Master's Thesis

Random Access Scheduling without Message Passing: A Collision-based AIMD Approach

Seunghyun Lee

Department of Electrical and Computer Engineering

(Computer Engineering)

Graduate School of UNIST

2016

Random Access Scheduling without Message Passing: A Collision-based AIMD Approach

Seunghyun Lee

Department of Electrical and Computer Engineering

(Computer Engineering)

Graduate School of UNIST

ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY

Random Access Scheduling without Message Passing: A Collision-based AIMD Approach

A thesis/dissertation

submitted to the Graduate School of UNIST

in partial fulfillment of the

requirements for the degree of

Master of Science

Seunghyun Lee

6. 24. 2016

Approved by

Advisor

Changhee Joo

Random Access Scheduling without Message Passing: A Collision-based AIMD Approach

Seunghyun Lee

This certifies that the thesis/dissertation of Seunghyun Lee is approved.

6. 24. 2016

Signature

Changhee Joo

Signature

Kyunghan Lee

Signature

Hyoil Kim

Abstract

Wireless scheduling has been extensively studied in the literature. Since MaximumWeighted Scheduling has been developed and shown to achieve the optimal performance, there have been many efforts to overcome its complexity issue. Random access has attracted much attention due to its potential for low complexity and distributed control, which are desirable for scheduling in multi-hop wireless networks. Although several interesting random access scheduling schemes have been shown to be provably efficient, they suffer in practice from high packet delays or severe performance degradation due to the control overhead to exchange information between neighboring links. In this paper, we develop a novel random access scheduling scheme that does not need message passing. We pay attention to the interplay between the links and control their access probabilities targeting at a certain collision rate. We employ the Additive Increase Multiplicative Decrease (AIMD) algorithm for convergence, and show that our proposed scheme can achieve the same performance bound as the previous random access schemes with high control overhead. We verify our results through simulations and show that our proposed scheme achieves the performance close to that of the centralized greedy algorithm.

Contents

Figure Contents

Ⅰ. INTRODUCTION

Scheduling in wireless networks is a process of granting users access to shared medium for transmission. Within the same frequency channel, simultaneous transmissions from multiple nodes may cause mutual interference if they are close to each other, in which case, none can transmit data at an acceptable rate, resulting in a *collision*. On the other hand, multiple sender-receiver pairs can successfully transmit at the same time, provided that they are sufficiently distant from each other, which can significantly improve the spectrum efficiency. We denote a set of the links that consists of non-interfering transmissions by a *feasible schedule*, and define the wireless scheduling problem as finding a sequence of feasible schedules that maximize the network performance.

Developing an optimal scheduling solution is a difficult task due to the non-linear, non-convex property of wireless interference. It has been known that Maximum Weighted Scheduling (MWS) achieves the optimal throughput by selecting the feasible schedule that maximizes the queue weighted rate sum [5]. However, it requires a centralized control with high computational complexity. In the primary interference model, where two links sharing a node cannot make simultaneous transmissions, the scheduling decision process that finds the schedule of the maximum queue weighted rate sum has $O(|V|^3)$ complexity, where $|V|$ denotes the number of nodes, and it is an NP-Hard problem in general [1].

There have been several efforts to achieve throughput optimality with lower complexity. A family of scheduling schemes called *Pick-and-Compare* randomly pick a feasible schedule and compare it with the previous schedule, and choose the better one as the next schedule [14], [15], [16]. They are shown to be throughput optimal with $O(1)$ complexity. However, they need frequent exchanges of control messages across the network to compare the performance, which often will result in substantial performance degradation in practice.

Recently, optimal scheduling solutions without message passing have been developed by exploiting the carrier-sensing technique. Continuous-time Carrier-Sensing Multiple Access (CSMA) scheduling scheme [2] operates without control messages and finds the optimal distribution of feasible schedules by exploiting the Glauber dynamics. An important weakness of the continuous-time CSMA is the assumption of perfect carrier-sensing, under which a link can sense the signal transmission of its neighbor and suppress its transmission immediately, so that there is no collision. Such an assumption is, however, infeasible since a few microseconds delay in the hardware and the signal propagation delay is unavoidable in practice. Ni et al. have developed Q-CSMA that achieves the same optimal distribution of feasible schedules as the continuous CSMA, but operates in discrete-time systems without the assumption of the perfect carrier-sensing [3]. Although Q-CSMA also achieves the optimal throughput, it needs again the control message passing, and empirically suffer from poor delay performance for the convergence to the optimal distribution, which will hinder its practical use.

There have been another effort to reduce the complexity and to develop provably efficient scheduling solutions that are amenable to distributed implementation, leading to the development of the Constant-time scheduling schemes [6], [7], [8], [9]. These ALOHA-like adaptive scheduling schemes let each link randomly access the shared medium based on the information it collects from its neighborhood by passing control messages. They are shown to achieve a fraction of the optimal throughput and provide an explicit trade-off between the complexity and the throughput performance. Although they can be implemented in a distributed fashion and empirically achieve high spectrum efficiency, the requirement of explicit message passing may cause a significant amount of overhead and thus substantially degrade the overall performance [7].

In more practical settings, IEEE 802.11 DCF that controls contention using the backoff timer has been studied to achieve the optimal throughput and the fairness. Under the principle that the links in the same contention domain should have an identical backoff time, the authors in [13] have proposed to copy the backoff time of a station to the others in the contention domain. In single-hop networks, several studies has shown that the optimal backoff time can be obtained by estimating the number of contending links. In [17], channel idle time and collision probability have been used to estimate the network size. In [18], Additive Increase and Multiplicative Decrease (AIMD) algorithm based on the idle event during the half of the contention period has been employed to estimate the number of contending links. In [19], the number of idle slots before a transmission has been used for the AIMD algorithm to estimate the number of contending links. The aforementioned results, however, are limited to single-hop scenarios, and their performance in multi-hop environment remains unclear.

Also, researches under the condition of the availability of multi-packet transmission/reception using multiple antenna systems have been studied. In [27], the authors have suggested the backoff algorithms for IEEE 802.11 using multi-packet reception, which maximize the system throughput. For maximizing the throughput of an 802.11 network, the minimum contention window when simultaneous communications through a number of directional/smart antennas is enabled has been studied in [25]. In [26], a MAC layer protocol has been proposed through the analytic research about the problems in IEEE 802.11ac WLANs caused by the nervous bandwidth resources.

