modified 802.11 DCF because they are indistinguishable from the corresponding curves of the modified fast adaptation algorithms. We can see that while the proposed fast adaptation algorithm can benefit from a low $K_{\min}$ value, for the modified algorithm, different $K_{\min}$ values only lead to different performance tradeoffs in the number of users.

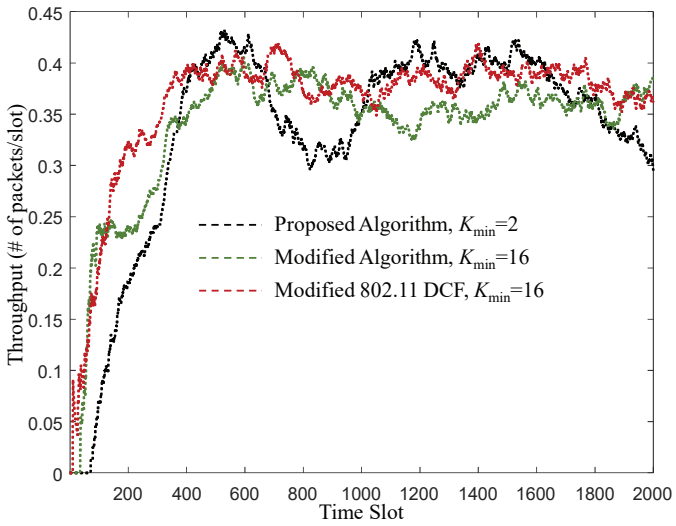

Fig. 5. Simulated throughput as a function of time slots.

In Figure 5, we set the number of users at 100, and illustrate the progression of the simulated throughput performance of the two fast adaptation algorithms as well as that of the modified 802.11 DCF protocol. The throughput data is evaluated using a moving average window of 300 time slots. It can be seen that convergence rates of the three adaptation algorithms are quite similar.

Example 2: In the second example, we consider the case when each user is equipped with multiple transmission options. More specifically, we assume that each user now has two rate options termed the "high" and "low" rate options. The length

of all packets, irrespective of their rates, remains that of one time slot. If all users transmit with the low rate option, the channel can support the parallel transmissions of no more than 64 packets, with each low rate packet carrying $1/64$ data units. If all users transmit with the high rate option, the channel can support the parallel transmissions of no more than 8 packets, with each high rate packet carrying $1/8$ data units.

We assume a simple multi-packet reception channel model. Let N_h and N_l be respectively the number of high rate and low rate packets being transmitted in parallel. We assume that all packets should be received successfully if the following inequality holds.

$$\frac{N_h}{8} + \frac{N_l}{64} \leq 1. \tag{20}$$

Otherwise, no packet will be received. We assume that the data rate of a virtual packet should be equivalent to the combination of 3 high rate packets. Consequently, virtual packet reception should be regarded as successful if and only if

$$\frac{N_h}{8} + \frac{N_l}{64} \leq \frac{5}{8}. \tag{21}$$

Assume again that users intend to maximize the symmetric throughput[2]. For the fast adaptation algorithms, we design the theoretical transmission probability vector function $\boldsymbol{p}^*(\hat{K})$ by following the guideline illustrated in [3]. More specifically, we partition the range of $\hat{K}$ into 3 regions. We define $\{\hat{K}|\hat{K} \leq 12\}$ as the "Head" region, and define $\{\hat{K}|\hat{K} \geq 58\}$ as the "Tail" region. Assuming that users should only use the "high" rate option in the Head region and should only use the "low" rate option in the Tail region. By following the guideline presented in [3, Section 4.3], $\boldsymbol{p}^*(\hat{K})$ in the two regions are designed as

$$\boldsymbol{p}^*(\hat{K}) = \begin{cases} \frac{5.804}{\max\{5,\hat{K}\}+1.01} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \hat{K} \leq 12 \\ \frac{52.28}{\hat{K}+12.29} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \hat{K} \geq 58 \end{cases}. \tag{22}$$

