A Distributed Algorithm for Maximal Throughput and Optimal Fairness in Wireless Networks with a General Interference Model

Libin Jiang
Jean Walrand

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-38
http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-38.html

April 15, 2008

A Distributed Algorithm for Optimal Throughput and Fairness in Wireless Networks with a General Interference Model

Libin Jiang and Jean Walrand
Dept. Electrical Engineering & Computer Science, University of California at Berkeley
{ljiang,wlr}@eecs.berkeley.edu

***Abstract*—In multi-hop wireless networks, earlier research on joint scheduling and congestion control has suggested that MAC-layer scheduling is the bottleneck. Distributed scheduling for maximal throughput is difficult since the conflicting relationship between different links is complex. Existing works on maximal-throughput scheduling usually assumes synchronized time slots, and in each slot, a maximal-weighted "independent set" needs to be found or approximated. However, this is hard to implement in distributed networks. On the other hand, a distributed greedy protocol similar to IEEE 802.11 can only achieve a fraction of the throughput region. In this paper, we introduce an adaptive CSMA algorithm, which can achieve the maximal throughput distributedly under some assumptions. The intuitive idea is that each link should adjust its aggressiveness of transmission based on its backlog. Furthermore, we combine the algorithm with end-to-end flow control to achieve fairness among competing flows. The effectiveness of the algorithm is verified by simulations.**

***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 needs cooperation of different network layers. Transport layer needs to inject the right amount of traffic into the network based on the congestion level. And 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 relationship between different links can be complicated. Existing works on maximal-throughput scheduling usually assume synchronized time slots for transmission. [2] shows that Maximal-weight (MW) scheduling is throughput-optimal (that is, it can support any incoming rates within the capacity region). This has been applied to achieve 100% throughput in input-queued switches [3]. However, finding the maximal-weighted "Independent Set" ("IS")[1] in each time slot is NP-complete in general, and is hard even for centralized algorithms. [4] introduces a throughput-optimal randomized algorithm. Although it has low complexity in computation, a global comparison between the weights of two Independent Sets is need in each slot. This introduces communication overhead when applied to a distributed wireless network.

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), depending on the specific network topology and interference relationships. Reference [8] studied the impact of such imperfect scheduling on utility maximization in wireless networks.

In this paper, we introduce an adaptive CSMA (Carrier Sensing Multiple Access) algorithm for a general interference model. 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[2]. Since the algorithm utilizes the carrier-sensing capability, it may not be directly comparable to the throughput-optimal algorithms mentioned above. However, it does have a few distinct features:

- Each node only uses its local information (e.g., its backlog). No explicit communication or control messages are required among the nodes.
- It is based on CSMA random access, which is similar to the IEEE 802.11 protocol and is very easy to implement.
- Time is not divided into synchronous slots. Thus no synchronization of transmissions is needed.

Using a novel technique, we further combine this scheduling algorithm with end-to-end flow control to achieve fairness among competing flows, as well as full utilization of the network.

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. Therefore the throughput regions are different.

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" $G = \{\mathcal{V}, \mathcal{E}\}$, 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

[1]An IS is a set of links that can transmit at the same time without conflicting. The "weight" of an IS is the summation of the queue lengths of all transmitting links in this IS.

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

transmit at the same time iff there is an edge between them. Note that this includes the "one-hop interference model" and "two-hop interference model" [7] 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 is 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".

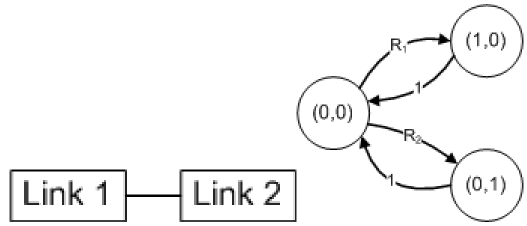
B. An idealized CSMA protocol and the average throughput

We generalize an idealized model of CSMA in [10]. 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. (We assume that a link can carrier-sense the transmission of any of its conflicting links. That is, there is no "hidden terminal" problem that causes collisions.) If none of its conflicting links is transmitting, then the transmitter of link k waits (or backoffs) for a random period of time which is exponentially distributed with mean $1/R_k$ and then starts its transmission[3]. 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, according to [10].) With the continuous distributions of the backoff time, the probability for two conflicting links to start transmission at the same time is 0, so we assume that there is no collision. (This is an idealized assumption, different from actual protocols such as IEEE 802.11. There, the backoff time of a transmitter is a multiple of "mini-slots". Because of this discretization fundamentally due to the limit on light speed, collisions occur if two transmitters finish their backoff at the same time.)