In this work, we develop distributed scheduling schemes that achieve high throughput performance in multi-hop wireless networks. Based on passively collected information, our proposed schemes adjust the channel access probability in an AIMD manner without message passing. Our main contribution can be summarized as follows:

- We develop distributed scheduling schemes that adjust the attempt probability dynamically with neither control message exchanges nor a priori knowledge of the traffic information. Unlike the previous works, we do not set the attempt probability from the estimation on the number of interfering neighbors. Instead, we try to meet a target collision probability.

- We show that under a mild assumption our proposed solution can theoretically achieve $\frac{1}{\Delta}$ fraction of the optimal throughput, where Δ denotes the interference degree.
- We verify the performance of our scheduling schemes and show that they achieve high performance comparable to the state-of-the-art distributed scheduling schemes with message passing through a number of simulations.

This paper is organized as follows. We describe our system model in Section Ⅱ. We explain the motivation of our work and develop a novel distributed random-access scheduling in Section Ⅲ. The proposed scheme is extended to the case with multiple contention opportunities in Section Ⅳ. We evaluate our schemes through simulations in comparison with the state-of-the-art distributed schedulers in Section Ⅴ, and we conclude in Section Ⅵ.

Ⅱ. SYSTEM MODEL

We consider a multi-hop wireless network graph $G = (V, E)$ with the set V of nodes and the set E of links. We assume that time is slotted, and a single frequency channel is shared by all the links. We consider the primary (or 1-hop) interference model, where any two links within 1-hop distance cannot transmit at the same time due to the mutual interference between them. Such two links are called a *neighbor* of each other. If any two neighboring links transmit simultaneously, a *collision* occurs and both transmissions will fail. Let N_l denote the set of neighbors of link l excluding itself, and let $N_l^+ \coloneqq N_l \cup \{l\}$. Also, let Δ_l denote the largest number of mutually non-interfering links in N_l^+, and let Δ denote the *interference degree* defined as $\Delta \coloneqq \max_{l \in E} \Delta_l$. For instance, we have $\Delta = 2$ under the primary interference model. Our result can be easily extended to more general K-hop interference models that define the link within K-hop distance as an interfering neighbor.

Let $A_l(t)$ be the number of packet arrivals at the beginning of time slot t at link l, λ_l denote the mean arrival rate, i.e., $\lambda_l \coloneqq E[A_l(t)]$, and $\vec{\lambda}$ denote its vector. Let $D_l(t)$ denote the actual number of packet departures from the queue of link l. Let c_l denote the capacity of link l when it makes a successful transmission. In this work, we assume unit link capacity, i.e., $c_l = 1$, but our results can be easily extended to the case of different link rates. The queue length Q_l of link l evolves as

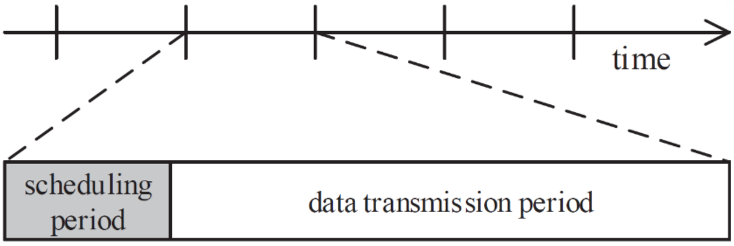


Fig. 1. Time structure of a slot.

$$Q_l(t+1) = Q_l(t) + A_l(t) - D_l(t). \tag{1}$$

In the sequel, we omit the time slot index t if there is no confusion.

The network is said to be *stable* when all the queues remain finite (or stable), or the packet arrival rate is less than the service rate $\lambda_l < E[D_l(t)]$ for all $l \in E$. The *capacity region* Λ is defined as the set of arrival rates that can be supported by some scheduling scheme. We measure the throughput performance of a scheduling scheme by the *efficiency ratio* defined as the largest fraction γ of the capacity region such that the scheduling scheme can stabilize the network for $\gamma\vec{\lambda}$ for all $\vec{\lambda} \in \Lambda$. If $\gamma = 1$, the scheduling scheme achieves the capacity region and it is said to be *throughput optimal*.

We divide a time slot into two periods: a scheduling period and a data transmission period. During the scheduling period, the scheduling scheme makes a decision about which set of non-interfering links will be active. Then the links in the chosen set transmit data during the transmission period.

In this work, we pay attention to random access scheduling due to its potential for low complexity and distributed control. In particular, we consider the following generic random access scheduler.

- During the scheduling period, each link l attempts the transmission of RTS (Request-To-Send) in a probabilistic manner, and the links which made the successful attempt (i.e., received CTS (Clear-To-Send)) transmit data during the transmission period.

Due to the randomness, it is possible for more than two neighboring links to attempt at the same time, in which case, a collision occurs and neither links can transmit data during the transmission period. Thus an appropriate setting of the attempt probability Pl is the key to achieving high performance: too small P_l reduces the spatial spectrum reuses and too large P_l will result in the waste of resource due to collisions.

It has been known that the state-of-the-art random access schemes [6], [7], [8] can guarantee up to $\frac{1}{\Delta}$ fraction of the optimal throughput under the K-hop interference model, and empirically achieve a near-optimal performance. Such schemes, however, have a common weakness – they require that each link collect the information of its neighboring links (e.g., queue length) to control the attempt probability. The overhead from the control message passing is often substantial [7], and in highly dense networks, may degrade the performance to an unacceptable level. To this end, we want to develop scheduling schemes that operate without control message passing while achieving good throughput performance.

Ⅲ. ADAPTIVE ATTEMPT PROBABILITY WITHOUT MESSAGE PASSING

In this section, we first present a static policy in a single-hop network, which will provide us an insight into the ideal attempt probability control. Then, we present the potential of the *collision probability control* in achieving good performance without message passing. We extend our ideas to the multi-hop networks, and achieve the performance comparable to the state-of-the-art scheduling schemes that require message passing for collecting the queue length information of the neighboring links.

Ⅲ-A. Motivation: A static scheme with saturated traffic

Consider a single-hop network with N links, where all the links are neighboring with each other and only one transmission can occur at a time. In this scenario, we assume that the links are always backlogged and have data to transmit.