Next, under the assumption that all users should have the same transmission probability vector, we define $\boldsymbol{p}^*_{\text{opt}}(\hat{K}) = p^*_{\text{opt}}(\hat{K})\boldsymbol{d}^*_{\text{opt}}(\hat{K})$ as the optimal transmission probability vector that maximizes the sum throughput of the system. For $\{\hat{K}|12 < \hat{K} < 58\}$, we first set transmission direction vectors $\boldsymbol{d}^*(\hat{K})$ at

$$\boldsymbol{d}^*(\hat{K}) = \boldsymbol{d}^*_{\text{opt}}(\hat{K}), \qquad \text{for } \hat{K} = 13, 14, 15. \tag{23}$$

We also choose $\boldsymbol{d}^*(\hat{K})$ such that $\boldsymbol{d}^*(\hat{K})$ transits linearly in $\hat{K}$ for $15 \leq \hat{K} \leq 58$. After that, we choose transmission probability $p^*(\hat{K})$ such that the resulting "theoretical channel contention measure" function $q^*_v(\hat{K})$ (whose definition can be found in Section II) should transit linearly in $\hat{K}$ for $12 \leq \hat{K} \leq 58$. Note that, while we skipped the reasoning of the $\boldsymbol{p}^*(\hat{K})$ function design in this paper, a detailed explanation of the design in a similar example can be found in [3, Section 4.3].

In Figure 6, we illustrate the throughput as a function of the number of users for several adaptation algorithms with

[2]Note that the design of $\boldsymbol{p}^*(\hat{K})$ function depends on the utility optimization objective, as explained in [3].

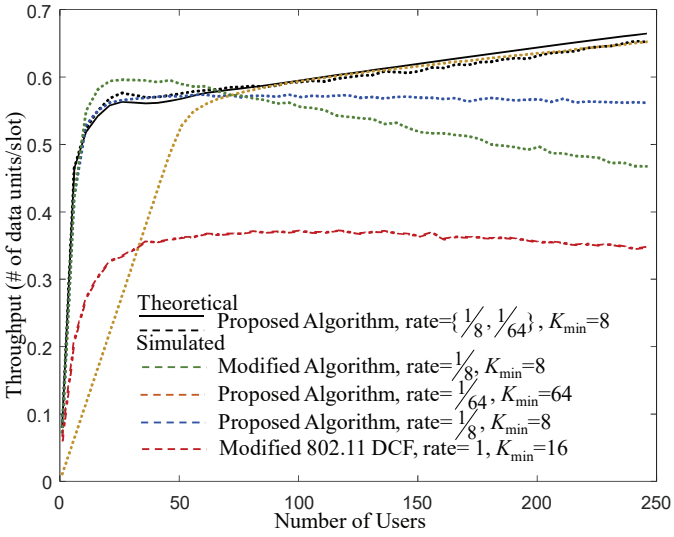

Fig. 6. Throughput as a function of the number of users.

$K_{\max} = 512$ and with appropriately chosen $K_{\min}$ values. The included algorithms are 1, the proposed fast adaptation algorithm with two rate options ($r = \{\frac{1}{8}, \frac{1}{64}\}$), 2, the proposed fast adaptation algorithm with only the high rate option ($r = \frac{1}{8}$), 3, the proposed fast adaptation algorithm with only the low rate option ($r = \frac{1}{64}$), 4, the modified fast adaptation algorithm with the high rate option ($r = \frac{1}{8}$), 5, the modified 802.11 DCF protocol with single user rate ($r = 1$), i.e., with each packet carrying one data unit. We can see that simulated performance of the proposed fast adaptation algorithm matches well with the theoretical result. This is also the case for other MAC algorithms although their theoretical performances are not shown in the figure.