It is not difficult to see that the transitions of the transmission states form a Continuous Time Markov Chain, which we call "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 $T_{k,1}x^i$ with a rate of R_k, where $T_{k,1}x^i$ refers to the state that differs from x^i only in the k'th element by changing x^i_k to 1. Similarly, state $T_{k,1}x^i$ 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 $T_{k,1}x^i$ does not exist.

Fig 1 gives an example network whose link contention graph 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, (1,0) means that only link 1 is transmitting, and state (0,1) means that only link 2 is transmitting. The state (1,1) is not feasible.

[3]If more than one backlogged links share the same transmitter, we can let these links compete for the medium as if they have different transmitters.



(a) Link contention graph (b) Markov Chain

Figure 1. Example network: link contention graph and corresponding Markov Chain.

Denote $r_k = \log(R_k)$. The stationary distribution of any feasible state x^i in the Markov Chain is

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

where

$$C(\mathbf{r}) = \sum_j \exp(\sum_{k=1}^K x^j_k r_k) \quad (2)$$

Note that the summation "$\sum_j$" is over all feasible states (or Independent Sets), and the vector $\mathbf{r} = (r_1, r_2, \cdots, r_K)$. This type of product-form distribution often appears in time-reversible Markov Chains [9]. (Later, we also write $p(x^i;\mathbf{r})$ as $p_i(\mathbf{r})$ for simplicity. They 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. Consider state x^i (where $x^i_k = 0$ and $x^i_m = 0, \forall m \in \mathcal{N}(k)$) and $T_{k,1}x^i$ again. From (1), we have

$$\frac{p(T_{k,1}x^i;\mathbf{r})}{p(x^i;\mathbf{r})} = \exp(r_k) = R_k$$

which is exactly the detailed balance equation between state x^i and $T_{k,1}x^i$. 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 ([9], page 189).

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

$$s_k(\mathbf{r}) = \sum_i x^i_k \cdot p(x^i;\mathbf{r}) \quad (3)$$

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 [10] shows that the stationary distribution (1) still holds.

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 $\lambda_k > 0, \forall k$. (Otherwise, the link(s) with zero arrival rate can be removed from the problem.) We say that $\lambda \in R^K$ 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 Independent Sets, 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 $\sum_i \bar{p}_i = 1$.

Define the following function (which is 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 $\mathbf{r}$ [11]. We will show in the following that

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 variable associated with the constraint $\mathbf{r} \geq 0$ 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 $d_k^* \geq 0$, we have $\lambda_k \leq s_k(\mathbf{r}^*)$. The inequality is strict only if $r_k^* = 0$, due to complementary slackness. ■

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, i.e., it's not at the boundary of the capacity region, 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 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) positive step size. 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 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 $r_k(t) = \alpha/b \cdot Q_k(t)$. 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 means slow changes of $\mathbf{r}$):

(1). If b is very large (but finite), then as the CSMA Markov Chain converges, $s'_k(t) \approx s_k(\mathbf{r}(t))$. (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.) 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 [14]. 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) [13] [12], 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.

III. Statistical Entropy and 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.

Assume that $\mathbf{r}$ is fixed, and let $u_i(\tau)$ be the probability of state x^i at time τ. Then the "statistical entropy" ([9], page 19) $H(\tau, \mathbf{r})$ of the CSMA Markov Chain, defined below, monotonically increases with time τ, and converges to its maximal value 0 when $u_i(\tau) \to p_i(\mathbf{r})$:

$$H(\tau, \mathbf{r}) := -\sum_i u_i(\tau) \log \frac{u_i(\tau)}{p_i(\mathbf{r})} \tag{9}$$

where $p_i(\mathbf{r})$ is the stationary distribution (1). The increase of $H(\tau, \mathbf{r})$ is due to the fact that the distribution $u_i(\tau)$ gets closer and closer to the stationary distribution.

