Low-Complexity and Distributed Energy Minimization in Multi-hop Wireless Networks

Longbi Lin, Xiaojun Lin, *Member, IEEE*, and Ness B. Shroff, *Fellow, IEEE*

***Abstract*— In this work, we study the problem of minimizing the total power consumption in a multi-hop wireless network subject to a given offered load. It is well-known that the total power consumption of multi-hop wireless networks can be substantially reduced by jointly optimizing power control, link scheduling, and routing. However, the known optimal cross-layer solution to this problem is centralized, and with high computational complexity. In this paper, we develop a low-complexity and distributed algorithm that is provably power-efficient. In particular, under the node exclusive interference model and with suitable assumptions on the power-rate function, we can show that the total power consumption of our algorithm is at most $(2+\epsilon)$-times as large as the power consumption of the optimal (but centralized and complex) algorithm, where ϵ is an arbitrarily small positive constant. Our algorithm is not only the first such distributed solution with provable performance bound, but its power-efficiency ratio is also tighter than that of another sub-optimal centralized algorithm in the literature.**

***Index Terms*— Energy Aware Routing, Duality, Mathematical Programming/Optimization, Cross-Layer Optimization.**

I. Introduction

There has been significant recent interest in developing control protocols for multi-hop wireless networks. Many applications can benefit from the deployment of these networks. For instance, sensors can form multi-hop wireless sensor networks [2] for a variety of applications, such as habitat monitoring and the management of sewer overflow events [3]. Vehicles can form multi-hop wireless networks to exchange safety messages and traffic information [4]. Wireless LAN devices can form multi-hop mesh networks to provide wireless broadband access [5].

A key issue in developing control protocols for multi-hop wireless networks is to reduce the energy or power consumption. This is obviously an important issue for battery-powered networks since the power consumption often limits the lifetime of the network. Even for networks with access to power sources, the transmission power of the communication links may still need to be properly controlled, e.g., due to health or regulatory concerns.

In this work, we are interested in the problem of minimizing the total power consumption of a multi-hop wireless network, subject to a given offered load. The authors of [6], [7] develop general solutions to minimize the total power consumption of the network by jointly optimizing power control, link scheduling, and routing. Although the algorithms in [6], [7] could be implemented in a distributed fashion when each link has an orthogonal channel, in general the algorithms there require centralized computation and high complexity when links interfere with each other. In this paper, we propose a new low-complexity and distributed solution to this problem under a widely-used interference model, called the node-exclusive interference model. Using this model, the work in [8] developed a centralized solution that yielded a 3-approximation ratio (i.e., the resultant power consumption is within a factor of 3 from the optimal power consumption). In contrast, in this paper, we will obtain a $(2+\varepsilon)$-approximation algorithm that is fully distributed, where $\varepsilon > 0$ is an arbitrarily small constant. In a more recent work [9], the authors also develop low-complexity sub-optimal energy minimization algorithms with provable efficiency ratios, under a more general model with multi-receiver diversity. However, the solution in [9] appears to achieve worse power-efficiency ratios than the solution in this paper. (For example, its approximation-ratio increases as the node-degree increases even under the node-exclusive interference model.)

Our solution approach is inspired by the recent progress in using imperfect scheduling algorithms to develop distributed cross-layer congestion control and scheduling algorithms in multi-hop wireless networks [10]–[12]. We first formulate the energy minimization problem into a special form that naturally leads to a distributed solution. We then map the solution to corresponding components of the cross-layer control protocol, and rigorously establish the stability and power-efficiency of the protocol.

Our work is also related to the study of energy-aware routing protocols for minimizing energy consumption and extending network lifetime [13]–[17]. These works assume that the system capacity is battery-limited instead of interference-limited and therefore do not consider scheduling constraints. In contrast, our work explicitly considers scheduling, jointly with power control and routing.

The intellectual contribution of this work is summarized as follows:

- We develop a low-complexity and distributed joint rout-

Manuscript received June 12, 2008; revised April 14, 2009; accepted July 17, 2009. Recommended by Associate Editor Patrick Thiran. This work has been partially supported by ARO MURI Award W911NF-07-10376 (SA08-03), NSF grants CNS-0626703, CNS-0721477, CNS-0721236, CNS-0721434, CCF-0635202, and AFOSR grant 60012034. An earlier version of this paper has appeared in the Proceedings of IEEE INFOCOM, 2007 [1].

Longbi Lin was with School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA, when the research of this work was carried out. He is currently with Morgan Stanley, USA (Email: linlongbi@yahoo.com).

Xiaojun Lin is with Center for Wireless Systems and Applications (CWSA) and School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (Email: linx@ecn.purdue.edu).

Ness B. Shroff is with Department of Electrical and Computer Engineering and Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 (Email: shroff@ece.osu.edu).

ing, power control, and scheduling algorithm for multi-hop wireless networks with provable power-efficiency ratio. Further, our algorithm can guarantee a better power-efficiency level than some existing centralized algorithms.

- To the best of our knowledge, our solution cannot be obtained by extending the known optimal solution in the literature [6], [7]. Instead, we develop an optimization approach to the energy minimization problem that naturally leads to distributed solutions[1]. We also develop rigorous techniques for proving the stability and power-efficiency of the resulted control protocol.

The rest of this paper is organized as follows: in Section II, we present the system model and formulate the energy minimization problem. We present the solution in Section III, and discuss how to map the algorithm to different network protocol components in Section IV. In Section V, we present our main analytical results on the stability and power-efficiency of the proposed protocol. Numerical results are provided in Section VI. Then, we conclude in Section VII.

II. Problem Formulation

We model a wireless multi-hop network by a directed graph $G(V,E)$, where V is the set of vertices representing the nodes, and E is the set of edges representing the communication links. We use $N_o(v)$ and $N_i(v)$ to denote the sets of outgoing and incoming links of node v, respectively. Their union $N(v)$ forms the set of all links incident on node v.

The system is time-slotted. We adopt the following node-exclusive interference model that is used to characterize FH-CDMA and UWB system with perfect orthogonal spreading codes and low power-spectrum density [11], [18]–[20]. Under this model, a node can only receive from or transmit to at most one node at any time-slot m. Further, each link is power-controlled. That is, if the node-exclusive interference constraint is satisfied, we assume that the possible data rate R_e of link e is a function of its own power assignment p_e. We use $p_e = h_e(R_e)$ to denote the power consumption for supporting data rate of R_e on link e. For every link e, it is assumed that $h_e(\cdot)$ is a non-decreasing and convex function on $[0, a_e]$ satisfying $h_e(0) = 0$, where a_e is the maximum rate supported on link e. An example of $h_e(\cdot)$ is the power-rate relationship in an Additive White Gaussian Noise (AWGN) channel.

