Forty-Sixth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 23-26, 2008

A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks

Libin Jiang and Jean Walrand
EECS Department, University of California at Berkeley
{ljiang,wlr}@eecs.berkeley.edu

***Abstract*—In multi-hop wireless networks, designing distributed scheduling algorithms to achieve the maximal throughput is a challenging problem because of the complex interference constraints among different links. Traditional maximal-weight (MW) scheduling, although throughput-optimal, is difficult to implement in distributed networks; whereas a distributed greedy protocol similar to IEEE 802.11 does not guarantee the maximal throughput. In this paper, we introduce an adaptive CSMA scheduling algorithm that can achieve the maximal throughput distributedly under some assumptions. Major advantages of the algorithm include: (1) It applies to a very general interference model; (2) It is simple, distributed and asynchronous. Furthermore, we combine the algorithm with end-to-end flow control to achieve the optimal utility and fairness of competing flows. The effectiveness of the algorithm is verified by simulations. Finally, we consider some implementation issues in the setting of 802.11 networks.**

***Index Terms*—Cross-layer optimization, joint scheduling and congestion control, maximal throughput, CSMA**

I. Introduction

In multi-hop wireless networks, it is important to efficiently utilize the network resources and provide fairness to competing data flows. This objective requires the cooperation of different network layers. The transport layer needs to inject the right amount of traffic into the network based on the congestion level and the MAC layer needs to serve the traffic efficiently to achieve high throughput. Through a utility optimization framework [1], this problem can be naturally decomposed into rate control at the transport layer and scheduling at the MAC layer.

It turns out that MAC-layer scheduling is the bottleneck of the algorithm [1]. In particular, it is not easy to achieve the maximal throughput through distributed scheduling, which in turn prevents full utilization of the wireless network. Scheduling is challenging since the conflicting relationships between different links can be complicated.

It is well known that maximal-weight (MW) scheduling [12] is *throughput-optimal* (that is, it can support any incoming rates within the capacity region; or it can achieve the "*maximal throughput*"). In MW scheduling, time is assumed to be slotted. In each slot, a set of non-conflicting links (called an "Independent Set", or "IS") that have the maximal weight are scheduled, where the "weight" is the summation of the queue lengths of these non-conflicting links. (This algorithm has also been applied to achieve 100% throughput in input-queued switches [13].) However, finding such a maximal-weighted IS is NP-complete in general and is hard even for centralized algorithms. So its distributed implementation is not trivial in wireless networks.

A few recent works proposed throughput-optimal algorithms for certain interference models. For example, Eryilmaz et al. [2] proposed a polynomial-complexity algorithm for the "two-hop interference model"[1]. Modiano et al. [3] introduced a gossip algorithm for the "node-exclusive model"[2]. The extensions to more general interference models, as discussed in [2] and [3], usually involves extra challenges. Sanghavi et al. [4] introduced an algorithm that can approach the throughput capacity (with increasing overhead) for the node-exclusive model.

On the other hand, by using a distributed greedy protocol similar to IEEE 802.11, reference [7] shows that only a fraction of the throughput region can be achieved (after ignoring collisions). The size of the fraction depends on the network topology and interference relationships. Reference [8] studied the impact of such imperfect scheduling on utility maximization in wireless networks.

Our first contribution in this paper is to introduce a *distributed* adaptive CSMA (Carrier Sensing Multiple Access) algorithm for a general interference model. It is inspired by CSMA but may be applied to more general resource sharing problems (i.e., not limited to wireless networks). We show that if packet collisions are ignored (as in the above references), the algorithm can achieve maximal throughput, if the adaptation is slow enough[3]. The algorithm may not be directly comparable to the throughput-optimal algorithms mentioned above since it utilizes the carrier-sensing capability. But it does have a few distinct features:

- Each node only uses its local information (e.g., its backlog). No explicit control messages are required among the nodes.
- It is based on CSMA random access, which is similar

This work is supported by MURI grant BAA 07-036.18.

[1]In this model, a transmission over a link $(n; m)$ is successful iff none the one-hop neighbors of n and m is in any conversation at the time.

[2]In this model, a transmission over a link $(n; m)$ is successful iff neither n nor m is in another conversation at the time.

[3]However, the algorithm works well with a wide range of step sizes in our simulations.

978-1-4244-2926-4/08/$25.00 ©2008 IEEE

to the IEEE 802.11 protocol and is easy to implement.

- Time is not divided into synchronous slots. Thus no synchronization of transmissions is needed.

In a related work, Marbach et al. [9] studied a model of CSMA with collisions. It was shown that under the "node-exclusive" interference model, CSMA can be made asymptotically throughput-optimal in the limiting regime of large networks with a small sensing delay.

Our second contribution is to combine the proposed scheduling algorithm with end-to-end flow control using a novel technique, to achieve fairness among competing flows as well as maximal throughput (sections III, IV). The performance is evaluated by simulations (section V). Finally, we considered some practical issues (e.g., packet collisions) in the setting of 802.11 networks (section VI).

There is extensive research in joint MAC and transport-layer optimization, for example [5] and [6]. Their studies have assumed the slotted-Aloha random access protocol in the MAC layer, instead of the CSMA-like protocol we consider here. Other related works assume physical-layer models which are quite different from ours. For example, [10] considered CDMA interference model; and [11] considered time-varying wireless channel.

II. Adaptive CSMA for Maximal Throughput

A. Interference model

First we describe the general interference model we will consider in this paper. Assume there are K *links* in the network, where each *link* is an (ordered) transmitter-receiver pair. The network is associated with a link contention graph (or "LCG") $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of vertexes (each of them represents a link) and $\mathcal{E}$ is the set of edges. Two links cannot transmit at the same time (i.e., "conflict") iff there is an edge between them. Note that this framework includes the "node-exclusive model" and "two-hop interference model" mentioned above as two special cases.

Assume that G has N different Independent Sets ("IS", not confined to "Maximal Independent Sets"). Denote the i'th IS as $x^i \in \{0,1\}^K$, a 0-1 vector that indicates which links are transmitting in this IS. The k'th element of x^i, $x^i_k = 1$ if link k is transmitting, and $x^i_k = 0$ otherwise. We also refer to x^i as a "transmission state", and x^i_k as the "transmission state of link k".

B. An idealized CSMA protocol and the average throughput

We use an idealized model of CSMA as in [15][16]. In this subsection, assume that the links are always backlogged. If the transmitter of link k senses the transmission of any conflicting link (i.e., any link m such that $(k,m) \in \mathcal{E}$), then it keeps silent. If none of its conflicting links is transmitting, then the transmitter of link k waits (or backs-off) for a random period of time which is exponentially distributed with mean $1/R_k$ and then starts its transmission[4]. During the backoff if some conflicting link starts transmitting, then link k suspends its backoff and resumes it after the conflicting transmission is over. The transmission time of link k is exponentially distributed with mean 1. (The assumption on exponential distribution can be relaxed [16].) Assume that the sensing time is negligible (in particular, assume an infinite speed of light), then with the continuous distributions of the backoff time, the probability for two conflicting links to start transmission at the same time is 0. So in the model of [15][16], collisions are ignored. (In section VI, however, we will discuss adaptations of our algorithm which consider collisions in an 802.11 network.) Also, it is assumed that the carrier-sensing mechanism works well such that there is no hidden-terminal (HT) problem. In other words, any pair of conflicting links (in particular, their transmitters) can sense the transmission of each other. (This is possible if the range of carrier-sensing is large enough [17].[5])

There are a few reasons for using this model in our context, although it makes some simplifying assumptions about collisions and the HT problem: (1) It is simple, tractable, and captures the essence of CSMA/CA; (2) Even without considering collisions and hidden-terminals, distributed scheduling to achieve maximal throughput is not an easy problem, as discussed in the Introduction section. In this paper, we would like to focus on the scheduling problem, without mixing it too much with the other problems specific in wireless networks and trying to solve them at once. Similar approaches have been taken in related works, for example [7], [1]; (3) The scheduling algorithm we propose here is inspired by CSMA, but it may be applied to more general resource sharing problems (i.e., not limited to wireless networks).

