Greenput: a Power-saving Algorithm That Achieves Maximum Throughput in Wireless Networks

Cheng-Shang Chang, *Fellow, IEEE,* Duan-Shin Lee, *Senior Member, IEEE*, and Chia-Kai Su

***Abstract*—The dynamic frame sizing algorithm is a throughput-optimal algorithm that can achieve maximum network throughput without the knowledge of arrival rates. Motivated by the need for energy-efficient communication in wireless networks, in this paper we propose a new dynamic frame sizing algorithm, called the *Greenput* algorithm, that takes power allocation into account. In our Greenput algorithm, time is partitioned into frames and the frame size of each frame is determined based on the backlogs presented at the beginning of a frame. To obtain a good delay-energy efficiency tradeoff, the key insight of our Greenput algorithm is to reduce transmit power to save energy when the backlogs are low so as not to incur too much packet delay. For this, we define a threshold parameter $T_{\max}$ (as the minimum time to empty the backlogs with maximum power allocation) and the Greenput algorithm enters the (mixed) power-saving mode when the backlogs are below the threshold. Using a large deviation bound, we prove that our Greenput algorithm is still throughput-optimal. In addition to the stability result, we also perform a fluid approximation analysis for energy efficiency and average packet delay when $T_{\max}$ is very large. To show the delay-energy efficiency trade-off, we conduct extensive computer simulations by using the Shannon formula as the channel model in a wireless network. Our simulation results show that both energy efficiency and average packet delay are quite close to their fluid approximations even when $T_{\max}$ is moderately large.**

I. Introduction

SCHEDULING algorithms that achieve maximum throughput for a network of constrained links (queues) have been an ongoing research topic for a long period of time. One of the most renowned algorithms, known as the maximum weighted matching (MWM) algorithm proposed for the discrete-time setting in the pioneering work [1], identifies the most suitable set of transmitting links according to the queue length information available at each time slot. Such an algorithm is known to be *throughput-optimal* as it can stabilize a network of queues as long as the rates of arrival traffic fall within the capacity region (that is the convex hull of the sets of transmitting rates). This is done without the need of knowing the arrival rates. The MWM algorithm was further extended to various switched processing systems (see e.g., [2]–[6]). In [7], [8], the issues of power allocation and channel states (of transmitting links) were also taken into account. In particular, a Dynamic Routing and Power Control (DRPC) policy was proposed in [7] to stabilize the queues by solving a joint routing and power allocation problem. Most of these works assumed infinite buffers. The effect of finite buffer-size on the performance of a network of queues was addressed in [9], [10]. A standard approach to prove the stability of the MWM algorithm in these works [1]– [10] is to first consider a Lyapunov function and then show the existence of a negative drift of the Lyapunov function when the MWM algorithm (with back-pressure routing) is used. Another interesting approach is to use the carrier-sense multiple access (CSMA)-type random access algorithm to achieve the maximum throughput in ad hoc wireless networks (see e.g. [11], [12]). This approach usually requires a timescale separation assumption that assumes the CSMA Markov chain converges to its steady-state distribution instantaneously compared to the timescale of adaptation of the CSMA parameters.

The dynamic frame sizing (DFS) algorithm [13]–[16] is another class of throughput-optimal algorithms. Unlike the MWM algorithm, the DFS algorithm is a frame-based algorithm. At the beginning of each frame, an optimization problem is solved to determine the frame size. For a single-hop (wireless or wired) network with multiple links (queues), the frame size is chosen to be the minimum amount of time, known as the minimum clearance time, to clear the backlogs observed at the links at the beginning of the frame. As such, the backlogs at the beginning of a frame are then bounded above by the arrivals during the previous frame, and a packet that arrives in one frame will depart the network in the next frame. Thus, as long as the expected size of each frame is finite, the expected backlog at each link remains finite. Unlike the Lyapunov approach for the proof of the stability of the MWM algorithm, the approach for proving the stability of the DFS algorithm in [13]– [16] is based on a large deviation bound for the minimum clearance time (the recurrent time of a regenerative process). Such a bound is derived by comparing the minimum clearance time with the time to drain the backlog at each link with a feasible rate that is larger than its arrival rate (even though the actual arrival rate is *not* known). To extend the DFS algorithm to a multiple-hop network with per flow queueing, a hierarchical smooth schedule (as an extension of the smooth schedule in [17]) is used in [15] for providing guaranteed rate services inside the network. By doing so, the number of packets in any internal buffer is bounded by a deterministic constant. Further extension to multicasting flows with network coding was shown in [16].

Motivated by the need for energy-efficient communication in wireless networks [18], [19], there has been tremendous interest in the study of various tradeoff mechanisms to achieve energy efficiency in each protocol layer (see the recent survey paper [20] and references therein). One of the main objectives

C.-S. Chang, D.-S. Lee, and C.-K. Su are with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.
E-mail: {cschang,ptcheng}@ee.nthu.edu.tw, lds@cs.nthu.edu.tw, josan88115@gapp.nthu.edu.tw.

Manuscript received November 13, 2016; revised July 14, 2017.

of this paper is to study the delay-energy efficiency tradeoff in wireless networks by using the DFS algorithm. For this, we propose a new dynamic frame sizing algorithm, called the *Greenput* algorithm, that takes power allocation into account. To obtain a good delay-energy efficiency tradeoff, the key insight of our Greenput algorithm is to reduce transmit power to save energy when the backlogs are low and this should not incur too much packet delay.

In the two recent papers [21], [22], the authors considered the minimum-time scheduling problem and the minimum-energy scheduling problem to empty the backlogs in wireless networks. They showed various characterizations of optimal schedules for these two problems. In particular, the time division multiple access (TDMA) schedule (that has the least interference) is optimal under various channel assumptions. Inspired by these two important papers, our idea for a generic power-saving DFS algorithm is to use the minimum-time schedule with maximum power allocation in [21] when the backlogs are large, and then switch to the minimum-energy schedule in [22] when the backlogs are small. To determine whether the backlogs are small, we define a threshold $T_{\max}$ (as the minimum time to empty the backlogs with maximum power allocation) and we only use the minimum-energy schedule (for saving energy) when the solution of the minimum-time schedule with maximum power allocation is smaller than the threshold $T_{\max}$. In addition to the threshold $T_{\max}$, we also define another parameter $T_{\min}$, called the minimum frame length for empty queues. When there is no backlog at the beginning of a frame, all the links in the network remain idle for a period of time $T_{\min}$. The parameter $T_{\min}$ is known as the vacation time in the queueing context [23]. We prove such a generic DFS algorithm (see Algorithm 1 for more detailed descriptions) with the two key parameters $T_{\min}$ and $T_{\max}$ is also throughput-optimal for any traffic that is within the capacity of the maximum power allocation.

One of the major difficulties for solving the minimum-energy schedule in [22] is its computational complexity. If one uses the well-known Shannon formula for the white Gaussian noise (AWGN) channel, then the transmit rate is a non-linear function of the transmit power. As such, the minimum-energy scheduling problem is a non-linear programming problem. To cope with such a computational complexity problem, we first show that there exists an optimal simple TDMA schedule for the minimum-energy scheduling problem if there is no constraint on the frame size, i.e., $T_{\max} = \infty$. Such a result implies there might exist a good TDMA schedule if $T_{\max}$ is not too small. We then propose the greedy allocation algorithm (see Algorithm 2 for more detailed descriptions) to find a simple TDMA schedule for the minimum-energy scheduling problem. The computational complexity of the greedy algorithm is very low and it is optimal if the objective is separable concave [24].

By incorporating the greedy allocation algorithm in Algorithm 2 into the generic DFS algorithm in Algorithm 1, we then propose the *Greenput* algorithm (see Algorithm 3 for more detailed descriptions). In the Greenput algorithm, there are four operation modes for each frame: (i) empty queue mode, (ii) power-saving mode, (iii) maximum power mode, and (iv) mixed power-saving mode. The empty queue mode is chosen when there is no backlog at the beginning of a frame. The power-saving mode is used when the backlogs can be cleared before $T_{\max}$ by a simple TDMA schedule. The maximum power mode is used when the backlogs cannot be cleared before $T_{\max}$ by using the maximum power allocation. Finally, the mixed power-saving mode is used when the backlogs can be cleared before $T_{\max}$ by using the maximum power allocation but they cannot be cleared before $T_{\max}$ by a simple TDMA schedule. Like the generic DFS algorithm, the Greenput algorithm is also throughput-optimal for any traffic that is within the capacity of the maximum power allocation. For such an algorithm, we also derive a fluid approximation analysis for energy efficiency and average packet delay (as a function of the arrival rates) when a large $T_{\max}$ is in the one second order of magnitude. To show the delay-energy efficiency trade-off of the Greenput algorithm, we conduct extensive computer simulations by using the Shannon formula as the channel model in a wireless network. Our simulation results show that average packet delay is roughly the same as the threshold $T_{\max}$ under various traffic loads. Also, energy efficiency (measured in joules/bit) is quite close to the fluid approximation even when $T_{\max}$ is moderately large. As such, one can choose an appropriate threshold $T_{\max}$ to obtain a good delay-energy efficiency tradeoff. Finally, we remark that the Greenput algorithm is not designed specifically for the current cellular networks, which make scheduling decisions at fixed time intervals.

The rest of this paper is organized as follows. In section Section II, we describe our mathematical model for wireless networks. We first propose the generic DFS algorithm in Section III. We then propose the greedy allocation algorithm and the Greenput algorithm in Section IV. We prove the stability results and derive approximations for energy efficiency in Section V. In Section VI, we conduct various simulations to show the delay-energy efficiency trade-off of the two key parameters. In Section VII, we conclude this paper by addressing some further extensions. In Table I, we provide a list of the notations that are used in the paper.

II. The Mathematical Model

A. Wireless network models

In this section, we first introduce the mathematical model and the notations that will be used in the paper.

Consider a network with N links, indexed from 1 to N. Let $\mathcal{H}$ be the collection of nonempty subsets of $\{1, 2, \ldots, N\}$. A member $g \in \mathcal{H}$ is called a *group* (of links). Clearly, $|\mathcal{H}| = 2^N - 1$. Also, let $\mathbf{p} = (p_1, p_2, \ldots, p_N)$ be the power allocation vector with p_i being the power allocated to link i. In this paper, we assume that there is maximum power $p_{i,\max}$ for link i, i.e., $p_i \le p_{i,\max}$ for all $i = 1, 2, \ldots, N$. Let $r_i(g, \mathbf{p})$ be the transmit rate of link i when the links in group g are allowed to transmit at a given time with the power allocation vector $\mathbf{p}$. Note that p_i is the power *allocated* to link i and it may not be the power *actually* consumed by link i. For $i \in g$, link i transmits with the power p_i. On the other hand, for $i \notin g$, link i is inactive and there is no power consumption of link i. As such, we define $r_i(g, \mathbf{p}) = 0$ for $i \notin g$.