We can see from Figure 6 that throughput of all the fast adaptation algorithms is significantly higher than that of the modified 802.11 DCF. Note that in this example, maximum sum throughput achievable in any time slot equals 1 data unit, irrespective of the rate options chosen by the users. Therefore, throughput gain of the fast adaptation algorithms is brought by their reduced packet collision probabilities because their rate options support multi-packet reception. Comparing the performances of the two proposed fast adaptation algorithms with either high rate option only or low rate option only, we can see that, while a low rate option with $r = \frac{1}{64}$ causes less packet collision than a high rate option with $r = \frac{1}{8}$, it leads to low throughput performance when the system only has a small number of users. The proposed fast adaptation algorithm with two rate options appears to be able to exploit the benefits of both options in the sense of achieving good throughput across the range of the number of users in the system. In addition, with an aggressive adaptation approach on the estimated number of users, and with a high rate option that supports multi-packet reception, the modified fast adaptation algorithm with only the high rate option can achieve the best throughput performance when the system has no more than 50 users. However, its throughput improvement compared with the corresponding proposed fast adaptation algorithm is

relatively small.

Next, we change the channel model to a multiple access channel with additive white Gaussian noise. Assume that signals of all users and all rate options should have the same receiving signal to noise ratio (SNR) of SNR $= 15$ dB[3]. Let us define single user rate r_s, high rate r_h, and low rate r_l in bits per symbol as

$$r_s = \frac{1}{2}\log(1+\text{SNR}),$$
$$r_h = \frac{1}{16}\log(1+8\text{SNR}),$$
$$r_l = \frac{1}{128}\log(1+64\text{SNR}). \quad (24)$$

Note that r_s, r_h, and r_l correspond respectively to the maximum symmetric information rate when there are 1, 8, and 64 users transmitting in parallel [14]. We assume that packets transmitted in the modified 802.11 DCF protocol have a rate of r_s, while packets transmitted in the fast adaptation algorithms have a rate of r_h when the high rate option is used, or a rate of r_l when the low rate option is used. When all users use the same rate option, maximum sum throughput supported by the channel corresponding to r_s, r_h, and r_l respectively equals $\frac{1}{2}\log(1+\text{SNR})$, $\frac{1}{2}\log(1+8\text{SNR})$, and $\frac{1}{2}\log(1+64\text{SNR})$ [14]. In this case, supporting multi-packet reception not only can reduce packet collision probability, but also can increase the sum throughput of the channel.

Let N_h and N_l be respectively the number of high rate and low rate packets being transmitted in parallel. We assume that all packets should be received successfully if[4]

$$N_h r_h + N_l r_l \le \frac{1}{2}\log(1+(N_h+N_l)\text{SNR}). \quad (25)$$

Otherwise, no packet will be received. We assume that virtual packet reception should be regarded as successful if and only if

$$(N_h+3)r_h + N_l r_l \le \frac{1}{2}\log(1+(N_h+N_l+3)\text{SNR}). \quad (26)$$

We maintain the design of the theoretical transmission probability vector function $\boldsymbol{p}^*(\hat{K})$ as before, but replace the throughput calculation and the virtual packet failure probability calculation in the example using the updated criteria given in (25) and (26). In Figure 7, we re-plotted the throughput performance of the same set of adaptation algorithms. We can see that, because of the maximum sum throughput gain brought by supporting multi-packet reception, the proposed fast adaptation algorithm with two rate options achieved a throughput that is consistently above three times that of the modified 802.11 DCF. Note that throughput comparison between the distributed MAC algorithms is fair in the sense that the receiving SNR of all users are kept at a constant irrespective of their rate choices.

[3]This is a typical SNR value in the "low signal" category in a WiFi network.

[4]Strictly speaking, with a finite codeword length, the packet reception criterion (25) and the virtual packet reception criterion (26) require that rates of the options should be slightly less than the values given in (24) [15]. However, such a minor difference does not change the general conclusion of the example.

