This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

# Energy-Conserving Scheduling in Multi-hop Wireless Networks with Time-Varying Channels

Yang Song[§], Chi Zhang[§], Yuguang Fang[§] and Zhisheng Niu[†]
Email:{yangsong@; zhangchi@; fang@ece.}ufl.edu; niuzhs@tsinghua.edu.cn
[§]Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida 32611
[†]Tsinghua National Lab for Information Science and Technology, Tsinghua University, Beijing 100084, P. R. China

***Abstract*—*MaxWeight* algorithm, a.k.a., *back-pressure* algorithm, has received much attention as a viable solution for dynamic link scheduling in multi-hop wireless networks. The basic principle of the *MaxWeight* algorithm is to select a set of interference-free links with the maximum overall link weights in the network, where the link weight is determined by the queue difference between the transmitter and the receiver. While the throughput-optimality of the *MaxWeight* algorithm is well understood in the literature, the energy consumption induced by the *MaxWeight* algorithm is less studied, which is of great interest in energy-constrained wireless networks such as wireless sensor networks.**

**In this paper, we propose an energy-conserving scheduling scheme, a.k.a., minimum energy scheduling (MES) algorithm for multi-hop wireless networks with stochastic traffic arrivals and time-varying channel conditions. We show that our algorithm is *energy optimal* in the sense that the proposed MES algorithm can achieve an energy consumption which is arbitrarily close to the global minimum solution. Moreover, the energy efficiency of the MES algorithm is achieved without losing the throughput-optimality. In other words, the proposed MES algorithm is still throughput optimal whereas the average consumed energy in the network is significantly reduced, as compared to the traditional *MaxWeight* algorithm. The theoretical results are substantiated via simulations.**

***Index Terms*—Multi-hop Wireless Networks; Performance Analysis; Stochastic Network Optimization; Energy Efficiency; Scheduling Algorithms**

## I. Introduction

There has been a lot of interest over the past few years in characterizing the network capacity region as well as designing efficient scheduling algorithms in multi-hop wireless networks. Due to the stochastic traffic arrivals and time-varying channel conditions, supporting high throughput and high quality communications in multi-hop wireless networks is inherently challenging. To utilize the scarce wireless bandwidth resource effectively, scheduling algorithms which can dynamically allocate the network resource, i.e., select active links, are investigated intensively in the community. For example, *MaxWeight* algorithm, a.k.a., *back-pressure* algorithm, has been extensively studied in the literature, e.g., [1]–[5], following the seminal work of [6]. The *MaxWeight* algorithm enjoys the merit of self-adaptability due to its online nature. In addition, *MaxWeight* algorithm is known to be throughput optimal [7]. That is to say, the *MaxWeight* algorithm can stabilize the network under arbitrary traffic load that can be stabilized by any other possible scheduling algorithms. Therefore, the *MaxWeight* algorithm attracts significant attention and becomes an indispensable component for link scheduling in network protocol designs, e.g., [8]–[10].

While the throughput-optimality of the *MaxWeight* algorithm is well understood, the energy consumption induced by the *MaxWeight* algorithm is less studied in the literature. However, due to the scarcity of energy supplies in wireless nodes, it is imperative to study the energy consumption of the scheduling algorithm which is of special interest in energy-constrained wireless networks such as wireless sensor networks. Is the throughput optimal *MaxWeight* scheduling algorithm also energy optimal? In this paper, we show that the answer to this question is *no*. The reason is that the vast energy consumptions during packet retransmissions are completely neglected by the *MaxWeight* algorithm. For example, in [11], an energy optimal control scheme is proposed where a minimum power expenditure is achieved. However, as in other related works, e.g., [7], the wireless channels in [11] are assumed to be *error-free*, i.e., all the transmissions are assumed to be successful. Nevertheless, in practice, wireless channels are error-prone and data transmissions are subject to random failures due to the hostile channel conditions. Therefore, before a packet can be successfully removed from the transmitter's queue, several transmissions may have occurred, including the original attempt and the posterior retransmissions, which deplete a significant amount of energy for the transmitter. However, in the traditional *MaxWeight* algorithm, such energy-consuming retransmissions induced by channel errors are overlooked. Intuitively, from energy-saving perspective, the possibility that a particular link is selected for transmissions should rely on not only the queue difference between the transmitter and the receiver, which is the design rationale of the traditional *MaxWeight* algorithm, but also the potential energy consumptions of retransmissions induced by erroneous channels. We will make this intuition precise and rigorous in the following sections.

This work was partially supported by the U.S. National Science Foundation under grants CNS-0916391, CNS-0716450, CNS-0721744 and CNS-0626881, and China 111 Project under grant B08038. The work of Z. Niu was also partially supported by the National Basic Research Program of China (973 Program: 2007CB310607) and the Distinguished Young Scholar Program of Natural Science Foundation of China (60925002).

Y. Fang is also a Changjiang Scholar Chair Professor with National Key Laboratory of Integrated Services Networks, Xidian University, China.

In this paper, we propose a minimum energy scheduling (MES) algorithm which consumes an amount of energy that can be pushed arbitrarily close to the global minimum solution. In addition, the energy efficiency attained by the MES algorithm incurs no loss of throughput-optimality. The proposed MES algorithm significantly reduces the overall energy consumption compared to the traditional *MaxWeight* algorithm and remains throughput optimal. Therefore, our proposed MES algorithm is more favorable for dynamic link scheduling and network protocol designs in energy-constrained wireless networks such as wireless sensor networks.

The rest of this paper is organized as follows. Section II describes the system model used in this paper. The proposed MES algorithm is introduced in Section III where the performance analysis is provided. Simulation results are shown in Section IV and Section V concludes this paper.