TABLE I: List of Notations

N	The number of links
$\mathcal{H}$	The collection of nonempty subsets of $\{1, 2, \ldots, N\}$
λ_i	The arrival rate of link i
Λ	The arrival rate vector
μ_i	The stable service rate of link i
$\mu_{\min}$	The minimum stable service rate
ρ	The traffic intensity in (8)
p_i	The power allocated to link i
$\mathbf{p}$	The power allocation vector
$p_{i,\max}$	The maximum power of link i
$\mathbf{p}_{\max}$	The maximum power vector
g	a group of links
$r_i(g, \mathbf{p})$	The transmit rate of link i when the links in group g are allowed to transmit with the power allocation vector $\mathbf{p}$
T_n	The length of the n^{th} frame
T'_n	The minimum clearance time with maximum power allocation in (14)
T''_n	The schedule length with minimum energy in (21)
$y_i(n)$	The backlog of link i at the beginning of the n^{th} frame
$A_i(s,t)$	The amount of data that arrives at link i in $[s,t]$
$T_{\max}$	The threshold for power-saving mode
$T_{\min}$	The minimum frame length for empty queues (vacation time)
t_g	The transmission time of group g
t^*_g	The optimal transmission time of group g in (5)
t'_g	The optimal transmission time of group g in (11)
t''_g	The optimal transmission time of group g in (16)
θ^*	The unique positive solution of (27)
$h_{i,j}$	The channel gain between link i and link j
W	The bandwidth of the band limited channel
σ^2	The noise variance
$\gamma_i(g, \mathbf{p})$	The SINR of link i when the links in group g are allowed to transmit with the power allocation vector $\mathbf{p}$
γ^*	The SINR threshold for a successful transmission

We make the following two rate monotonic assumptions:

(A1) If $\alpha \geq 1$, then $r_i(g, \alpha\mathbf{p}) \geq r_i(g, \mathbf{p})$ for all $i \in g$ and $g \in \mathcal{H}$.

(A2) If $g \subset g'$ and $i \in g$, then $r_i(g, \mathbf{p}) \geq r_i(g', \mathbf{p})$.

The first assumption is rather intuitive and it says that transmit rate of a link is increased if the power allocated to each link is increased *proportionally*. The second assumption is also intuitive. Given the same amount of power allocation, the corresponding links in the group with a smaller number of links have larger rates. This is because there is less interference in the group with a smaller number of links.

As described in [21], one commonly used model for a wireless network is the signal-to-interference-and-noise-ratio (SINR) model in which a transmission at a given rate is successful if the SINR at the receiver exceeds a certain threshold. Specifically, consider N transmitters and N receivers. There is a link between each pair of transmitters and receivers. Let $H = (h_{ij})$ be an $N \times N$ channel matrix with h_{ij} being the channel gain between transmitter i and receiver j. In a band limited channel with an additive white Gaussian noise, the SINR for link i in group g is given by

$$\gamma_i(g, \mathbf{p}) = \frac{p_i h_{ii}}{\sum_{j \in g, j \neq i} p_j h_{ji} + W\sigma^2}, \tag{1}$$

where W is the bandwidth of the band limited channel and σ^2 is the noise variance. Let γ^* be the SINR threshold for a successful transmission. Then

$$r_i(g, \mathbf{p}) = \begin{cases} 1, & \text{if } \gamma_i(g, \mathbf{p}) \geq \gamma^*, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Note from (1) that $\gamma_i(g, \alpha\mathbf{p}) \geq \gamma_i(g, \mathbf{p})$ for $\alpha \geq 1$ (as the noise variance σ^2 is effectively reduced to σ^2/α in the SINR). In view of (2), the assumption in (A1) is satisfied. On the other hand, if $g \subset g'$ and $i \in g \cap g'$, then $\gamma_i(g, \mathbf{p}) \geq \gamma_i(g', \mathbf{p})$ (as there is less interference in the SINR). Thus, the assumption in (A2) is also satisfied.

The second commonly used model is the Shannon formula for the additive white Gaussian noise (AWGN) channel:

$$r_i(g, \mathbf{p}) = W \log_2(1 + \gamma_i(g, \mathbf{p})). \tag{3}$$

One can also easily verify (from the monotonic properties of the SINR discussed in the threshold model) that both assumptions in (A1) and (A2) are also satisfied for this Shannon formula model.

B. Admissible arrival traffic under maximum power allocation

In [21], Angelakis et al. considered the minimum-time link scheduling problem to clear the backlogs presented at the links in a wireless network as soon as possible. In their minimum-time link scheduling problem, there are no future arrivals at the links. In order to define the notion of throughput, we first extend the minimum-time link scheduling problem to the setting with future arrivals in this section.

For each link, we assume that there is a queue to store backlogged data. The traffic that arrives at link i is assumed to be a Poisson process with rate λ_i. Let

$$\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N). \tag{4}$$

The vector Λ is called the arrival rate vector in this paper.

Let $\mathbf{p}_{\max} = (p_{1,\max}, \ldots, p_{N,\max})$ be the power allocation vector when each link is allocated with the maximum power. Consider the expected number of arrivals at link i, where $i = 1, 2, \ldots, N$, in a unit time interval. Since the arrival process is Poisson, the expected number of arrivals is λ_i. Now consider the following minimum-time link scheduling problem to empty the "backlogs" $(\lambda_1, \lambda_2, \ldots, \lambda_N)$:

$$\min_{t_g} \quad \sum_{g \in \mathcal{H}} t_g \tag{5}$$

$$\text{s.t.} \quad \lambda_i \leq \sum_{g \in \mathcal{H}} t_g \cdot r_i(g, \mathbf{p}_{\max}), \ i = 1, 2, \ldots, N, \tag{6}$$

$$t_g \geq 0, \ g \in \mathcal{H}. \tag{7}$$

The variable t_g in (5) is the (normalized) transmission time of group g (with respect to one unit of time interval). As such, it is a dimensionless quantity. Note that the inequality in (6) imposes a constraint that the backlog accumulated at link i must be cleared by the service time assigned to link i by the scheduler. As the minimum-time link scheduling problem is a linear programming problem, it was shown in Lemma 2 of [21] (or an application of the Carathéodory theorem) that under (A2) there exists an optimal scheduling solution using at most N groups. Specifically, let $\{t^*_g, g \in \mathcal{H}\}$ be an optimal solution of this linear programming problem and $\mathcal{H}^*$

be the collection of nonzero elements in $\{t_g^*, g \in \mathcal{H}\}$. Then $|\mathcal{H}^*| \leq N$. From the minimum-time link scheduling problem, we define the intensity ρ of an arrival rate vector Λ as follows:

$$\rho = \sum_{g \in \mathcal{H}^*} t_g^*. \tag{8}$$

The input traffic with the arrival rate vector Λ is said to be *admissible* if $\rho < 1$. The capacity region (with the maximum power allocation) is the union of the admissible arrival rate vectors. For the admissible traffic with the arrival rate vector Λ, we define a *stable* service rate vector $\mu = (\mu_1, \mu_2, \ldots, \mu_N)$ with

$$\mu_i = \frac{\sum_{g \in \mathcal{H}^*} t_g^* \cdot r_i(g, \mathbf{p}_{\max})}{\rho},\ i = 1, 2, \ldots, N. \tag{9}$$

In view of (6), we have for $\rho < 1$ that

$$\mu_i \geq \frac{\lambda_i}{\rho} > \lambda_i,\ i = 1, 2, \ldots, N. \tag{10}$$

Thus, if the arrival rates λ_i, $i = 1, 2, \ldots, N$, are known, one can stabilize the queues by providing stable service rates. In fact, there are many scheduling policies in the literature that can be used for stabilizing each queue in the network (see e.g., the rate proportional processor sharing (RPPS) scheme in [25] and the service curve earliest deadline first (SCED) scheme in [26]). On the other hand, if the arrival rates are *not known*, the maximum weighted matching (MWM) algorithm in [1] and the dynamic frame sizing algorithm in [13]–[15] can be used for stabilizing each queue in the network with the maximum power allocation. One of the main objectives of this paper is to take power allocation into consideration and propose power-saving algorithms that not only achieve maximum throughput but also have better energy efficiency for transmitting data.

III. Generic Dynamic Frame Sizing Algorithm with Power Saving

In [13]–[15], the dynamic frame sizing (DFS) algorithm has been used for stabilizing various queueing networks without knowing the arrival rates. In this section, we incorporate power saving into the DFS algorithm.

A. Determining the frame size

The basic idea of the DFS algorithm in [13]–[15] is quite simple. In such an algorithm, time is partitioned into frames, where the frame size is not fixed and is determined at the beginning of each frame. The frame size in [13]–[15] is set to be the minimum time to empty all the backlogs observed at the beginning of the frame. As long as the expected size of each frame is finite, the expected backlog at each queue remains finite. To incorporate power saving into the DFS algorithm, our idea is to use maximum power allocation when the backlogs are large and switch to a power-saving mode when the backlogs are small. The question is then how to determine whether the backlogs are large or small. Our approach is to set a *threshold* $T_{\max}$ for the minimum clearance time (which is defined as the minimum time to empty the backlogs with maximum power allocation).

Now we present the method for determining the frame size at the beginning of each frame in the generic DFS algorithm with power saving. Let T_n be the length of the n^{th} frame and $\tau_n = \sum_{\ell=1}^{n-1} T_\ell$ be the beginning time epoch of the n^{th} frame. The *backlog* at τ_n at link i is denoted by $y_i(n)$. Let $Y(n) = (y_1(n), y_2(n), \ldots, y_N(n))$ be the backlog vector at the beginning of the n^{th} frame.

If there is no backlog at the beginning of the n^{th} frame, i.e., $y_i(n) = 0$ for all $i = 1, 2, \ldots, N$, then we simply set the size of the n^{th} frame, denoted by T_n, to be some constant $T_{\min}$. The constant $T_{\min}$, called the *minimum frame length for empty queues* in this paper, is also known as the server vacation time in the queueing context [23]. Intuitively, the backlogs could be very large when a server takes a long vacation. On the other hand, if there are backlogs at the beginning of the n^{th} frame, we first consider the minimum-time scheduling problem that minimizes the clearance time with maximum power allocation:

$$\min_{t_g} \quad \sum_{g \in \mathcal{H}} t_g \tag{11}$$

$$\text{s.t.} \quad y_i(n) \leq \sum_{g \in \mathcal{H}} t_g \cdot r_i(g, \mathbf{p}_{\max}), i = 1, 2, \ldots, N. \tag{12}$$

$$t_g \geq 0,\ g \in \mathcal{H}. \tag{13}$$

Let $\{t'_g, g \in \mathcal{H}\}$ be an optimal solution of this linear programming problem and $\mathcal{H}'$ be the set of nonzero elements of $\{t'_g, g \in \mathcal{H}\}$. As discussed for the minimum-time scheduling problem in (5), we know that $|\mathcal{H}'| \leq N$. Now let

$$T'_n = \sum_{g \in \mathcal{H}'} t'_g \tag{14}$$

be the minimum time to clear the backlogs at the beginning of the n^{th} frame with the maximum power allocation. If T'_n is not less than the threshold $T_{\max}$, then we consider the backlogs are large and we do not go into the power-saving mode. In this case, we simply set

$$T_n = T'_n. \tag{15}$$

Otherwise, we conclude the backlogs are small and we switch to the power-saving mode. In the power-saving mode, we consider another minimum-energy scheduling problem that was previously addressed in [22]:

$$\min_{t_g, \mathbf{p}_g} \quad \sum_{g \in \mathcal{H}} t_g (\sum_{i \in g} p_{g,i}) \tag{16}$$

$$\text{s.t.} \quad y_i(n) \leq \sum_{g \in \mathcal{H}} t_g \cdot r_i(g, \mathbf{p}_g),\ i = 1, 2, \ldots, N, \tag{17}$$

$$\sum_{g \in \mathcal{H}} t_g \leq T_{\max}, \tag{18}$$

$$t_g \geq 0, g \in \mathcal{H}, \tag{19}$$

$$\mathbf{p}_g = (p_{g,1}, \ldots, p_{g,N}) \leq \mathbf{p}_{\max}. \tag{20}$$

Since $T'_n < T_{\max}$, we know from (A1) that there is a feasible solution of this minimum-energy scheduling problem. Let $\{t''_g, g \in \mathcal{H}\}$ be an optimal solution of this optimization problem and

$$T''_n = \sum_g t''_g \tag{21}$$

be the time to clear the backlogs not later than $T_{\max}$ with minimum energy. In this case, we set $T_n = T_n''$.