Under our generic scheduling scheme, each link l can attempt for transmission during the scheduling period with probability P_l. Suppose that each link has a single attempt opportunity at each time slot and has an identical attempt probability, i.e., $P_l = P$ with some constant P. The overall throughput U can be easily calculated as

$$U = \sum_{l \in E} P_l \left(\prod_{k \in N_l} (1 - P_k) \right) = NP(1-P)^{N-1}. \qquad (2)$$

From $\frac{dU}{dP} = 0$, we can obtain the optimal probability P^* that maximizes the overall throughput, i.e., $P^* = \frac{1}{N}$ [19]. We also note that the conditional collision probability given a link's attempt can be calculated as $P(collision|attempt) = 1 - (1 - P^*)^{N-1}$, which approaches $1 - \frac{1}{e}$ as N increases under the optimal control.

This result implies that in the single-hop network, the performance can be maximized by setting the attempt probability to $\frac{1}{\mathrm{N}}$ or by setting the conditional collision probability to $1 - \frac{1}{e}$. Motivated by this facts, we expect that the performance in the multi-hop network can be maximized too, by setting the target value of attempt probability or the conditional collision probability as shown in the single-hop network. Note that the former approach requires the information of the number of backlogged links, which can be obtained by explicit message passing between the neighboring links. Several works have tried to estimate the number of the neighboring links without explicit message exchanges in single-hop networks [17], [18], [19]. However, in multi-hop networks, the links will have a different estimate of the neighboring links depending on the topology and the traffic, and their control may not lead to good performance. In contrast, the conditional collision probability can be obtained from a link's own experience without control message passing. To this end, we develop our random access scheduling scheme, under which each link adjusts the attempt probability to maintain its conditional collision probability close to $1-\frac{1}{e}$.

Ⅲ-B. Additive Increase and Multiplicative Decrease algorithm

Unlike the single-hop wireless network, the links in multi-hop networks will experience different interference depending on their location in the network topology and on the traffic of their nearby links. In such an environment, the static scheme in Section Ⅲ-A cannot achieve the optimal throughput. For example, in a star topology network with N links, the central node will have $N-1$ neighboring links while the other nodes will have only one neighboring link. Thus if we set the attempt probability as in the single-hop network, the central node will have the attempt probability of $\frac{1}{N-1}$ and will suffer from poor throughput performance.

Instead of setting the attempt probability to $\frac{1}{|N_l|}$, where $|\cdot|$ denotes the set cardinality, we adaptively control the attempt probability aiming at the conditional collision probability of $1-\frac{1}{e}$. To elaborate, we adopt the Additive Increase Multiplicative Decrease (AIMD) algorithm for dynamic controls. The AIMD algorithm is a distributed algorithm that has been used to allocate resource in a fair manner, e.g., in TCP congestion control to adjust the congestion window size. The application of the AIMD algorithm to adaptive controls for the medium access is not new. There have been several studies to apply it to the medium access control based on the estimation of the number of neighboring links in single-hop networks [17], [18], [19]. In this work, we also use the AIMD algorithm but with a different flavor to support multi-hop networks.

We let each link modulate its attempt probability such that the probability increases linearly in time and decreases multiplicatively upon the occurrence of collision. Specifically, each link l updates the attempt probability P_l at each time slot as

$$P_l = \begin{cases} \dfrac{P_l}{\beta_l}, & \text{on a collision and } \hat{P}_l^c > 1-\dfrac{1}{e} \\ \max\{1, P_l+\alpha_l\}, & \text{otherwise} \end{cases} \tag{3}$$

where $\alpha_l (>0)$ and $\beta_l (>1)$ are two configuration parameters, and $\hat{P}_l^c$ denotes the (average) estimation on the conditional collision probability. Note that the collision can be detected by the failure of receiving ACK or CTS, and the term $\hat{P}_l^c$ consult the number of collisions occurred for actually estimated value. We refer to the above adaptive algorithm as Constant-AIMD (C-AIMD) and set $\beta_l = 2$ as in TCP. Since the AIMD algorithm has been known to operate in a reliable manner in a variety of network scenarios [10], we assume that the AIMD control converges to the steady state as in [19].

Once the system is in its steady state, each link will experience the same conditional collision probability $1 - \frac{1}{e}$, and thus, the expected probability of successful transmission $\bar{P}_l^s$ will be given as

$$\bar{P}_l^s = \frac{\bar{P}_l}{e}, \tag{4}$$

where $\bar{P}_l$ denotes average attempt probability of link l. Note that given any $\vec{\lambda} \in \Lambda$ (strictly inside), there exists a stationary static scheduling scheme ϕ_l that achieves $\phi_l \geq \lambda_l - \epsilon$ for some small $\epsilon > 0$ [20]. Eq. (2) implies that C-AIMD can achieve the fraction $\frac{1}{e}$ of the capacity region by setting $\bar{P}_l = \phi_l$, which, however, may require a priori knowledge of the arrival rate or the schedule distribution under an optimal scheduler.

In most practical scenarios, the information of the arrival rate and the schedule distribution of an optimal scheduler will not be available. Without such information, several random access scheduling schemes are shown to achieve the efficiency ratio up to $\frac{1}{\Delta}$ based on the queue length information of neighboring links [6], [7], [8]. For example, the Queue-Length based Constant-Time (QBCT) random access scheduler [7] controls the attempt probability of link l proportional to the ratio x_l of its queue length and the maximum sum of its neighboring links' queue lengths:

$$x_l := \frac{Q_l}{\max_{k \in N_l^+} \sum_{j \in N_k^+} Q_j}. \tag{5}$$

Note that in a simple two-link network, with links l and k, the following equation holds under QBCT:

$$\frac{\bar{P}_l}{Q_l} = \frac{\bar{P}_k}{Q_k}. \tag{6}$$

Inspired by this, we extend C-AIMD such that the attempt probabilities of the two neighboring links satisfy (3) as explained below. Note that the attempt probability of C-AIMD will have the saw-like behavior in the steady state as the TCP's congestion window control. Let us consider its *typical* movement, as shown in Fig. 2. Let P_l^{max} and P_l^{min} denote the peak and bottom attempt probability, respectively, in the typical movement. The mean attempt probability $\bar{P}_l$ will be $\frac{1}{2}(P_l^{max} + P_l^{min})$. Let T_l denote the cycle period of the movement and let X_l denote the average number of attempts of link l during one cycle time T_l.