## II. System Model

We consider a static multi-hop wireless network denoted by a directed graph $(\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ denotes the set of links in the network. We use $|X|$ to represent the cardinality of set $X$. Time is slotted by $t = 0, 1, 2, \cdots$ and in every time slot, the instantaneous channel state of a link $(a, b) \in \mathcal{L}$ is denoted by $S_{a,b}(t)$ where $a$ and $b$ are the transmitter and the receiver of the link. In this paper, we use $\mathbf{S}(t)$ to denote the channel state vector of the whole network. Note that $\mathbf{S}(t)$ remains constant within one time slot, however, it is subject to changes on time slot boundaries. We assume that $\mathbf{S}(t)$ has a finite but potentially large number of possible values and evolves following an irreducible Markovian chain with well defined steady state distributions. Nevertheless, the steady state distribution and the transition probabilities are unknown to the network. Given an instantaneous channel state $S_{a,b}(t)$, the transmission of a packet on link $(a, b)$ is successful with a probability of $p_{a,b}(t)$, if link $(a, b)$ is active and suffers no interference from concurrent transmissions. From the network's perspective, at each time slot $t$, an interference-free link schedule, denoted by $I(t)$, is selected from a feasible set $\Omega(t)$, which is constrained by the underlying interference model, e.g., $K$-hop interference model [12], as well as other limitations such as duplex constraints and peak power limitations. Denote $\widetilde{u_{a,b}}(t)$ as the *nominal* link rate if link $(a, b)$ is selected by the network and the channel is error-free, i.e., $p_{a,b}(t) = 1$. The *actual* data rate of link $(a, b)$ is hence represented by $u_{a,b}(t) = \widetilde{u_{a,b}}(t) p_{a,b}(t)$. We assume that $u_{a,b}(t)$ is upper bounded by a constant $u^{\max}$ for all $(a, b) \in \mathcal{L}$. In practice, $u^{\max}$ can be determined by the number of antennas equipped in a single node as well as the coding/modulation schemes available to the network. We denote $\mathbf{u}(t)$ as the link rate vector of the network at time $t$. Apparently, $\mathbf{u}(t)$ is a function of both $I(t)$ and $\mathbf{S}(t)$.

The network consists of $|\mathcal{F}|$ flows indexed by $f = 1, 2, \cdots, |\mathcal{F}|$. Each flow $f$ is associated with a routing path $R_f = [n_0^f, n_1^f, \cdots, n_{\iota_f}^f]$ where $n_j^f, j = 0, \cdots, \iota_f$ denotes the nodes on the path of flow $f$. At each time slot $t$, the stochastic exogenous arrivals of flow $f$, i.e., the number of new packets that are initiated by node $n_0^f$, is denoted by $A_f(t)$. For the ease of exposition, we assume that $A_f(t)$ is i.i.d. for every time slot with an average rate of $\lambda_f$. In addition, the arrival processes of all flows are assumed to be independent. We further assume that the maximum number of new packets generated by a flow during one time slot is upper bounded, i.e., $A_f(t) \leq A^{\max}, \forall f, t$. We emphasize that the i.i.d. assumption incurs no loss of generality and our model can be extended to cases where $A_f(t)$ is non-stationary in a straightforward fashion, as in [1].

Every node in the network maintains a separate queue for each flow that passes through it. Denote $Q_n^f(t)$ as the queue backlog at time $t$, for node $n$, where $f$ is one of the flows that traverses through $n$, i.e., $n \in R_f$. We assume that $Q_n^f(t) = 0, \forall t$ if $n = n_{\iota_f}^f$. That is to say, if a packet reaches the destination, we consider the packet as leaving the network immediately. Define the *overflow function* of $Q_n^f$ as

$$O(M) = \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \Pr(Q_n^f(\tau) > M). \tag{1}$$

The queue is stable if [1]

$$\lim_{M \to \infty} O(M) \to 0 \tag{2}$$

and the network is stable if all individual queues are stable concurrently.

For notation brevity, we define the network admission rate vector as $\boldsymbol{\lambda} = \{\lambda_{n,f}, \forall n, f\}$ where $\lambda_{n,f}$ is the average exogenous arrival rate of flow $f$ on queue[1] $Q_n^f$. We have $\lambda_{n,f} = \lambda_f$ if $n = n_0^f$ and $\lambda_{n,f} = 0$ otherwise. Denote $\Lambda$ as the *network capacity region* [6], namely, the set of all feasible admission rate vectors, i.e., $\boldsymbol{\lambda}$, that the network can support, in the sense that there exists a scheduling algorithm which stabilizes the network under traffic load $\boldsymbol{\lambda}$. It is shown in [6] and [1] that $\Lambda$ is convex, closed and bounded.

Without loss of generality, we consider an energy consumption model as follows. Recall that the successful transmission probability of link $(a, b)$ is $p_{a,b}(t)$. Therefore, for a particular packet to be transmitted from $a$ to $b$, on average, a number of $\frac{1}{p_{a,b}(t)}$ transmissions are needed in order to "erase" this packet from the queue of node $a$. From the transmitter's perspective, however, every transmission, either the original attempt or retransmissions, costs the same amount of energy. Denote $\alpha_{a,b}$ as the energy needed to transmit a packet on link $(a, b)$. For example, $\alpha_{a,b}$ can be proportional to the distance between node $a$ and $b$. Therefore, in order to transmit a packet successfully on $(a, b)$, node $a$ needs to spend a total energy of $\frac{\alpha_{a,b}}{p_{a,b}(t)}$ on average. For the receiver $b$, we assume that the energy depletion on *successful* packets receptions are dominant, i.e., the energy spent for overhearing and short ACK messages are neglected. Denote $\beta_{a,b}$ as the energy consumed for a successful packet reception in demodulation

[1] Note that with a slight abuse of notation, we use $Q_n^f$ to denote both the queue itself and the number of packets in the queue.

and decoding on node $b$. With a data rate[2] of $u_{a,b}(t)$ on link $(a,b)$, the overall energy spent during time slot $t$ is given by

$$\begin{aligned} G_{a,b}(t) &= u_{a,b}(t)\left(\frac{\alpha_{a,b}}{p_{a,b}(t)} + \beta_{a,b}\right) \\ &= \widetilde{u_{a,b}}(t)\left(\alpha_{a,b} + \beta_{a,b} p_{a,b}(t)\right). \end{aligned} \tag{3}$$

Note that due to the stochastic nature of wireless channels, $G_{a,b}$ is a random variable. We stress that the simple energy consumption model above is not essential and our analysis can be extended to other more complex forms of energy models straightforwardly, as will be shown in the next section.

In addition, we assume that the energy consumption of the whole network during one single time slot is upper bounded, i.e.,

$$\sum_{(a,b)\in\mathcal{L}} G_{a,b}(t) \le G^{\max}, \forall t. \tag{4}$$

Therefore, to minimize the energy consumption, the objective of the network is to find a scheduling algorithm which solves

Minimum Energy Scheduling Problem:

$$\text{minimize} \quad C = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \overline{G}(t) \tag{5}$$

*s.t.*

the network remains stable, and

$$\overline{G}(t) = E\left(\sum_{(a,b)\in\mathcal{L}} h_{a,b}(t)\left(\frac{\alpha_{a,b}}{p_{a,b}(t)} + \beta_{a,b}\right)\right) \tag{6}$$

is the *expected* overall network energy consumption during time slot $t$, with respect to the randomness of arrival processes and channel variations. Note that $h_{a,b}(t)$ is the actual number of successfully transmitted packets on link $(a,b)$ during time slot $t$ and[3] $h_{a,b}(t) \le u_{a,b}(t)$.