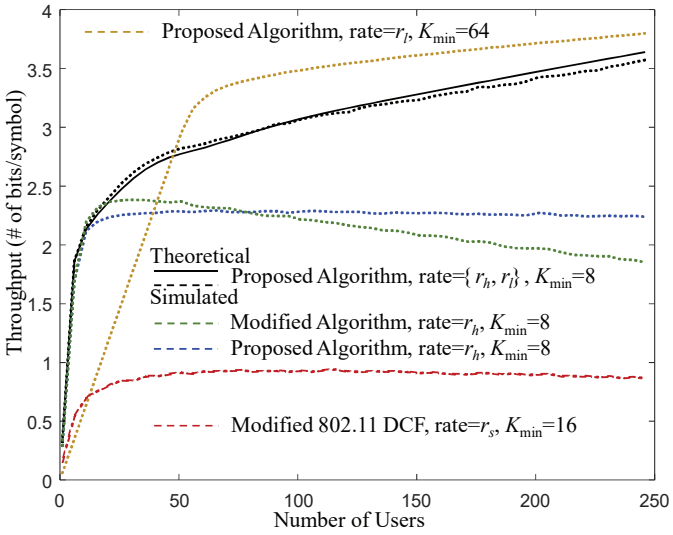

Fig. 7. Throughput as a function of the number of users. $r_s = \frac{1}{2}\log(1+\text{SNR})$, $r_h = \frac{1}{16}\log(1+8\text{SNR})$, $r_l = \frac{1}{128}\log(1+64\text{SNR})$, SNR $= 15$ dB.

IV. Random Scheduling Approach

In 802.11 DCF, when a long message becomes available at a user, the user should first send a control message RTS to the receiver. If the message is received successfully, the receiver should reply with a CTS message. All other users hearing the RTS/CTS exchange should remain idle, and the channel is then reserved exclusively for the long message transmission. We term this mechanism the "random scheduling approach" because, while it schedules undisturbed long message transmissions, such scheduling activities are initiated randomly by distributed users without coordinated multi-user planning. In this section, we further extend the distributed MAC algorithm to include the random scheduling approach, and to discuss its benefit when the receiver supports multi-packet reception.

For 802.11 DCF, the random scheduling approach enables long messages to be transmitted at the maximum single user rate. Therefore, if communication is dominated by long message transmissions, overall rate of the system becomes close to the single user rate. When multi-packet reception is supported at the receiver, however, maximum sum rate of the channel can increase in the number of participating users. Since long message transmissions are scheduled randomly, the number of participating users in each scheduled transmission is a random variable. Consequently, even when communication is dominated by scheduled transmissions, calculating the overall sum rate of the system can still be a tricky task.

Let us assume that short packets of the users are still transmitted using a distributed MAC protocol such as those introduced in Section III. Assume that an RTS message can be embedded in any short packet, irrespective of the chosen transmission option. Because the distributed MAC protocol may support multi-packet reception, it is possible that multiple RTS messages can be received simultaneously. In the following, we present a random scheduling approach that can be inserted as an intermediate Step 4.5 into any of the distributed MAC algorithms introduced in Section III.

Proposed Random Scheduling Approach:

Step 4.5) In each time slot, the receiver checks whether an RTS message is received. If so, the receiver should respond with a CTS message using a feedback channel. The CTS message should include the total number of RTS messages received in parallel, and the corresponding sizes of long messages (in number of bits) involved in the transmission requests.

Upon hearing a CTS message, users should calculate the length of the scheduled long message transmission assuming that messages should be sent in parallel at the supported maximum sum rate. Scheduled long message transmission should then begin with appropriately designed channel codes. Other users hearing the RTS/CTS exchange should pause their communication activities, and should remain idle during this process.

Once the long messages are received successfully, the receiver should send an Acknowledgement (ACK) message to confirm the reception.

It is not difficult to carry out theoretical performance analysis of the random scheduling approach if we focus on the throughput of the long message transmissions only. Such a focus helps to remove from the results the impact of the sizes of messages, and therefore can make the results relatively easy to understand. Let R_s be the expected sum throughput of a scheduled long message transmission, in bits per symbol. Denote by P_N the probability of scheduling the parallel transmissions of N long messages. Denote the sum throughput in bits per symbol of a scheduled transmission of N messages by R_N. Let S be the expected size of a long message. R_s can be obtained by

$$R_s = \frac{E[\#\text{ of message bits}]}{E[\#\text{ of channel symbols}]} = \frac{\sum_N P_N N S}{\sum_N P_N \frac{NS}{R_N}} = \frac{\sum_N P_N N}{\sum_N P_N \frac{N}{R_N}}. \quad (27)$$

Given a particular distributed MAC algorithm, P_N can be further calculated using the channel model and the stationary transmission probabilities of the users, as we will illustrate in the following example.