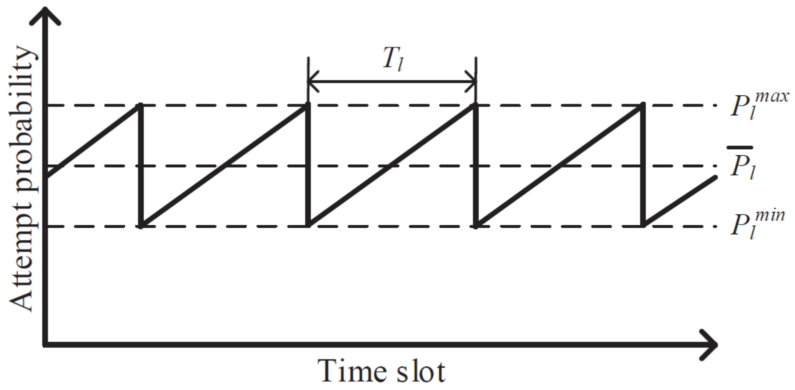


Fig. 2. Average behaviors of the attempt probability P_l in the steady state are shown. The probability follows the typical saw-like movement of AIMD in range $[P_l^{min}, P_l^{max}]$ with mean $\overline{P}_l$.

From the AIMD algorithm (1), we will have $P_l^{min} = \frac{P_l^{max}}{\beta_l}$, and thus

$$\overline{P}_l = \frac{1}{2}\left(P_l^{max} + P_l^{min}\right) = \frac{\beta_l + 1}{2\beta_l} \cdot P_l^{max}. \quad (7)$$

Again from the AIMD algorithm, the cycle period T_l can be obtained as

$$T_l = \frac{P_l^{max} - P_l^{min}}{\alpha_l} = \frac{2(\beta_l - 1)}{\alpha_l(\beta_l + 1)} \cdot \overline{P}_l. \quad (8)$$

Note that during one cycle period, link l will make $X_l = T_l \cdot \overline{P}_l$ attempts, and its conditioned collision probability can be written as

$$P(collision|attempt) = \frac{1}{X_l} = \frac{\alpha_l \beta_l (\beta_l + 1)}{(\beta_l - 1) \cdot 2\beta_l \cdot \overline{P}_l^2} = \frac{3\alpha_l}{2\overline{P}_l^2}\ , \quad (9)$$

where the last equality holds for $\beta = 2$. Since out C-AIMD algorithm will maintain the conditional collision probability close to $1 - \frac{1}{e}$, we can obtain that

$$\overline{P}_l = \sqrt{\frac{3}{2}\alpha_l \cdot \frac{e}{e-1}}\ . \quad (10)$$

The result implies that given $\beta = 2$, the steady-state attempt probability of C-AIMD can be controlled by changing the increasing rate α_l. Further, by setting

$$\alpha_l = C \cdot Q_l^2\ , \quad (11)$$

for some constant C, the scheduling scheme will satisfy the condition (3). We denote this extension of C-AIMD with (5) by Queue-length based AIMD (Q-AIMD). We emphasize that Q-AIMD also works without message passing since each link only needs its own queue length information. The setting of parameter C will be of interest for Q-AIMD. In Section Ⅴ, we show through simulations that Q-AIMD performs well for a wide range of C.

Ⅳ. MULTIPLE CONTENTION OPPORTUNITIES

In Section Ⅲ, we assumed that there is only one attempt (or contention) opportunity during the scheduling period. This can be extended by dividing the scheduling period into M mini-slots (numbered from 0 to $M-1$). In general, one can expect that as M increases, the collision probability decreases resulting in higher throughput performance, and on the other hand, the scheduling overhead increases degrading the performance in practice. In this section, we optimize the performance of random access scheduling schemes taking into account the multiple contention opportunities.

Exploiting the carrier-sensing or signal overhearing technique as in [6], [7], [8], we consider the following generic random access scheduling scheme with multiple mini-slots:

- At each mini-slot M, link l attempts transmission with probability $P_l(m)$ until link l itself or one of its neighbors makes an attempt.
- There are three possibilities:
  1) If a neighbor of link l makes an attempt (and link l does not), link l can overhear the attempt by using the carrier-sensing technique and will not attempt in the remaining mini-slots.
  2) If link l makes an attempt and all its neighbors do not, link l will transmit data during the transmission period.
  3) If link l and some of its neighbors make an attempt at the same mini-slot, they collide with each other and none of them can transmit data during the transmission period.

For Q-AIMD, we set the attempt probability at each mini-slot as

$$P_l = \frac{\mu}{M} y_l \ , \tag{12}$$

where $0 < \mu < M$ is a constant and y_l is under the AIMD control, i.e.,

$$y_l = \begin{cases} \dfrac{y_l}{\beta_l}, & on\ a\ collision\ and\ \hat{P}_l^c > 1 - \dfrac{1-\frac{\mu}{M}}{1-P_l} \\ \max\{1, y_l + \alpha_l\}, & otherwise. \end{cases} \tag{13}$$

As before, we rely on the stable operations of the AIMD algorithm [10], and assume that the AIMD algorithm successfully stabilizes the system to the steady state as in [19]. We focus on finding the optimal value of μ to maximize the system performance. We start with the following set Ω of the arrival rates.

$$\Omega = \left\{ \vec{\lambda}_l \middle| \sum_{k \in N_l^+} \lambda_k \leq \Delta \right\}. \tag{14}$$

Clearly, $\Lambda \subset \Omega$ since at most Δ links can be active simultaneously in N_l^+. The following lemma specifies a sufficient condition to achieve $\frac{Z}{\Delta}$ fraction of the capacity region for some Z.

Lemma 1: If a scheduling scheme has the successful transmission probability $\{P_l^s\}$, which satisfies

$$\sum_{k\in N_l^+} P_k^s \geq Z \quad \text{for all} \quad l, \tag{15}$$

for some constant Z, it achieves the efficiency ratio no smaller than $\frac{Z}{\Delta}$.

The lemma can be proven by showing that for any arrival rate $\vec{\lambda}$ strictly inside $\frac{Z}{\Delta}$ fraction of Ω, we have $\sum_{k\in N_l^+} \lambda_k - \sum_{k\in N_l^+} P_k^s < 0$ for all l, and thus the total queue length over N_l^+ will decrease. Since $\Lambda \subset \Omega$, the result follows. We omit the proof detail.

Proposition 2: For sufficiently large number M of mini-slots, Q-AIMD can achieve the efficiency ratio of $\frac{1}{\Delta}(1-e^{-\mu})$.