Each packet may take multiple hops to be delivered from source to destination. Let T_{vd} denote the long-term average data rate of the flow that needs to be supported from source node v to destination node d. We use D to denote the set of destinations.

The joint energy minimization problem is now formulated as follows:

$$\textbf{(*)} \quad \min_{\vec{f},\vec{R}} \quad \lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M}\sum_{e\in E} h_e(R_e(m)), \qquad (1)$$

$$\begin{aligned}
\text{subject to} \quad & R_e(m) \le a_e \text{ for all } e \text{ and } m, \\
& \vec{R}(m) \text{ satisfies the node-exclusive} \\
& \qquad \text{constraint for all time-slots } m, && (2) \\
& \sum_{d\in D} f_e^d = \lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M} R_e(m), \\
& \qquad \text{for all links } e, && (3) \\
& \sum_{e\in N_o(v)} f_e^d - \sum_{e\in N_i(v)} f_e^d - T_{vd} \ge 0, \\
& \qquad \text{for all } d \in D \text{ and nodes } v \ne d, && (4)
\end{aligned}$$

where $R_e(m), m = 1, 2, ...$ is the rate assigned to link e at time slot m, $\vec{R}(m) = [R_e(m)]$, $\vec{f} = [f_e^d]$ and the quantity f_e^d can be interpreted as the average data rate on link e allocated for destination d. The objective function in (1) corresponds to the long-term average energy consumed by all links. The constraints in (3) require that the long-term average data rate, determined by the power allocation, should be able to support the total average data rate ($\sum_{d\in D} f_e^d$) on each link. The constraints in (4) require that the total outgoing flow of a node should be able to support the total incoming flow plus the locally-generated flow, for all destinations. In the rest of the paper, we will refer to the above problem as Problem $(*)$.

III. Solution Methodology

A. Approximating the Energy Minimization Problem

The optimal solutions developed in [6], [7] could be used to solve Problem $(*)$. However, their solutions contain a scheduling component with high computational complexity. In order to compute at which power and at what time each link e should be activated, these solutions need to solve a complex global optimization problem in each time-slot.

In this paper, in order to obtain a low-complexity and distributed solution, we take a different approach. We first approximate $(*)$ by another optimization problem that is easier to solve. The following Lemma [8] provides the first step in this direction.

Lemma 1: There exists a power-optimal solution that solves Problem $(*)$ such that for all time-slots m *when link* e *is activated*, the instantaneous data rate $R_e(m)$ is independent of m.

Lemma 1 follows from the convexity of the function $h_e(\cdot)$ [8]. According to this lemma, we only need to consider those solutions for which there exists a single value R_e, such that $R_e(m) = R_e$ holds *for all time-slots* m *when link* e *is activated.* As a result, $\lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M} R_e(m)$ is equal to the product of R_e and the fraction of time that link e is activated. Therefore, by (3), the objective function of Problem $(*)$ can now be written as

$$\sum_{e\in E}\frac{\sum_{d\in D} f_e^d}{R_e} h_e(R_e),$$

[1] We note however that the solutions in [6], [7] are for more general system models than the node-exclusive interference model studied in this paper.

where $\frac{\sum_{d\in D} f_e^d}{R_e}$ is the fraction of time-slots that link e is activated.

Further, using the results from low-complexity scheduling [10]–[12], we have

- *Fact 1:* In the optimal solution to $(*)$, we must have
$$\sum_{e\in N(v)} \frac{\sum_{d\in D} f_e^d}{R_e} \le 1, \text{ for all nodes } v \in V.$$
- *Fact 2:* Under the node-exclusive interference model, if
$$\sum_{e\in N(v)} \frac{\sum_{d\in D} f_e^d}{R_e} \le \frac{1}{2} - \eta, \text{ for all nodes } v \in V,$$
where $\eta > 0$ is an arbitrarily small positive constant, then a maximal schedule [10]–[12] can be computed such that each link is activated for $\frac{\sum_{d\in D} f_e^d}{R_e}$ fraction of time-slots. We will discuss more about the role of maximal scheduling in our solution in Section IV-B.

Based on these two facts, in the rest of the paper, we will replace the scheduling constraints (2) by
$$\sum_{e\in N(v)} \frac{\sum_{d\in D} f_e^d}{R_e} \le \beta, \text{ for all } v \in V. \tag{5}$$

Problem $(*)$ can then be approximated by the following problem:

$$\text{(A)} \qquad \min_{\vec{f},\vec{R}} \quad \sum_{e\in E} \frac{\sum_{d\in D} f_e^d}{R_e} h_e(R_e), \tag{6}$$
$$\text{subject to} \quad \text{(4) and (5)}$$
$$(\vec{f},\vec{R}) \in X,$$

where $X = \{(\vec{f},\vec{R}) : 0 \le R_e \le a_e, f_e^d \ge 0, \text{ for all links } e \text{ and destinations } d\}$.

Not only is the above formulation (A) easier to solve, it also produces natural bounds for proving the power efficiency ratio of our solution. Indeed, solving (A) with $\beta = 1$ provides a lower bound on the minimum power of Problem $(*)$, while $\beta = \frac{1}{2} - \eta, \eta > 0$ provides an upper bound. Hence, if we can solve Problem (A) with $\beta = \frac{1}{2} - \eta$, we can then obtain a solution with a provable power-efficiency ratio. This efficiency ratio can be derived by assuming a second-order approximation of the power-rate function $h_e(\cdot)$ as in [8]. Specifically, assume that the data rate R_e in an AWGN channel is given by
$$R_e = W \log_2 \left[1 + \frac{\sigma_e p_e}{N_0 W}\right],$$
where W is the available bandwidth, σ_e is the channel gain of link e, N_0 is the noise spectral density, and p_e is the transmission power. The power-rate function $h_e(R_e)$ is then given by
$$p_e = h_e(R_e) = \frac{N_0 W}{\sigma_e}\left(2^{R_e/W} - 1\right). \tag{7}$$
Using a second-order approximation,
$$2^{R_e/W} \approx 1 + \frac{R_e \ln 2}{W} + \frac{1}{2}\frac{R_e^2 \ln^2 2}{W^2},$$
the objective function in (6) can be approximated by
$$\sum_{e\in E} \frac{\sum_{d\in D} f_e^d}{R_e} h_e(R_e) \approx \sum_{e\in E}\left[\frac{N_0 \ln 2}{\sigma_e}\sum_{d\in D} f_e^d + \frac{N_0 \ln^2 2}{2W\sigma_e}(\sum_{d\in D} f_e^d) R_e\right]. \tag{8}$$
Let $(\vec{f},\vec{R})$ be the optimal solution to Problem (A) with $\beta = 1$. It is evident that $(\vec{f},(2+\varepsilon)\vec{R})$, where $\varepsilon = 1/(\frac{1}{2} - \eta) - 2$, is a feasible solution to Problem (A) with $\beta = \frac{1}{2} - \eta$. According to (8), this feasible solution results in power consumption that is (approximately) at most $(2+\varepsilon)$ times the optimal value of Problem (A) with $\beta = 1$. Since the optimal value of Problem (A) with $\beta = 1$ is a lower bound on the minimum power from Problem $(*)$, we conclude that, if we can solve Problem (A) with $\beta = \frac{1}{2} - \eta$, the power-efficiency ratio of the resulting solution is upper-bounded by $(2+\varepsilon)$.