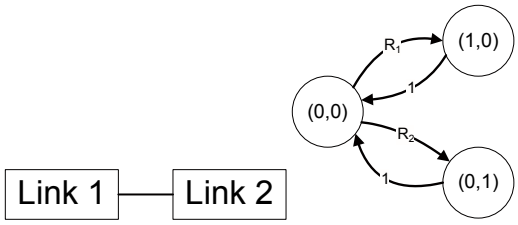
It is not difficult to see that the transitions of the transmission states form a Continuous Time Markov Chain, which is called *CSMA Markov Chain*. Denote link k's neighboring set $\mathcal{N}(k) := \{m : (k,m) \in \mathcal{E}\}$. If in state x^i, link k is not active ($x^i_k = 0$) and all of its conflicting links are not active (i.e., $x^i_m = 0, \forall m \in \mathcal{N}(k)$), then state x^i transits to state $x^i + \mathbf{e}_k$ with a rate of R_k, where $\mathbf{e}_k$ is the K-dimension vector whose k'th element is 1 and all other elements are 0's. Similarly, state $x^i + \mathbf{e}_k$ transits to state x^i with a rate of 1. However, if in state x^i, any link in its neighboring set $\mathcal{N}(k)$ is active, then state $x^i + \mathbf{e}_k$ does not exist.

Fig 1 gives an example network whose LCG is shown in (a). There are two links, with an edge between them, which means that they cannot transmit together. Fig 1 (b) shows the corresponding CSMA Markov Chain. State (0,0) means that no link is transmitting, state (1,0) means that only link 1 is transmitting, and (0,1) means that only link 2 is transmitting. The state (1,1) is not feasible.

Denote $r_k = \log(R_k)$. We call r_k the "transmission

[4]If more than one backlogged links share the same transmitter, the transmitter maintains independent backoff timers for these links.

[5]A related problem that affects the performance of wireless networks is the exposed-terminal (ET) problem. Reference [17] proposed a protocol to address HT and ET problems in a systematic way. We assume in this paper that the HT and ET are negligible with the use of such a protocol. Note that however, although ET problem may reduce the capacity region, it does not affect the applicability of our model, since we can define an edge between two links in the LCG as long as they can sense the transmission of each other, even if this results in ET.



(a) Link contention graph (b) Markov Chain

Fig. 1. Example: link contention graph and corresponding Markov Chain.

aggressiveness" ("TA") of link k. The stationary distribution of any feasible state x^i in the Markov Chain is

$$p(x^i;\mathbf{r}) = \frac{\exp(\sum_{k=1}^{K} x_k^i r_k)}{C(\mathbf{r})} \tag{1}$$

where

$$C(\mathbf{r}) = \sum_j \exp(\sum_{k=1}^{K} x_k^j r_k) \ . \tag{2}$$

Note that the summation "$\sum_j$" is over all feasible states (or IS's), and the vector $\mathbf{r} = (r_1, r_2, \cdots, r_K)$. (Later, we also write $p(x^i;\mathbf{r})$ as $p_i(\mathbf{r})$ for simplicity. These notations are interchangeable throughout the paper.) For example, in Fig 1, the probabilities of state (0,0), (1,0) and (0,1) are $1/(1+R_1+R_2)$, $R_1/(1+R_1+R_2)$ and $R_2/(1+R_1+R_2)$ in the stationary distribution.

Indeed, we can verify the detailed balance equation is satisfied [14]. Consider state x^i (where $x_k^i = 0$ and $x_m^i = 0, \forall m \in \mathcal{N}(k)$) and $x^i + \mathbf{e}_k$ again. From (1), we have

$$\frac{p(x^i + \mathbf{e}_k;\mathbf{r})}{p(x^i;\mathbf{r})} = \exp(r_k) = R_k$$

which is exactly the detailed balance equation between state x^i and $x^i + \mathbf{e}_k$. Similar relations hold for any two states that differ in only one element. And all infeasible states have probability zero.

Also, $\sum_i p(x^i;\mathbf{r}) = 1$. Therefore (1) is the stationary distribution of the Markov Chain given $\mathbf{r}$. Furthermore, the Markov Chain is time-reversible since the detailed balance equations hold. In fact, the Markov chain is a reversible "spatial process" and its stationary distribution (1) is a Markov Random Field ([14], page 189). (This means that for each link k, given the transmission states of its conflicting links, the state of link k is conditionally independent of all other links.)

Consequently, the *normalized* throughput (or service rate) of link k is

$$s_k(\mathbf{r}) = \sum_i x_k^i \cdot p(x^i;\mathbf{r}) \ . \tag{3}$$

Even if the distributions of the waiting time and transmission time are not exponential distributed but have the same means ($1/R_k$ and 1), reference [16] shows that the stationary distribution (1) still holds. That is, the stationary distribution is insensitive.

C. Adaptive CSMA for maximal throughput

Assume i.i.d. traffic arrival at each link k with a *normalized* arrival rate λ_k. And denote the vector of arrival rates as $\lambda \in R_+^K$. Without loss of generality, assume that $\lambda_k > 0, \forall k$. (The link(s) with zero arrival rate can be removed from the problem.) We say that λ is *feasible* if and only if $\lambda = \sum_i \bar{p}_i \cdot x^i$ for some probability distribution $\bar{\mathbf{p}} \in \mathcal{R}_+^N$ satisfying $\bar{p}_i \geq 0$ and $\sum_i \bar{p}_i = 1$. That is, λ is a convex combination of the IS's, such that it is possible to serve the arriving traffic with some transmission schedule. We say that λ is *strictly feasible* iff it is in the interior of the capacity region, i.e., iff it can be written as $\lambda = \sum_i \bar{p}_i \cdot x^i$ where $\bar{p}_i > 0$ and $\sum_i \bar{p}_i = 1$. Denote the set of strictly feasible λ as $\mathcal{C}$.

Define the following function (the "log likelihood function" if we estimate the parameter $\mathbf{r}$ with the observation $\bar{p}_i$)

$$\begin{aligned} F(\mathbf{r}) &:= \sum_i \bar{p}_i \log(p_i(\mathbf{r})) \\ &= \sum_i \bar{p}_i [\sum_{k=1}^{K} x_k^i r_k - \log(C(\mathbf{r}))] \\ &= \sum_k \lambda_k r_k - \log(\sum_j \exp(\sum_{k=1}^{K} x_k^j r_k)) \end{aligned}$$

where $\lambda_k = \sum_i \bar{p}_i x_k^i$ is the traffic arrival rate at link k.

Consider the following optimization problem

$$\sup_{\mathbf{r}\geq 0} F(\mathbf{r}) \ . \tag{4}$$

Since $\log(p(x^i;\mathbf{r})) \leq 0$, we have $F(\mathbf{r}) \leq 0$. Therefore $\sup_{\mathbf{r}\geq 0} F(\mathbf{r})$ exists. Also, $F(\mathbf{r})$ is concave in $\mathbf{r}$ [18]. We show that the following proposition holds.