In the next section, we will propose a minimum energy scheduling (MES) algorithm which minimizes the average network energy consumption asymptotically subject to network stability. In addition, the proposed MES algorithm is throughput optimal, in the sense that the MES algorithm can ensure the network stability for all feasible network admission rate vectors in the network capacity region. Restated, the set of feasible arrival rates supported by the MES algorithm is the superset of all other possible scheduling algorithms, including those with a priori knowledge on the futuristic arrivals and channel conditions.

## III. Minimum Energy Scheduling Algorithm

### A. Algorithm Description

The minimum energy scheduling (MES) algorithm is given as follows.

**MES ALGORITHM:**

**At every time slot $t$:**

- Every link $(a,b)\in\mathcal{L}$ finds the flow $f^*$ which maximizes
$$\max_{f:(a,b)\in R_f} \left(2Q_a^f(t) - \frac{J\alpha_{a,b}}{p_{a,b}(t)} - 2Q_b^f(t) - J\beta_{a,b}\right) \tag{7}$$
where $J$ is a positive constant which is tunable as a system parameter.
- Every link $(a,b)\in\mathcal{L}$ calculates the link weight as
$$H_{a,b}(t) = \left[2Q_a^{f^*}(t) - \frac{J\alpha_{a,b}}{p_{a,b}(t)} - 2Q_b^{f^*}(t) - J\beta_{a,b}\right]^+ \tag{8}$$
where $[x]^+$ denotes $\max(x,0)$.
- For the network, a link schedule $I^*(t)$ is selected which solves
$$\max_{I(t)\in\Omega(t)} \sum_{(a,b)\in\mathcal{L}} u_{a,b}(t) H_{a,b}(t) \tag{9}$$

**END**

*Remark 1:* Note that the proposed MES algorithm is different from the *MaxWeight* algorithm proposed in [6], [13]. In the original *MaxWeight* algorithm, the weight of a particular link $(a,b)$ is the queue difference between node $a$ and $b$. However, in the MES algorithm, as indicated by (8), the link weight $H_{a,b}(t)$ is related to the potential energy consumptions on this link under the current channel condition as well. More specifically, the link weight in the MES algorithm is the queue difference subtracted by a weighted energy consumption factor, i.e., $J\left(\frac{\alpha_{a,b}}{p_{a,b}(t)} + \beta_{a,b}\right)$, where $J$ represents the weight. Intuitively, if the current channel is unfavorable, i.e., $p_{a,b}(t)$ is small, the energy required for a successful transmission is remarkably large due to the retransmissions and thus the link should be selected less likely. Therefore, the link weight in the MES algorithm, i.e., $H_{a,b}(t)$, can be viewed as a balance between the queue difference and the energy consumptions under the current channel condition. More specifically, in the original *MaxWeight* algorithm, if a link has a larger queue difference, the link is more likely to be selected for transmissions. However, in the MES algorithm, both the queue difference and the energy expenditure are taken into consideration. In addition, as indicated by (7), by replacing $J\left(\frac{\alpha_{a,b}}{p_{a,b}(t)} + \beta_{a,b}\right)$ with other metrics, our MES algorithm can incorporate other forms of energy consumption models straightforwardly.

*Remark 2:* Observe that if $J = 0$, the MES algorithm reduces to the original *MaxWeight* algorithm, i.e., the energy consumptions during retransmissions are omitted. Therefore, the *MaxWeight* algorithm is a special case of our MES algorithm. On the other hand, if we let $J \to \infty$, the performance of the MES algorithm can be pushed arbitrarily close to the global minimum solution, as will be shown analytically in Section III-C. However, as illustrated in (7) and (8), when $J$ increases, the network becomes more reluctant to transmissions, for the sake

[2] The unit of data rate in this paper is defined as packets/slot. It is worth noting that other units such as bits/slot are also applicable.

[3] The inequality holds when node $a$ has less packets to transmit than the allocated data rate $u_{a,b}(t)$.

of energy conservation, unless the accumulated queue backlogs are significantly large. Intuitively, a larger average queue size induces a longer experienced delay for transmissions. Therefore, the system parameter $J$ is essentially a control knob which provides a tradeoff between the energy-optimality and the experienced delay in the network.

*Remark 3:* Note that similar to the traditional *MaxWeight* algorithm, the calculation in (9) is centralized. However, following [6], much progress has been made in easing the computational complexity and deriving decentralized solutions for the centralized *MaxWeight* algorithm, e.g., [3], [5], [9], [10], [12], [14]–[21]. It is worth noting that solving (9) is equivalent to finding the maximum weight independent set in the conflict graph, which is combinatorial in nature and thus is intrinsically difficult to solve. A natural heuristic, denoted by *GreedyMax*, is described as follows. First, the link with the highest weight in the network is selected. Next, the link with the second highest weight which does not involve conflicts with any previously selected links, is selected and the iteration continues until no link can be added. Based on *GreedyMax*, a novel pre-partition based approach is introduced in [19]. The authors prove that if the topology of the network satisfies certain conditions, a.k.a., local pooling factor conditions, *GreedyMax* can achieve the same throughput as *MaxWeight*. Furthermore, tree based topology are shown to satisfy the local pooling conditions. In light of this, [19] utilizes graph algorithms to partition the whole network into trees where each tree is allocated an orthogonal channel. As a result, the whole network can operate with simple *GreedyMax* algorithm and achieve the same throughput as *MaxWeight*. This line of research is further simplified by a recent work of [18] where the author proves that a *local* greedy maximal algorithm can obtain the same performance as the global *GreedyMax* algorithm. Specifically, in each time slot, a link only needs to compare its weight with local neighboring links to decide a feasible transmission schedule. Therefore, in tandem with the tree pre-partition method in [19], the complex scheduling algorithm in (9) can be implemented in a fully distributed fashion. Another feasible direction is to utilize the distributed random access approximation schemes, e.g., [5], [9], [21]. For example, in [9], each node in the network utilizes an IEEE 802.11 MAC protocol where the channel access probability is dynamically adjusted in accordance to the link weight, i.e., $H_{a,b}(t)$ in our scenario. The effectiveness of such random access based distributed approximations is studied extensively in [10] and [9] . We note that, although distributed implementation is not the focus of this paper, our proposed MES algorithm can be approximated well by the solutions suggested in the above papers.