Remark: Problem formulation (A) also appears in [8]. However, our solution is different from this point on. As mentioned earlier, their solution is a centralized one. Further, because of some additional approximation steps, the power efficiency ratio of the solution in [8] is 3 (assuming the same second-order approximation of the power-rate function as in (8)). In contrast, in this paper we will convert Problem (A) to a convex form, which allows us to develop a distributed solution with a better power-efficiency ratio of $2 + \epsilon$.

In the rest of the paper, we assume that Problem (A) is strictly feasible for some $\beta = 1/2 - \eta$, i.e., there exists $(\vec{f},\vec{R}) \in X$ such that the constraints (4) and (5) are satisfied with strict inequality. Note that in practice this assumption can easily be satisfied by picking the maximum data rate a_e to be sufficiently large.

B. Handling the Non-Convexity

In Problem (A), the objective function and the constraint (5) are non-convex. Problems of this type are considered to be difficult in general. To overcome this difficulty, the following change of variable is performed:
$$t_e = \frac{\sum_{d\in D} f_e^d}{R_e}, \quad \text{for all links } e \in E.$$
The parameter t_e can be interpreted as the *fraction of time-slots* for which link e is activated. Due to the constraint (5), there is no loss of optimality by assuming that $t_e \le 1$ for each link e. Later in Section IV-A, we will interpret t_e also as the *offered load* on link e. The latter interpretation will become more appropriate in Section IV-A when we deal with $t_e(m)$ for each time-slot in the dual solution. With this change of variable, we can denote the long-term average power consumption of link e as a function of t_e and $\vec{f_e} = [f_e^d, d \in D]$, i.e., $\Theta_e(\vec{f_e}, t_e) = t_e h_e(\frac{\sum_{d\in D} f_e^d}{t_e})$, for $t_e > 0$. To define the value of the function Θ_e for $t_e = 0$, note that $\sum_{d\in D} f_e^d / t_e = R_e \le a_e$. Hence, the only feasible point when $t_e = 0$ is when $\sum_{d\in D} f_e^d$ is also equal to 0. Let
$$Y_e = \{(\vec{f_e}, t_e) : f_e^d \ge 0, \text{ for all } d \in D; 0 \le t_e \le 1; \sum_{d\in D} f_e^d \le a_e t_e\}. \tag{9}$$

We can then define the function $\Theta_e(\vec{f}_e, t_e)$ on the domain Y_e as

$$\Theta_e(\vec{f}_e, t_e) = \begin{cases} 0, & t_e = 0, \\ t_e h_e\left(\frac{\sum_{d\in D} f_e^d}{t_e}\right), & 0 < t_e \le 1. \end{cases} \tag{10}$$

Note that this definition ensures that the function Θ_e is continuous on its domain.

Let $\vec{t} = [t_e, e \in E]$, and let Y denote the Cartesian product of Y_e for all e, i.e.,

$$\begin{aligned} Y &= \{(\vec{f}, \vec{t}) : f_e^d \ge 0, \text{ for all edges } e \text{ and destinations } d; \\ &\qquad 0 \le t_e \le 1, \sum_{d\in D} f_e^d \le a_e t_e, \text{ for all edges } e\}. \end{aligned}$$

Using the above notation, Problem (A) can be transformed into

$$\begin{aligned} \text{(B)} \qquad \min_{\vec{f},\vec{t}} \quad & \sum_{e\in E} \Theta_e(\vec{f}_e, t_e), & (11) \\ \text{subject to} \quad & \sum_{e\in N(v)} t_e \le \beta, \text{ for all nodes } v \in V, & (12) \\ & \sum_{e\in N_o(v)} f_e^d - \sum_{e\in N_i(v)} f_e^d - T_{vd} \ge 0, & \\ & \qquad \text{for all } d \in D \text{ and nodes } v \ne d, & (13) \\ & (\vec{f}, \vec{t}) \in Y. & \end{aligned}$$

We emphasize that Problem (A) and Problem (B) are equivalent. We now show that Problem (B) is a convex program. We need the following lemma.

Lemma 2: Assume that $h(r)$ is a convex function on $r \ge 0$. Let

$$\theta(f, t) = \begin{cases} 0, & t = 0, \\ t h\left(\frac{f}{t}\right), & t > 0. \end{cases} \tag{14}$$

Then $\theta(f, t)$ is also convex on the domain $C = \{(f, t) : 0 \le f \le at, 0 \le t \le 1\}$.

The proof is available in our online technical report [21] and is related to the *perspective* of the function $h(r)$ [22, Chapter 3.2.6, p89]. From Lemma 2, each term $\Theta_e(\vec{f}_e, t_e)$ in the objective function of Problem (B) is convex, and hence the entire problem is a convex program. We can then use the following duality approach to solve the problem.