Proposition 1: If $\sup_{\mathbf{r}\geq 0} F(\mathbf{r})$ is attainable (i.e., there exists *finite* $\mathbf{r}^* \geq 0$ such that $F(\mathbf{r}^*) = \sup_{\mathbf{r}\geq 0} F(\mathbf{r})$), then $s_k(\mathbf{r}^*) \geq \lambda_k, \forall k$. That is, the service rate is not less than the arrival rate when $\mathbf{r} = \mathbf{r}^*$.

Proof: Let $\mathbf{d} \geq 0$ be a vector of dual variables associated with the constraints $\mathbf{r} \geq 0$ in problem (4), then the Lagrangian is $\mathcal{L}(\mathbf{r};\mathbf{d}) = F(\mathbf{r}) + \mathbf{d}^T\mathbf{r}$. At the optimal solution $\mathbf{r}^*$, we have

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{r}^*;\mathbf{d}^*)}{\partial r_k} &= \lambda_k - \frac{\sum_j x_k^j \exp(\sum_{k=1}^{K} x_k^j r_k^*)}{C(\mathbf{r}^*)} + d_k^* \\ &= \lambda_k - s_k(\mathbf{r}^*) + d_k^* = 0 \end{aligned} \tag{5}$$

where $s_k(\mathbf{r})$, according to (3), is the service rate (at stationary distribution) given $\mathbf{r}$. Since $d_k^* \geq 0$, $\lambda_k \leq s_k(\mathbf{r}^*)$. ■

The following condition, proved in the Appendix, ensures that $\sup_{\mathbf{r}\geq 0} F(\mathbf{r})$ is attainable.

Proposition 2: If the arrival rate λ is strictly feasible, then $\sup_{\mathbf{r}\geq 0} F(\mathbf{r})$ is attainable.

Combining Proposition 1 and 2, we know that for any strictly feasible λ, there exists a finite $\mathbf{r}^*$ such that $s_k(\mathbf{r}^*) \geq \lambda_k, \forall k$. To see why "strict feasibility" is necessary, consider the network in Fig. 1. If $\lambda_1 = \lambda_2 = 0.5$ (not strictly feasible), then only when $r_1 = r_2 \to \infty$, the service rates $s_1(\mathbf{r}) = s_2(\mathbf{r}) \to 0.5$ but cannot reach 0.5.

Since $\partial F(\mathbf{r})/\partial r_k = \lambda_k - s_k(\mathbf{r})$, a simple gradient algorithm to solve (4) is

$$r_k(t+1) = [r_k(t) + \alpha(t) \cdot (\lambda_k - s_k(\mathbf{r}(t)))]_+ \tag{6}$$

where $\alpha(t)$ is some (small) step sizes. The algorithm is easy for *distributed* implementation in wireless networks, because

link k can adjust r_k based on its *local information*: arrival rate λ_k and service rate $s_k(\mathbf{r}(t))$. (If the arrival rate is larger than the service rate, then r_k should be increased, and vice versa.) Note that however, the arrival and service rates are generally random variables in actual networks, unlike in (6).

Let link k adjust r_k every b time units. So t is incremented by 1 every b time units. For convenience, assume that at link k, the arrived traffic between moment $t-1$ and t is stored in a temporary buffer, and is added to the queue at moment t. Let $\lambda'_k(t)$ be the average arrival rate between moment $t-1$ and t, and let $s'_k(t)$ be the average service rate between moment t and $t+1$. Then the dynamics of Q_k (the queue length at the *transmitter* of link k) is

$$Q_k(t+1) = [Q_k(t) + b \cdot (\lambda'_k(t) - s'_k(t))]_+ \tag{7}$$

where $\lambda'_k(t)$ and $s'_k(t)$ are generally random variables. We design the following distributed algorithm

Algorithm 1: Adjust the TA (transmission aggressiveness) in CSMA

$$r_k(t+1) = [r_k(t) + \alpha \cdot (\lambda'_k(t) - s'_k(t))]_+ \tag{8}$$

where α is a small constant step size. Let $r_k = 0$ when $Q_k = 0$. Then, by (8) and (7), Algorithm 1 is simply

$$r_k(t) = \alpha/b \cdot Q_k(t). \tag{9}$$

We can see that if $\mathbf{r}$ is stable (i.e., *does not go to infinity*), then the queues are also stable (which means that the arriving traffic can be served). Consider the following two cases (both mean slow changes of $\mathbf{r}$):

(1) If b is very large (but finite), then as the CSMA Markov Chain converges, $s'_k(t) \approx s_k(\mathbf{r}(t))$.[6] Also, $\lambda'_k(t) \approx \lambda_k$. Then (8) is a gradient algorithm to solve (4). Since the step size is constant, $\mathbf{r}$ may not converge to $\mathbf{r}^*$, but to a neighborhood of $\mathbf{r}^*$ if α is small enough. This is not an issue since we only require that $\mathbf{r}$ (and the queues) does not go to infinity.

(2) If b is of typical length but α is very small, then vector $\mathbf{r}$ (and the stationary distribution $p_i(\mathbf{r}), \forall i$) changes slowly. Assume that the distribution of the transmission states can "track" the slowly-varying stationary distribution, i.e., $E[s'_k(t)] = s_k(\mathbf{r}(t))$, then algorithm (8) is a stochastic gradient algorithm (with constant step-size) [20] [19], which can stabilize $\mathbf{r}$ (and the queues) if α is small enough.

In practice, on the other hand, the above choices may not be preferable since they slow down the system and reduce its responsiveness to variations of queue lengths. In case (1), it may take a long time for the CSMA Markov chain to converge, especially in large networks. So a more practical approach is to adjust $\mathbf{r}$ faster, without waiting for the convergence of the Markov chain in each iteration. Our simulations show good performance, suggesting that very slow adaptation is not necessary (although sufficient) for maximal throughput. However, identifying the exact conditions on the step size to ensure stability is a challenging problem and deserves future research.

[6]A subtle point: If between moment t and $t+1$, the queue of link k' becomes empty, then link k' can continue to transmit dummy packets with $r_{k'}(t)$ until $t+1$. This ensures that the average service rate is still $s_k(\mathbf{r}(t))$ for all k.

III. The Primal-dual relationship

In the previous section we have described the adaptive CSMA algorithm to support any strictly-feasible arrival rates. For joint scheduling and flow control, however, directly using the above expression of service rate (3) will lead to a non-convex problem. This section gives another look at the problem and also helps to avoid the difficulty.

Rewrite (4) as

$$\begin{array}{ll} \max_{\mathbf{r},\mathbf{z}} & \{\sum_k \lambda_k r_k - \log(\sum_j \exp(z_j))\} \\ \text{s.t.} & z_j = \sum_{k=1}^K x_k^j r_k, \forall j \\ & r_k \geq 0, \forall k. \end{array} \tag{10}$$

For each $j = 1, 2, \ldots, N$, associate a dual variable u_j to the constraint $z_j = \sum_{k=1}^K x_k^j r_k$. Write the vector of dual variables as $\mathbf{u} \in \mathcal{R}_+^N$. Then it is not difficult to find the dual problem of (10) as follows. (We omit the computation here due to the limit of space.)

$$\begin{array}{ll} \max_{\mathbf{u}} & -\sum_i u_i \log(u_i) \\ \text{s.t.} & \sum_i (u_i \cdot x_k^i) \geq \lambda_k, \forall k \\ & u_i \geq 0, \sum_i u_i = 1. \end{array} \tag{11}$$

where the objective function is the entropy of the distribution $\mathbf{u}$, $H(\mathbf{u}) := -\sum_i u_i \log(u_i)$. [7]

Also, if for each k, we associate a dual variable r_k to the constraint $\sum_i (u_i \cdot x_k^i) \geq \lambda_k$ in problem (11), then one can compute that the dual problem of (11) is the original problem $\max_{\mathbf{r} \geq \mathbf{0}} F(\mathbf{r})$ (This is shown in the Appendix as a by-product of the proof of Proposition 2). This is not surprising, since in convex optimization, the dual problem of dual problem is often the original problem.