The algorithm for determining the frame size at the beginning of the n^{th} frame is summarized in Algorithm 1.

Algorithm 1 The generic dynamic frame sizing algorithm with power saving

Input: The backlog vector at the beginning of the n^{th} frame $Y(n) = (y_1(n), y_2(n), \ldots, y_N(n))$

Output: The length of the n^{th} frame T_n

1: If there is no backlog at the beginning of the n^{th} frame, then set $T_n = T_{\min}$.
2: Otherwise, solve the minimum-time scheduling problem in (11). If $T_n' \geq T_{\max}$, then set $T_n = T_n'$.
3: Otherwise, go to the power-saving mode and solve the minimum-energy scheduling problem in (16). Set $T_n = T_n''$.

Though the idea of the generic DFS algorithm with power saving in Algorithm 1 is quite simple, its computational complexity might be high. This is because the minimum-energy scheduling problem is not a linear programming problem (as $r_i(g, \mathbf{p}_g)$ is not linear in $\mathbf{p}_g$). As discussed in [22], finding an optimal solution of the minimum-energy scheduling problem may be difficult. In the next section, we will propose a much simplified version of the dynamic frame sizing algorithm with power saving.

IV. The GreenPut algorithm

A. The existence of an optimal simple TDMA schedule

For the minimum-energy scheduling problem, it was shown in [22] that time division multiple access (TDMA) scheduling (which avoids simultaneous transmissions) is optimal for some cases. In particular, if the constraint in (18) is removed, i.e., $T_{\max} = \infty$, then it was shown in Theorem 1 of [22] that TDMA is optimal for the Shannon formula model. The intuition behind this is that there is no interference in a TDMA schedule and thus power can be saved to maintain the same level of SINR. In Theorem 1 below, we further extend such a result to a more general setting. For a TDMA schedule, the transmit rate for link i only depends on the power allocated to link i. For ease of presentation, we simplify the notation for $r_i(\{i\}, \mathbf{p})$ in a TDMA schedule by $r_i(p_i)$. Also, we will call a TDMA schedule *simple* if each link is selected for transmission at most once in the schedule.

Theorem 1: Consider the minimum-energy scheduling problem in (16) with the constraint in (18) being removed, i.e., $T_{\max} = \infty$. Suppose that the assumption (A2) holds and for each link i, the transmit rate $r_i(p_i)$ is concave in the allocated power p_i. Then there is an optimal simple TDMA schedule.

The proof of Theorem 1 is given in Appendix A, available in the online supplemental material.

B. A greedy allocation algorithm for generating a simple TDMA schedule

The results in [22] and Theorem 1 motivate us to consider using simple TDMA schedules in the power-saving mode. Instead of solving the minimum-energy scheduling problem in (16), we propose a greedy allocation algorithm in Algorithm 2 to find a simple TDMA schedule with the length (frame size) not greater than $T_{\max}$.

Algorithm 2 Greedy allocation algorithm for finding a simple TDMA schedule in the power-saving mode

Input: The backlog vector at the beginning of the n^{th} frame $Y(n) = (y_1(n), y_2(n), \ldots, y_N(n))$

Output: A simple TDMA schedule with $\sum_{i=1}^{N} t_i \leq T_{\max}$ (if it exists)

1: Compute $T = \sum_{i=1}^{N} y_i(n)/r_i(p_{i,\max})$. If $T > T_{\max}$, then there does not exist a feasible TDMA schedule.
2: Otherwise, Set $t_i = y_i(n)/r_i(p_{i,\max}), i = 1, 2, \ldots, N$.
3: Let $R = T_{\max} - T$ and $\Delta = R/K$, where K is a constant.
4: For each Δ, allocate it sequentially to the link that results in the largest energy reduction for transmitting the data on its link.

Now we explain how Algorithm 2 works. In Step 1, we first check whether there exists a simple TDMA schedule that can clear the backlogs before $T_{\max}$ by allocating maximum power to each link. If such a simple TDMA schedule exists, we know from (A1) that we can allocate the remaining time to the links in the TDMA schedule so that the transmit power at each link of this TDMA schedule can be further reduced to save energy. In Step 2, we set the (minimum) transmission time of each link to be the time to empty its backlog with maximum power allocation. In Step 3, we calculate the remaining time that we can allocate before exceeding $T_{\max}$. We then divide it into K time units for further allocation. It is well-known (see e.g., [24]) that for a discrete resource allocation problem, the greedy (or marginal allocation) algorithm that assigns each available time unit sequentially to the link that results in the largest energy reduction from an additional allocation among all links is optimal if the objective of the total energy consumption is *separable concave* in the allocation of time units (such a concave condition can be shown to be satisfied under an appropriate assumption for $r_i(p_i)$, $i = 1, 2, \ldots, N$). In Step 4, we carry out the greedy allocation of time units. We note that the frame size is exactly $T_{\max}$ when the sequential allocation in Step 4 is completed. Also, increasing K further increases the saving of energy. But this is at the cost of increasing computational complexity.

C. The proposed algorithm

In view of the generic DFS algorithm in Algorithm 1 and the greedy allocation algorithm for finding a feasible TDMA schedule in Algorithm 2, there is still one more case that needs to be resolved. It is the case when the following two conditions hold: (i) there does not exist a feasible TDMA schedule (with the schedule length not greater than $T_{\max}$) and (ii) the schedule that uses the maximum power allocation yields a schedule length shorter than $T_{\max}$. In this case, there must be group transmission(s) with group size larger than 1 (as otherwise there should be a feasible TDMA schedule). As such, we can separate the schedule that uses the maximum power allocation

into two subschedules: (i) group transmission(s) with group size larger than 1 and (ii) group transmission(s) with group size equal to 1. For the group transmission(s) with group size equal to 1, we can further reduce the transmit power by allocating more time as in Algorithm 2. Specifically, as in Section III-A, we let $\{t'_g, g \in \mathcal{H}'\}$ be an optimal solution of the minimum-time scheduling problem in (11). Also, let

$$\mathcal{H}'_+ = \{g : g \in \mathcal{H}', |g| > 1|\} \tag{22}$$

be the set of group transmission(s) with group size large than 1 in $\mathcal{H}'$. Since the schedule with the maximum power allocation yields a schedule length shorter than $T_{\max}$, we must have

$$\sum_{g \in \mathcal{H}'_+} t'_g \leq \sum_{g \in \mathcal{H}'} t'_g < T_{\max}. \tag{23}$$

Now let

$$\tilde{y}_i(n) = y_i(n) - \sum_{g \in \mathcal{H}'_+} t'_g \cdot r_i(g, \mathbf{p}_{\max}) \tag{24}$$

be the remaining backlog for link i after the subschedule for group transmission(s) with group size large than 1. Similarly, let

$$\tilde{T}_{\max} = T_{\max} - \sum_{g \in \mathcal{H}'_+} t'_g \tag{25}$$

be the remaining time that we can allocate for the simple TDMA schedule (after the subschedule for group transmission(s) with group size large than 1). Then we can use Algorithm 2 to find a feasible TDMA schedule by using $\tilde{T}_{\max}$ and $\tilde{y}_i(n)$'s as its inputs.

Now we are ready to describe our Greenput algorithm. The details are outlined in Algorithm 3. The algorithm has the following four modes: (i) empty queue mode, (ii) power-saving mode, (iii) maximum power mode, and (iv) mixed power-saving mode. The empty queue mode is chosen when there is no backlog at the beginning of a frame, and its frame length is $T_{\min}$. The power-saving mode is used when the backlogs can be cleared before $T_{\max}$ by a simple TDMA schedule, and its frame length is exactly $T_{\max}$. The maximum power mode is used when the backlogs cannot be cleared before $T_{\max}$ by using the maximum power allocation. In this mode, its frame length is not smaller than $T_{\max}$. Finally, the mixed power-saving mode is used when the backlogs can be cleared before $T_{\max}$ by using the maximum power allocation but they cannot be cleared before $T_{\max}$ by a simple TDMA schedule. Note that the mixed power-saving mode consists of two subschedules: one with the maximum power allocation and the other with a simple TDMA schedule. The sum of the lengths of these two subschedules is $T_{\max}$ and thus the frame length in the mixed power-saving mode is also $T_{\max}$.

Algorithm 3 The Greenput algorithm

Input: The backlog vector at the beginning of the n^{th} frame $Y(n) = (y_1(n), y_2(n), \ldots, y_N(n))$

Output: The length of the n^{th} frame T_n

1: **(Empty queue mode)** If there is no backlog at the beginning of the n^{th} frame, then set $T_n = T_{\min}$.
2: Otherwise, run Algorithm 2.
3: **(Power-saving mode)** If Algorithm 2 returns a feasible TDMA schedule $\{t_i, i = 1, 2, \ldots, N\}$, set $T_n = \sum_{i=1}^{N} t_i = T_{\max}$ and run this TDMA schedule for this frame.
4: Otherwise, solve the minimum-time scheduling problem in (11).
5: **(Maximum power mode)** Suppose that we obtain an optimal minimum-time schedule $\{t'_g, g \in \mathcal{H}'\}$ and $\sum_{g \in \mathcal{H}'} t'_g \geq T_{\max}$. Set $T_n = \sum_{g \in \mathcal{H}'} t'_g$ and run the minimum-time schedule for this frame.
6: Otherwise, separate the minimum-time schedule into two subschedules: the subschedule with group size larger than 1 and the subschedule with group size equal to 1.
7: **(Mixed power-saving mode)** Keep the subschedule with group size larger than 1 and replace the other subschedule by the TDMA schedule from Algorithm 2 by using $\tilde{y}_i(n)$, $i = 1, 2, \ldots, n$ in (24) and $\tilde{T}_{\max}$ in (25) as its inputs. Set $T_n = T_{\max}$ to be the sum of the length of the subschedule with group size larger than 1 and the length of the new TDMA subschedule.

V. Performance Analysis

A. Stability for Admissible Traffic

In this section, we show that for any admissible Poisson arrival traffic, the expected frame sizes under both the generic DFS algorithm in Algorithm 1 and the Greenput algorithm in Algorithm 3 are finite. Thus, all the queues are stable for such Poisson traffic.

Here we make three specific assumptions on the input traffic.

(A3) All the arrivals are independent Poisson processes. Specifically, the amount of data that arrives at link i during the time interval $[s, t]$, denoted by $A_i(s, t)$, is a Poisson random variable with mean $\lambda_i(t - s)$. Without loss of generality, we assume that $\lambda_i > 0$ for all $i = 1, 2, \ldots, N$.

(A4) Assume that the arrival rates λ_i, $1 \leq i \leq N$, are unknown to the network.

(A5) The input traffic is admissible, i.e., the intensity ρ defined in (8) is strictly smaller than 1.

Theorem 2: Assume that (A1-5) hold. Let $\mu_{\min} = \min_{1 \leq i \leq N} \mu_i$ with μ_i being the stable service rate in (9) and $C = \max[T_{\min}, T_{\max}]$. Then, under either the generic DFS algorithm in Algorithm 1 or the Greenput algorithm in 3, we have for $n > 1$

$$\log E[e^{\theta^* T_n}] \leq \frac{2 \log N + 2C\theta^*}{(1 - \rho)} \tag{26}$$

where θ^* is the unique positive solution of

$$\frac{e^{\theta/\mu_{\min}} - 1}{\theta/\mu_{\min}} = \frac{1 + \rho}{2\rho}. \tag{27}$$

As a result, the expectation of the frame size $E[T_n]$ is bounded by $\frac{2 \log N + 2C\theta^*}{\theta^*(1-\rho)}$.

As a direct consequence of Theorem 2, the expected backlog in each queue is also finite. The proof of Theorem 2 is given in Appendix B, available in the online supplemental material.