Proof: Based on Lemma 1, we can characterize the performance of Q-AIMD with multiple mini-slots by finding a Z such that $\sum_{k\in N_l^+} P_k^s \geq Z$ for all l under Q-AIMD. Note that from (6), the conditional collision probability satisfies that $1-\prod_{k\in N_l}(1-P_k) = 1-\frac{1-\frac{\mu}{M}}{1-P_k}$, which leads to

$$\prod_{k\in N_l^+}(1-P_k) = 1-\frac{\mu}{M} \ . \tag{16}$$

Then the probability P_l^s that link l successfully transmits data packet in a time slot can be calculated as $P_l^s = \sum_{m=0}^{M-1} P_l^s[m]$, where $P_l^s[m]$ denotes the probability that link l successfully attempts at mini-slot m, i.e.,

$$P_l^s[0] = P_l \cdot \prod_{k\in N_l}(1-P_k) \quad \text{,for} \ \ m=0 \ , \tag{17}$$

$$P_l^s[m] = P_l \cdot \prod_{k\in N_l}(1-P_k) \cdot \prod_{k\in N_l^+}(1-P_k)^m \quad \text{,for} \ \ m>0 \ . \tag{18}$$

Hence, we can obtain that

$$\begin{aligned} P_l^s &= \frac{P_l}{1-P_l}\sum_{m=0}^{M-1}\prod_{k\in N_l^+}(1-P_k)^{m+1} \\ &\geq P_l \sum_{m=0}^{M-1}\left(1-\frac{\mu}{M}\right)^{m+1} \\ &= \frac{\mu}{M} y_l \cdot \frac{\left(1-\frac{\mu}{M}\right)\left(1-\left(1-\frac{\mu}{M}\right)^M\right)}{1-\left(1-\frac{\mu}{M}\right)} \end{aligned}$$

$$= y_l \cdot \left(1 - \frac{\mu}{M}\right) \cdot \left(1 - \left(1 - \frac{\mu}{M}\right)^M\right), \tag{19}$$

where the inequality holds from (8) and $\frac{1}{1-P_l} \geq 1$. Thus for sufficiently large M, we will have

$$P_l^s \geq y_l \cdot (1 - e^{-\mu}) \ . \tag{20}$$

From (8) and the fact that $\prod_{k \in N_l^+}(1 - P_k) \geq 1 - \sum_{k \in N_l^+} P_k$, we can obtain that $\prod_{k \in N_l^+} y_k \geq 1$. Combining it with (9), we can obtain that

$$\sum_{k \in N_l^+} P_l^s \geq \sum_{k \in N_l^+} y_l (1 - e^{-\mu}) \geq 1 - e^{-\mu} \ . \tag{21}$$

Lemma 1 and (10) imply that Q-AIMD can achieve the efficiency ratio no smaller than $\frac{1-e^{-\mu}}{\Delta}$ for sufficiently large M.

By setting μ as an increasing function of M, e.g., $\mu = \log M$, the lower bound on the achievable efficiency ratio can be arbitrarily close to $\frac{1}{\Delta}$. For Q-AIMD, we set $\mu = 1$. We also consider our scheme with $\mu = \log M$, which is denoted by Q-AIMD+.

We highlight that Q-AIMD achieves the same performance bound as QBCT, while it does not require the neighboring links' queue length information and has significantly lower overhead in practice.

Ⅴ. SIMULATION

In this section, we numerically evaluate the performance of our scheduling schemes in both a grid network and a randomly generated network. We compare them with those of the state-of-the-art distributed scheduling schemes.

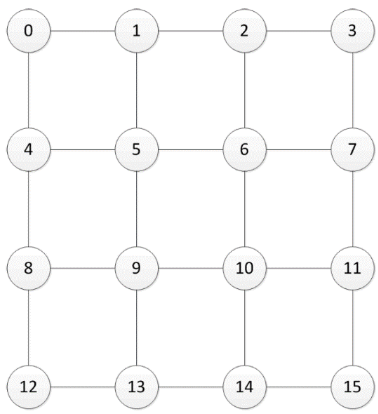


Fig. 3. Grid network topology.

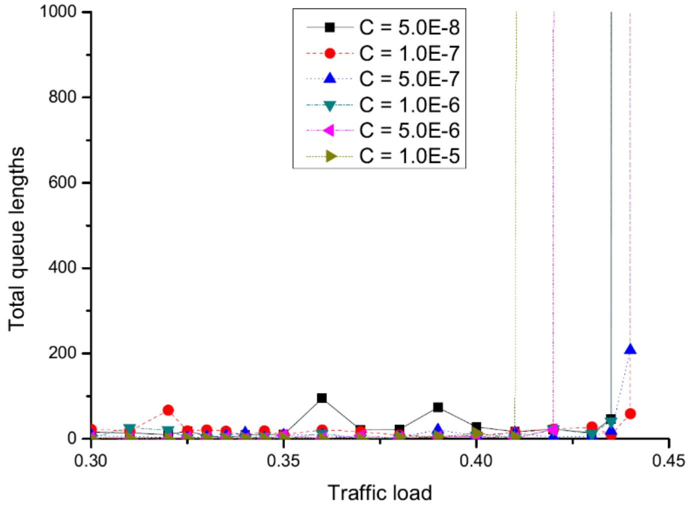


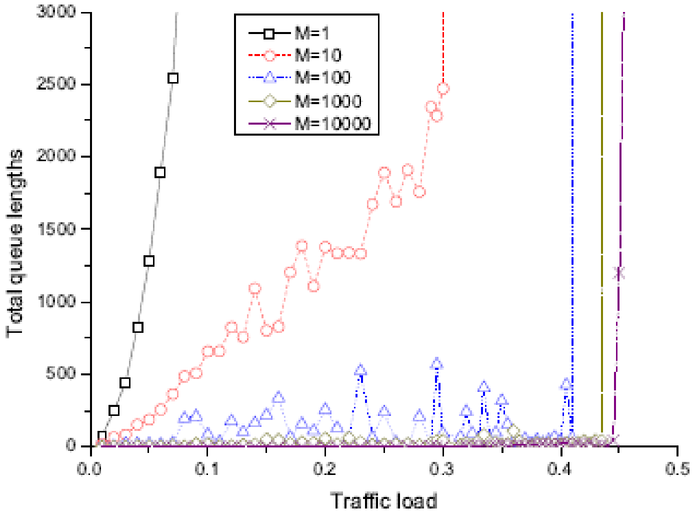
Fig. 4. Performance of Q-AIMD+ with different values of C.