Example 2 (Continued): Let us consider Example 2 presented in Section III with the Gaussian multiple access channel. By adding the random scheduling approach to the distributed MAC protocols presented in Section III, we will calculate the throughput of the long message transmissions in these protocols. We still maintain the same level of receiving SNR for the signal of each user in their long message transmissions. We assume that each user has a constant probability, denoted by p_s, to embed an RTS message in a short packet, irrespective of the chosen transmission option.

For the modified 802.11 DCF protocol, because rate of the scheduled long message transmission is fixed at the single user rate, we have $R_S = \frac{1}{2}\log(1+\text{SNR})$.

For the proposed and the modified fast adaptation algorithms, stationary short packet transmission probability of each user is given by

$$p_t = \sum_{i=0}^{c} b_{i,0}. \quad (28)$$

Let d_h and d_l be the conditional probabilities that a user chooses the high rate option and the low rate option, respectively. Let N_h and N_l be respectively the number of high rate short packets and low rate short packets being transmitted in a time slot. Define $I(N_h, N_l)$ be the indicator function that $I(N_h, N_l) = 1$ if and only if N_h and N_l satisfy (25). Given the number of users K, we can calculate P_N, which is the probability that N RTS messages are received successfully in parallel by the receiver, as follows.

$$P_N = \sum_{\substack{N_h, N_l \\ I(N_h,N_l)=1 \\ N \le N_h+N_l \le K}} \binom{K}{N_h}\binom{K-N_h}{N_l} \times (p_t d_h)^{N_h}(p_t d_l)^{N_l}(1-p_t)^{(K-N_h-N_l)}\binom{N_h+N_l}{N} \times p_s^N (1-p_s)^{(N_h+N_l-N)}. \quad (29)$$

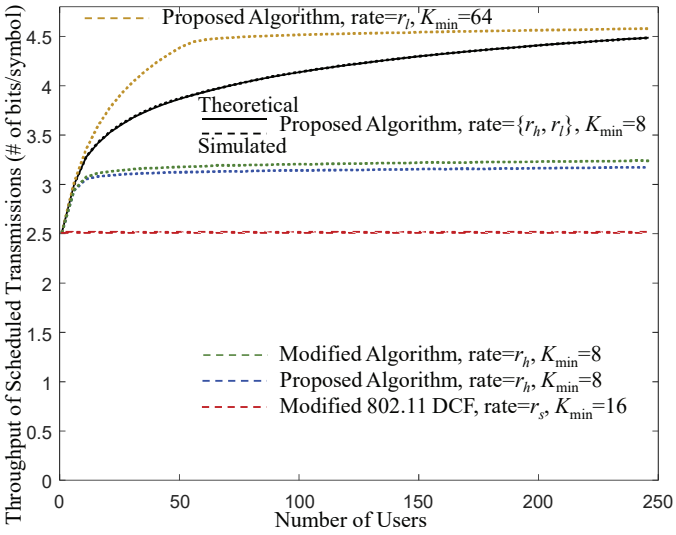

Fig. 8. Throughput of scheduled transmissions as a function of the number of users. $r_s = \frac{1}{2}\log(1+\text{SNR})$, $r_h = \frac{1}{16}\log(1+8\text{SNR})$, $r_l = \frac{1}{128}\log(1+64\text{SNR})$, SNR = 15 dB, $p_s = 0.4$.