B. Approximation analysis for energy efficiency and average packet delay

In this section, we perform a fluid approximation analysis for energy efficiency and average packet delay in the Greenput algorithm under the condition that $T_{\max}$ is large.

As shown in Algorithm 3, there are four operation modes. Let us also assume that $T_{\min} << T_{\max}$ and thus the effect of the empty queue mode can be neglected. For the other three modes, only the maximum power mode can lead to a frame with frame size larger than $T_{\max}$. We claim that this cannot happen very often for any admissible traffic. To see this, note from the law of large numbers that the backlog at link i during a time interval of length $T_{\max}$ is concentrated on $\lambda_i T_{\max}$ with high probability. For any admissible traffic (with $\rho = \sum_{g\in\mathcal{H}^*} t_g^* < 1$ in (8)), the backlog will be cleared before $T_{\max}$ with the maximum power allocation. As such, there is a "drift" that pushes the frame size below $T_{\max}$. In fact, the stability result in the previous section provides a formal proof for that. With this in mind, we only have two operation modes left when $T_{\max}$ is very large: the power-saving mode and the mixed power-saving mode. For both modes, the frame sizes are exactly $T_{\max}$.

Now consider the following two cases:

Case 1. $\sum_{i=1}^N \lambda_i/r_i(p_{i,\max}) \le 1$:

In this case, we claim that Algorithm 3 is in the power-saving mode with high probability. To see this, note that the backlog at link i during a time interval of length $T_{\max}$ is concentrated on $\lambda_i T_{\max}$ with high probability. Since $\sum_{i=1}^N \lambda_i/r_i(p_{i,\max}) \le 1$, this backlog can then be cleared by using a simple TDMA schedule with the length $T_{\max}$ from Algorithm 2. As $T_{\max}$ is assumed to be very large, we may normalize the backlog at each link by $T_{\max}$ and consider the fluid approximation that views backlogs as fluids. Specifically, we let $\hat{t}_i$, $i = 1, 2, \ldots, n$ be the output schedule from Algorithm 2 with the input backlog vector $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $T_{\max} = 1$. It is clear from Algorithm 2 that $\hat{t}_i \ge \lambda_i/r_i(p_{i,\max})$ and $\sum_{i=1}^n \hat{t}_i = 1$. The power selected by link i, denoted by $\hat{p}_i$, is $r_i^{-1}(\lambda_i/\hat{t}_i)$, where $r_i^{-1}(\cdot)$ is the inverse function of $r_i(\cdot)$. Thus, the total energy consumed by such a TDMA schedule with a unit length is $\sum_{i=1}^N \hat{t}_i \cdot \hat{p}_i$ and the energy efficiency in this case is then

$$\frac{\sum_{i=1}^N \lambda_i}{\sum_{i=1}^N \hat{t}_i \cdot \hat{p}_i}. \tag{28}$$

Case 2. $\sum_{i=1}^N \lambda_i/r_i(p_{i,\max}) > 1$:

In this case, Algorithm 3 is in the mixed power-saving mode with high probability. There are two subschedules for this case: the first subschedule uses the maximum power allocation for groups with group size larger than 1 and the second subschedule uses a TDMA schedule. Analogous to the argument in Case 1, we can normalize the backlog at each link by $T_{\max}$ and consider the fluid approximation that views backlogs as fluids. The remaining fluid after the first subschedule is

$$\check{\lambda}_i = \lambda_i - \sum_{g\in\mathcal{H}^*,|g|>1} t_g^* \cdot r_i(g, \mathbf{p}_{\max}),$$

where $\{t_g^* > 0, g \in \mathcal{H}^*\}$ is an optimal solution in (5). Similarly, the remaining time for the second subschedule is

$$1 - \sum_{g\in\mathcal{H}^*,|g|>1} t_g^*.$$

Let $\check{t}_i$, $i = 1, 2, \ldots, n$ be the output schedule from Algorithm 2 with the input backlog vector $\check{\Lambda} = (\check{\lambda}_1, \check{\lambda}_2, \ldots, \check{\lambda}_n)$ and $T_{\max} = 1 - \sum_{g\in\mathcal{H}^*,|g|>1} t_g^*$. It is clear from Algorithm 2 that $\check{t}_i \ge \check{\lambda}_i/r_i(p_{i,\max})$ and $\sum_{i=1}^n \check{t}_i = 1 - \sum_{g\in\mathcal{H}^*,|g|>1} t_g^*$. The power selected by link i, denoted by $\check{p}_i$, is $r_i^{-1}(\check{\lambda}_i/\check{t}_i)$. Thus, the total energy consumed by such a TDMA subschedule is $\sum_{i=1}^n \check{t}_i \cdot \check{p}_i$. By adding the energy consumed by the first subschedule, the energy efficiency in this case is then

$$\frac{\sum_{i=1}^N \lambda_i}{\sum_{g\in\mathcal{H}^*,|g|>1} t_g^* \cdot (\sum_{i\in g} p_{g,i}) + \sum_{i=1}^N \check{t}_i \cdot \check{p}_i}. \tag{29}$$

Now we turn our attention to the fluid approximation analysis of average packet delay. When a packet arrives at the network in the n^{th} frame, it can only be served in the $(n+1)^{th}$ frame. Thus, the delay of a packet arriving in the n^{th} frame consists of two parts: the waiting time in the n^{th} frame and the completion time in the $(n+1)^{th}$ frame. Since the frame size is $T_{\max}$ with high probability when $T_{\max}$ is large, the average waiting time in the n^{th} frame is roughly $T_{\max}/2$ as Poisson arrivals are uniformly distributed in that frame. On the other hand, the average completion time in the $(n+1)^{th}$ frame is also roughly $T_{\max}/2$ in a TDMA schedule that randomly selects the order to serve a large number of packets in the TDMA schedule. Thus, we conclude that the average packet delay is roughly $T_{\max}$ when $T_{\max}$ is large. This approximation for the average packet delay will be further verified by computer simulations in Section VI.

VI. Simulations

In this section, we use computer simulations to test the performance of the Greenput algorithm in Algorithm 3. There are two key performance metrics: energy efficiency (measured by the average amount of transmitted bits to consume a unit of energy in joules) and average packet delay. In the following simulations, we vary the threshold $T_{\max}$ and the minimum frame size for empty queues $T_{\min}$ to see their effects on these two performance metrics under various traffic loads. Each data point in the simulation results for energy efficiency and packet delay is obtained from averaging over 10,000 frames. Also, the constant K in the greedy allocation algorithm (Algorithm 2) is set to 10.

A. Simulation settings

We consider an area of two adjacent rectangles, each with $500m \times 500m$. Within each rectangle, we place a base station at the center of the rectangle, i.e., the coordinates of the first base station is $(250, 250)$ and the coordinates of the second base station is $(750, 250)$. We then randomly place $N = 3$ users in the area of $1000m \times 500m$. In our simulation, the coordinates of these three users are (743,255), (721,222) and (247,245), respectively. Each user is then connected to the base

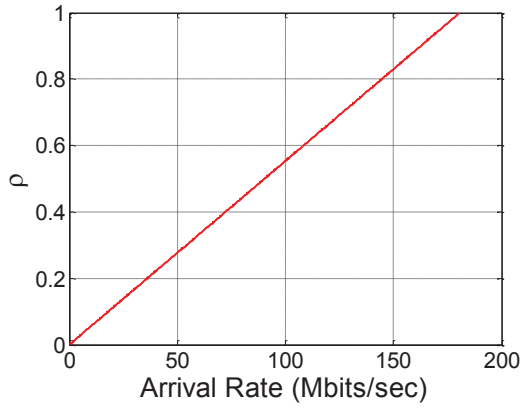

Fig. 1: The mapping between the Poisson arrival rate λ and the traffic intensity ρ.

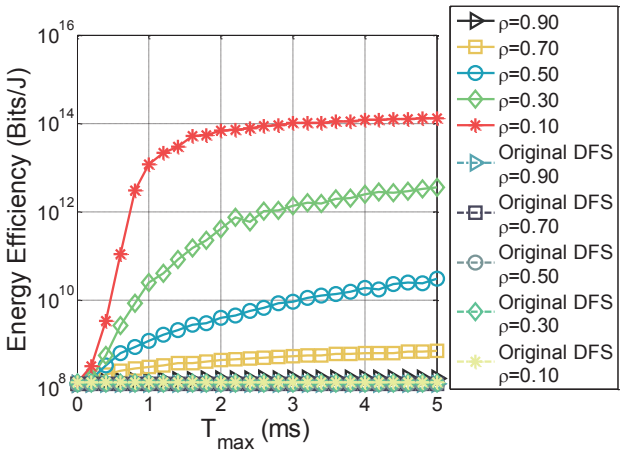

Fig. 2: Energy efficiency as a function of $T_{\max}$ for $T_{\min} = 1ms$.

station that is closer to her/him. Each user then sets up a link to its base station and thus there are three links. The channels for these links are modelled by the Shannon formula in (3) with the channel gain

$$h_{i,j} \propto (d_{i,j})^{-\beta},$$

where $d_{i,j}$ is the distance between the transmitter of link i and the receiver of link j, and $\beta = 4$. In our simulations, we also assume the network is stationary, i.e., all the users and the base stations stay in fixed locations and thus $d_{i,j}$'s do not change over time. Moreover, for the ease of coordinated transmissions, we only consider down links, i.e., transmitters are the two base stations and receivers are the three users. The arrival processes of all the three links are independent Poisson processes with the same rate.

The maximum power of all the links are the same and they are all set to be 1W (0dbW). The noise spectral density is set to be -204 dBW/Hz. These parameter settings are similar to those in [28]. Also, we set the packet size to be 1500 bytes (12000bits), which is roughly the maximum size of an Ethernet packet. In Table II, we show the transmit rates of the three links computed from the Shannon formula in (3) when the links in a group g are allowed to transmit with the maximum power allocation. We then use (8) to compute the traffic intensity ρ. In Figure 1, we plot the traffic intensity ρ as a function of the arrival rate (Mbits/sec). The function is linear as we assume that all the Poisson processes have the same rate and all the packets are of the same size.

B. Energy efficiency

In our first simulation, we choose the minimum frame size for empty queues $T_{\min}$ to be $1ms$. In Fig. 2, we show the simulation results for energy efficiency (as a function of the threshold $T_{\max}$) under various traffic loads. Clearly, when $T_{\max}$ is set to 0, the Greenput algorithm is always in the maximum power mode when there are backlogs. As such, all the data are transmitted with the maximum power allocation. This corresponds to the first data point in Fig. 2. Thus, as shown in Fig. 2, the energy efficiency for $T_{\max} = 0$ is the worst and it is basically the same under various traffic loads, ranging from 0.1 to 0.9. As expected, increasing the threshold $T_{\max}$ increases energy efficiency as it increases the chance for the algorithm to leave the maximum power mode and enter either the power-saving mode or the mixed power-saving mode. As such, more data are transmitted under TDMA schedules and that increases energy efficiency. Another interesting observation from Fig. 2 is that energy efficiency of Algorithm 3 decreases when traffic load increases. This is because the algorithm is operated in the power-saving mode (that uses the simple TDMA schedules) most of the time when the traffic load is very light. When the traffic load is further increased, it then gradually migrates into the mixed power-saving mode in which a subschedule uses the maximum power allocation. Finally, there seems a limit for energy efficiency when $T_{\max}$ is large. In Table III, we show that this limit indeed matches the fluid approximation in Section V-B under various loads when we further increase $T_{\max}$ to $0.1s, 0.3s$ and $0.5s$. We remark that we intentionally choose large values for $T_{\max}$ to study the asymptotic behavior of the energy efficiency metric. Normally, $T_{\max}$ should be of the order of a few milliseconds. Otherwise, it would be disruptive to certain network applications.