*Remark 4:* If an admission control mechanism is implemented in the network to regulate the overwhelming arrivals, in order to achieve an average minimum rate provision for each flow, we can utilize the concept of *virtual queues* introduced in [4], [7], where for each flow $f$, we define a virtual queue $\mathcal{Y}_f(t)$ which is initially empty, i.e.,

$$\mathcal{Y}_f(0) = 0, \forall f \tag{10}$$

and the virtual queue updating dynamic is defined as

$$\mathcal{Y}_f(t+1) = [\mathcal{Y}_f(t) - R_f(t)]^+ + \delta_f \quad \forall f \tag{11}$$

where $R_f(t)$ is the admitted traffic for flow $f$ at time slot $t$ and $\delta_f$ is a feasible minimum rate requirement of flow $f$. Intuitively, if each of the virtual queues in the system is stable, the average arrival rate should be less than the average departure rate of the virtual queue and hence we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} R_f(t) \geq \delta_f, \forall f \tag{12}$$

which is exactly the desired minimum rate requirement for flow $f$. Moreover, the virtual queues are easy to implement. For example, the source node of flow $f$ can maintain a software based counter to measure the backlog updates of virtual queue $\mathcal{Y}_f(t)$. Therefore, the minimum rate requirements can be incorporated into the proposed scheme straightforwardly.

### B. Throughput-optimality

We first show that the proposed MES algorithm is throughput optimal, as given in the following theorem.

*Theorem 1:* The proposed MES algorithm is throughput optimal, i.e., for an arbitrary network admission rate vector $\boldsymbol{\lambda}$ which is inside of the network capacity region $\Lambda$, MES stabilizes the network under $\boldsymbol{\lambda}$.

*Proof:* We first provide the queue updating equation of $Q_n^f$. Note that a packet is removed from the transmitter's queue if and only if it is received by the receiver successfully. Therefore, the queue updating dynamic is given by

$$Q_n^f(t+1) \leq [Q_n^f(t) - u_{n,f}^{out}(t)]^+ + u_{n,f}^{in}(t) + A_{n,f}(t) \tag{13}$$

where $u_{n,f}^{out}(t)$ and $u_{n,f}^{in}(t)$ are the allocated data rate on the outgoing link and the incoming link of node $n$, with respect to flow $f$. Note that $u_{n,f}^{out}(t) = \widetilde{u}_{n,f}^{out}(t) p_{n,c}^{out}(t)$ and $p_{n,c}^{out}(t)$ is the current successful transmission probability on this link. Note that $u_{n,f}^{out}(t) = 0, \forall t$ if node $n$ is the destination node of flow $f$ and $u_{n,f}^{in}(t) = 0, \forall t$ if node $n$ is the source node of flow $f$. Furthermore, $A_{n,f}(t) = A_f(t)$ if $n$ is the source node of flow $f$ and $A_{n,f}(t) = 0$ otherwise.

From (13), we have

$$\begin{aligned}(Q_n^f(t+1))^2 - (Q_n^f(t))^2 &\leq \left((u^{\max})^2 + (u^{\max} + A^{\max})^2\right) \\ &-2Q_n^f(t) \left(u_{n,f}^{out}(t) - u_{n,f}^{in}(t) - A_{n,f}(t)\right).\end{aligned} \tag{14}$$

Next, we sum (14) over the whole network on all data queues and obtain

$$\begin{aligned}&\sum_{n,f} (Q_n^f(t+1))^2 - \sum_{n,f} (Q_n^f(t))^2 \\ &\leq B - 2\sum_{n,f} Q_n^f(t) \left(u_{n,f}^{out}(t) - u_{n,f}^{in}(t) - A_{n,f}(t)\right)\end{aligned} \tag{15}$$

where

$$B = |\mathcal{N}||\mathcal{F}|\left((u^{\max})^2 + (u^{\max} + A^{\max})^2\right) \tag{16}$$

is a constant.

Denote $\boldsymbol{Q}(t) = \{Q^f_n(t), \forall f, n\}$ as the instantaneous queue backlogs in the network. We take the conditional expectation with respect to $\boldsymbol{Q}(t)$ on (15) and have

$$\begin{aligned}&E\left(\sum_{n,f}(Q^f_n(t+1))^2|\boldsymbol{Q}(t)\right) - E\left(\sum_{n,f}(Q^f_n(t))^2|\boldsymbol{Q}(t)\right)\\ &\le B - 2\sum_{n,f} Q^f_n(t)E\left(u^{out}_{n,f}(t) - u^{in}_{n,f}(t) - A_{n,f}(t)|\boldsymbol{Q}(t)\right).\end{aligned}$$

Define

$$G^{MES}_Q(t) = E\left(\sum_{n,f}\left(h^{out}_{n,f}(t)\frac{\alpha^{out}_{n,f}}{p^{out}_{n,f}(t)} + h^{in}_{n,f}(t)\beta^{in}_{n,f}\right)|\boldsymbol{Q}(t)\right)$$

where $\alpha^{out}_{n,f}$ ($\beta^{in}_{n,f}$) denotes the energy needed for a packet transmission (reception) on the outgoing (incoming) link of node $n$, with respect to flow $f$.

In addition, we define

$$\overline{G^{MES}}(t) = E\left(\sum_{n,f}\left(h^{out}_{n,f}(t)\frac{\alpha^{out}_{n,f}}{p^{out}_{n,f}(t)} + h^{in}_{n,f}(t)\beta^{in}_{n,f}\right)\right)$$

as the expected network-wide energy consumption during time slot $t$, by following the proposed MES algorithm. Apparently, we have

$$E\left(G^{MES}_Q(t)\right) = \overline{G^{MES}}(t).$$

Next, we add both sides by $JG^{MES}_Q(t)$ where $J$ is a positive constant, and have

$$\begin{aligned}&E\left(\sum_{n,f}(Q^f_n(t+1))^2|\boldsymbol{Q}(t)\right) - E\left(\sum_{n,f}(Q^f_n(t))^2|\boldsymbol{Q}(t)\right)\\ &+JG^{MES}_Q(t)\\ &\le B - 2\sum_{n,f} Q^f_n(t)E\left(u^{out}_{n,f}(t) - u^{in}_{n,f}(t) - A_{n,f}(t)|\boldsymbol{Q}(t)\right)\\ &+JG^{MES}_Q(t)\\ &\le B - 2\sum_{n,f} Q^f_n(t)E\left(u^{out}_{n,f}(t) - u^{in}_{n,f}(t) - A_{n,f}(t)|\boldsymbol{Q}(t)\right)\\ &+JE\left(\sum_{n,f}\left(u^{out}_{n,f}(t)\frac{\alpha^{out}_{n,f}}{p^{out}_{n,f}(t)} + u^{in}_{n,f}(t)\beta^{in}_{n,f}\right)|\boldsymbol{Q}(t)\right)\end{aligned} \tag{17}$$