Plugging (1) into (9), we have

$$\begin{array}{rl} & H(\tau, \mathbf{r}) \\ = & -\sum_i u_i(\tau)[\log(u_i(\tau)) - \sum_k x_k^i r_k + \log(C(\mathbf{r}))] \\ = & -\sum_i u_i(\tau)\log(u_i(\tau)) + \sum_k r_k \mu_k(\tau) - \log(C(\mathbf{r})) \end{array} \tag{10}$$

where $C(\mathbf{r})$ is determined by (2), and $\mu_k(\tau) := \sum_i u_i(\tau) \cdot x_k^i$ is the average service rate of link k at time τ.

Now we claim that the adaptive CSMA algorithm in the last section also solves the following convex optimization problem.

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

To see this, let y_k be the dual variable associated with the constraint $\sum_i u_i \cdot x_k^i \geq \lambda_k$. Then a partial Lagrangian is

$$\begin{array}{rcl} L(\mathbf{u}; \mathbf{y}) & = & -\sum_i u_i \log(u_i) + \sum_k [y_k(\sum_i u_i \cdot x_k^i - \lambda_k)] \\ & = & -\sum_i u_i \log(u_i) + \sum_k (y_k \mu_k) - \sum_k (y_k \lambda_k) \end{array} \tag{12}$$

Regard the dual variable y_k as r_k (which we will see is indeed the case). Then, given the dual variables $\mathbf{r}$, $u_i = p_i(\mathbf{r})$ maximizes the statistical entropy (10) of the CSMA Markov Chain. So, $u_i = p_i(\mathbf{r})$ also maximizes (subject to $u_i \geq 0, \sum_i u_i = 1$) the Lagrangian $L(\mathbf{u}; \mathbf{y}) = L(\mathbf{u}; \mathbf{r})$, since the terms $C(\mathbf{r})$ and $-\sum_k (y_k \lambda_k) = -\sum_k (r_k \lambda_k)$ do not depend on $\mathbf{u}$. Meanwhile, the local algorithm of link k, $r_k \leftarrow [r_k + \alpha(\lambda_k - \mu_k)]_+$ is a sub-gradient algorithm to update the dual variables. So, the overall algorithm is solving (11) by searching for the saddle point of $L(\mathbf{u}, \mathbf{r})$.

Interestingly, the dual problem of (11) is (4) (section VIII-A in the Appendix contains a derivation), and vice versa (see section VIII-B in the Appendix). The purpose of discussing "statistical entropy" here is to help reveal the physical meaning of u_i: it is the probability of the state x^i. (Otherwise, u_i's show up as some dual variables of problem (4), whose meaning is not evident.) We will use this observation in the next section.

IV. Joint scheduling and rate control

In this section, we combine end-to-end rate control with the CSMA scheduling algorithm to achieve fairness among competing flows, as well as full utilization of the network. 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 it 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:

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

Notice that the objective function is not exactly the total utility, but with 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). The 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

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

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

Iterate:

- Link k chooses to serve 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 (12)).
- Rate control: For each flow m, if link k is its source link, then the transmitter of link k solves $\max_{f_m}\{v_m(f_m) - q_{km} f_m\}$. This maximizes $\mathcal{L}(\mathbf{u},\mathbf{f};\mathbf{q})$ over $\mathbf{f}$.
- 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 $k = \delta(m)$.

B. Approaching the maximal utility

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

$$\beta \cdot W' := \max_{\mathbf{u},\mathbf{s},\mathbf{f}} - \sum_i u_i \log(u_i) + \sum_m V_m(f_m) \quad (16)$$

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 at most 2^K possible states, then, $0 \leq -\sum_i u_i \log(u_i) \leq \log 2^K = K \cdot \log 2$. So when β is large, the "importance" of the total utility dominates the objective function of (16). As a result, the solution of (16) approximately achieves the maximal utility. Denote the highest total utility achievable as $\bar{W}$, i.e.,

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

subject to the same constraints as in (13). Assume that $\mathbf{u} = \bar{\mathbf{u}}$ when (17) is solved. We show in the Appendix that