What is interesting is that both $\mathbf{r}$ and $\mathbf{u}$ have concrete physical meanings. We have seen that r_k is the TA of link k. Also, u_i can be regarded as the stationary probability of state i in the CSMA Markov Chain given the dual variable $\mathbf{r}$. This observation will be useful in later sections. A convenient way to guess this is by observing the constraint $\sum_i (u_i \cdot x_k^i) \geq \lambda_k$. If u_i is the probability of state i, then the constraint simply means that the service rate of link k, $\sum_i (u_i \cdot x_k^i)$, is larger than the arrival rate.

Proposition 3: Given some (finite) TA's of the links (that is, given the dual variable $\mathbf{r}$ of problem (11)), the stationary distribution of the CSMA Markov Chain maximizes the partial Lagrangian $\mathcal{L}(\mathbf{u}; \mathbf{r}) = -\sum_i u_i \log(u_i) + \sum_k r_k (\sum_i u_i \cdot x_k^i - \lambda_k)$ over all possible distributions $\mathbf{u}$. Also, Algorithm 1 can be viewed as a (stochastic) subgradient algorithm to update the dual variable $\mathbf{r}$ in order to solve problem (11).

Proof: Given some finite dual variables $\mathbf{r}$, a partial Lagrangian of problem (11) is

$$\mathcal{L}(\mathbf{u}; \mathbf{r}) = -\sum_i u_i \log(u_i) + \sum_k r_k (\sum_i u_i \cdot x_k^i - \lambda_k).$$

Denote $\mathbf{u}^*(\mathbf{r}) = \arg\max_{\mathbf{u}} \mathcal{L}(\mathbf{u}; \mathbf{r})$, where $\mathbf{u}$ is a distribution. Since $\sum_i u_i = 1$, if we can find some w, and $\mathbf{u}^*(\mathbf{r}) > 0$

[7]In fact, there is a more general relationship between ML estimation problem such as (4) and Maximal-Entropy problem such as (11) [21].

(i.e., in the interior of the feasible region) such that

$$\frac{\partial \mathcal{L}(\mathbf{u}^*(\mathbf{r});\mathbf{r})}{\partial u_i} = -\log(u_i^*(\mathbf{r})) - 1 + \sum_k r_k x_k^i = w, \forall i,$$

then $\mathbf{u}^*(\mathbf{r})$ is the desired distribution. The above conditions are

$$u_i^*(\mathbf{r}) = \exp(\sum_k r_k x_k^i - w - 1), \forall i. \text{ and } \sum_i u_i^*(\mathbf{r}) = 1.$$

By solving the two equations, we find that $w = \log[\sum_j \exp(\sum_k r_k x_k^j)] - 1$ and

$$u_i^*(\mathbf{r}) = \frac{\exp(\sum_k r_k x_k^i)}{\sum_j \exp(\sum_k r_k x_k^j)}, \forall i \tag{12}$$

satisfy the conditions.

Note that in (12), $u_i^*(\mathbf{r})$ is exactly the stationary probability of state i in the CSMA Markov Chain given the TA of all links. So Algorithm 1 can be viewed as a stochastic subgradient algorithm to search for the optimal dual variable. Indeed, given $\mathbf{r}$, $u_i^*(\mathbf{r})$ maximizes $\mathcal{L}(\mathbf{u};\mathbf{r})$; then, $\mathbf{r}$ can be updated by the subgradient algorithm $r_k \leftarrow [r_k + \alpha(\lambda_k - \sum_i u_i^*(\mathbf{r}) x_k^i)]_+$, which is the deterministic version of Algorithm 1. The whole system is trying to solve problem (11) or (4). ■

IV. Joint Scheduling and Rate Control

Now, we combine end-to-end rate control with the CSMA scheduling algorithm to achieve fairness among competing flows as well as maximal throughput. Here, the input rates are distributedly adjusted by the source of each flow.

A. Formulation

Assume there are M flows, and let m be the index ($m = 1, 2, \ldots, M$). Define $a_{mk} = 1$ if flow m uses link k, and $a_{mk} = 0$ otherwise. Let f_m be the rate of flow m, and $v_m(f_m)$ be the "utility function" of this flow, which is assumed to be increasing and strictly concave. Assume all links have the same PHY data rates (it is easy to extend the algorithm to different PHY rates).

Assume that each link k maintains a separate queue for each flow that traverses it. Then, the service rate of flow m by link k, denoted by s_{km}, should be no less than the incoming rate of flow m to link k. For flow m, if link k is its first link (i.e., the source link), we say $\delta(m) = k$. In this case, the constraint is $s_{km} \geq f_m$. If $k \neq \delta(m)$, denote flow m's upstream link of link k by $up(k,m)$, then the constraint is $s_{km} \geq s_{up(k,m),m}$, where $s_{up(k,m),m}$ is equal to the incoming rate of flow m to link k. We also have $\sum_i u_i \cdot x_k^i = \sum_{m:a_{mk}=1} s_{km}, \forall k$, i.e., the total service rate of link k is divided among the flows.

Then, consider the following optimization problem:

$$\begin{array}{ll} \max_{\mathbf{u},\mathbf{s},\mathbf{f}} & -\sum_i u_i \log(u_i) + \sum_{m=1}^M v_m(f_m) \\ \text{s.t.} & s_{km} \geq 0, \forall k, m : a_{mk} = 1 \\ & s_{km} \geq s_{up(k,m),m}, \forall m, k : a_{mk} = 1, k \neq \delta(m) \\ & s_{km} \geq f_m, \forall m, k : k = \delta(m) \\ & \sum_i u_i \cdot x_k^i = \sum_{m:a_{mk}=1} s_{km}, \forall k \\ & u_i \geq 0, \sum_i u_i = 1. \end{array} \tag{13}$$

Notice that the objective function is not exactly the total utility, but it has an extra term $-\sum_i u_i \log(u_i)$. In section IV-B, we will introduce a method to approach the maximal utility. Associate dual variables $q_{km} \geq 0$ to the 2nd and 3rd lines of constraints of (13). Then a partial Lagrangian (subject to $s_{km} \geq 0$, $\sum_i u_i \cdot x_k^i = \sum_{m:a_{mk}=1} s_{km}$ and $u_i \geq 0, \sum_i u_i = 1$) is

$$\begin{array}{rl} & \mathcal{L}(\mathbf{u},\mathbf{s},\mathbf{f};\mathbf{q}) \\ = & -\sum_i u_i \log(u_i) + \sum_{m=1}^M v_m(f_m) \\ & + \sum_{m,k:a_{mk}=1,k\neq\delta(m)} q_{km}(s_{km} - s_{up(k,m),m}) \\ & + \sum_{m,k:,k=\delta(m)} q_{km}(s_{km} - f_m) \\ = & -\sum_i u_i \log(u_i) \\ & + \sum_{m=1}^M v_m(f_m) - \sum_{m,k:k=\delta(m)} q_{km} f_m \\ & + \sum_{k,m:a_{mk}=1} s_{km}[(q_{km} - q_{down(k,m),m})] \end{array} \tag{14}$$

where $down(k,m)$ means flow m's downstream link of link k (Note that $down(up(k,m),m) = k$). If k is the last link of flow m, then define $q_{down(k,m),m} = 0$.