In Figure 8, we set $p_s = 0.4$, and illustrated the throughput of the scheduled long message transmissions in bits per symbol for various adaptation algorithms. Compared with the modified 802.11 DCF protocol, throughput gains of the other distributed MAC algorithms come purely from their capability of scheduling multiple long message transmissions in parallel. Therefore, it is not surprising to see that the proposed MAC algorithm with a single low rate option achieved the maximum throughput gain because the short packet part of the MAC algorithm supports the most number of packets being received in parallel. For MAC algorithms with a single rate option, throughput of their scheduled transmissions becomes flat once the number of users increases beyond a threshold. Throughput of the proposed MAC algorithm with two rate

options, however, continue to increase in the number of users in a wide range of system sizes. This is because, as users increase their probabilities of using the low rate option, the number of users involved in successful multi-packet receptions also increases. The proposed MAC algorithm with two rate options can clearly bridge the throughput performance of the corresponding algorithms with single rate options, although such a transition happens slowly in the number of users.

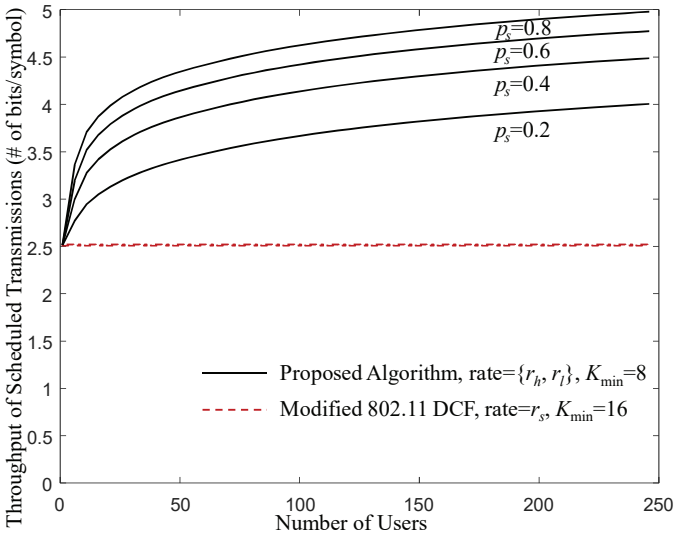

Fig. 9. Throughput of scheduled transmissions of the proposed MAC algorithm with two rate options for various p_s values.

In Figure 9, we plotted the throughput of scheduled transmissions of the proposed MAC algorithm with two rate options for various p_s values. As expected, the throughput increases in p_s because a higher p_s value leads to a higher probability of scheduling a large number of parallel long message transmissions. Note that, to further exploit the throughput gain due to parallel transmissions of multiple users, one can revise the RTS/CTS random scheduling approach to group long message transmissions requested in multiple successive time slots, and hence to increase the number of participating users in each scheduled transmission event. Investigations on such a revision, however, is beyond the scope of this paper.

In a practical environment, depending on the relative sizes of long and short messages, overall throughput performance achieved by a MAC algorithm should lie between the throughput of its scheduled transmissions and its short packet transmissions. The probability p_s of requesting a scheduled long message transmission depends on the traffic load. There are other factors such as transmission delay, sizes of the control messages, sizes of the headers of packets, etc, should be taken into consideration. These factors are ignored in this paper to give a relatively clean image about the performance of the MAC algorithms.

V. Discussions and Conclusion

With the development of distributed channel coding theory [3][4][5][6] and the proposal of the enhanced physical-link layer interface [6][3], exploitation of advanced wireless capability at the link layer is enabled in the following two senses. First, by equipping each link layer user with multiple transmission options, distributed MAC algorithms can exploit the benefit of multi-packet reception with improved throughput due to reduced packet collision. Second, the enhanced physical-link layer interface also enabled realistic link layer channel model that can be derived from the physical layer channel and packet coding details [6][3]. Consequently, MAC algorithms can exploit special channel properties such as improved sum throughput due to parallel transmissions of multiple packets. While these benefits were investigated in the development of a stochastic approximation-based distributed MAC algorithm presented in [3], the focus was put on developing the optimization framework and investigating its optimality and convergence properties.