Proposition 3: 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,

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

V. Reducing the queueing delay

Consider the scenario with a given arrival rate vector λ in the interior of the capacity region. In the optimization problem (11), we have imposed the constraint that $\sum_i u_i \cdot x_k^i \geq \lambda_k, \forall k$. That is, the service rate is not less than the arrival rate. If for some link k, the two are equal at the optimal point, i.e., $\sum_i u_i^* \cdot x_k^i = \lambda_k$, then the queue length will remain at a certain level and not decrease to 0. Although this still stabilize the queue, traffic suffers from queueing delay when traversing this link. To reduce the delay, consider a modified version of problem (11):

$$\begin{array}{ll}\max_{\mathbf{u}} & -\sum_i u_i \log(u_i) + c\sum_k \log(w_k) \\ \text{s.t.} & \sum_i (u_i \cdot x_k^i) \geq \lambda_k + w_k, \forall k \\ & u_i \geq 0, \sum_i u_i = 1 \\ & w_k \geq 0, \forall k\end{array} \quad (19)$$

where $0 < c < 1$ is a small constant. Note that we have added the new variables w_k, and require $\sum_i u_i \cdot x_k^i \geq \lambda_k + w_k$. In the objective function, the term $c \cdot \log(w_k)$ is a penalty function to avoid w_k being too close to 0.

Since λ is in the interior of the capacity region, there is a vector λ' also in the interior and satisfying $\lambda' > \lambda$ component-wise. So, the above problem is strictly feasible, i.e., there exist $\mathbf{u} > 0, \mathbf{w} > 0$ satisfying the constraints. So in the optimal solution, we have $w_k^* > 0, \forall k$ (otherwise the objective function is $-\infty$). Thus $\sum_i u_i^* \cdot x_k^i \geq \lambda_k + w_k^* > \lambda_k$. This means that the service rate is strictly larger than the arrival rate, bringing the extra benefit that the queue lengths tend to decrease to 0.

The dual problem of (19) is

$$\begin{array}{ll}\max_{\mathbf{r}} & F(\mathbf{r}) - c\sum_k \log(r_k) + K \cdot c \cdot [\log(c) - 1] \\ \text{s.t.} & r_k \geq 0, \forall k\end{array} \quad (20)$$

where $K \cdot c \cdot [\log(c) - 1]$ is a constant and thus can be ignored in the optimization problem.

So the localized (gradient) algorithm at link k is:

Algorithm 3: Enhanced Algorithm 1 to reduce queueing delays

$$r_k(t+1) = [r_k(t) + \alpha \cdot (\lambda_k'(t) + c/r_k(t) - s_k'(t))]_+ \quad (21)$$

where α is some small step size. As before, even when link k' has no backlog (i.e., zero queue length), we let it send dummy packet with its current aggressiveness $r_{k'}$. This ensures that the average service rate of link k is $s_k(\mathbf{r}(t))$ for all k.

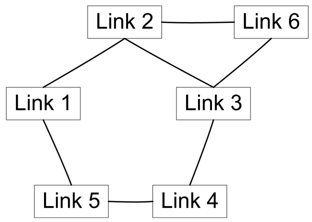
With Algorithm 3, $\mathbf{r}$ oscillate around a neighborhood of $\mathbf{r}^*$, the optimal solution of (20). But the dynamics of Q_k is determined by (7), whose Right-hand-side does not have the positive term $c/r_k(t)$. Since r_k is stable, Q_k is not only stable, but also tends to zero.

For the end-to-end utility maximization (without a given arrival rate vector), we can use similar techniques to modify problem (13).

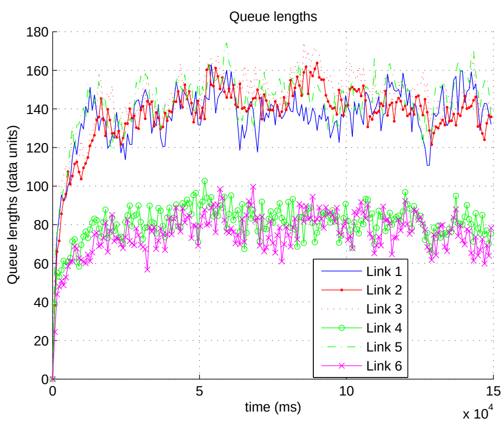
VI. Simulations

A. i.i.d. input traffic with fixed average rates

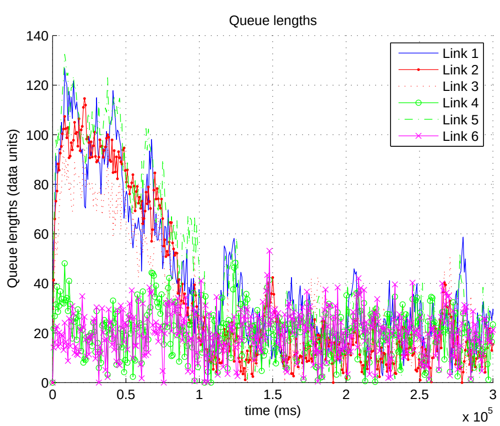
In our Matlab simulations, the transmission time of all links be exponentially distributed with mean 0.5ms, and the backoff time of link k be exponentially distributed with mean