C. Distributed Algorithm Based on Lagrange Duality

To use the duality approach to solve Problem (B), we first form the Lagrangian:

$$\begin{aligned} & L(\vec{f}, \vec{t}, \vec{\mu}, \vec{q}) \\ = \; & \sum_{e\in E} \Theta_e(\vec{f}_e, t_e) + \sum_{v\in V} \mu_v \left(\sum_{e\in N(v)} t_e - \beta \right) \\ & + \sum_{v\in V} \sum_{d\in D} q_v^d \left(\sum_{e\in N_i(v)} f_e^d + T_{vd} - \sum_{e\in N_o(v)} f_e^d \right), \end{aligned}$$

where $\vec{\mu} = [\mu_v, v \in V] \ge 0$ and $\vec{q} = [q_v^d, v \in V, d \in D] \ge 0$ are the Lagrange multipliers for the constraints (12) and (13), respectively. For ease of notation, we define $q_d^d = 0$, for all d. By rearranging the order of the summations, the above equation can be transformed into the following:

$$\begin{aligned} & L(\vec{f}, \vec{t}, \vec{\mu}, \vec{q}) \\ = \; & \sum_{e\in E} c_e(\vec{f}_e, t_e) - \beta \sum_{v\in V} \mu_v + \sum_{v\in V} \sum_{d\in D} (q_v^d T_{vd}), \end{aligned}$$

where

$$\begin{aligned} c_e(\vec{f}_e, t_e) &= \Theta_e(\vec{f}_e, t_e) + (\mu_{x(e)} + \mu_{r(e)}) t_e \\ & \quad - \sum_{d\in D} (q_{x(e)}^d - q_{r(e)}^d) f_e^d, \end{aligned} \tag{15}$$

and $x(e)$ and $r(e)$ are the transmitting node and receiving node, respectively, of link e.

The dual objective function is then

$$\begin{aligned} D(\vec{\mu}, \vec{q}) &= \min_{(\vec{f},\vec{t})\in Y} L(\vec{f}, \vec{t}, \vec{\mu}, \vec{q}), \\ &= \sum_{e\in E} \left[\min_{(\vec{f}_e, t_e)\in Y_e} c_e(\vec{f}_e, t_e) \right] - \beta \sum_{v\in V} \mu_v \\ & \quad + \sum_{v\in V} \sum_{d\in D} (q_v^d T_{vd}), \end{aligned} \tag{16}$$

where Y_e is given by (9). In other words, the minimization of the Lagrangian can now be decomposed into minimization subproblems for each link. Note that all the information needed in minimizing $c_e(\vec{f}_e, t_e)$ is local to link e.

The dual optimization problem is

$$\text{(C)} \qquad \max_{\vec{\mu}\ge 0, \vec{q}\ge 0} D(\vec{\mu}, \vec{q}). \tag{17}$$

Let Θ^* denote the optimal solution to Problem (B). Assuming that the primal problem (B) is strictly feasible, the Slater condition can be verified. We can then conclude that there is no duality gap as in standard results [22, Chapter 5.2.3, p226].

Theorem 3: **(Strong Duality)**

Assume that Problem (B) is strictly feasible (and hence its optimal value Θ^* is finite.) Then there is no duality gap, i.e., $\Theta^* = \max_{\vec{\mu}\ge 0, \vec{q}\ge 0} D(\vec{\mu}, \vec{q})$.

The next step is to solve the dual problem in a distributed fashion. We can show that $D(\vec{\mu}, \vec{q})$ is concave and a subgradient of $D(\cdot, \cdot)$ at $(\vec{\mu}, \vec{q})$ is given by the following vector $\vec{g}$ such that the components of $\vec{g}$ are given by

$$\begin{aligned} [\vec{g}]_{\mu_v} &= \sum_{e\in N(v)} t_e - \beta, \\ [\vec{g}]_{q_v^d} &= \sum_{e\in N_i(v)} f_e^d + T_{vd} - \sum_{e\in N_o(v)} f_e^d, \end{aligned}$$

where $\vec{f}$ and $\vec{t}$ solve (16). We can then use the following subgradient-ascent method to solve the dual problem.

Distributed Energy Minimization Algorithm

At each iteration m,

1) At link e, the data rate $\vec{f}_e$ and the link assignment t_e are determined by:

$$
\begin{aligned}
(\vec{f}_e(m), t_e(m)) &= \operatorname*{argmin}_{(\vec{f}_e,t_e)\in Y_e} c_e(\vec{f}_e, t_e, m) \\
= \operatorname*{argmin}_{(\vec{f}_e,t_e)\in Y_e} &\Big[\Theta_e(\vec{f}_e, t_e) \\
&+ (\mu_{x(e)}(m) + \mu_{r(e)}(m))t_e \\
&- \sum_{d\in D}(q^d_{x(e)}(m) - q^d_{r(e)}(m))f^d_e\Big]. \quad (18)
\end{aligned}
$$

2) At node v, the dual variables are updated by:

$$
\mu_v(m+1) = \left\{\mu_v(m) + \alpha_m\left[\sum_{e\in N(v)} t_e(m) - \beta\right]\right\}^+ \quad (19)
$$

$$
\begin{aligned}
q^d_v(m+1) = \Bigg\{ q^d_v(m) + \alpha_m \Bigg[&\sum_{e\in N_i(v)} f^d_e(m) + T_{vd} \\
&- \sum_{e\in N_o(v)} f^d_e(m)\Bigg]\Bigg\}^+, \quad (20)
\end{aligned}
$$

where $\{\alpha_m\}$ is a sequence of positive stepsizes.

Remark: In the above algorithm, we use the same stepsize for updating $\vec{\mu}$ and $\vec{f}$ at each iteration. This is simply for ease of notation: the stepsizes can be different for each dual variable.

The above exercise of using Lagrange duality is standard. However, there are a number of questions that are not answered by the above algorithm alone. First, how does the algorithm translate to practical network protocol components? Second, note that the primal problem is not strictly convex. Specifically, the objective function of Problem (B) contains a linear term (see (10)). Hence, the dual objective function is not always differentiable. In practice, in order to dynamically track the changes of the network condition (e.g., as the offered load T^d_v changes), it is typical that a constant stepsize is used. As a consequence of the lack of differentiability of the dual objective function, the algorithm will not always be able to converge to a single operating point. Rather, the dual variables $\mu_v(m)$ and $q^d_v(m)$ could oscillate around their corresponding optimal values. Further, even if the dual variables are close to the optimal values, the primal variables $\vec{f}(m)$ and $\vec{t}(m)$ may not. Hence, it is not immediately clear what level of performance this algorithm will be able to achieve. In the next two sections, we will carefully address these questions and quantify the performance levels of the resulting protocol.

IV. Mapping to Network Protocol Components

In this section, we will map the Distributed Energy Minimization Algorithm to various protocol components. Recall that in each iteration m the Distributed Energy Minimization Algorithm updates the variables $\vec{f}(m), \vec{t}(m)$, $\vec{\mu}(m)$ and $\vec{q}(m)$ according to Equations (18)-(20). We will basically identify *iteration* m with *time-slot* m, and use the values of these variables as the control decision at *time-slot* m.