Fix the vectors $\mathbf{u}$ and $\mathbf{q}$ first, we solve for s_{km} in the sub-problem

$$\begin{array}{ll} \max_{\mathbf{s}} & \sum_{k,m:a_{mk}=1} s_{km}[(q_{km} - q_{down(k,m),m})] \\ \text{s.t.} & s_{km} \geq 0, \forall k, m : a_{mk} = 1 \\ & \sum_{m:a_{mk}=1} s_{km} = \sum_i (u_i \cdot x_k^i), \forall k. \end{array} \tag{15}$$

The solution is easy to find: at link k, for an $m' \in \arg\max_{m:a_{mk}=1}(q_{km} - q_{down(k,m),m})$, let $s_{km'} = \sum_i u_i \cdot x_k^i$; and let $s_{km} = 0, \forall m \neq m'$. In other words, each link schedules a flow with the maximal "back-pressure" $q_{km} - q_{down(k,m),m}$. (This is similar to [1] and related references therein.) Since the value of $q_{down(k,m),m}$ can be obtained from a one-hop neighbor, this algorithm is distributed.

Plug the solution of (15) back into (14), we get

$$\begin{array}{rl} \mathcal{L}(\mathbf{u},\mathbf{f};\mathbf{q}) & = [-\sum_{i=1} u_i \log(u_i) + \sum_k z_k(\sum_i u_i \cdot x_k^i)] \\ & \quad + [\sum_{m=1}^M v_m(f_m) - \sum_{m,k:k=\delta(m)} q_{km} f_m] \end{array}$$

where $z_k := \max_m(q_{km} - q_{down(k,m),m})$ is the maximal back-pressure at link k. So a distributed algorithm to solve (13) is

Algorithm 2: Joint scheduling and rate control

Initially, assume that all queues are empty, and set $q_{km} = 0, \forall k, m$. Then iterate:

- Link k transmits the head-of-line packet from a flow with the maximal back-pressure $z_k = \max_{m:a_{mk}=1}(q_{km} - q_{down(k,m),m})$ when it gets the opportunity to transmit.
- Link k lets $r_k = z_k$ in the CSMA operation. This is because given $\mathbf{z}$, the optimal $\mathbf{u}$ (that maximizes $\mathcal{L}(\mathbf{u},\mathbf{f};\mathbf{q})$ over $\mathbf{u}$) is the stationary distribution of the CSMA Markov Chain with $r_k = z_k$ (similar to the proof of Proposition 3)[8].
- Rate control: For each flow m, if link k is its source link, then the transmitter of link k let $f_m = \arg\max_{f'_m}\{\beta \cdot v_m(f'_m) - q_{km} f'_m\}$. This maximizes $\mathcal{L}(\mathbf{u},\mathbf{f};\mathbf{q})$ over $\mathbf{f}$.

[8]Similar to section II-C, if we do not wait for the convergence of the CSMA Markov Chain in each iteration, then the resulting service rates $\mathbf{s}$ are random variables in Item 4 of Algorithm 2.

- The dual variables q_{km} (maintained by the transmitter of each link) are updated by a sub-gradient algorithm: $q_{km} \leftarrow [q_{km} + \alpha(s_{up(k,m),m} - s_{km})]_+$ if $k \neq \delta(m)$; and $q_{km} \leftarrow [q_{km} + \alpha(f_m - s_{km})]_+$ if $k = \delta(m)$. Note that by doing this, $q_{km} \propto Q_{km}$, where Q_{km} is the queue length of flow m at link k.

Remark: Using similar derivations, the adaptive CSMA algorithm can be combined with optimal routing, anycast or multicast. So it is a modular MAC-layer protocol which can work with other algorithms in transport layer and network layer.

B. Approaching the maximal utility

Define $V_m(f_m) := \beta \cdot v_m(f_m)$, where $\beta > 0$ is a weighting factor. And we use the above algorithm to solve

$$\max_{\mathbf{u},\mathbf{s},\mathbf{f}} \{-\textstyle\sum_i u_i \log(u_i) + \sum_m V_m(f_m)\} \tag{16}$$

subject to the same constraints as in (13). Assume that when the optimum is achieved, the flow rates $\mathbf{f} = \hat{\mathbf{f}}$, and $\mathbf{u} = \hat{\mathbf{u}}$.

Notice that $-\sum_i u_i \log(u_i)$, the entropy of the distribution $\mathbf{u}$, is bounded. Since there are $N \leq 2^K$ possible states, then, $0 \leq -\sum_i u_i \log(u_i) \leq \log N \leq \log 2^K = K \cdot \log 2$. So when β is large, the "importance" of the total utility dominates the objective function of (16). (This is similar in spirit to the weighting factor used in [11].) As a result, the solution of (16) approximately achieves the maximal utility. Denote the highest total utility achievable as $\bar{W}$, i.e.,

$$\bar{W} := \max_{\mathbf{u},\mathbf{s},\mathbf{f}} \textstyle\sum_m v_m(f_m) \tag{17}$$

subject to the same constraints as in (13). It is not difficult to show the following bound [25].

Proposition 4: The difference between the total utility ($\sum_{m=1}^{M} v_m(\hat{f}_m)$) resulting from solving (16) and the maximal total utility $\bar{W}$ is bounded. The bound of difference decreases with the increase of β. In particular,

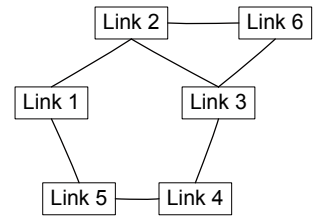
$$\bar{W} - (K \cdot \log 2)/\beta \leq \textstyle\sum_m v_m(\hat{f}_m) \leq \bar{W}. \tag{18}$$

V. Simulations

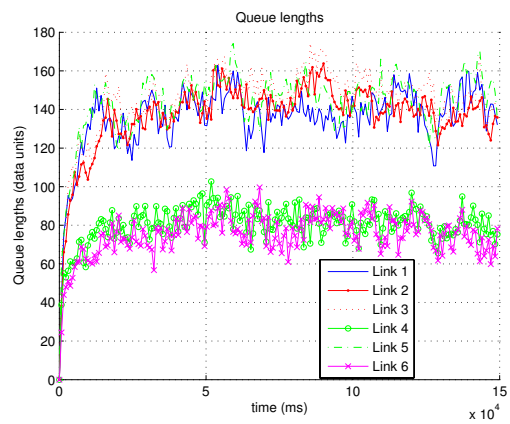
A. CSMA scheduling: i.i.d. input traffic with fixed average rates

In our C++ simulations, the transmission time of all links is exponentially distributed with mean 0.5ms, and the backoff time of link k is exponentially distributed with mean $0.5/\exp(r_k)$ ms. Here we have proportionally decreased the two mean values, which does not affect the stationary distribution (1). Assume that the full speed of transmission of each link (without contentions from other links) is 1(data unit)/ms. (For example, the link transmits 0.6 unit of data in 0.6 ms.) Initially, all queues are empty, and the initial value of r_k is 0 for all k. r_k is then adjusted using Algorithm 1 once every $b = 5ms$, with a constant step size $\alpha = 0.23$.