In Fig. 3, we plot the energy efficiency computed by using the fluid approximation in Section V-B as a function of traffic intensity ρ. In light load, we see that the logarithm of the energy needed to transmit a bit is almost linear in the traffic intensity. To see the insight behind this, we note that most of the frames in the Greenput algorithm are expected to be in the power-saving mode under light load. As such, most packets are transmitted under TDMA schedules with the same frame length $T_{\max}$. In view of the Shannon formula in (3) and (1), the transmit rate of a link is roughly proportional to the log of the transmit power of that link (when there is no interference). As such, in order to clear all the backlogs within $T_{\max}$, one needs to increase the transmit rates in proportion to the increase of the traffic intensity. This then implies that the logarithm of the energy needed to transmit a bit is almost linear in the traffic

TABLE II: The transmit rates of the three links when the links in a group g are allowed to transmit with the maximum power allocation

Transmit rates (Mbits/sec)							
Link \ Group	{1}	{2}	{3}	{1,2}	{1,3}	{2,3}	{1,2,3}
1	165.48	0	0	5.00	116.81	0	5.00
2	0	120.91	0	5.00	0	70.97	5.00
3	0	0	176.70	0	128.62	128.62	123.61

TABLE III: Energy efficiency (bits/joule) measured for $T_{\max} = 0.1s, 0.3s$ and $0.5s$ and the fluid approximation in Section V-B

ρ	$0.1s$	$0.3s$	$0.5s$	Approximation
0.1	1.936e+14	1.956e+14	1.958e+14	1.965e+14
0.3	1.130e+13	1.199e+13	1.208e+13	1.230e+13
0.5	2.051e+11	2.188e+11	2.222e+11	2.274e+11
0.7	2.552e+09	2.789e+09	2.831e+09	2.915e+09
0.9	1.835e+08	1.850e+08	1.853e+08	1.856e+08

intensity.

Another interesting observation is when the traffic intensity ρ exceeds 0.83, where we observe a change of operating modes. For $\rho \geq 0.83$, most of the frames in the Greenput algorithm are expected to be in the mixed power-saving mode. It might appear to be quite counterintuitive at the first look to see that the transition from the power-saving mode into the mixed power-saving mode near $\rho = 0.83$ results in more energy efficient schedules. This is because the TDMA schedule obtained from Algorithm 2 may not be the optimal schedule for the minimum-energy scheduling problem (and it is only shown to be optimal in Theorem 1 when the constraint on $T_{\max}$ is removed). A simultaneous transmission of two links in the mixed power-saving mode could be more energy efficient. To see this, note that user 1 placed at (743,255) is in fact quite close to the second base station at (750,250) and user 3 placed at (247,245) is also quite close to the first base station at (250,250). Thus, the interference of a simultaneous transmission of link 1 and link 3 is quite limited. As shown in Table II, a simultaneous transmission of link 1 and link 3 (with maximum power allocation) yields 116.81 Mbits/sec for link 1 and 128.62 Mbits/sec for link 3. This is only slightly smaller than 165.48 Mbits/sec for link 1 to be transmitted alone and 176.70 Mbits/sec for link 3 to be transmitted alone. As such, a simultaneous transmission of link 1 and link 3 can greatly reduce the backlogs at links 1 and 3 in a short period of time and thus leave more time for low energy TDMA transmissions for the remaining backlogs in the same frame.

In Fig. 4 and 5, we report the numbers of frames in the four modes among the 10,000 frames we simulated for $\rho = 0.5$ and 0.9, respectively. As predicted by the fluid approximation in Section V-B, almost every frame should be operated in the power-saving mode for $\rho = 0.5$ and in the mixed power-saving mode for $\rho = 0.9$ when $T_{\max}$ is very large. As shown in Fig. 4, it is clear that almost every frame is operated in the power-saving mode for $T_{\max} \geq 2.5ms$. The "convergence" for $\rho = 0.5$ to the fluid approximation is very fast in Fig. 4. However, for $\rho = 0.9$, we can only see from Fig. 5 that the

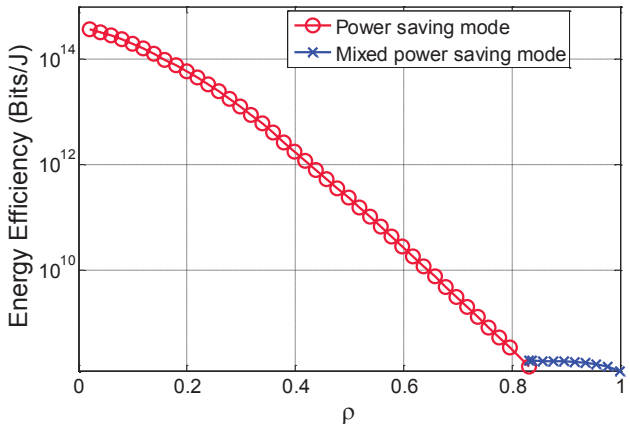

Fig. 3: Energy efficiency as a function of traffic intensity ρ for $T_{\min} = 1ms$.

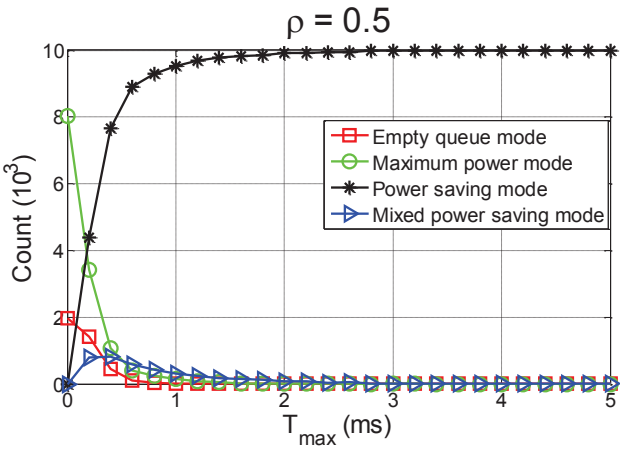

Fig. 4: The numbers of frames in the four modes among the 10,000 frames for $\rho = 0.5$ and $T_{\min} = 1ms$.

number of frames in the mixed power-saving mode increases as $T_{\max}$ increases in the range of $0ms \leq T_{\max} \leq 5ms$. On the other hand, both the numbers of frames of the maximum power mode and the power-saving mode decreases slowly as $T_{\max}$ increases (for $T_{\max} \geq 1ms$). The "convergence" for $\rho = 0.9$ to the fluid approximation is very slow in Fig. 5. To verify the "convergence" for $\rho = 0.9$, we further run simulations for $T_{\max} = 0.1s, 0, 3s$ and $0.5s$. Among the 10,000 frames, only the first two frames are not in the mixed power-saving mode (as we start from an empty network).

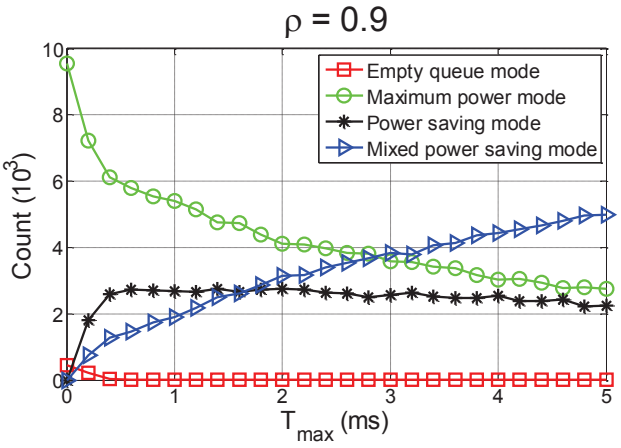

Fig. 5: The numbers of frames in the four modes among the 10,000 frames for $\rho = 0.9$ and $T_{\min} = 1ms$.

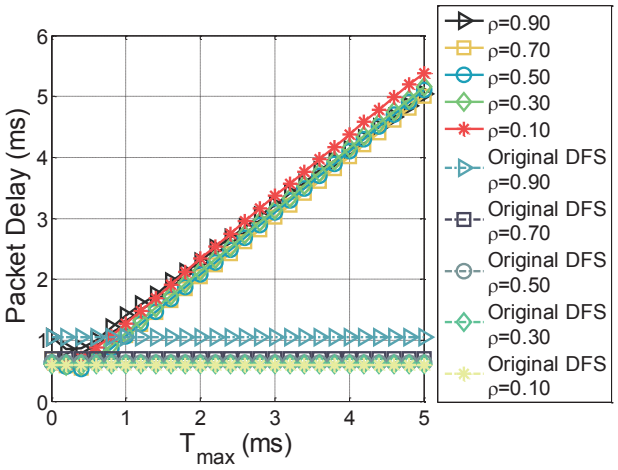

Fig. 6: Packet delay as a function of $T_{\max}$ for $T_{\min} = 1ms$.

C. Packet delay

In Fig. 6, we show the average packet delay (as a function of $T_{\max}$) for various traffic loads. As explained in Section V-B, when $T_{\max}$ is large, the average packet delay is roughly $T_{\max}$ for all the traffic loads in our simulations. Note that the average packet delay for $\rho = 0.1$ is slightly larger than $T_{\max}$. This might be due to the fact that there are only a small number of packets served in a frame. For instance, if there are only two packets served in a frame of length $T_{\max}$, then the first packet is completed at $T_{\max}/2$ and the second packet is completed at $T_{\max}$. Thus, the average completion time for these two packets is $2T_{\max}/3$, instead of being $T_{\max}/2$ predicted in Section V-B for a large number of packets served in a frame.

One interesting phenomenon in Fig. 6 is the average packet delay is not linear in $T_{\max}$ in the range of $0 \leq T_{\max} \leq T_{\min} = 1ms$. Moreover, the average packet delay is decreasing in $T_{\max}$ when $T_{\max}$ is close to 0. This appears to be quite counterintuitive at the first look. To see the insight of this, we note that Algorithm 3 is always operated in the maximum power mode when $T_{\max} = 0$. After carefully examining our simulation results for $T_{\max} = 0$, we find that the frame sizes for the frames in the maximum power mode are much shorter than $T_{\min}$. As such, there might be no packet arrivals in a frame operated in the maximum power mode and the next frame is then operated in the empty queue mode with the frame length $T_{\min} = 1ms$. Those packets that arrive in the next frame now have to wait on average $T_{\min}/2 = 0.5ms$ before they can be served. By increasing $T_{\max}$ slightly away from 0, we reduce the probability of getting into the empty queue mode and thus reduce the packet delay induced by waiting in a frame operated in the empty queue mode.

D. Comparison with the original DFS algorithm and the MWM algorithm

In this section, we compare the Greenput algorithm with the original DFS algorithm (without power saving) [13]–[16] and the maximum weighted matching (MWM) algorithm [1]. The MWM algorithm is a discrete-time algorithm. To select the group to transmit in a time slot, the MWM algorithm solves the following maximum weighted matching problem in (30) at the beginning of each time slot:

$$\max_{g} \arg \sum_{i=1}^{N} y_i(t) r_i(g, \mathbf{p}_{\max}). \tag{30}$$

The length of a time slot for the MWM algorithm in our simulation is set to be 0.1 ms. The setting for the Greenput algorithm is the same as that in Section VI-A with $T_{\min} = 1ms$. The traffic intensity ρ is set to be 0.5 in this simulation (and the results for the other selections of ρ are similar). As shown in Fig. 7, it is clear that both the energy efficiency of the original DFS algorithm and that of the MWM algorithm are much lower than that of our algorithm. This is because both the original DFS algorithm and the MWM algorithm are operated in the maximum power mode all the time. On the other hand, the packet delay of the MWM algorithm in Fig. 8 is smaller than the original DFS algorithm and the Greenput algorithm. As the MWM algorithm solves the maximum weighted matching problem in (30) in every time slot, it is more effective in reducing packet delay (when the length of a time slot is much smaller than the length of a frame in the original DFS algorithm). From Fig. 7 and Fig. 8, it is clear that the Greenput algorithm is capable of providing the energy efficiency-delay tradeoff by adjusting the parameter $T_{\max}$. As such, if a system can tolerate more delay, it can be more energy efficient under the Greenput algorithm.