A. Routing, Power Control and Link Assignment

In Step 1 of the Distributed Energy Minimization Algorithm, each link solves (18) by minimizing $c_e(\vec{f}_e, t_e, m)$. We now investigate the structure of this minimization problem, and we will show that this step corresponds to the routing, power control and link-assignment protocol components. We first introduce the following transformation:

$$
f^d_e(m) = R^d_e(m)t_e(m). \quad (21)
$$

Recall that $f^d_e(m)$ is an estimate (at iteration m) of the average data rate on link e allocated for destination d, and $t_e(m)$ is an estimate (at iteration m) of the fraction of time-slots that link e is activated. Thus, $R^d_e(m)$ can be viewed as an estimate of the instantaneous data rate allocated on link e for destination d. Substituting the above equation into $c_e(\vec{f}_e, t_e, m)$, we have (dropping the time index m when there is no source of confusion):

$$
c_e(\vec{f}_e, t_e) = t_e l_e(\vec{R}_e), \quad (22)
$$

where $\vec{R}_e = [R^d_e, d \in D]$, and $l_e(\vec{R}_e)$ is defined as

$$
\begin{aligned}
l_e(\vec{R}_e) &= h_e(\sum_{d\in D} R^d_e) + (\mu_{x(e)} + \mu_{r(e)}) \\
&\quad - \sum_{d\in D}\left[(q^d_{x(e)} - q^d_{r(e)})R^d_e\right]. \quad (23)
\end{aligned}
$$

Since $t_e \geq 0$, to minimize (22), we should first minimize $l_e(\vec{R}_e)$ as a function of $\vec{R}_e$ over $0 \leq \sum_{d\in D} R^d_e \leq a_e$. Note that function $h_e(\cdot)$ takes as input parameter the sum of the data rates allocated for all the destinations on this link. In other words, from the viewpoint of power consumption, it is indifferent to which destination the data rate is allocated for, as long as the total data rate is the same. Further, we can interpret $q^d_v(m)$ in (20) as the *backlog* at node v for destination d (since it captures the cumulative difference between the input rate and output rate at node v for destination d). Then, the minimum of $l_e(\vec{R}_e)$ is attained when *all the data rates are allocated to the destination with the maximum positive backlog difference.* In other words, if we let

$$
\hat{d} = \operatorname*{argmax}_{d\in D}(q^d_{x(e)} - q^d_{r(e)}), \quad (24)
$$

then to find the optimal value of $\vec{R}_e$, we should let $R^d_e = 0$ for all $d \neq \hat{d}$. With this observation, the minimization of $l_e(\vec{R}_e)$ can be reduced to a minimization problem of a single-variable function, i.e.,

$$
\begin{aligned}
\min_{0\leq R_e\leq a_e} \qquad l_e(R_e) &= h_e(R_e) + (\mu_{x(e)} + \mu_{r(e)}) \\
&\quad - \left[\max_{d\in D}(q^d_{x(e)} - q^d_{r(e)})\right] R_e. \quad (25)
\end{aligned}
$$

Let $R_e(m)$ denote the optimal solution to (25). We can then set $R^d_e(m) = R_e(m)$ if $d = \hat{d}$, and $R^d_e(m) = 0$, otherwise. For example, if $h_e(x) = \exp(x) - 1$, and $\max_{d\in D}(q^d_{x(e)}(m) - q^d_{r(e)}(m)) > 0$, then $R^{\hat{d}}_e(m) = [\log(q^{\hat{d}}_{x(e)}(m) - q^{\hat{d}}_{r(e)}(m))]^+$, and $R^d_e(m) = 0$ for all other destinations $d \neq \hat{d}$.

Now that $\vec{R}_e$ has been chosen to minimize $l_e(\vec{R}_e)$ in (23), the next step is to determine the value of t_e over the interval

$[0, 1]$ to minimize $c_e(\vec{f}_e, t_e) = t_e l_e(\vec{R}_e)$. Clearly, the optimal t_e value is

$$t_e(m) = \begin{cases} 1, & \text{if } l_e(R_e(m)) \leq 0, \\ 0, & \text{if } l_e(R_e(m)) > 0. \end{cases} \tag{26}$$

Note that among the three terms of $l_e(R_e)$ in (25), the term $h_e(R_e)$ can be viewed as the *power cost* due to power consumption; the term $(\mu_{x(e)}(m)+\mu_{r(e)}(m))t_e$ can be viewed as the *scheduling cost* due to the constraint on the fraction of the time each link can be scheduled; and the term $\sum_{d\in D}(q^d_{x(e)}(m) - q^d_{r(e)}(m))f^d_e$ can be viewed as the *utility* of transporting data on link e. Hence, according to (26) we will assign $t_e(m) = 1$ only when *the utility of transporting data to next hop is no smaller than the power cost plus the scheduling cost*. Substituting into (21), the values of $\vec{f}(m)$ can then be set as $f^d_e(m) = R_e(m)$ if $t_e(m) = 1$ and $d = \hat{d}$, and $f^d_e(m) = 0$, otherwise.

So far we have derived the values of $\vec{f}(m)$ and $\vec{t}(m)$ at iteration m according to (18). We now use the values of these variables as the control of the various protocol components in time-slot m. The minimization of $c_e(\vec{f}_e, t_e, m)$ on each link then naturally translates into the following protocol components. At each time-slot m,

1) *Routing:* Choose only the flow $\hat{d}(m)$ with maximum positive backlog difference (cf. (24)). This is the flow that should receive service.
2) *Power control:* Choose $R_e(m)$ to minimize $l_e(R_e(m))$ in (25). Then, $h_e(R_e(m))$ is the power assignment that link e should use, and $R_e(m)$ is the corresponding data rate assigned to link e when it is turned on.
3) *Link assignment:* Choose $t_e(m)$ to minimize $t_e(m)l_e(R_e(m))$ in such a way that $t_e(m)$ takes its maximum value 1 if the optimal $l_e(R_e(m))$ is less than or equal to 0; and 0 otherwise. This determines the amount of time that link e should be on.