There are 6 links in the "Network 1", whose LCG is shown in Fig. 2 (a). (Each link only needs to know the set of links which conflict with itself.) Define $0 \leq \rho < 1$ as the "load factor", and let $\rho = 0.99$ in this simulation.



(a) Link Contention Graph



(b) Queue lengths, with constant step size. The vector $\mathbf{r}$ is not shown since it is proportional to the queue length.

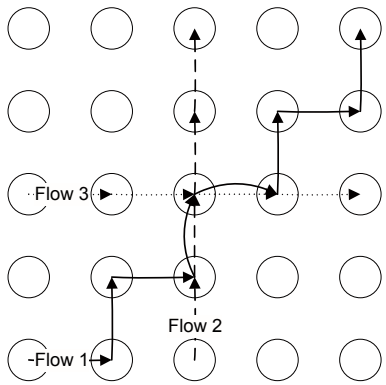
Fig. 2. Adaptive CSMA Scheduling with fixed input rates (Network 1)

The arrival rate vector is set to λ=ρ*[0.2*(1,0,1,0,0,0) + 0.3*(1,0,0,1,0,1) + 0.2*(0,1,0,0,1,0) + 0.3*(0,0,1,0,1,0)] = ρ*(0.5,0.2,0.5,0.3,0.5,0.3) (data units/ms). We have multiplied ρ to a convex combination of some Maximal IS's to ensure that λ is in the interior of the capacity region. Fig. 2 (b) shows the evolution of the queue lengths. They are stable (do not go to infinity) despite some oscillations. The vector $\mathbf{r}$ is not shown since it is just α/b times the queue lengths.

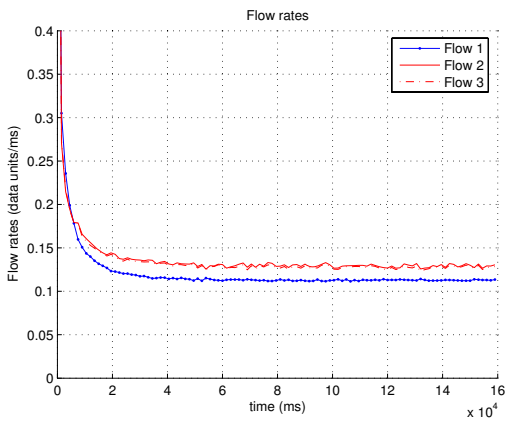
Since $\mathbf{r}$ oscillates around a neighborhood of $\mathbf{r}^*$ (the optimal solution of (4)), the queue lengths $\mathbf{Q} = b/\alpha \cdot \mathbf{r}$ oscillate near $b/\alpha \cdot \mathbf{r}^*$. Since $\mathbf{r}^*$ is generally not zero, the queue lengths are generally not around zero, which causes some queueing delays. In [25], we introduced an enhancement of Algorithm 1 to reduce the delay. Apart from throughput-optimality, it also keeps the queues short.

B. Joint scheduling and rate control

In Fig 3, we simulate a more complex network ("Network 2"). We also go one step further than Network 1 by giving the actual locations of the nodes, not only the LCG. Fig 3 (a) shows the network topology, where each circle represents a node. The nodes are arranged in a grid for convenience, and the distance between two adjacent nodes (horizontally or vertically) is 1. Assume that the transmission range is 1, so that a link can only be formed by two adjacent nodes. Assume that two links cannot transmit simultaneously if there



(a) Network 2 and flow directions



(b) Flow rates

Fig. 3. Flow rates in Network 2 (Grid Topology) with Joint scheduling and rate control

are two nodes, one in each link, being within a distance of 1.1 (In IEEE 802.11, for example, DATA and ACK packets are transmitted in opposite directions. This model has considered the interference among the two links in both directions). The paths of 3 multi-hop flows are plotted. The utility function of each flow is $\log(\cdot)$. The weighting factor is $\beta = 3$. (Note that the input rates are adjusted by the flow control algorithm instead of being specified as in the last subsection.)

Fig 3 (b) shows the evolution of the flow rates (using Algorithm 2). We see that they become relatively constant after an initial convergence. By directly solving (17) centrally, we find that the theoretical optimal flow rates for the three flows are 0.1111, 0.1333 and 0.1333 (data unit/ms), very close to the simulation results. The queue lengths are also stable but not shown here due to the limit on space.

VI. Implementation Considerations in 802.11 Networks

A. Packet Collisions

In the idealized CSMA model we used, the distribution of backoff time is continuous and there is no collision. This allows us to focus on the scheduling problem without worrying about the contention resolution problem. (The resulting performance can serve as a benchmark in the following.) However in practice, the backoff time is a multiple of mini-slots, where each mini-slot cannot be arbitrarily small (since the sensing time is not zero). Given the discrete distribution of backoff time, collisions are possible to occur. Clearly, 100% throughput cannot be achieved in this case. In this section we consider this practical issue and propose adaptations of our algorithm for 802.11 networks. (Note that collisions affect the performance of all random access protocols, not only the one proposed here. On the other hand, ignoring collisions in no way implies maximal throughput. For example, the 802.11-like distributed greedy algorithm is not throughput-optimal [7].)

Assume that for link k, the average transmission time is T. Then the average backoff time is T/R_k. Denote W_k as the Contention Window (CW) that gives the same average backoff time (Recall that the distribution of the backoff time is not important, as long as it has the correct mean). Since a random number is uniformly picked from 0 to $W_k - 1$, then the average backoff time is $t_m \cdot (W_k - 1)/2$, where t_m is the length of a mini-slot. (For simplicity, we do not consider Binary Exponential Backoff, or BEB, in this calculation.) Equating the two quantities gives

$$W_k = \frac{T}{R_k}\frac{2}{t_m} + 1. \tag{19}$$

We know that larger CW's lead to lower collision probabilities. By equation (19), for given R_k's, small mini-slot t_m or large transmission time T can lead to large CW. (If $t_m \to 0$ or $T \to +\infty$, then collisions can be ignored and we return to the previous model.) However, t_m is limited by the speed of light. The mean transmission time T can be made large, but should not be too large in practice since that will increase access delays. So, it seems more realistic to impose an upper bound, r_{max}, to all r_k's. This gives W_k's a lower bound: $W_k \geq 2T/(\exp(r_{max}) \cdot t_m) + 1$. For example, assume $T = 1ms$. And a mini-slot in 802.11a is $t_m := 9\mu s$. If we require that $r_k \leq r_{max} = 2$. Then $W_k \geq 31$. This gives reasonably low collision probabilities if the number of nodes in a collision domain is not too high [22].

In the following, we introduce two methods of adapting Algorithm 2 for 802.11 networks with collisions.

1) Method 1: Approximating Algorithm 2 by directly bounding $\mathbf{r}$*:* As an approximation, we modify the second item in Algorithm 2 to "Link k lets $r_k = \min\{z_k, r_{max}\}$ in the CSMA operation". We call the modified algorithm as **Algorithm 3**. (Note that the source of flow m still takes q_{km}, where $k = \delta(m)$, as the "price" for rate control. No upper bound is imposed on the price.) Smaller r_{max} tends to give lower total utility of the flows.

For any flow m, the solution of $\max_{f_m}\{v_m(f_m) - q_{km} f_m\}$ tends to 0 as q_{km} tends to infinity. Then it is easy to see that with Algorithm 3, the queue lengths of all links are stable. Otherwise the input rate will be reduced to 0 which leads to a contradiction.