Denote the R.H.S. of (17) as $\Theta$. It is of great importance to observe that, from the algorithm description of the MES algorithm above, at every time slot $t$, the MES algorithm essentially minimizes the R.H.S. of (17) over all possible scheduling algorithms.

Since $\boldsymbol{\lambda}$ lies in the interior of the network capacity region $\Lambda$, it immediately follows that there exists a small positive constant $\epsilon > 0$ such that

$$\boldsymbol{\lambda} + \boldsymbol{\epsilon} = \{(\lambda_{1,1}+\epsilon), \cdots, (\lambda_{n,f}+\epsilon), \cdots\} \in \Lambda. \tag{18}$$

By invoking Corollary 3.9 in [7], we claim that there exists a randomized scheduling policy, denoted by $RA$, which stabilizes the network while providing a data rate of

$$E\left(\overline{u^{out}_{n,f}}(t) - \overline{u^{in}_{n,f}}(t)|\boldsymbol{Q}(t)\right) = \lambda_{n,f} + \epsilon, \forall t \tag{19}$$

where $\overline{u^{out}_{n,c}}(t)$, $\overline{u^{in}_{n,c}}(t)$ are the link data rates induced by the randomized policy $RA$. Therefore, we have

$$\begin{aligned}\Theta^{RA} &= B - 2\epsilon\textstyle\sum_{n,f} Q^f_n(t)\\ &+JE\left(\textstyle\sum_{n,f}\left(\overline{u^{out}_{n,f}}(t)\frac{\alpha^{out}_{n,f}}{p^{out}_{n,f}(t)} + \overline{u^{in}_{n,f}}(t)\beta^{in}_{n,f}\right)|\boldsymbol{Q}(t)\right).\end{aligned} \tag{20}$$

Note that the last term in (20) is the actual energy consumption by $RA$ algorithm during time slot $t$. Following (4), we have

$$\Theta^{RA} \le B - 2\epsilon\sum_{n,f} Q^f_n(t) + JG^{\max}. \tag{21}$$

In light of (17), we have

$$\begin{aligned}&E\left(\textstyle\sum_{n,f}(Q^f_n(t+1))^2|\boldsymbol{Q}(t)\right) - E\left(\textstyle\sum_{n,f}(Q^f_n(t))^2|\boldsymbol{Q}(t)\right)\\ &+JG^{MES}_Q(t) \le \Theta^{MES} \le \Theta^{RA}\\ &\le B - 2\epsilon\textstyle\sum_{n,f} Q^f_n(t) + JG^{\max}.\end{aligned} \tag{22}$$

Next, we take the expectation with respect to $\boldsymbol{Q}(t)$ on (22) and have

$$\begin{aligned}&E\left(\textstyle\sum_{n,f}(Q^f_n(t+1))^2\right) - E\left(\textstyle\sum_{n,f}(Q^f_n(t))^2\right)\\ &+J\overline{G^{MES}}(t) \le B - 2\epsilon E\left(\textstyle\sum_{n,f} Q^f_n(t)\right) + JG^{\max}.\end{aligned} \tag{23}$$

Note that the above inequality holds for any time slot $t$. Hence, we sum over from time slot 0 to $T-1$ and obtain

$$\begin{aligned}&E\left(\textstyle\sum_{n,f}(Q^f_n(T))^2\right) - E\left(\textstyle\sum_{n,f}(Q^f_n(0))^2\right)\\ &+J\textstyle\sum_{t=0}^{T-1}\overline{G^{MES}}(t)\\ &\le TB - 2\epsilon\textstyle\sum_{t=0}^{T-1}\sum_{n,f}E\left(Q^f_n(t)\right) + TJG^{\max}.\end{aligned} \tag{24}$$

Next, we rearrange terms and divide (24) by $T$ and have

$$\begin{aligned}&2\epsilon\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n,f}E\left(Q^f_n(t)\right)\\ &\le B + JG^{\max} + \frac{E\left(\sum_{n,f}(Q^f_n(0))^2\right)}{T}\\ &-J\frac{1}{T}\sum_{t=0}^{T-1}\overline{G^{MES}}(t) - \frac{E\left(\sum_{n,f}(Q^f_n(T))^2\right)}{T}.\end{aligned} \tag{25}$$

Note that the last two terms of (25) are both non-positive. By taking $\limsup_{T\to\infty}$ on both sides of (25), we attain

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n,f}E\left(Q^f_n(t)\right) \le \frac{B + JG^{\max}}{2\epsilon} < \infty. \tag{26}$$

Therefore, for every individual queue $Q_n^f$, we have

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}E\left(Q_n^f(t)\right)\le\frac{B+JG^{\max}}{2\epsilon}<\infty. \tag{27}$$

Finally, by invoking Markov inequality, we have

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\Pr\left(Q_n^f(t)>M\right)<\frac{B+JG^{\max}}{2\epsilon M}. \tag{28}$$

Therefore, by taking $\lim_{M\to\infty}$, we obtain the stability result of the MES algorithm and thus completes the proof. ■

### C. Asymptotic Energy-optimality

In this section, we show that the MES algorithm yields an asymptotic optimal solution to (5), i.e., the average energy consumption induced by the MES algorithm can be arbitrarily close to the global minimum solution of (5), by selecting a sufficiently large value of $J$.

*Theorem 2:* Define $C^{MES}$ and $C^*$ as the average energy consumption induced by the MES algorithm and the optimal (minimum) solution of (5), respectively. The performance of the MES algorithm is given by

$$C^{MES}\le C^*+\frac{B}{J} \tag{29}$$

where $B$ is defined in (16). Therefore, by choosing $J\to\infty$, the performance of the MES algorithm can be pushed arbitrarily close to the optimum solution $C^*$.