We first consider a grid network with 16 nodes and 24 undirected links, as shown in Fig. 3. As assumed earlier, time is slotted. At each time slot, link l has an i.i.d. packet arrival with probability $\lambda_l = \rho \cdot r_l$, where r_l denotes a traffic vector coefficient randomly chosen among $\{0.2, 0.4, 0.6, 0.8\}$ and ρ denotes the traffic load. We assume that the links have a unit capacity and are constrained under the primary interference model, i.e., two links sharing a node cannot transmit at the same time.

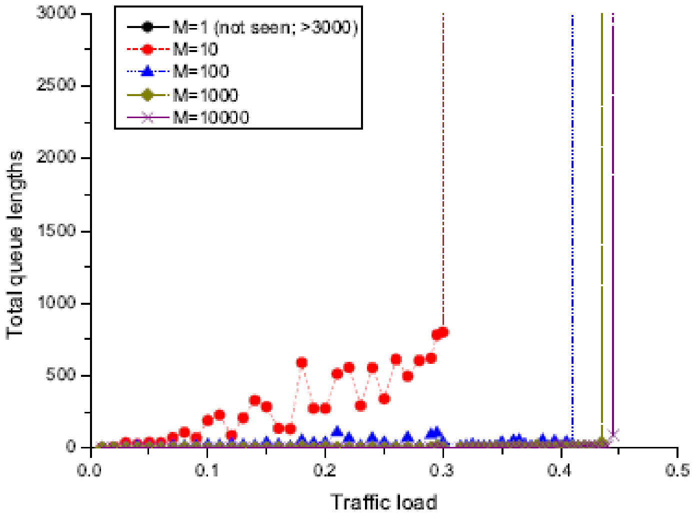
We run each simulation for 100,000 time slots, and measure the total queue lengths (summed over all the links) after termination, to estimate the throughput indirectly. By comparing the change amount of the queue lengths as the traffic load ρ varies, the limitation of acceptable arrival rate for each scheduling scheme can be known. When ρ is small, the scheduling schemes can support the arrival rates and keep the queue lengths small. As the traffic load ρ increases across the boundary of the achievable region, we will observe that the queue lengths soar sharply.

Fig. 4 shows the performance of Q-AIMD+ with different values of C, which controls the increase rate of the attempt probability as shown in (5). We fix $\beta_l = 2$ and change C in range $[5 \cdot 10^{-8}, 10^{-5}]$ to investigate the impact of aggressive increment of the attempt probability. The number of mini-slots is set to $M = 1000$. The results show that Q-AIMD+ performs best when $C \in [10^{-7}, 5 \cdot 10^{-7}]$, which implies that there is an optimal setting for the dynamics of AIMD. Finding the optimal parameters β and C for Q-AIMD+ is beyond the scope of this paper and remains as an interesting open problem. We use $C = 10^{-7}$ for our scheduling schemes in the sequel.

Since the number M of mini-slots denotes the length of the contention period, it can be easily expected that, as M increases, the probability of collision decreases and thus the performance of the scheduling schemes improves. Fig. 5 illustrates the results with different M under Q-AIMD and Q-AIMD+. The results show that for both Q-AIMD and Q-AIMD+, at least 100 mini-slots are necessary



(a) Q-AIMD



(b) Q-AIMD+

Fig. 5. Performance with different numbers M of mini-slots.

for reasonable performance, and the marginal performance gain decreases as M increases. Especially, when $M > 1000$, additional mini-slots lead to a marginal performance gain. The comparison between Q-AIMD and QAIMD+ shows that both have similar throughput performance in terms of the achievable region, Q-AIMD+ outperforms Q-AIMD in delay performance by maintaining queue lengths smaller (except when $M = 1$; Q-AIMD+ has very large queue lengths > 3000 even for a small ρ.).

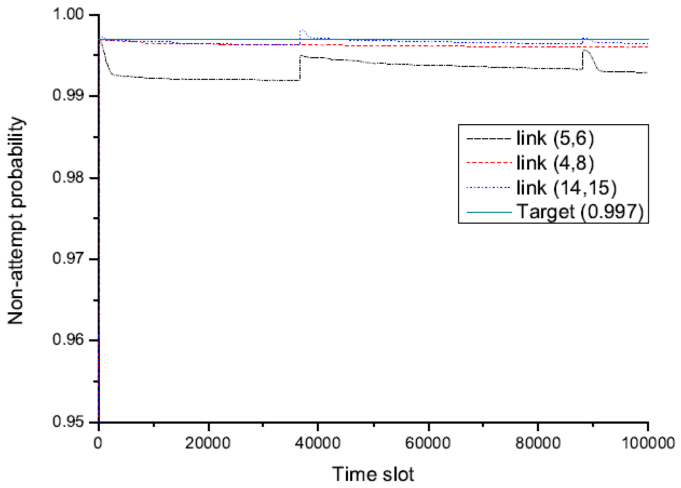


Fig. 6. Non-attempt probabilities of links under Q-AIMD+. By the AIMD control, the probabilities remains near the target value 0.997.

Fig. 6 shows the traces of the non-attempt probabilities of some links in the grid network, i.e., the measurements of $\prod_{k\in N_l^+}(1-P_k)$ for link $l=$ link $(5,6)$, link $(4,8)$, and link $(14,15)$ under Q-AIMD+ with $M=1000$ and $\rho=0.38$. It shows that they are close to the target $1-\frac{\mu}{M}=0.997$ as we intended in (8). Note that the number of neighboring links decreases in the order of links $(5,6),(4,8),(14,15)$, and the results imply that the link in the more crowded area is likely to have a smaller non-attempt probability.

We evaluate the performance of our schemes in comparison with other state-of-the-art distributed scheduling schemes, such as Greedy algorithm, QBCT [7] and Q-CSMA [3]. The Greedy (maximal) scheduling finds a maximal schedule in decreasing order of queue length conforming to the interference constraints, i.e., longest-queue-first. It can be implemented in a distributed manner with $O(|E|\log|E|)$ complexity, and empirically shown to achieve near-optimal performance [7]. Hence, we use it as the reference to the optimal performance. Q-CSMA is a distributed scheduling scheme that achieves the optimal throughput [3]. It carefully controls the transition of link activities using the carrier-sensing technique and maintains the stationary schedule distribution of an optimal solution. It requires control message passing during the scheduling period to manage the schedule transition in a distributed fashion. We omit the results of the static scheme in Section Ⅲ-A due to its low performance, but include the results of C-AIMD that sets all the links with identical parameter values of $\alpha_l=0.01$ and $\beta_l=2$. We set $M=1000$.