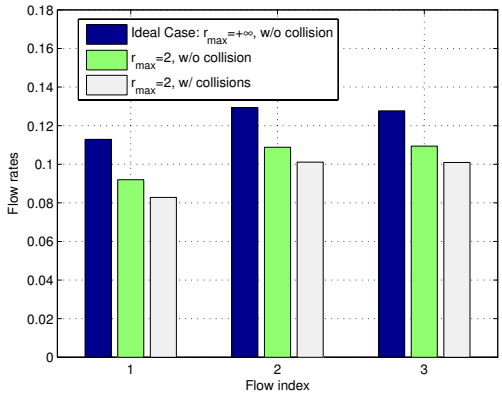


Fig. 4. Flow rates in Network 2 (Grid Topology) with $r_{max} = 2$, compared to the ideal case

We perform simulations in the grid topology (Network 2), with $T = 1ms$, and compare the resulting flow rates in three cases: (1) The idealized case where there is no upper bound of $\mathbf{r}$ (i.e., $r_{max} = +\infty$) and there is no collision (with continuous backoff time); (2) the case with $r_{max} = 2$ but no collision (with continuous backoff time) and (3) the case with $r_{max} = 2$, and collisions (i.e., with discrete backoff time and Binary Exponential Backoff). In (3), the contention window of link k is computed by plugging $r_k(t)$ into equation (19) and rounding the result to the nearest integer, where $r_k(t)$ is determined by Algorithm 3. The results are shown in Fig 4. We observe that the systems are stable. And as expected, the flow rates are reduced compared to the ideal case. However, the reduction is reasonable considering the small r_{max} value and packet collisions.

2) Method 2: Choosing a suitable weighting factor β: In this alternative method, we don't bound $\mathbf{r}$ directly. Instead, by choosing a suitable weighting factor β of the total utility, all r_k is guaranteed to be smaller than r_{max}, if certain conditions are satisfied. (In [23], a similar approach is used to control the amount of backlog in the network.)

Proposition 5: Assume that each utility function $v_m(f_m)$ satisfies that $v'_m(0) < V < \infty$, i.e., the derivative at 0 is bounded by V. (For example, if $v_m(f_m) = \log(1 + f_m)$, then $V = 1$.) Then by setting $\beta = r_{max}/V$ in the following Algorithm 4 (a minor revision of Algorithm 2), we have $r_k \leq r_{max}, \forall k$ at all time.

Algorithm 4: Modified joint scheduling and rate control algorithm

Initially, assume that all queues are empty, and set $q_{km} = 0, \forall k, m$. Then iterate:

- Link k computes the maximal back-pressure $z_k = \max_{m:a_{mk}=1}(q_{km} - q_{down(k,m),m})$. When link k gets the opportunity to transmit, (1) if $z_k > 0$, it transmits the head-of-line packet of a flow $m' \in \arg\max_{m:a_{mk}=1}(q_{km} - q_{down(k,m),m})$; (2) if $z_k \leq 0$, then it transmits a dummy packet.
- Link k lets $r_k = (z_k)_+$ in the CSMA operation.
- Rate control: For each flow m, if link k is its source link, then the transmitter of link k let $f_m = \arg\max_{f'_m}\{\beta \cdot v_m(f'_m) - q_{km} f'_m\}$.
- The dual variables q_{km} are updated in the same way as Algorithm 2.

Due to the limit of space, the proof is included in [24].

B. Discrete TA and a real-world implementation

Although r_k is continuous in our model, one may find it convenient to quantize r_k into a set of discrete values in real implementation. Each discrete value corresponds to a different Contention Window (smaller r_k corresponds to larger CW), and this can be easily mapped to the "service classes" in IEEE 802.11e. Note that here the prioritization is based on the back-pressure instead of service type originally defined in 802.11e. Indeed, in [26], a similar protocol is implemented with 802.11e hardware and shows superior performance compared to normal 802.11. (Different from our work, however, [26] only focuses on implementation study. Also, the CW's there are set in a more heuristic way.)

VII. Conclusion

In this paper, we have proposed a distributed CSMA scheduling algorithm, and showed that it is throughput-optimal in wireless networks with a general interference model. We have utilized the Markov Random Field property of CSMA networks in order to obtain the distributed algorithm and the maximal throughput. Furthermore, we have combined it with end-to-end flow control to approach the optimal utility, and showed its connection with maximal backpressure scheduling. The algorithm is easy to implement, and the simulation results are encouraging.

The adaptive CSMA algorithm is a modular MAC-layer protocol that can work with other algorithms in transport layer and network layer. For example, it can be combined with optimal routing, anycast and multicast using protocols similar to Algorithm 2.

We also considered some practical issues when implementing the algorithm in an 802.11 setting. To avoid excessive collisions, transmissions should not be too aggressive. This leads to reasonable performance reduction compared to the idealized model.

As mentioned before, the current proof of the throughput optimality is based on the stationary distribution of the CSMA Markov chain. This is certainly sufficient, but may not be necessary according to our simulations. However, identifying the exact conditions on the step sizes to ensure stability is difficult since they may depend on network size, network topology, and arrival rates, etc. This is a direction for future research.

References

[1] X. Lin, N.B. Shroff, R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," IEEE Journal on Selected Areas in Communications, 2006.

[2] A. Eryilmaz, A. Ozdaglar and E. Modiano, "Polynomial Complexity Algorithms for Full Utilization of Multi-hop Wireless Networks," Proceedings of IEEE Infocom, 2007

[3] E. Modiano, D. Shah, and G. Zussman, "Maximizing Throughput in Wireless Networks via Gossiping," ACM SIGMETRICS Performance Evaluation Review, vol. 34 , no. 1, June 2006

[4] S. Sanghavi, L. Bui, R. Srikant, "Distributed Link Scheduling with Constant Overhead," Proceedings of the 2007 ACM SIGMETRICS

[5] J. W. Lee, M. Chiang, and R. A. Calderbank, "Utility-Optimal Random-Access Control," IEEE Transactions on Wireless Communications, vol. 6, no. 7, pp. 2741-2751, July 2007.

[6] P. Gupta, A.L. Stolyar, "Optimal Throughput Allocation in General Random Access Networks," Conference on Information Sciences and Systems (CISS'06), Princeton, March 22-24.

[7] X. Wu, R. Srikant, "Bounds on the Capacity Region of Multi-Hop Wireless Networks under Distributed Greedy Scheduling," IEEE INFOCOM, 2006

[8] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," In Proceedings of IEEE Infocom, Miami, FL, March 2005.

[9] P. Marbach, A. Eryilmaz and A. Ozdaglar, "Achievable Rate Region of CSMA Schedulers in Wireless Networks with Primary Interference Constraints," IEEE Conference on Decision and Control, 2007.

[10] Y. Xi and E. M. Yeh, "Throughput Optimal Distributed Control of Stochastic Wireless Networks," WiOpt'06

[11] M. J. Neely, E. Modiano, C. P. Li, "Fairness and Optimal Stochastic Control for Heterogeneous Networks," In Proceedings of IEEE Infocom, 2005.

[12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," IEEE Transactions on Automatic Control, 36:1936-1948, December 1992.

[13] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, "Achieving 100% throughput in an input-queued switch," IEEE Transactions on Communications, 1999.

[14] F. P. Kelly, "Reversibility and Stochastic Networks," Wiley, 1979.