*Proof:* First, denote the optimal sequence of link rates, which generates the optimum solution $C^*$, as $\boldsymbol{u}^*(0),\boldsymbol{u}^*(1),\cdots,\boldsymbol{u}^*(t),\cdots$. Next, let us consider a *deterministic* policy, denoted by $DE$, which allocates exactly the optimum link data rates on every time slot $t$. Similar to (22), we have

$$E\left(\textstyle\sum_{n,f}(Q_n^f(t+1))^2|\boldsymbol{Q}(t)\right)-E\left(\textstyle\sum_{n,f}(Q_n^f(t))^2|\boldsymbol{Q}(t)\right) +JG_Q^{MES}(t)\le\Theta^{MES}\le\Theta^{DE} \tag{30}$$

where

$$\begin{aligned}&\Theta^{DE}\\&=B-2\sum_{n,f}Q_n^f(t)E\left(u_{n,f}^{*out}(t)-u_{n,f}^{*in}(t)-A_{n,f}(t)|\boldsymbol{Q}(t)\right)\\&+JE\left(\sum_{n,f}\left(u_{n,f}^{*out}(t)\frac{\alpha_{n,f}^{out}}{p_{n,f}^{out}(t)}+u_{n,f}^{*in}(t)\beta_{n,f}^{in}\right)|\boldsymbol{Q}(t)\right)\end{aligned} \tag{31}$$

It is worth noting in (31), only the data rates are replaced by the ones generated by the $DE$ algorithm whereas all other values remain the same. Note that in this case, the optimal data rates $u_{n,f}^{*out}(t)$ and $u_{n,f}^{*in}(t)$ are known as a priori by the $DE$ algorithm and thus are constants with respect to $Q_n^f(t)$.

We take the expectation on both sides of (30) and have

$$\begin{aligned}&E\left(\sum_{n,f}(Q_n^f(t+1))^2\right)-E\left(\sum_{n,f}(Q_n^f(t))^2\right)\\&+J\overline{G^{MES}}(t)\le B\\&-2E\left(\sum_{n,f}Q_n^f(t)E\left(u_{n,f}^{*out}(t)-u_{n,f}^{*in}(t)-\lambda_{n,f}|\boldsymbol{Q}(t)\right)\right)\\&+JE\left(\sum_{n,f}\left(u_{n,f}^{*out}(t)\frac{\alpha_{n,f}^{out}}{p_{n,f}^{out}(t)}+u_{n,f}^{*in}(t)\beta_{n,f}^{in}\right)\right).\end{aligned} \tag{32}$$

We emphasize that, however, in this scenario, at arbitrary time slot $t$, the value of

$$u_{n,f}^{*out}(t)-u_{n,f}^{*in}(t)-\lambda_{n,f} \tag{33}$$

could be either non-positive or nonnegative. In other words, the relationship of (19) does not hold. To circumvent this, we first sum (32) over time slots $t=0,\cdots,T-1$ and divide it by $T$. Thus, we attain

$$\begin{aligned}&J\frac{1}{T}\sum_{t=0}^{T-1}\overline{G^{MES}}(t)\le B\\&+J\frac{1}{T}\sum_{t=0}^{T-1}E\left(\sum_{n,f}\left(u_{n,f}^{*out}(t)\frac{\alpha_{n,f}^{out}}{p_{n,f}^{out}(t)}+u_{n,f}^{*in}(t)\beta_{n,f}^{in}\right)\right)\\&+\frac{E\left(\sum_{n,f}(Q_n^f(0))^2\right)}{T}-2\frac{1}{T}\sum_{t=0}^{T-1}\Xi(t)\end{aligned}$$

where

$$\Xi(t)=E\left(\sum_{n,f}Q_n^f(t)E\left(u_{n,f}^{*out}(t)-u_{n,f}^{*in}(t)-\lambda_{n,f}|\boldsymbol{Q}(t)\right)\right). \tag{34}$$

By taking $\limsup_{T\to\infty}$, we have

$$\begin{aligned}&J\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\overline{G^{MES}}(t)\le B+JC^*\\&-2\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\Xi(t).\end{aligned} \tag{35}$$

Note that

$$\lim(AB)=\lim(A)\lim(B) \tag{36}$$

if $\lim(A)$ and $\lim(B)$ exist and are bounded. Recall that in the previous section, we have shown that

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n,f}E\left(Q_n^f(t)\right) \tag{37}$$

exists and is finite. Moreover, since $u_{n,f}^{*out}(t)$ and $u_{n,f}^{*in}(t)$ are the optimum solution to (5), the network stability is achieved under $\boldsymbol{\lambda}$. Therefore, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} u_{n,f}^{*out}(t) \geq \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} u_{n,f}^{*in}(t) + \lambda_{n,f}$$

since otherwise, the network stability cannot be ensured by $\boldsymbol{u}^*$. Finally, we conclude that

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \Xi(t) \tag{38}$$

is nonnegative and thus the last term of (35) can be omitted. Consequently, by dividing $J$ on both sides of (35), we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \overline{G^{MES}}(t) \leq \frac{B}{J} + C^*. \tag{39}$$

Therefore, Theorem 2 holds. ■

*Remark 5:* We note that the system parameter $J$ represents the tradeoff between the average network delay and the energy expenditure of the system. In other words, if the incoming flows are tolerant to a larger experienced delay, the network is able to utilize the proposed MES algorithm to reduce the overall energy consumption significantly while accommodating the network traffic. In practice, the parameter of $J$ should be a sufficiently large yet upper bounded value which is determined by the delay tolerance of the network flows. The proposed MES algorithm, parameterized by $J$, hence provides an energy efficient solution if a longer network delay can be tolerated. The question of how to dynamically tune the value of $J$ according to different network applications is interesting and remains as future research.

## IV. Simulations

We consider a multi-hop wireless network illustrated in Figure 1. There are three flows in the network, denoted by *Flow 1, 2, 3*. The routing paths of flows are specified by $R_1 = \{A, B, C, D\}$, $R_2 = \{F, G, C, D\}$ and $R_3 = \{E, F, G, H\}$. The exogenous arrival processes are Bernoulli processes with an average rate of 5 packets per slot for all three flows. Without loss of generality, we assume a two-hop interference model which represents the general IEEE 802.11 MAC protocols [5], [18]. The nominal link rate of a wireless link is assumed to be 20 packets per slot. For a particular link, there are three equally possible channel states, i.e., *Good, Medium, Bad* where the corresponding successful transmission probabilities are $0.8, 0.6, 0.3$, respectively.