In this paper, we extended the distributed MAC algorithm presented in [3] further toward a practical distributed MAC protocol by adding the components of "fast adaptation algorithm" and "random scheduling approach". The extensions were carefully designed such that theoretical performance obtained using a Markov model, originally proposed in [12], matches well with the simulated performance of the MAC algorithm. The key significance of these systematic extensions is that they established a connection, although an implicit one, that relates a stochastic approximation-based MAC algorithm to a more practical version whose performance can be theoretically evaluated and simulated.

Such a connection is bidirectional. In the reverse direction, the connection can help to improve the understandings about practical distributed MAC protocols by investigating their corresponding stochastic approximation-based optimization models. For example, 802.11 DCF can be related to a distributed optimization algorithm, where an unknown number of homogeneous users intend to maximize the symmetric system throughput with their theoretical transmission probability function being set at $p^*(\hat{K}) = \frac{1}{\hat{K}+0.5}$, $\hat{K}$ is the estimated number of users. As explained in Section II, such a design leads to a theoretical channel contention measure function $q_v^*(\hat{K})$ that does not satisfy the convergence condition presented in [3]. Such an understanding is consistent with observations on the convergence issue of 802.11 DCF, often characterized as unfairness problems, widely reported in the literature [16].

In the forward direction, the connection enabled meaningful performance evaluation and comparison for a large number of distributed MAC algorithms, developed in [3] based on a theoretical optimization framework with clearly designed utility function, realistic channel model, and proven convergence. Throughput evaluations of the extended distributed MAC algorithms can further support investigations on the design of their transmission options, channel contention measures, as well as their backoff algorithms. However, we want to point out that, the extended MAC algorithms are still not yet practical, due to unavailability of low complexity distributed channel codes, as well as the lack of support of the collision avoidance mechanism implemented in 802.11 DCF.

REFERENCES

[1] J. Luo, *Performance of Distributed Medium Access Control with an Enhanced Physical-Link Layer Interface,* IEEE WCNC, Glasgow, UK, Mar. 2023.

[2] A. Ephremides and B. Hajek, *Information Theory and Communication Networks: An Unconsummated Union,* IEEE Trans. Inform. Theory, Vol. 44, pp. 2416-2434, Oct. 1998.

[3] Y. Tang, F. Heydaryan, J. Luo, *Distributed Coding in A Multiple Access Environment,* Foundations and Trends in Networking, Vol. 12, pp. 260-412, Now Publishers, 2018.

[4] J. Luo and A. Ephremides, *A New Approach to Random Access: Reliable Communication and Reliable Collision Detection,* IEEE Trans. Inform. Theory, Vol. 58, pp. 379-423, Feb. 2012.

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

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

[7] Y. Tang, F. Heydaryan, and J. Luo, *Distributed Multiple Access with A General Link Layer Channel,* Ad Hoc Networks, Vol. 85, pp. 120-130, Mar. 2019.

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

[9] V. Borkar and S. Meyn. *The O.D.E Method for Convergence of Stochastic Approximation and Reinforcement Learning,* SIAM J. Contrl. and Opt., Vol. 38, pp. 447-469, Jan. 2000.

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

[11] IEEE Computer Society LAN Man Standards Committee, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,* IEEE Standard 802.11-1999, New York, NY, IEEE 1999.

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

[13] L. Dai and X. Sun, *A Unified Analysis of IEEE 802.11 DCF Networks: Stability, Throughput and Delay,* IEEE Trans. Mobile Computing, Vol. 12, pp. 1558-1572, Aug. 2013.

[14] T. Cover and J. Thomas, *Elements of Information Theory,* 2nd. Ed., John Wiley & Sons, New Jersey, 2005.

[15] Y. Polyanskiy, H. V. Poor, and S. Verdu, *Channel Coding Rate in The Finite Blocklength Regime,* IEEE Trans. Inform. Theory, Vol. 56, pp. 2307-2359, May 2010.

[16] G. Celik, G. Zussman, W. Khan, and E. Modiano, *MAC for Networks with Multipacket Reception Capability and Spatially Distributed Nodes,* IEEE Trans. Mobile Comp., Vol. 9, pp. 226-240, Feb. 2010.