(a) Link Contention Graph

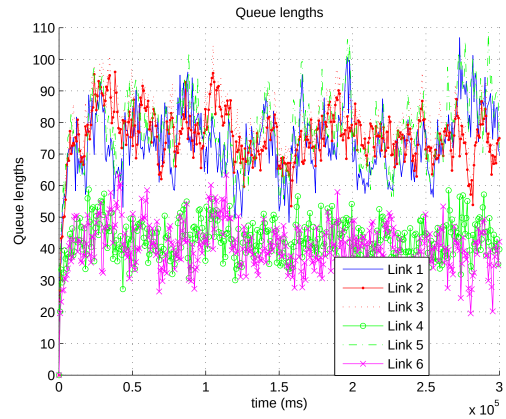


(b) Queue lengths, with constant step size. The vector **r** is not shown since it is proportional to the queue length.

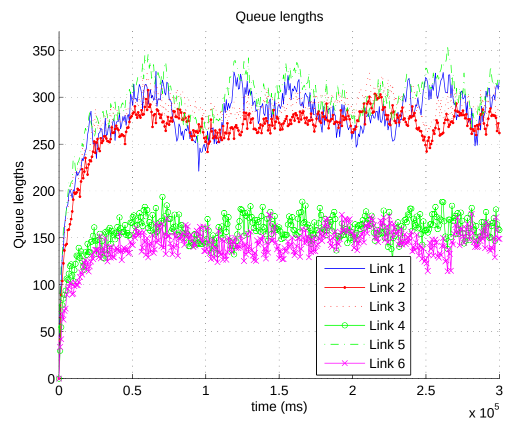


(c) Queue lengths (with Algorithm 3)

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



(a) Queue lengths, with step size 2α.



(b) Queue lengths, with step size 0.5α.

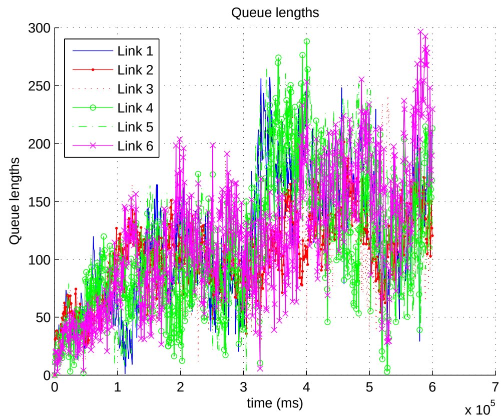
Figure 3. Comparison of different constant step sizes

$0.5 \cdot \exp(r_k)$ per ms. Here we have proportionally decreases 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, if one transmission takes 0.6ms, then 0.6 units of data is transmitted.) 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 $\alpha = 0.23$.

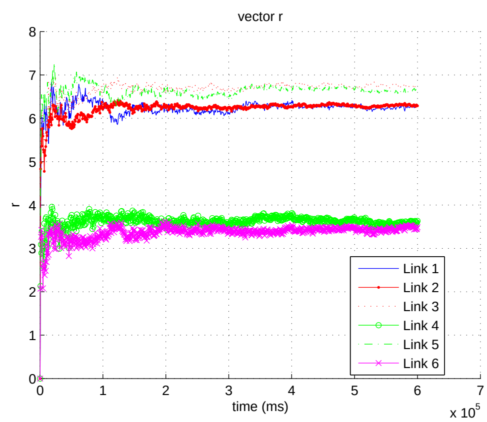
There are 6 links in the "Network 1", whose link contention graph is shown in Fig. 2 (a). (Each link only needs to know the set of links which conflict with itself.) The arrival rate vector is set to λ=0.99*[0.2*(1,0,1,0,0,0) + 0.3*(1,0,0,1,0,1) + 0.2*(0,1,0,0,1,0) + 0.3*(0,0,1,0,1,0)] = 0.99*(0.5,0.2,0.5,0.3,0.5,0.3) (data units/ms). We have multiplied 0.99 to a convex combination of some Maximal Independent Sets 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 **r** is not shown since it is just α/b times the queue lengths. Fig. 2 (c) shows the evolution of queue lengths with Algorithm 3 (the small constant $c = 0.01$), which drives the queue lengths to around zero, thus significantly reducing the queueing delays.