E. Delay-energy efficiency tradeoff

From Fig. 2 and Fig. 6, it seems reasonable to choose $T_{\max} = 1.5ms$ to obtain a good delay-energy efficiency tradeoff (for $T_{\min} = 1ms$). As shown in Fig. 2, the energy efficiency at $T_{\max} = 1.5ms$ is quite close to its fluid limit when the traffic load is moderate or high. On the other hand, the average packet delay is also close to $T_{\max} = 1.5ms$ for various traffic loads. Increasing $T_{\max}$ further beyond $1.5ms$ increases the average packet delay linearly and it does not increase energy efficiency very much.

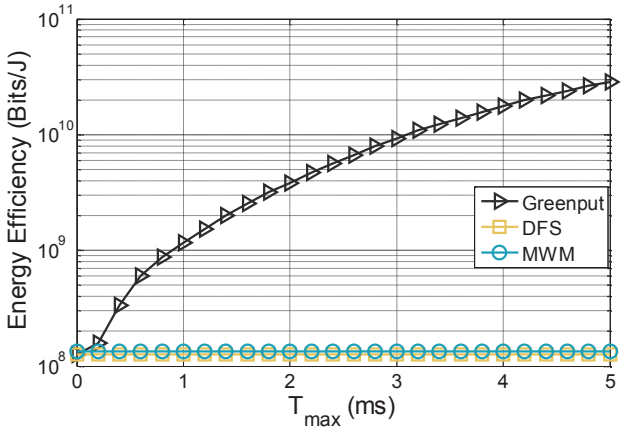

Fig. 7: Comparing energy efficiency with the original DFS algorithm and the MWM algorithm.

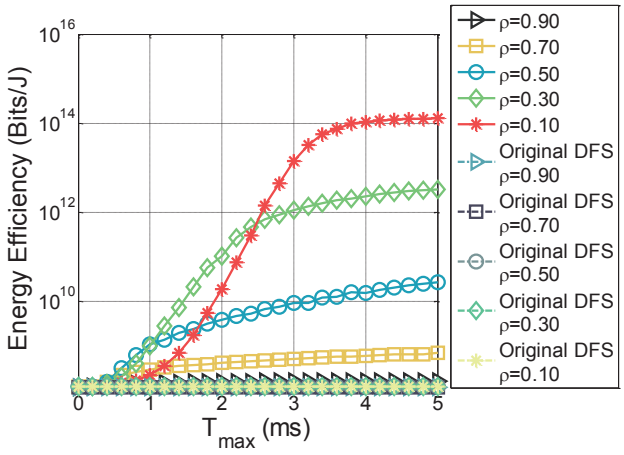

Fig. 9: Energy efficiency as a function of $T_{\max}$ for $T_{\min} = 10ms$.

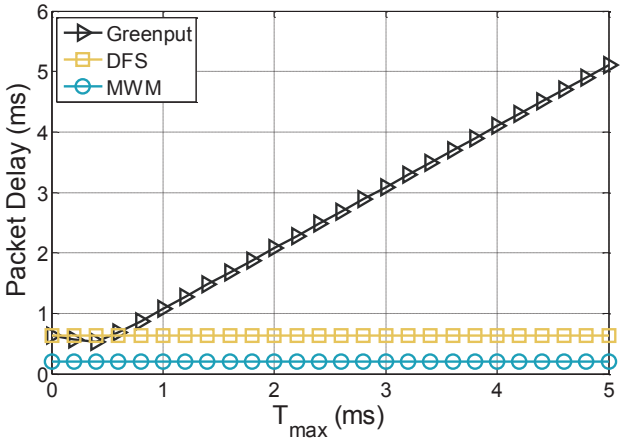

Fig. 8: Comparing packet delay with the original DFS algorithm and the MWM algorithm.

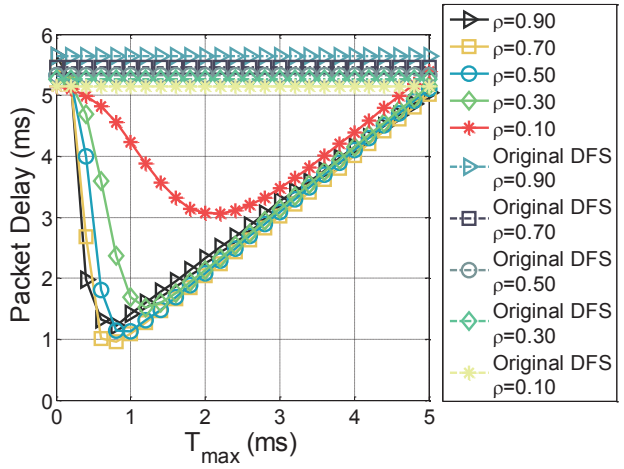

Fig. 10: Packet delay as a function of $T_{\max}$ for $T_{\min} = 10ms$.

F. The effect of $T_{\min}$

To understand the effect of $T_{\min}$, we increase $T_{\min}$ from $1ms$ to $10ms$ in the next simulation. The simulation results are shown in Fig. 9 for energy efficiency and Fig. 10 for packet delay. Note that the energy efficiency curves in Fig. 2 and Fig. 9 are not that different when the load is moderate or high. However, the energy efficiency curves for light loads are quite different. In particular, for $\rho = 0.1$, the energy efficiency for $T_{\min} = 10ms$ is worse than that for $T_{\min} = 1ms$. This is because it is quite likely to see an empty queue when the load is extremely light. When there is an empty queue at the beginning of a frame, a longer vacation time $T_{\min}$ incurs a larger backlog (after the vacation) and thus it is more likely to enter the maximum power mode in the next frame when $T_{\max}$ is small. As such, when $T_{\min}$ is large and $T_{\max}$ is small, it would not be energy efficient under light load.

To see the effect of $T_{\min}$ on packet delay, we observe from Fig. 2 and Fig. 9 that the average packet delay for $T_{\min} = 10ms$ is substantially larger than that for $T_{\min} = 1ms$ when $T_{\max} = 0$. This is because the Greenput algorithm is always in the maximum power mode when there is an nonempty backlog at the beginning of a frame when $T_{\max} = 0$. Now suppose that the n^{th} frame is in the maximum power mode. As observed from our simulation results, the frame size in the maximum power mode, i.e., the time to clear the backlog under the maximum power mode, is much smaller than $T_{\min}$. Thus, it is quite likely that there are no arrivals during the n^{th} frame. As such, the next frame, i.e., the $(n+1)^{th}$ frame, is in the empty queue mode and a large $T_{\min}$ incurs large delay for packets that arrive in the $(n+1)^{th}$ frame. This then increases the average packet delay.

Finally, we note that in the range of $0 \leq T_{\max} \leq T_{\min} = 10ms$ in Fig. 10, the average packet delay is not linear in $T_{\max}$. In fact, the average packet delay is decreasing in $T_{\max}$ when $T_{\max}$ is close to 0. Such a phenomenon also appears in Fig. 6 and it can be explained similarly as in Section VI-C.

G. The effect of a single persistent TCP flow

In this paper, we assume that the buffer for each link is infinite so that it can be shown to be throughput optimal. How-

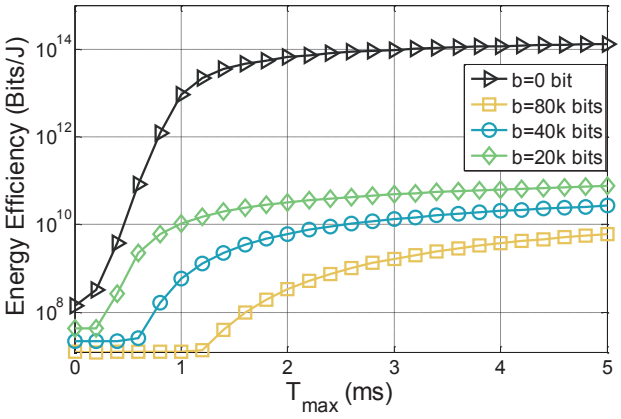

Fig. 11: Energy efficiency in the presence of a TCP flow.

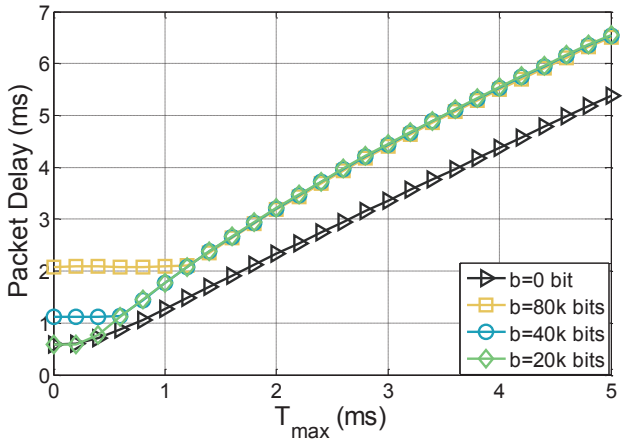

Fig. 12: Packet delay in the presence of a TCP flow.

ever, in practice the buffer is finite, and the upper layer flows usually run between TCP endpoints. To further understand the effect of TCP flows to our Greenput algorithm, we add a single persistent TCP to our simulation. For this, we assume that a buffer with b bits is allocated to such a TCP flow. As TCP flows are greedy by nature, we also assume that such a buffer is persistently backlogged at the beginning of each frame. In addition to the three users in the simulation, the TCP user (user 4) is placed at (497,119). The buffer size of the TCP flow b is set to be 80k bits, 40k bits, 20k bits, respectively, in our simulation. The arrival processes of the other three users are all independent Poisson processes with $\lambda_i = 67.56$ packets/sec, for $i = 1, 2, 3$. All the other parameters are the same as those in Section VI-A with $T_{\min} = 1ms$.

In Fig. 11 and Fig. 12, we show the simulation results for energy efficiency and packet delay of the Greenput algorithm in the presence of a single TCP flow. One interesting observation is that the presence of a single TCP flow with a finite buffer b does not affect the stability of the system. Its presence only affects energy efficiency and packet delay. This is because the Greenput algorithm clears the backlogs in every frame. As the buffer size of the TCP flow is b bits, the number of bits that arrive during a frame from the TCP flow is exactly b bits. Thus, the arrival rate of the TCP flow is inversely proportional to the frame size and it approaches 0 when the frame size is increased to ∞. From Fig. 11, we see that energy efficiency drops when b is increased. This is because a larger b implies a larger arrival rate of the TCP flow and that pushes the algorithm to be operated in the maximum power mode more often. On the other hand, we see from Fig. 12 that packet delay is not greatly affected by the presence of the TCP flow as the Greenput algorithm still clears the backlogs of all the links in each frame with the frame size roughly equal to $T_{\max}$ in a stable system.