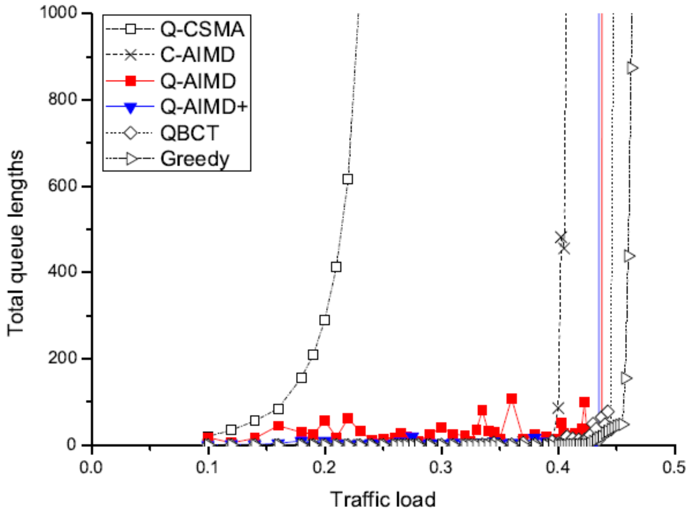


Fig. 7. Comparison with other scheduling schemes in the grid networks.

Fig. 7 shows the results. The Greedy scheduling scheme achieves the best performance up to traffic load 0.46, which is supposed to the bound of the capacity region. The performance of QBCT and our proposed schemes closely follows that of the Greedy scheduling scheme. We emphasize that the Greedy scheme is a centralized algorithm with higher complexity, and that both the Greedy scheme and QBCT require control message exchanges to collect the information of neighboring links. On the other hand, Q-AIMD and Q-AIMD+ work without such overhead. It is notable that Q-AIMD and Q-AIMD+ achieve the performance comparable to QBCT but without explicit control message exchanges. For Q-CSMA, although it has been shown to be throughput-optimal, it has empirical performance much lower than the others and suffers from poor delay performance. Further, Q-CSMA also needs control message exchanges in the process of its scheduling decision.

We also consider a network where 10 nodes are randomly deployed in a 6×6 area as shown in Fig. 8. Two nodes within distance 3 can communicate directly, which results in a total of 19 links. Fig. 9

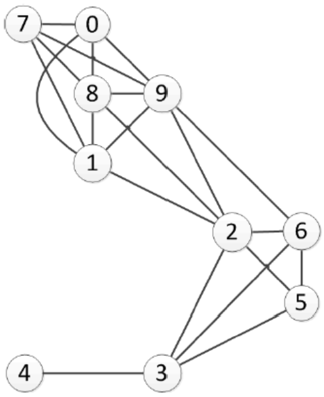


Fig. 8. Random network topology.

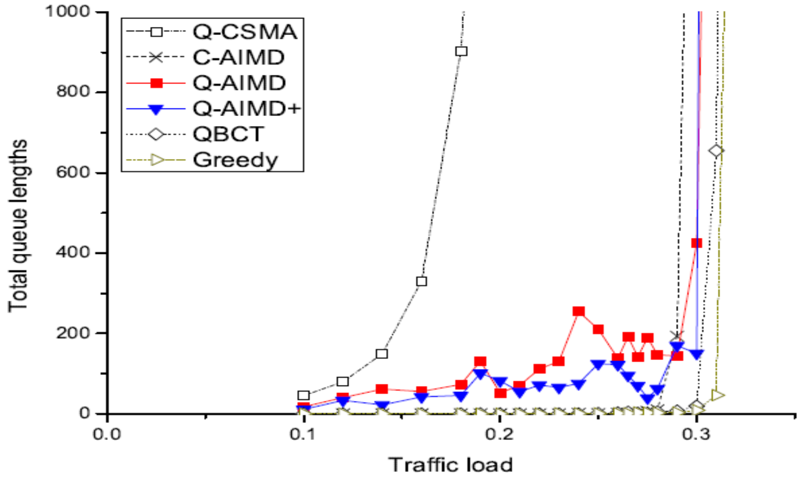


Fig. 10. Comparison of different scheduling schemes in a random network

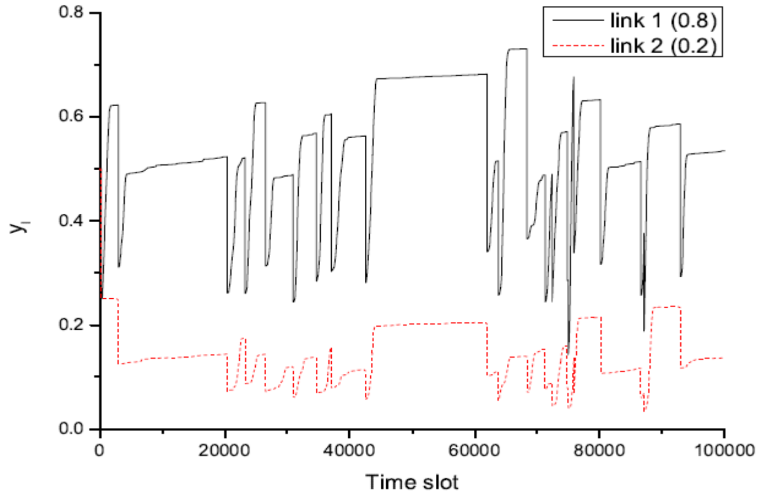


Fig. 9. Trace of the attempt probabilities of two links under Q-AIMD+, when the arrival rates are different.

depicts the performance of the scheduling schemes. The results are similar to those in the grid network.