Fig. 3 (a), (b) show the queue lengths with constant step sizes 2α and 0.5α, respectively. In both cases, the queues are stable, and the queue lengths are roughly inversely proportional to the step size (since **r** is about the same to achieve the same service rate).

Fig 4 show the case with diminishing step size: $\alpha(t)$ approaches 0 when t increases. Although **r** tends to converge,

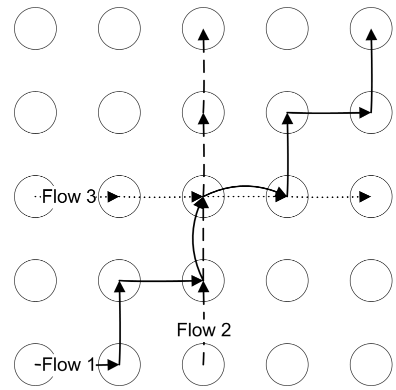


(a) Queue lengths

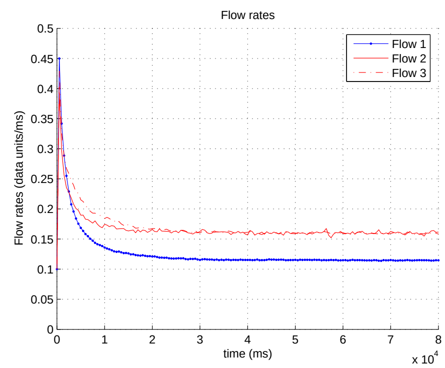


(b) The vector $\mathbf{r}$

Figure 4. Diminishing step sizes



(a) Network 2 and flow directions



(b) Flow rates

Figure 5. Flow rates in Network 2 (Grid Topology) with Joint scheduling and rate control

the queue lengths oscillate violently. This is because with such step sizes, $\mathbf{r}$ loses track of the queue lengths (i.e., they are not proportional to each other as in the case of constant step sizes.) Also, as $\alpha(t)$ becomes very small when t becomes large, $\mathbf{r}$ is not responsive to the variations of queue lengths. At that time if the arrival rates change, the algorithm is slow to adapt to that. This is why we choose constant step sizes.

B. Joint scheduling and rate control

In Fig 5, 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 link contention graph. Fig 5 (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 are two nodes, one in each link, are 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 functions of all flows are $\log(\cdot)$. The scaling factor $\beta = 4.5$. (Note that ihe input rates are adjusted by the flow control algorithm instead of being specified as in the last subsection.)

Fig 5 (b) shows the evolution of the flow rates (using Algorithm 2). We see that they become relatively smooth after an initial convergence. By directly solving (17), we find that the theoretical optimal flow rates for the three flows are 0.1111, 0.1667 and 0.1667 (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.

VII. Conclusion

In this paper, we have proposed and studied a fully distributed scheduling algorithm, and showed that it can support any strictly feasible arrival rates in wireless networks with a general interference model. Furthermore, we have combined it with end-to-end flow control to approach the optimal utility. The algorithm is based on CSMA and is easy to implement. The simulation results are encouraging.

As mentioned before, the current proof of the throughput optimality is based on the stationary distribution of the CSMA

Markov chain. Having the stationary distribution is sufficient, but may not be necessary to achieve the maximal throughput. However, identifying the exact conditions on the step sizes is difficult since they may depend on network size, network topology, and arrival rates, etc. So this problem needs further study.

In this paper we have made a few idealized assumptions. In the future, we would like to study the effects of some practical issues, such as packet collisions and hidden terminal problem, on the performance of the algorithm. For example, if link k transmits too aggressively (large r_k), it may cause excessive collisions, so an upper bound of r_k is needed in practice.

REFERENCES

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

[2] 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.

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

[4] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," In Proceedings of IEEE Infocom, pages 533-539, 1998.

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

[10] 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

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

[12] Herbert Robbins. Sutton Monro, "A Stochastic Approximation Method", The Annals of Mathematical Statistics, Vol. 22, No. 3, 400-407. Sep., 1951.

[13] 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.