Therefore, when link e is turned on (by the scheduling component to be discussed in Section IV-B), it will then use the power level $h_e(R_e(m))$ (and the corresponding data rate $R_e(m)$) to transfer packets for destination $\hat{d}(m)$ from its transmitting node to its receiving node in the corresponding time-slot. (Note that each link will carry packets for at-most one destination at each time-slot m.) Once the decisions of the above protocol components are determined based on the dual variables $\mu_v(m)$ and $q^d_v(m)$, these dual variables are then updated according to (19) and (20). The only remaining problem, however, is that not all links with $t_e(m) = 1$ can be activated immediately because they may violate the node-exclusive interference constraints. We next address this issue.

B. The Maximal-Matching Scheduling Component

As we have seen thus far, the duality approach exploits the problem structure and decomposes the primal problem into sub-problems that immediately translate to protocol components. However, in the above discussions, although we have considered the scheduling constraint in the form of (12), we have not studied the actual schedule of activating the links. In particular, the links that are assigned $t_e(m) = 1$ may in fact interfere with each other, and hence cannot all be activated immediately in time-slot m. A scheduling algorithm is then needed to schedule (at least some of) these links for activation at a later time. In particular, for each link e and each time-slot m such that $t_e(m) = 1$, define $\delta_e(m) \geq m$ as the time-slot when the scheduling algorithm can actually activate the link e for this particular link-assignment instance. For obvious reasons, we require a one-to-one mapping from each time-slot m with $t_e(m) = 1$ to $\delta_e(m)$, and a natural ordering of $\delta_e(m)$ such that $\delta_e(m_1) < \delta_e(m_2)$ for every $m_1 < m_2$ and $t_e(m_1) = t_e(m_2) = 1$. Hence, the inverse mapping of $\delta_e(m)$ is well-defined, and we denote it by $\delta_e^{-1}(m)$. For those time-slots m when link e is not activated, they do not correspond to any time-slot m' with $t_e(m') = 1$ and $\delta_e(m') = m$. In this case, we define $\delta_e^{-1}(m) = -1$. Consider the following equation,

$$Q^d_v(m+1) = \Big\{Q^d_v(m) + \alpha_m \Big[T^d_v + \sum_{e\in N_i(v)} f^d_e(\delta_e^{-1}(m)) - \sum_{e\in N_o(v)} f^d_e(\delta_e^{-1}(m))\Big]\Big\}^+. \tag{27}$$

It is easy to see that when $\alpha_m = \alpha$ and $Q^d_v(0) = 0$, $Q^d_v(m)/\alpha$ provides an upper bound on the real queue maintained at node v for packets destined to node d. (It is an upper bound because the actual number of incoming packets to node v are less than or equal to $\sum_{e\in N_i(v)} f^d_e(\delta_e^{-1}(m))$.) Note that here we have used the convention that $f^d_e(-1) = 0$ for all d and e, which is consistent with the value of $f^d_e(m)$ for those time-slots when $t_e(m) = 0$. Thus, we need to design an algorithm for determining $\delta_e(m)$ such that: (i) the real queues are bounded; and (ii) the energy consumption is close to the minimum value of Problem (B).

We address this problem by mapping it to a scheduling problem for stability (similar to those in [10], [11], [23]–[26]) as follows. Consider a virtual system with the same topology as the original system, except that each link e has a unit capacity. Whenever $t_e(m) = 1$, we offer a virtual packet with unit length to link e. Hence, the process $t_e(m), m = 1, 2, ...$ represents the virtual offered load to link e. We let each virtual packet remember the time-slot that it arrives. Note that under the node-exclusive interference model, at any time slot the feasible schedule must be a matching. (A matching of a graph is a subset of the links such that no two links share a common node.) At each time-slot m, once a matching in the virtual system is chosen, each matched link e will then serve one virtual packet. In the original system, this service corresponds to assigning $\delta_e(m') = m$, where m' is the arrival time-slot of the head-of-line virtual packet just served. In other words, in the original system link e is activated at time-slot m, and we use the value of $f^d_e(m')$ to serve packets on link e at this time-slot m.

With this construction, the amount of backlog and the delay in the virtual system then correspond to the number of link-assignments pending to be scheduled and the scheduling delay, respectively, in the original system. Intuitively, if we can design a scheduling algorithm that can keep the backlog and delay of the virtual system to be bounded, then the real queue $Q_v^d(m)$ will be bounded as long as the dual variables $q_v^d(m)$ are bounded. Further, such a bounded scheduling delay will not alter the long-term average power consumption. (This argument will be made precise in Section V-B.)

In this paper, we are interested in a simple scheduling algorithm called *maximal-matching*. This algorithm will schedule a matching with backlogged links such that no more links can be added without violating the node-exclusive interference constraint. More precisely, denote the virtual backlog at link e as

$$\nu_e(m) = \sum_{s=1}^{m-1} \mathbf{1}_{\{t_e(s)=1\}} - \sum_{s=1}^{m-1} \mathbf{1}_{\{\delta_e^{-1}(s) \neq -1\}}.$$

Clearly, $\nu_e(m) \geq 0$ for all m. Let $\vec{\nu}(m) = [\nu_e(m)]$. Denote $\mathcal{M}(m)$ as the actual set of links that are scheduled in the virtual system at time-slot m. Then, for the maximal-matching scheduling policy, at least one of the following statements must be true. For any link e that has non-zero virtual backlog (i.e., $\nu_e(m) \geq 1$):

- either $e \in \mathcal{M}(m)$;
- or $l \in \mathcal{M}(m)$ for some interfering link $l \in N(x(e))$;
- or $l \in \mathcal{M}(m)$ for some interfering link $l \in N(r(e))$;

where $x(e)$ and $r(e)$ are the transmitting node and receiving node, respectively, of link e. (Note that links with zero virtual backlog are not scheduled.) The evolution of the virtual backlog $\nu_e(m)$ is then given by

$$\nu_e(m+1) = \nu_e(m) + t_e(m) - \mathbf{1}_{\{e \in \mathcal{M}(m)\}}. \tag{28}$$

Let λ_e denote the average number of packets that arrive to link e in the virtual system. It is well-known that, under the node-exclusive interference model, when the offered load satisfies $\sum_{e \in N(v)} \lambda_e < 1/2$ for all nodes v, then maximal-matching is guaranteed to produce a schedule such that the virtual backlog at each link does not grow to infinity. In the Distributed Energy Minimization Algorithm, if we choose $\beta = 1/2 - \eta$ for some $\eta > 0$, then as long as the dual variable $\mu_v(m)$ is bounded, it implies that the long-term offered-load to the virtual system satisfies $\sum_{e \in N(v)} \lambda_e \leq \beta$ for all nodes v. Hence, the virtual backlog of each link under the maximal matching policy will be finite. As we will show in Section V, we will then obtain a scheduling algorithm that both keeps the real queue $Q_v^d(m)$ bounded, and achieves power-efficiency close to the solution of Problem (B).