For the ease of exposition, we assume that $\alpha_{a,b} = \beta_{a,b} = 50\mu J$ for all links in the network [22]. We investigate the average energy consumption induced by the *MaxWeight* algorithm and the proposed MES algorithm with different values of $J$, i.e., 50, 100, 200, 500, 800, 1000, 2000, 5000, 8000, 10000, 12000, 14000, 16000, 18000, 20000, 25000. Every simulation is executed for 10000 time slots.

Figure 2 depicts the average energy consumption per time slot for the *MaxWeight* algorithm and the MES algorithm with different values of $J$. As illustrated in Figure 2, when $J = 0$, the MES algorithm reduces to the original *MaxWeight* algorithm and yields the same amount of energy expenditure. However, as $J$ increases, the energy consumption induced by the MES algorithm decreases remarkably. In addition, as shown analytically in Theorem 2, the MES algorithm approaches to the global minimum energy expenditure gradually as $J$ increases. To achieve a better understanding on the impact of larger values of $J$, we compare the average queue backlogs of all data queues, with $J = 50$, 150, 350 and 500, in Figure 3 and Figure 4, where the queues in the network are indexed by 1 to 9 in the order of $Q_A^1$, $Q_B^1$, $Q_C^1$, $Q_F^2$, $Q_G^2$, $Q_C^2$, $Q_E^3$, $Q_F^3$ and $Q_G^3$. It is worth noting that, as $J$ grows, the average queue backlogs in the network increases correspondingly. Following Little's Law, larger queue backlogs yield longer network delays. Therefore, while the network-wide energy expenditure is noticeably reduced, a larger value of $J$ yields a longer average delay in the network. As a consequence, a tradeoff between the energy-optimality and the experienced delay can be attained by tuning $J$ properly.

So far, throughout this paper, we have assumed that the energy in each node is constrained yet sufficiently large. Next, we investigate the performance of the MES algorithm in multi-

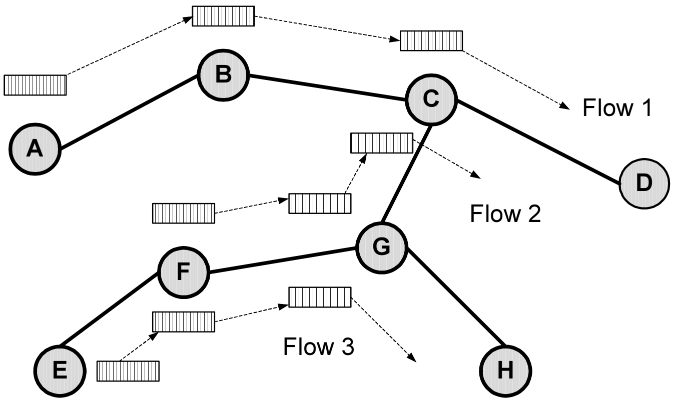

Fig. 1. Network topology with interconnected queues.

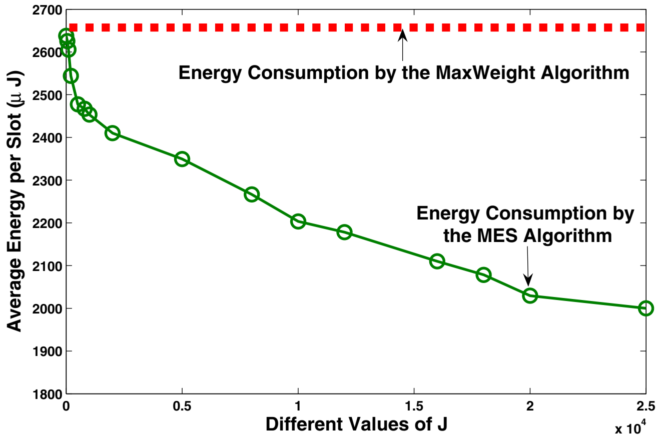

Fig. 2. Comparison of the energy consumptions of the *MaxWeight* algorithm and the MES algorithm.

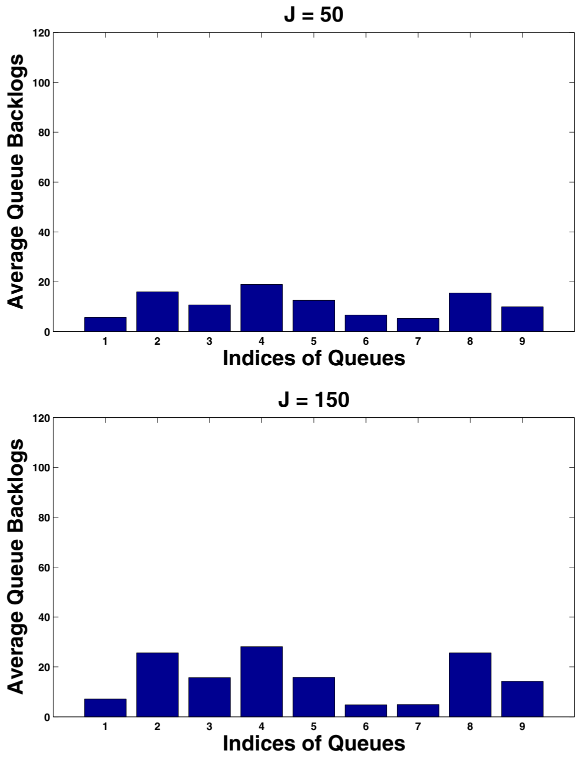

Fig. 3. The average queue backlogs in the network for $J = 50$ and 150.