Finally, we present the evolution of the attempt probability under Q-AIMD+. We consider a simple network that consists of two links, with arrival rates $\vec{\lambda} = \rho \cdot \vec{r}$, $\rho = 0.8$, and $\vec{r} = (0.8, 0.2)$. Fig. 10 show the trace of the attempt probabilities of the two links. They converge to the steady state by the AIMD algorithm, and show a typical saw-like movement as expected. Since the two links have different arrival rates, they have different average attempt probabilities, 0.534 and 0.145, respectively, and different average queue lengths, 12.948 and 9.746, respectively. The decrease of the attempt probabilities of the two links is synchronized since a collision will be observed by the both links. On the other hand, link 1 has higher attempt probabilities than link 2 since it has a larger increment step from the larger queue lengths.

Ⅵ. CONCLUSION

The scheduling problem attracts great attention due to its significance to find a set of simultaneous transmissions that result in efficient spectrum use without incurring severe interference in multi-hop wireless networks. In this paper, we develop novel scheduling schemes that graft the concept of AIMD and maintain a target collision probability. The proposed solution successfully removes the overhead incurred by the control message passing, and achieves good throughput performance comparable to the centralized GMS without a priori knowledge of the traffic demand. We evaluate our proposed schemes through simulations, whose results demonstrate that the achievable regions of our schemes are similar to those of the state-of-the-art distributed scheduling schemes that need a large amount of overhead for the control message passing.

REFERENCES

[1] Joo, Changhee, et al. "On the complexity of scheduling in wireless networks." EURASIP Journal on Wireless Communications and Networking 2010 (2010): 8.

[2] Jiang, Libin, and Jean Walrand. "A distributed CSMA algorithm for throughput and utility maximization in wireless networks." IEEE/ACM Transactions on Networking (TON) 18.3 (2010): 960-972.

[3] Ni, Jian, Bo Tan, and Rayadurgam Srikant. "Q-CSMA: queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks." Networking, IEEE/ACM Transactions on 20.3 (2012): 825-836.

[4] Choi, Jin-Ghoo, et al. "Distributed link scheduling under SINR model in multihop wireless networks." IEEE/ACM Transactions on Networking (TON) 22.4 (2014): 1204-1217.

[5] Tassiulas, Leandros, and Anthony Ephremides. "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks." Automatic Control, IEEE Transactions on 37.12 (1992): 1936-1948.

[6] Lin, Xiaojun, and Shahzada B. Rasool. "Constant-time distributed scheduling policies for ad hoc wireless networks." Automatic Control, IEEE Transactions on 54.2 (2009): 231-242.

[7] Joo, Changhee, and Ness B. Shroff. "Performance of random access scheduling schemes in multi-hop wireless networks." IEEE/ACM Transactions on Networking (TON) 17.5 (2009): 1481-1493.

[8] Gupta, Abhinav, Xiaojun Lin, and R. Srikant. "Low-complexity distributed scheduling algorithms for wireless networks." IEEE/ACM Transactions on Networking (TON) 17.6 (2009): 1846-1859.

[9] Joo, Changhee. "On random access scheduling for multimedia traffic in multihop wireless networks with fading channels." Mobile Computing, IEEE Transactions on 12.4 (2013): 647-656.

[10] Chiu, Dah-Ming, and Raj Jain. "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks." Computer Networks and ISDN systems 17.1 (1989): 1-14.

[11] Joo, Changhee, Xiaojun Lin, and Ness B. Shroff. "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks." IEEE/ACM Transactions on Networking (TON) 17.4 (2009): 1132-1145.

[12] Papadimitriou, Christos H., and Kenneth Steiglitz. Combinatorial optimization: algorithms and complexity. Courier Corporation, 1998.

[13] Bharghavan, Vaduvur, et al. "MACAW: a media access protocol for wireless LAN's." ACM SIGCOMM Computer Communication Review. Vol. 24. No. 4. ACM, 1994.

[14] Tassiulas, Leandros. "Linear complexity algorithms for maximum throughput in radio networks and input queued switches." INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Vol. 2. Ieee, 1998.

[15] Modiano, Eytan, Devavrat Shah, and Gil Zussman. "Maximizing throughput in wireless networks via gossiping." ACM SIGMETRICS Performance Evaluation Review. Vol. 34. No. 1. ACM, 2006.

[16] Sanghavi, Sujay, Loc Bui, and R. Srikant. "Distributed link scheduling with constant overhead." ACM SIGMETRICS Performance Evaluation Review. Vol. 35. No. 1. ACM, 2007.

[17] Cali, Federico, Marco Conti, and Enrico Gregori. "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism." Selected Areas in Communications, IEEE Journal on 18.9 (2000): 1774-1786.

[18] Tay, Y. C., Kyle Jamieson, and Hari Balakrishnan. "Collision-minimizing CSMA and its applications to wireless sensor networks." Selected Areas in Communications, IEEE Journal on 22.6 (2004): 1048-1057.

[19] Heusse, Martin, et al. "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs." ACM SIGCOMM Computer Communication Review. Vol. 35. No. 4. ACM, 2005.

[20] Stolyar, Alexander L. "Maxweight scheduling in a generalized switch: State space collapse and

workload minimization in heavy traffic." Annals of Applied Probability (2004): 1-53.

[21] Li, Ke, et al. "Additive-Increase Multiplicative-Decrease MAC Protocol with Multi-Packet Reception." Wired/Wireless Internet Communication. Springer Berlin Heidelberg, 2013. 15-28.

[22] Cai, Songlin, Yong Liu, and Weibo Gong. "Analysis of an AIMD based collision avoidance protocol in wireless data networks." Decision and Control, 2003. Proceedings. 42nd IEEE Conference on. Vol. 1. IEEE, 2003.

[23] Liu, Jiaping, et al. "Towards utility-optimal random access without message passing." Wireless Communications and Mobile Computing 10.1 (2010): 115-128.

[24] Huang, Jianwei, Mung Chiang, and Vincent WS Wong. "Utility-optimal random access without message passing." Wireless Communications, IEEE Transactions on 8.3 (2009): 1073-1079.

[25] Babich, F., and M. Comisso. "Optimum contention window for 802.11 networks adopting directional communications." Electronics Letters 44.16 (2008): 994-995.

[26] Ji, Baofeng, et al. "Efficient MAC protocol design and performance analysis for dense WLANs." Wireless Networks 20.8 (2014): 2237-2254.

[27] Jin, Hu, Jun-Bae Seo, and Victor Leung. "Cooperative Pseudo-Bayesian Backoff Algorithms for Unsaturated CSMA Systems with Multi-Packet Reception." Mobile Computing, IEEE Transactions on 14.2 (2015): 302-315.