[15] X. Wang and K. Kar, "Throughput Modelling and Fairness Issues in CSMA/CA Based Ad-Hoc Networks," Proceedings of IEEE Infocom 2005, Miami, March 2005.

[16] S.C. Liew, C. Kai, J. Leung, B. Wong, "Back-of-the-Envelope Computation of Throughput Distributions in CSMA Wireless Networks," submitted for publication, http://arxiv.org//pdf/0712.1854

[17] L. Jiang, S. C. Liew, "Improving Throughput and Fairness by Reducing Exposed and Hidden Nodes in 802.11 Networks," IEEE Transactions on Mobile Computing, Vol. 7, No. 1, pp. 34-49, Jan. 2008.

[18] S. Boyd and L. Vandenberghe, "Convex Optimization", Cambridge University Press, 2004.

[19] H. Robbins and S. Monro, "A Stochastic Approximation Method," The Annals of Mathematical Statistics, Vol. 22, No. 3, 400-407. Sep., 1951.

[20] J. Zhang, D. Zheng, and M. Chiang, "The impact of stochastic noisy feedback on distributed network utility maximization," IEEE Transactions on Information Theory, vol. 54, no. 2, pp. 645-665, February 2008.

[21] P. Whittle, "Systems in stochastic equilibrium," John Wiley & Sons, Inc., 1986, New York, NY, USA.

[22] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE Journal on Selected Areas in Communications, 18(3):535–547, 2000.

[23] M. J. Neely and R. Urgaonkar, "Cross Layer Adaptive Control for Wireless Mesh Networks," Ad Hoc Networks (Elsevier), vol. 5, no. 6, pp. 719-743, August 2007.

[24] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," Journal version, submitted.

[25] L. Jiang and J. Walrand, "A Distributed Algorithm for Maximal Throughput and Optimal Fairness in Wireless Networks with a General Interference Model," EECS Technical Report, UC Berkeley, Apr. 2008 (http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-38.html).

[26] A. Warrier, S. Ha, P. Wason and I. Rhee, "DiffQ: Differential Backlog Congestion Control for Wireless Multi-hop Networks," Technical Report, Dept. Computer Science, North Carolina State University, 2008.

APPENDIX: PROOF THE PROPOSITION 2

Consider the convex optimization problem (11), where λ is strictly feasible (i.e., $\lambda = \sum_i \bar{p}_i \cdot x^i$ for some $\bar{p}_i > 0, \forall x^i$ and $\sum_i \bar{p}_i = 1$). Problem (11) is clearly feasible and the feasible region is closed and convex. The objective function (the entropy) is bounded in the feasible region. So, the optimal value is bounded.

We now check whether the Slater condition [18] (page 226-227) is satisfied. Since all the constraints in (11) are linear, we only need to check whether there exists a *feasible* $\mathbf{u}$ which is in the relative interior [18] of the domain $\mathcal{D}$ of the objective function $-\sum_i u_i \log(u_i)$, which is $\mathcal{D} = \{\mathbf{u} | u_i \geq 0, \sum_i u_i = 1\}$. Since $\lambda = \sum_i \bar{p}_i \cdot x^i$ where $\bar{p}_i > 0, \forall i$ and $\sum_i \bar{p}_i = 1$, letting $\mathbf{u} = \bar{\mathbf{p}}$ satisfies the requirement. Therefore the Slater condition is satisfied. As a result, there exist (finite) dual variables $y_k^* \geq 0, w_i^* \geq 0, z^*$ such that the Lagrangian

$$\begin{aligned} & \mathcal{L}(\mathbf{u}; \mathbf{y}^*, \mathbf{w}^*, z^*) \\ = \; & -\sum_i u_i \log(u_i) + \sum_k y_k^* (\sum_i u_i \cdot x_k^i - \lambda_k) \\ & + z^* (\sum_i u_i - 1) + \sum_i w_i^* u_i \end{aligned} \tag{20}$$

is maximized by the optimal solution $\mathbf{u}^*$, and the maximum is attained.

We first claim that the optimal solution satisfies $u_i^* > 0, \forall i$. Suppose $u_i^* = 0$ for all i's in a non-empty set $\mathcal{I}$. For convenience, denote $\bar{\mathbf{p}}$ as the vector of $\bar{p}_i$'s. Since both $\mathbf{u}^*$ and $\bar{\mathbf{p}}$ are feasible to the problem (11), any point on the line segment between them is also feasible. Then, if we slightly move $\mathbf{u}$ from $\mathbf{u}^*$ along the direction of $\bar{\mathbf{p}} - \mathbf{u}^*$, the change of the objective function $h(\mathbf{u}) := -\sum_i u_i \log(u_i)$ (at $\mathbf{u}^*$) is proportional to

$$\begin{aligned} & (\bar{\mathbf{p}} - \mathbf{u}^*)^T \nabla h(\mathbf{u}^*) \\ = \; & \sum_i (\bar{p}_i - u_i^*)[-\log(u_i^*) - 1] \\ = \; & \sum_{i \notin \mathcal{I}} (\bar{p}_i - u_i^*)[-\log(u_i^*) - 1] + \sum_{i \in \mathcal{I}} \bar{p}_i [-\log(u_i^*) - 1]. \end{aligned}$$

For $i \notin \mathcal{I}$, $u_i^* > 0$, so $\sum_{i \notin \mathcal{I}} (\bar{p}_i - u_i^*)[-\log(u_i^*) - 1]$ is bounded. But for $i \in \mathcal{I}$, $u_i^* = 0$, thus $-\log(u_i^*) - 1 = +\infty$. Also, since $\bar{p}_i > 0$, we have $(\bar{\mathbf{p}} - \mathbf{u}^*)^T \nabla h(\mathbf{u}^*) = +\infty$. This means that $h(\mathbf{u})$ increases when we slightly move $\mathbf{u}$ away from $\mathbf{u}^*$ towards $\bar{\mathbf{p}}$. Thus, $\mathbf{u}^*$ is not the optimal solution.

Therefore $u_i^* > 0, \forall i$. By complementary slackness, $w_i^* = 0$. So the term $\sum_i w_i^* u_i$ in (20) is 0. Since $\mathbf{u}^*$ maximizes $\mathcal{L}(\mathbf{u}; \mathbf{y}^*, \mathbf{w}^*, z^*)$, then

$$\frac{\partial \mathcal{L}(\mathbf{u}^*; \mathbf{y}^*, \mathbf{w}^*, z^*)}{\partial u_i} = -\log(u_i^*) - 1 + \sum_k y_k^* x_k^i + z = 0, \forall i.$$

Combining this and $\sum_i u_i^* = 1$, we have

$$u_i^* = \frac{\exp(\sum_k y_k^* x_k^i)}{\sum_j \exp(\sum_k y_k^* x_k^j)}, \forall i. \tag{21}$$

Plug (21) back into (20), we have $\max_{\mathbf{u}} \mathcal{L}(\mathbf{u}; \mathbf{y}^*, \mathbf{w}^*, z^*) = -F(\mathbf{y}^*)$. Since $\mathbf{u}^*$ and the dual variables $\mathbf{y}^*$ solves (11), $\mathbf{y}^*$ is the solution of $\min_{\mathbf{y} \geq \mathbf{0}} \{-F(\mathbf{y})\}$ (and the optimum is attained). So, $\sup_{\mathbf{r} \geq 0} F(\mathbf{r})$ is attained by $\mathbf{r} = \mathbf{y}^*$. The above proof also shows that (4) is the dual problem of (11).