[14] Xiaojun Lin and Ness B. Shroff, "Joint Rate Control and Scheduling in Multihop Wireless Networks," Technical Report, Purdue University, 2004

VIII. APPENDIX

A. Proof the Proposition 2

Consider the following convex optimization problem (same as (11))

$$\begin{aligned} \max_{\mathbf{u}} \quad & -\sum_i u_i \log(u_i) \\ \text{s.t.} \quad & \sum_i u_i \cdot x_k^i \geq \lambda_k, \forall k \\ & u_i \geq 0, \sum_i u_i = 1 \end{aligned} \quad (22)$$

where λ is in the interior of the capacity region, that is, it can be written as $\lambda = \sum_i \bar{p}_i \cdot x^i$ where $\bar{p}_i > 0, \forall x^i$ and $\sum_i \bar{p}_i = 1$.

The problem 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 and we denote it as d^*.

Since λ is in the interior of the capacity region, there exists $\lambda' > \lambda$ (element-wise) which is also in the interior. So λ' can be written as $\lambda' = \sum_i \bar{p}'_i \cdot x^i$ where $\bar{p}'_i > 0, \forall x^i$ and $\sum_i \bar{p}'_i = 1$ Note that by letting $u_i = \bar{p}'_i$, all the inqualities are strict, so (22) is strictly feasible, and thus satisfies the Slater condition [11] (page 226-227). As a result, there exist (finite) dual variables $y_k^* \geq 0, w_i^* \geq 0, z^*$ such that the Lagrangian

$$\begin{aligned} & \mathcal{L}(\mathbf{u}; y_k^*, w_i^*, z^*) \\ = \; & -\sum_i u_i \log(u_i) + \sum_k y_k^* (\sum_i u_i \cdot x_k^i - \lambda_k) \\ & + z^* (\sum_i u_i - 1) + \sum_i w_i^* u_i \end{aligned} \quad (23)$$

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

We first claim that the optimal solution satisfies $u_i^* > 0, \forall i$. So the constraints $u_i \geq 0$ is not binding. By complementary slackness, $w_i^* = 0$.

To show this, suppose $u_i^* = 0$ for any i 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 (22), 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.

Continuing the proof, because $w_i^* = 0$, (23) becomes

$$\begin{aligned} & \mathcal{L}(\mathbf{u}; \mathbf{y}^*, w_i^*, z^*) \\ = \; & -\sum_i u_i \log(u_i) + \sum_k y_k^* (\sum_i u_i \cdot x_k^i - \lambda_k) \\ & + z^* (\sum_i u_i - 1) \end{aligned} \quad (24)$$

Since $\mathbf{u}^*$ maximizes $\mathcal{L}(\mathbf{u}; y_k^*, w_i^*, z^*)$, we have

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

Then

$$u_i^* = \exp(\sum_k y_k^* x_k^i + z - 1), \forall i$$

Since $\sum_i u_i^* = 1$, then

$$u_i^* = \frac{\exp(\sum_k y_k^* x_k^i)}{\sum_j \exp(\sum_k y_k^* x_k^j)}, \forall i \quad (25)$$

Plug (25) back into (24), we have $\max_{\mathbf{u}} \mathcal{L}(\mathbf{u}; \mathbf{y}^*, w_i^*, z^*) = -F(\mathbf{y}^*)$. Since $\mathbf{u}^*$ and the dual variables $\mathbf{y}^*$ solves (22), $\mathbf{y}^*$ is the solution of

$$\min_{\mathbf{y} \geq \mathbf{0}} \{-F(\mathbf{y})\}$$

and the optimum is attained. So, by letting $\mathbf{r}^* = \mathbf{y}^*$, $\sup_{\mathbf{r} \geq 0} F(\mathbf{r})$ is attained. The above proof also shows that (4) is the dual problem of (22) (i.e., problem (11)).

B. Proof of the primal-dual relationship between problem (4) and (11)

During the proof in section VIII-A, we have shown that (4) is the dual problem of (11), if the vector of arrival rates is strictly feasible. In this section we show that, not surprisingly, (11) is the dual problem of (4). Also, u_i's are dual variables of (4).