The maximal-matching scheduling policy is easy to implement. Essentially, if link e is backlogged (in the virtual system) and its neighbors have not been scheduled, then link e itself should be scheduled. The maximal-matching scheduling policy can be implemented in a distributed fashion using the algorithm in [27]. This algorithm proceeds in rounds, where in each round it computes a (not necessarily maximal) matching, and then proceeds to the next round by removing those links that either have been matched or that have a neighboring link that is matched. The algorithm terminates when there are no links left. Then, the union of the matchings found in all rounds form a maximal matching. In each round, the algorithm only requires each node to exchange a small amount of control messages with neighboring nodes. The result of [27] shows that the average number of rounds required to compute a maximal matching is $O(\log |E|)$, where $|E|$ is the total number of links in the network. We refer the readers to [27], [28] for more details on the distributed implementation of maximal schedules.

By combining the Distributed Energy Minimization Algorithm and the maximal matching scheduling policy, we then obtain a cross-layer protocol for energy-minimization, which is summarized as follows. In this protocol, we use the convention that the computation at each link e is carried out at the transmitting node $x(e)$. We also implement the virtual backlog at link e as a FIFO queue.

Distributed Energy Minimization Protocol

At each time-slot m,

1) Each node v exchanges the value $\mu_v(m)$ and $q_v^d(m)$ with its immediate neighbors (that share a common link).
2) Link e finds the flow $\hat{d}(m)$ with the maximum positive backlog difference in (24).
3) Link e calculates the rate assignment $R_e(m)$ that minimizes $l_e(R_e(m))$ in (25). The corresponding power assignment is then $h_e(R_e(m))$.
4) A virtual packet with the information $\hat{d}(m)$ and $R_e(m)$ is appended to the end of the virtual queue at link e.
5) The algorithm in [27] is used to compute a maximal matching among those links with positive virtual backlogs.
6) For each link e that is chosen in the maximal matching, remove one virtual packet from the head of the virtual queue at link e. Let m' denote the time-slot that this virtual packet arrived.
7) Link e then uses the value $\hat{d}(m')$ and $R_e(m')$ as the routing and power control decision at time-slot m.
8) Each node v updates the dual variables μ_v and q_v^d according to (19) and (20).

Note that to carry out the above control protocol, each link only needs to know the dual-variables at its end-points. Further, to update the dual variables, each node only needs to know the control decisions at the links incident to it. Taking into account the overhead of the distributed implementation of maximal matching (discussed earlier), the number of messages that each node needs to exchange with its immediate neighbors in each iteration is of the order $O(\bar{\aleph} \log |E|)$ where $\bar{\aleph}$ is the maximum node-degree. In the next section, we will carefully quantify the stability and power-efficiency of the cross-layer protocol proposed above.

V. Performance Analysis

In this section, we will answer the following two questions. First, can the protocol developed in Section IV support the offered load given by $[T_v^d]$? Second, what is the power-efficiency of the proposed cross-layer control protocol? We note that these questions cannot be answered by standard results in convex optimization and duality theory alone. The reason is because the introduction of the maximal-matching scheduling component in Section IV-B leads to some complication in analyzing the dynamics of the protocol. In particular, a link with $t_e(m) = 1$ may need to be activated at a later time $\delta_e(m)$. This delayed activation leads to a discrepancy between the value of $q_v^d(m)$ and that of the real queue. For ease of exposition, in this section we will first ignore the maximal-matching scheduling component, and study the properties of the control variables $f_e^d(m)$ and $t_e(m)$ computed by (18)-(20) under the assumption that all links with $t_e(m) = 1$ can be activated immediately in time-slot m. We then remove this unrealistic assumption, and relate the properties of $f_e^d(m)$ and $t_e(m)$ to the actual performance of the protocol when the maximal-matching scheduling component is used.

A. Properties of $f_e^d(m)$ and $t_e(m)$ with Constant Stepsizes

Since the algorithm in (18)-(20) is a standard subgradient-ascent algorithm for the dual problem, we would expect that the dual variables will converge to a neighborhood of some optimal value. However, the primal variables $(\vec{f}(m), \vec{t}(m))$ will likely oscillate. For example, $t_e(m)$ is either 0 or 1 according to Equation (26). A natural question to ask then is the following: *In what sense are the primal variables $f_e^d(m)$ and $t_e(m)$ optimal?*

The following theorem answers this question. Recall that Θ^* is the minimum value of Problem (B). Let

$$\bar{W} = 4|E|^2 + |V| + 8\left(\sum_{e\in E} a_e\right)^2 + 2\sum_{v\in V}\sum_{d\in D} T_{vd}^2. \quad (29)$$

Theorem 4: Let the stepsizes in the Distributed Energy Minimization Algorithm be equal to a constant, i.e., $\alpha_m = \alpha$, for all time-slots m. Let Φ be the set of $(\vec{\mu}, \vec{q})$ that maximizes $D(\vec{\mu}, \vec{q})$, and define the distance metric $d((\vec{\mu}, \vec{q}), \Phi) \triangleq \min_{(\vec{\mu}^*, \vec{q}^*)\in\Phi} \|(\vec{\mu}, \vec{q}) - (\vec{\mu}^*, \vec{q}^*)\|$. Given any $\varepsilon > 0$, there exists some $\bar{\alpha}_0 > 0$ such that, for any $\alpha \le \bar{\alpha}_0$ and any initial implicit costs $(\vec{\mu}(0), \vec{q}(0))$, there exists a time M_0 such that for all $m > M_0$,

$$d((\vec{\mu}(m), \vec{q}(m)), \Phi) < \varepsilon. \quad (30)$$

Further, for any $m \ge \frac{2V(\vec{\mu}(0), \vec{q}(0))}{\alpha^2 \bar{W}}$, we have

$$\frac{1}{m}\sum_{\tau=1}^{m}\sum_{e\in E} \Theta_e(\vec{f}_e(\tau), t_e(\tau)) \le \Theta^* + \alpha\bar{W}. \quad (31)$$