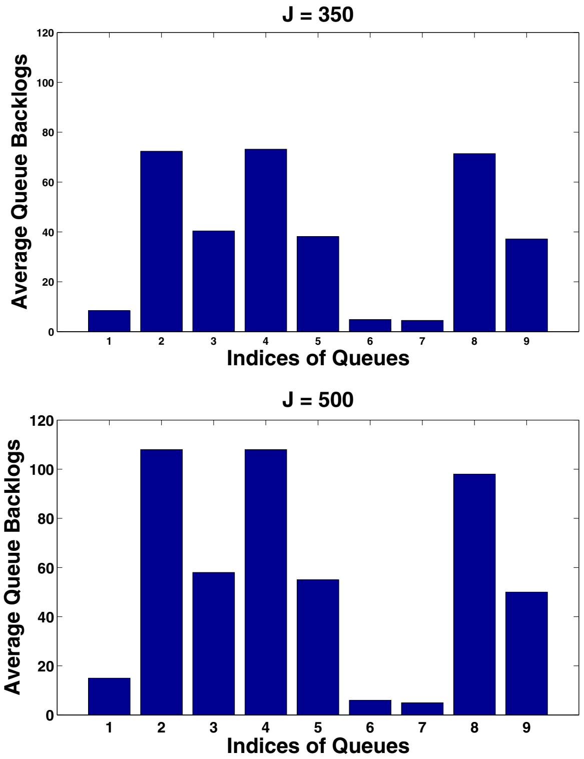

Fig. 4. The average queue backlogs in the network for $J = 350$ and 500.

hop wireless networks where each node has a limited and finite amount of energy. More specifically, the same network in Figure 1 is considered, however, we assume that each node in the network has a battery with an initial energy of 1 Joule. We compare the performance of the *MaxWeight* algorithm with the MES algorithm for different values of $J$, in terms of the *network lifetime*, which is defined as the time instance when the first node in the network depletes the battery completely. The values of $J$ are 50, 200, 500, 800, 1000, 2000, 5000, 8000, 10000, 15000 and 20000.

Figure 5 pictorially compares the network lifetime induced by the *MaxWeight* algorithm and that of the MES algorithm for different values of $J$. Similar to the previous scenario, when $J$ is small, the MES algorithm yields similar performance, in terms of the network lifetime, compared to the *MaxWeight* algorithm. However, when $J$ increases, the MES algorithm significantly outperforms the traditional *MaxWeight* algorithm. It is worth noting that when $J = 20000$, the MES algorithm prolongs the network lifetime by more than twice as much as that of the *MaxWeight* algorithm! By the same token, a tradeoff between the network lifetime and the experienced delay of the network can be controlled effectively by tuning the value of $J$.

## V. Conclusions

In this paper, we investigate the energy consumption issue of the scheduling algorithms in multi-hop wireless networks. We show that the traditional *MaxWeight* algorithm, which is well known to be throughput optimal, is not energy optimal due to the overlook of the energy consumptions induced by inevitable packet retransmissions. In light of this, we propose a minimum energy scheduling (MES) algorithm which significantly reduces the energy consumption compared to the original *MaxWeight* algorithm. In addition, we analytically show that the MES algorithm is essentially energy optimal in the sense that the average energy expenditure of the MES algorithm can be pushed arbitrarily close to the global minimum solution. Moreover, the improvement on the energy efficiency is achieved without losing the throughput-optimality. Therefore, the proposed MES algorithm is of great importance for network protocol designs in energy-constrained multi-hop wireless networks such as wireless sensor networks.

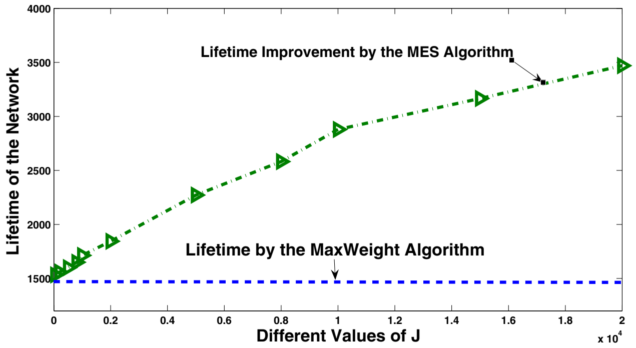

Fig. 5. Comparison of the lifetime of the *MaxWeight* algorithm and the MES algorithm.

## REFERENCES

[1] M. J. Neely, “Dynamic power allocation and routing for satellite and wireless networks with time varying channels,” Ph.D. dissertation, Masssachusetts Institute of Technology, LIDS, 2003.

[2] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control,” *IEEE INFOCOM*, 2005.

[3] X. Lin and N. Shroff, “The impact of imperfect scheduling on cross-layer rate control in wireless networks,” *IEEE INFOCOM*, 2005.

[4] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Systems*, vol. 50, pp. 401–457, 2005.

[5] A. Gupta, X. Lin, and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” *IEEE INFOCOM*, 2007.

[6] 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*, vol. 37, pp. 1936–1949, 1992.

[7] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*. Foundations and Trends in Networking, 2006.

[8] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key, “Horizon: Balancing tcp over multiple paths in wireless mesh network,” *Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobicom)*, 2008.

[9] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, “Joint scheduling and congestion control in mobile ad-hoc networks,” *IEEE INFOCOM*, 2008.

[10] A. Warrier, S. Janakiraman, and I. Rhee, “Diffq: Practical differential backlog congestion control for wireless networks,” *IEEE INFOCOM*, 2009.

[11] M. J. Neely, “Energy optimal control for time varying wireless networks,” *IEEE Transactions on Information Theory*, vol. 52, pp. 2915–2934, July 2006.

[12] Y. Yi, A. Proutiere, and M. Chiang, “Complexity in wireless scheduling: Impact and tradeoffs,” *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2008.

[13] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 89–103, Jan. 2005.

[14] E. Modiano, D. Shah, and G. Zussman, “Maximizing throughput in wireless networks via gossiping,” *ACM SIGMETRICS*, 2006.

[15] L. Jiang and J. Walrand, “A distributed csma algorithm for throughput and utility maximization in wireless networks,” *Proc. Allerton Conf. of Comm, Control, and Computing*, 2008.

[16] S. Rajagopalan and D. Shah, “Reversible networks, distributed optimization, and network scheduling: What do they have in common?” *Proc. Conf. on Information Sciences and Systems (CISS)*, 2008.

[17] X. Wu, R. Srikant, and J. Perkins, “Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks,” *IEEE Transactions on Mobile Computing*, 2007.

[18] C. Joo, “A local greedy scheduling scheme with provable performance guarantee,” *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, 2008.

[19] A. Brzezinski, G. Zussman, and E. Modiano, “Distributed throughput maximization in wireless mesh networks via pre-partitioning,” *IEEE/ACM Transactions on Networking*, 2008.

[20] C. Joo, X. Lin, and N. Shroff, “Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks,” *IEEE INFOCOM*, 2008.

[21] A. Stolyar, “Dynamic distributed scheduling in random access networks,” *Journal of Applied Probability*, vol. 45, pp. 297–313, 2008.

[22] T. Hou, Y. Shi, and H. D. Sherali, “Rate allocation in wireless sensor networks with network lifetime requirement,” *Proceedings of the Fifth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2004.