H. The effect of time-varying channel quality

In our simulation setting, we use the Shannon formula in (3) to model the transmission rate in the AWGN channel. There we assume that the channel quality is constant in time. However, in reality, the channel quality can vary in time due to many reasons. To further understand the effect of time-varying channel quality to the performance of the Greenput algorithm, we model the channel condition by a continuous-time two-state Markov chain. Specifically, a channel has two states: the ON state, and the OFF state. They correspond to the good channel condition and the bad channel condition, respectively. We assume that the expected sojourn time in the ON (resp. OFF) state is equal to $1/\mu_{ON}$ (resp. $1/\mu_{OFF}$). In this simulation, we set $1/\mu_{ON}$ to be 8 ms, and consider three choices of $1/\mu_{OFF}$: 0.25ms, 2ms and 8ms. The transmission rate when the channel is in the ON state is determined by the Shannon formula in (3). When the channel is in the OFF state, we assume that the transmission rate is only 1/10 of that in the ON state. We choose the intensity $\rho = 0.1$ so that even in a bad channel condition, the overall system utilization is still within the capacity region. The other parameters are the same as those in Section VI-A with $T_{\min} = 1ms$.

In Fig. 13 and Fig. 14, we show the simulation results for energy efficiency and packet delay of the Greenput algorithm with a time-varying channel. It is clear that when we increase the expected sojourn time in the OFF state, i.e., $1/\mu_{OFF}$, the energy efficiency decreases and the expected delay increases. This is because the system is operated at a point near the capacity in the OFF state, and thus the system is in the maximum power mode most of the time when the channel is in the OFF state. On the other hand, we see from Fig. 14 that the expected delay increases moderately when $1/\mu_{OFF}$ is increased to 8ms. This is because the Greenput algorithm still clears the backlogs of all the links in each frame even though the transmission rate in the OFF state is only 1/10 of that in the ON state.

VII. Conclusion

Motivated by the need of energy efficient communication in wireless networks, in this paper we proposed the Greenput algorithm that takes power allocation into account. As the previous DFS algorithms in [13]–[16], the Greenput algorithm is also throughput-optimal for any finite threshold $T_{\max}$ and

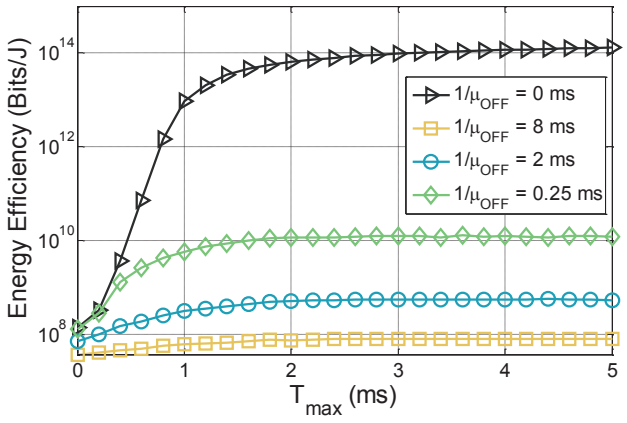

Fig. 13: Energy efficiency when channel quality varies.

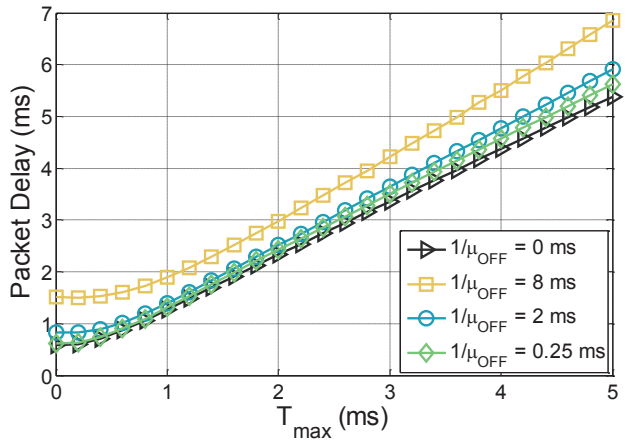

Fig. 14: Packet delay when channel quality varies.

minimum frame length $T_{\min}$. In addition to the stability result, we also performed a fluid approximation analysis for energy efficiency and average packet delay when $T_{\max}$ is very large. To show the effects of $T_{\max}$ and $T_{\min}$ on the delay-energy efficiency trade-off, we conducted extensive computer simulations by using the Shannon formula as the channel model in a wireless network. Our simulation results showed that both energy efficiency and average packet delay are quite close to the fluid approximations even when $T_{\max}$ is moderately large.

There are several possible extensions of this work:

(i) Arrival process: here we assume that the arrival processes are independent Poisson processes. Such an assumption can be easily extended to the stochastic processes in [13]. Such an extension includes finite state Markov arrival processes, renewal processes, and autoregressive processes.

(ii) Multihop networks: here we only consider single-hop wireless networks. To extend our work to a multihop network with per flow queueing for multicast flows, one might consider the hierarchical smooth schedule in [15] to provide guaranteed rate services inside the network. By doing so, the number of packets in any internal buffer is bounded by a deterministic constant.

(iii) Traffic isolation: Like the previous DFS algorithms, the Greenput algorithm does not provide traffic isolation. When the traffic is not admissible, the expected frame size cannot be bounded. As all the links are coupled through the minimum-time scheduling problem that determines the frame size at the beginning of each frame, the performance could be very bad for all the links.

(iv) The tradeoff between energy efficiency and delay in practical systems: in our analysis, we assume that (A1) and (A2) hold. Though these two assumptions are valid for many channel models, including the AWGN channel used in the simulation setting, they may not be true for practical systems. It would be of interest to study the tradeoff between energy efficiency and delay in practical systems, such as the OFDMA systems (see e.g., [29], [30]).

(v) Multiple TCP flows: in Section VI-G, we studied the effect of a single persistent TCP flow. We observed that energy efficiency drops when the buffer size b is increased. This is because a larger TCP buffer pushes the Greenput algorithm to be operated in the maximum power mode more often. For the case that there are K persistent TCP flows with buffer sizes $b_1, b_2, \ldots, b_K$, the effect of these K persistent TCP flows to the Greenput algorithm is the same as a single persistent TCP flow with buffer $\sum_{k=1}^{K} b_k$ (as there are $\sum_{k=1}^{K} b_k$ bits backlogged at the beginning of each frame). However, if TCP flows are not persistent, then the buffers may not be full at the beginning of each frame. For such a more realistic scenario, the performance of the Greenput algorithm requires further study.

REFERENCES

[1] 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. 31, no. 12, pp. 1936–1948, 1992.

[2] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260–1267, 1999.

[3] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proceedings of IEEE INFOCOM 2000.*

[4] M. Armony and N. Bambos, "Queueing dynamics and maximal throughput scheduling in switched processing systems," *Queueing systems*, vol. 44, no. 3, pp. 209–252, 2003.

[5] A. L. Stolyar, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," *Annals of Applied Probability*, vol. 14, no. 1, pp. 1–53, 2004.

[6] P. Chaporkar and S. Sarkar, "Stable scheduling policies for maximizing throughput in generalized constrained queueing networks, " *Proceedings of IEEE INFOCOM 2006.*

[7] 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, no. 1, pp. 89–103, 2005.

[8] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[9] P. Giaccone, E. Leonardi, and D. Shah, "Throughput region of finite-buffered networks," *IEEE Transaction on Parallel and Distributed Systems,* vol. 18, no. 2, Feb. 2007.

[10] L. B. Le, E. Modiano, and N. B. Shroff, "Optimal control of wireless networks with finite buffers, " *Proceedings of IEEE INFOCOM 2010.*

[11] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol.18, no. 3, pp. 960–972, 2010.

[12] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 825–836, 2012.

[13] C. -S. Chang, Y. -H. Hsu, J. Cheng, and D. -S. Lee, "A dynamic frame sizing algorithm for CICQ switches with 100% throughput, " *Proceedings of IEEE INFOCOM 2009.*
[14] C. M. Lien and C. S. Chang, "Generalized dynamic frame sizing algorithm for finite-internal-buffered networks, " *IEEE Communication Letters,* vol. 13, no. 9, Sep. 2009.
[15] C.-M. Lien, C.-S. Chang, J. Cheng, and D.-S. Lee, "Maximizing throughput in wireless networks with finite internal buffers," *Proceedings of IEEE INFOCOM 2011.*
[16] C.-M. Lien, C.-S. Chang, and D.-S. Lee, "A universal stabilization algorithm for multicast flows with network coding," *IEEE Transactions on Communications*, vol. 61, no. 2, pp. 712–721, 2013.
[17] S.-M. He, S.-T. Sun, H.-T. Guan, Q. Zheng, Y.-J. Zhao, and W.Gao, "On guaranteed smooth switching for buffered crossbar switches," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 718-731, June 2008.
[18] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 30–37, 2011.
[19] C. Han, T. Harrold, S. Armour, I. Krikidis, S. Videv, P. M. Grant, H. Haas, et al. "Green radio: radio techniques to enable energy-efficient wireless networks," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 46–54, 2011.
[20] R. Mahapatra, Y. Nijsure, G. Kaddoum, N. U. Hassan and C. Yuen, "Energy efficiency tradeoff mechanism towards wireless green Communication: a survey," *IEEE Communacations Surveys & Tutorials*, vol. 18, no. 1, pp. 686–705, 2016.
[21] V. Angelakis, A. Ephremides, Q. He and D. Yuan, "Minimum-time link scheduling for empty wireless systems: solutions characterization and algorithmic framework," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1083–1100, 2014.
[22] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier and A. Ephremides, "Minimum-energy link scheduling for empty wireless netowrks," *Proceedings of WiOpt 2015*, pp. 207–212.
[23] S. W. Fuhrmann and R. B. Cooper, "Stochastic decompositions in the M/G/1 queue with generalized vacations," *Operations Research*, vol. 33, no. 5, pp. 1117–1129, 1985.
[24] A. Federgruen and H. Groenevelt, "The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality," *Operations Research*, vol. 34, no. 6, pp.909-918, 1986.
[25] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: the multiple node case," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 137-150, 1994.
[26] H. Sariowan, R.L. Cruz and G.C. Polyzos, "Scheduling for Quality of Service Guarantees via Service Curves," *Proceedings of the International Conference on Computer Communications and Networks*, 1995.
[27] C. S. Chang, *Performance Guarantees in Communication Networks,* London: Springer-Verlag, 2000.
[28] G. Sharma, R. R. Mazumdar and N. B. Shroff, "On the complexity of scheduling in wireless networks," *Proceedings of the 12th annual international conference on Mobile computing and networking(MOBICOM)*, pp. 227–238, 2006.
[29] D. W. K. Ng, E. S. Lo, and R. Schober, "Wireless information and power transfer: Energy efficiency optimization in OFDMA systems," *IEEE Transactions on Wireless Communications*, vol. 12, no. 12, pp. 6352–6370, 2013.
[30] L. Venturino, A. Zappone, C. Risi, S. and Buzzi, "Energy-efficient scheduling and power allocation in downlink OFDMA networks with base station coordination," *IEEE transactions on wireless communications*, vol. 14, no. 1, pp. 1–14, 2015.

Cheng-Shang Chang (S'85-M'86-M'89-SM'93-F'04) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1983, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA, in 1986 and 1989, respectively, all in electrical engineering.

From 1989 to 1993, he was employed as a Research Staff Member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Since 1993, he has been with the Department of Electrical Engineering, National Tsing Hua University, Taiwan, where he is a Tsing Hua Distinguished Chair Professor. He is the author of the book Performance Guarantees in Communication Networks (Springer, 2000) and the coauthor of the book Principles, Architectures and Mathematical Theory of High Performance Packet Switches (Ministry of Education, R.O.C., 2006). His current research interests are concerned with network science, high-speed switching, communication network theory, and mathematical modeling of the Internet.