With the assumption of strict feasibility, the optimal value of (4) is attainable. By defining $z_j = \sum_{k=1}^K x_k^j r_k$ (this z_j is unrelated to the dual variable z used in section VIII-A), we re-write (4) as

$$\begin{aligned} \max \quad & \{\sum_k \lambda_k r_k - \log(\sum_j \exp(z_j))\} \\ \text{s.t.} \quad & z_j = \sum_{k=1}^K x_k^j r_k, \forall j \\ & r_k \geq 0, \forall k \end{aligned}$$

The Lagrangian is

$$\begin{aligned} \mathcal{L}(\mathbf{r}, \mathbf{z}; \mathbf{u}, \mathbf{v}) &= \sum_k \lambda_k r_k - \log(\sum_j \exp(z_j)) \\ &\quad + \sum_j u_j (z_j - \sum_{k=1}^K x_k^j r_k) + \sum_k v_k r_k \end{aligned}$$

where $\mathbf{u}$ and $\mathbf{v} \geq 0$ are vectors of dual variables.

Given $\mathbf{u}, \mathbf{v}$, assume that $\mathcal{L}(\mathbf{r}, \mathbf{z}; \mathbf{u}, \mathbf{v})$ is maximized by $\mathbf{z}'$ and $\mathbf{r}'$, then

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{r}', \mathbf{z}'; \mathbf{u}, \mathbf{v})}{\partial z_j} &= -\frac{\exp(z_j')}{\sum_i \exp(z_i')} + u_j = 0 \\ \frac{\partial \mathcal{L}(\mathbf{r}', \mathbf{z}'; \mathbf{u}, \mathbf{v})}{\partial r_k} &= \lambda_k - \sum_j x_k^j u_j + v_k = 0 \end{aligned}$$

That is,

$$\begin{aligned} \frac{\exp(z_j')}{\sum_i \exp(z_i')} &= u_j \quad , \forall j \\ \sum_j x_k^j u_j &\geq \lambda_k, \forall k \end{aligned}$$

Then,

$$\begin{aligned} & \max_{\mathbf{r}, \mathbf{z}} \mathcal{L}(\mathbf{r}, \mathbf{z}; \mathbf{u}, \mathbf{v}) \\ = \; & \mathcal{L}(\mathbf{r}', \mathbf{z}'; \mathbf{u}, \mathbf{v}) \\ = \; & -\log(\sum_j \exp(z_j')) + \sum_j u_j z_j' \\ & + \sum_k r_k' (\lambda_k - \sum_j x_k^j u_j + v_k) \\ = \; & -\log(\sum_j \exp(z_j')) + \sum_j u_j z_j' \\ = \; & \sum_j u_j [z_j' - \log(\sum_j \exp(z_j'))] \\ = \; & \sum_j u_j \log(u_j) \end{aligned}$$

where the second last step follows from $\sum_j u_j = \sum_j \exp(z_j') / \sum_i \exp(z_i') = 1$.

Therefore, the dual problem of (4) is

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_j u_j \log(u_j) \\ \text{s.t.} \quad & \sum_j x_k^j u_j \geq \lambda_k, \forall k / \\ & u_j \geq 0, \sum_j u_j = 1 \end{aligned}$$

which is equivalent to (11). Also, $\mathbf{u}$ are some dual variables of (4).

C. Proof of Proposition 3

Since $\beta \cdot W'$ is the optimal value of problem (16), we have

$$\beta \cdot W' \geq -\sum_i \bar{u}_i \log(\bar{u}_i) + \beta \cdot \bar{W}$$

Because $\beta \cdot W' = -\sum_i \hat{u}_i \log(\hat{u}_i) + \beta \sum_{m=1}^M v_m(\hat{f}_m)$, then

$$\beta[\sum_{m=1}^M v_m(\hat{f}_m) - \bar{W}] \geq -\sum_i \bar{u}_i \log(\bar{u}_i) + \sum_i \hat{u}_i \log(\hat{u}_i)$$

Notice that $|-\sum_i \bar{u}_i \log(\bar{u}_i) + \sum_i \hat{u}_i \log(\hat{u}_i)| \leq K \cdot \log 2$, so

$$\beta[\sum_{m=1}^M v_m(\hat{f}_m) - \bar{W}] \geq -K \cdot \log 2$$

Also, clearly $\bar{W} \geq \sum_{m=1}^M v_m(\hat{f}_m)$, so

$$-\frac{K \cdot \log 2}{\beta} \leq \sum_{m=1}^M v_m(\hat{f}_m) - \bar{W} \leq 0 \quad (26)$$

Therefore, when β is large, the total utility resulting from solving (16) is close to the optimal total utility.