Dr. Chang served as an Editor for Operations Research from 1992 to 1999, an Editor for the *IEEE/ACM TRANSACTIONS ON NETWORKING* from 2007 to 2009, and an Editor for the *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING* from 2014 to 2017. He is currently serving as an Editor-at-Large for the *IEEE/ACM TRANSACTIONS ON NETWORKING*. He is a member of IFIP Working Group 7.3. He received an IBM Outstanding Innovation Award in 1992, an IBM Faculty Partnership Award in 2001, and Outstanding Research Awards from the National Science Council, Taiwan, in 1998, 2000, and 2002, respectively. He also received Outstanding Teaching Awards from both the College of EECS and the university itself in 2003. He was appointed as the first Y. Z. Hsu Scientific Chair Professor in 2002. He received the Merit NSC Research Fellow Award from the National Science Council, R.O.C. in 2011. He also received the Academic Award in 2011 and the National Chair Professorship in 2017 from the Ministry of Education, R.O.C. He is the recipient of the 2017 IEEE INFOCOM Achievement Award.

Duan-Shin Lee (S'89-M'90-SM'98) received the B.S. degree from National Tsing Hua University, Taiwan, in 1983, and the MS and Ph.D. degrees from Columbia University, New York, in 1987 and 1990, all in electrical engineering. He worked as a research staff member at the C&C Research Laboratory of NEC USA, Inc. in Princeton, New Jersey from 1990 to 1998. He joined the Department of Computer Science of National Tsing Hua University in Hsinchu, Taiwan, in 1998. Since August 2003, he has been a professor. He received a best paper award from the Y.Z. Hsu Foundation in 2006. His current research interests are social networks, network science, game theory and data science. He is a senior IEEE member.

Chia-Kai Su received his B.S. degree in the Department of Communications Engineering at Feng Chia University, Taichung, Taiwan, in 2015. He is currently a MS student in the Institute of Communications Engineering at National Tsing Hua University, Hsinchu, Taiwan.

APPENDIX

APPENDIX A

In this section, we prove Theorem 1.

Suppose $\{(t_g^0, \mathbf{p}_g^0), g \in \mathcal{H}\}$ is an optimal schedule for the minimum-energy scheduling problem in (16) with the constraint in (18) being removed. We will construct an optimal TDMA schedule from this schedule so that the amount of energy consumed by this TDMA schedule is not greater than that by the original optimal schedule. If the original optimal schedule is not a TDMA schedule, then there exists some group g with $t_g^0 > 0$ and $|g| > 1$. Assume that the group g contains links $i_1, i_2, \ldots, i_{|g|}$ and the amount of data transmitted during the time interval of length t_g^0 for these links are $d_{i_1}, d_{i_2}, \ldots, d_{i_{|g|}}$, respectively. Clearly, we have

$$d_{i_k} \leq t_g^0 \cdot r_{i_k}(g, \mathbf{p}_g^0),\ k = 1, 2, \ldots, |g|. \tag{31}$$

Also, the amount of energy for the original optimal schedule to transmit the amount of data d_{i_k} on link i_k is

$$t_g^0 \cdot p_{g,i_k}^0. \tag{32}$$

Now we consider another schedule that transmits $d_{i_1}, d_{i_2}, \ldots, d_{i_{|g|}}$ separately. Specifically, the amount of data d_{i_k} is transmitted by using $r_{i_k}(\{i_k\}, \mathbf{p}_g^0)$ for $k = 1, 2, \ldots, |g|$. For this new schedule, the duration to transmit $d_{i,k}$ is $d_{i,k}/r_{i_k}(\{i_k\}, \mathbf{p}_g^0)$ and the amount of energy to transmit $d_{i,k}$ is

$$\frac{d_{i,k}}{r_{i_k}(\{i_k\}, \mathbf{p}_g^0)} \cdot p_{g,i_k}^0.$$

Note from (A2) and (31) that

$$\frac{d_{i,k}}{r_{i_k}(\{i_k\}, \mathbf{p}_g^0)} \cdot p_{g,i_k}^0 \leq \frac{d_{i,k}}{r_{i_k}(g, \mathbf{p}_g^0)} \cdot p_{g,i_k}^0 \leq t_g^0 \cdot p_{g,i_k}^0. \tag{33}$$

Thus, the amount of energy consumed by this new schedule is not greater than that of the original schedule. Repeating the same argument for every group with more than one link yields a schedule that does not have simultaneous transmissions.

Note that in this new TDMA schedule there might be several transmissions of a particular link with different power allocations and time durations. As such, it may not be a simple TDMA schedule. The last step is to replace multiple transmissions of a particular link with different power allocations and time durations by a single transmission with a fixed power. By doing so, it becomes a simple TDMA schedule so that it can be a feasible solution of the minimum-energy problem in (16). Without loss of generality, suppose that for link i, there are transmissions with power $p_{i,\ell}$ and duration $t_{i,\ell}$, $\ell = 1, \ldots, L_i$. The amount of energy consumed by link i is

$$\sum_{\ell=1}^{L_i} t_{i,\ell} \cdot p_{i,\ell},$$

and the amount of data transmitted by link i is

$$\sum_{\ell=1}^{L_i} t_{i,\ell} \cdot r_i(p_{i,\ell}).$$

Now we replace these L_i transmissions on link i by a single transmission with power

$$\bar{p}_i = \frac{\sum_{\ell=1}^{L_i} t_{i,\ell} \cdot p_{i,\ell}}{\sum_{\ell=1}^{L_i} t_{i,\ell}}.$$

Since we assume that $r_i(p_i)$ is concave in the allocated power p_i, it follows that

$$r_i(\bar{p}_i) \geq \frac{\sum_{\ell=1}^{L_i} t_{i,\ell} \cdot r_i(p_{i,\ell})}{\sum_{\ell=1}^{L_i} t_{i,\ell}}.$$

This then implies that the time to transmit $\sum_{\ell=1}^{L_i} t_{i,\ell} \cdot r_i(p_{i,\ell})$ on the link i by a single transmission with power $\bar{p}_i$ is not greater than $\sum_{\ell=1}^{L_i} t_{i,\ell}$. As such, the amount of energy consumed by single transmission with power $\bar{p}_i$ is also not greater than before.

APPENDIX B

In this section, we prove Theorem 2.

For the proof of Theorem 2, we need the following upper bound on the frame sizes.

Lemma 3: For both the generic DFS algorithm in Algorithm 1 and the Greenput algorithm in Algorithm 3, we have

$$T_n \leq \max_{1 \leq i \leq N} \left\lceil \frac{y_i(n)}{\mu_i} \right\rceil + C, \tag{34}$$

where μ_i is the stable service rate for link i in (9).

Proof. Let us first consider the generic DFS algorithm in Algorithm 1. If $y_i(n) = 0$ for all $i = 1, 2, \ldots, N$, then we know $T_n = T_{\min} \leq C$ and the bound in (34) holds trivially. Now assume there exists some i such that $y_i(n) > 0$. If $T_n = T_n''$, then we also know that $T_n \leq T_{\max} \leq C$ from the constraint in (18) of the minimum-energy scheduling problem. Thus, we only need to consider the case that $T_n = T_n'$. In this case, T_n' is the minimum clearance time defined in (14). As such, T_n' is upper bounded by the time needed to empty all the backlogs by a suboptimal way that uses the stable service rate μ_i for each link i. Note from (10) and (A3) that $\mu_i > \lambda_i > 0$. In this case, we then have

$$T_n = T_n' \leq \max_{1 \leq i \leq N} \left\lceil \frac{y_i(n)}{\mu_i} \right\rceil. \tag{35}$$

For the Greenput algorithm, note that the length of a frame cannot be larger than C for the empty queue mode, the power-saving mode, and the mixed power-saving mode. For the maximum power mode, the inequality in (35) also holds. ■

Now we prove Theorem 2. The proof of Theorem 2 is basically the same as the proof for the large deviation bound of the minimum clearance time in [15]. For $\theta > 0$, we have from (34) that

$$\begin{aligned} e^{\theta T_{n+1}} &\leq \max_{1 \leq i \leq N} \exp\left[\theta \left(\frac{y_i(n+1)}{\mu_i} + C\right)\right] \\ &\leq \sum_{1 \leq i \leq N} \exp\left[\theta \left(\frac{y_i(n+1)}{\mu_i} + C\right)\right]. \end{aligned}$$

In view of Algorithms 1 and 3, the backlog $y_i(n+1)$ for link i is simply the arrivals in $[\tau_n, \tau_{n+1}]$.

$$y_i(n+1) = A_i(\tau_n, \tau_{n+1}), \tag{36}$$

where $A_i(\tau_n, \tau_{n+1})$ is a Poisson random variable with mean $\lambda_i T_n$. It then follows that

$$E[e^{\theta T_{n+1}}|T_n] \leq \sum_{1\leq i\leq N} e^{\theta C} \left\{ E\left[\exp\left(\frac{\theta A_i(\tau_n,\tau_{n+1})}{\mu_i}\right)\Big|T_n\right]\right\}. \quad (37)$$

Notice that

$$\begin{aligned} & \log E\left[\exp\left(\frac{\theta A_i(\tau_n,\tau_{n+1})}{\mu_i}\right)\Big|T_n\right] \\ = \;& \lambda_i T_n(e^{\theta/\mu_i}-1) \\ \leq \;& \rho\mu_i T_n(e^{\theta/\mu_i}-1), \end{aligned} \quad (38)$$

where we use (10) in the last inequality.

Hence, we have that

$$E\left[e^{\theta T_{n+1}}|T_n\right] \leq e^{\theta C}\sum_{1\leq i\leq N}\exp\left[\rho T_n\mu_i(e^{\theta/\mu_i}-1)\right]. \quad (39)$$

It was shown in [15] by Taylor's expansion that for fixed $\theta > 0$, $\mu_i(e^{\theta/\mu_i}-1)$ decreases monotonically in μ_i. As such,

$$\mu_i(e^{\theta/\mu_i}-1) \leq \mu_{\min}(e^{\theta/\mu_{\min}}-1) \quad (40)$$

for all $1 \leq i \leq N$. From (39) and (40), we have that

$$E\left[e^{\theta T_{n+1}}|T_n\right] \leq e^{\theta C}N\exp\left(\rho T_n\mu_{\min}(e^{\theta/\mu_{\min}}-1)\right). \quad (41)$$

Taking expectation on both sides of (41) yields

$$E[e^{\theta T_{n+1}}] \leq e^{\theta C}NE\left[\exp(\rho T_n\mu_{\min}(e^{\theta/\mu_{\min}}-1))\right]. \quad (42)$$

According to (27), we have that

$$\rho\mu_{\min}(e^{\theta^*/\mu_{\min}}-1) = \theta^*(\rho+1)/2,$$

and (42) can be rewritten (with θ being replaced by θ^*) as

$$E[e^{\theta^* T_{n+1}}] \leq e^{\theta^* C}NE[e^{\theta^* T_n(1+\rho)/2}]. \quad (43)$$

Since $\log E[e^{\theta T_n}]$ is convex in θ (see e.g., [27, Proposition 7.1.8]) and $\rho < 1$, we have that

$$\log E[e^{\theta^* T_n(1+\rho)/2}] \leq \frac{1+\rho}{2}\log E[e^{\theta^* T_n}]. \quad (44)$$

Using (44) and (43) yields

$$\log E[e^{\theta^* T_{n+1}}] \leq \log N + C\theta^* + \frac{1+\rho}{2}\log E[e^{\theta^* T_n}]. \quad (45)$$

Since $T_1 = T_{\min} \leq C$ (as the network is started from an empty system), one can verify (26) from induction by using (45). Finally, we use (26) to show the bound of the frame size in Theorem 2. Since $e^{\theta x}$ is convex in x, it follows from Jensen's Inequality that

$$E[T_n] \leq \frac{1}{\theta^*}\log E[e^{\theta^* T_n}] \leq \frac{2\log N + 2C\theta^*}{\theta^*(1-\rho)}. \quad (46)$$