The proof of Theorem 4 is provided in Appendix A. It shows that, when stepsizes are small, the dual variables eventually converge to within a small neighborhood of the optimal dual solution. Note that under the assumption that all links with $t_e(m) = 1$ can be activated immediately in time-slot m, the boundness of $q_v^d(m)$ immediately implies that the offered load $[T_v^d]$ are supported by the protocol. In other words, according to (20),

$$\sum_{s=m}^{m+\tau_1-1}\left\{\sum_{e\in N_i(v)} f_e^d(s) + T_v^d - \sum_{e\in N_o(v)} f_e^d(s)\right\}$$

must be bounded for all time-slots m and for all $\tau_1 > 0$. Hence, the constraint in (4) is satisfied when we take f_e^d in (4) as the long-term average of $f_e^d(m)$. Further, Theorem 4 shows that the power consumption determined by the primal variables $f_e^d(m)$ and $t_e(m)$ is close to the minimal value Θ^* of Problem (B). In other words, even though the primal variables $(\vec{f}(m), \vec{t}(m))$ may not converge, by using $(\vec{f}(m), \vec{t}(m))$ for each time-slot m, the long-term average of the resultant power consumption is arbitrarily close to Θ^*. Finally, Theorem 4 reveals a tradeoff between power-efficiency and the convergence speed (depending on the step-size α). A smaller α will drive the average power-consumption closer to Θ^*, although it will take a larger number of iterations before (31) holds.

B. Stability and Power-Efficiency with the Maximal-Matching Scheduling Component

Theorem 4 establishes the stability and optimality of the primal variables $f_e^d(m)$ and $t_e(m)$. However, as we discussed at the beginning of this section, due to the delayed-activation of the links, there is a discrepancy between the real queue and the value of $q_v^d(m)$. Hence, in order to ensure that the offered load $[T_v^d]$ is supported, we must prove that the real-queue is stable. Further, we must show that the delayed-activation of the links does not change the average energy consumption in the system. Towards this end, we first show that the delay in activating the links is bounded.

Lemma 5: Assume that the positive stepsizes α_m are fixed, i.e., $\alpha_m = \alpha$ for all time-slots m, where $0 < \alpha < \bar{\alpha}_0$ and $\bar{\alpha}_0$ is given in Theorem 4. Let Q_1 be the bound on the dual variables $\mu_v(m)$ for all nodes v and time-slots m. (Note that such a bound exists due to Theorem 4). Let

$$\bar{T}_0 = \frac{16Q_1}{\alpha(1/2-\beta)^3}\sum_{v\in V}|N(v)|^2,$$

Then for any link e and any m with $t_e(m) = 1$, the delay in activating the link e is no greater than $\bar{T}_0$, i.e.

$$\delta_e(m) - m \le \bar{T}_0.$$

The proof is provided in Appendix B. To obtain the above upper bound on the delay, we have assumed that, for each link e with a positive virtual backlog, either link e or *only one* of its neighboring links must be scheduled. As a result, the delay bound in Lemma 5 increases to infinity as β approaches $1/2$. In practice, it is possible that maximal matching can pick two neighboring links of link e at the same time. Hence, as we observe in the simulation results in Section VI, the actual delay is often much smaller than the above bound.

We can now state the main result of this section.

Theorem 6: Assume that the positive stepsizes α_m are fixed, i.e., $\alpha_m = \alpha$ for all time-slots m, where $0 < \alpha < \bar{\alpha}_0$ and $\bar{\alpha}_0$ is given in Theorem 4. Let Q_3 be the bound on the dual variables $q_v^d(m)$ for all v, d and m. (Note that such a bound exists due to Theorem 4.) With the maximal-matching scheduling policy stated earlier, the real queues $Q_v^d(m)$ at all nodes must be bounded at all time-slots by

$$Q_v^d(m) \le Q_3 + \alpha\bar{T}_0[2\sum_{e\in N_o(v)} a_e + T_v^d],$$

where $\bar{T}_0$ is the maximum delay given in Lemma 5. Hence, the offered-load $[T_v^d]$ is supported by the cross-layer control protocol. Further, the long-term average energy consumption

TABLE I
THE TWO FLOWS SUPPORTED BY THE NETWORK

	source	destination	data rate	paths
flow 1	1	7	250 kbps	1-7, 1-2-7
flow 2	3	6	500 kbps	3-2-6, 3-4-5-6

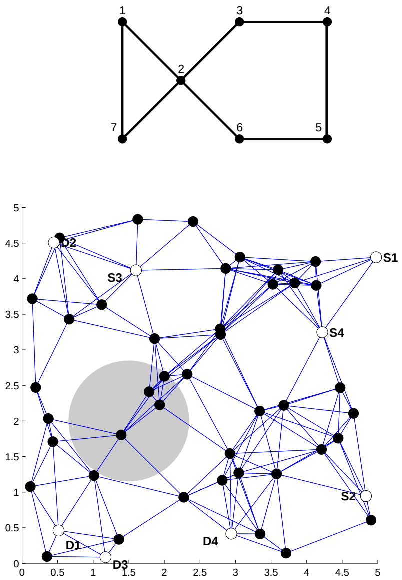


Fig. 1. Network topology

is no greater than $\Theta^* + \alpha\bar{W}$, where Θ^* is the minimal value of Problem (B), and $\bar{W}$ is the constant defined in Theorem 4.

The proof is in Appendix C. According to the discussion in Section III-A, by taking $\beta = \frac{1}{2} - \eta$, we then obtain a distributed solution whose power-efficiency ratio is upper-bounded by $(2+\varepsilon)$, where $\varepsilon = 1/(\frac{1}{2} - \eta) - 2$.

VI. Numerical Results

We first simulate a simple 7-node network (see top figure in Figure 1). The rate-power function is of the following form:

$$R_e = W \log_2 \left[1 + \frac{\sigma_e p_e}{N_0 W}\right],$$

where $W = 1.0$ MHz is the available bandwidth, $\sigma_e = 1.6 \times 10^{-13}$ is the channel gain of link e, $N_0 = 1.6 \times 10^{-18}$ mW/Hz is the noise spectral density, p_e is the transmission power, and R_e is the resultant instantaneous data rate of link e. The power-rate function $h_e(\cdot)$ is then given by (7). This network supports two flows, as shown in Table I.

The node-exclusive interference model is considered, and we use $\beta = (0.5 - 10^{-4})$ in Problem (B). To show that our proposed solution can adapt to variations in the input parameters, we apply the following changes in the system setting. At iteration $t = 4000$, the channel gain $\sigma_{(1,7)}$ of the direct link between node 1 and node 7 is decreased from 1.6×10^{-13} to 0.4×10^{-13}. At iteration $t = 8000$s, the data rate of flow 2 (from node 3 to node 6) is reduced from 500 kbps to 250 kbps.

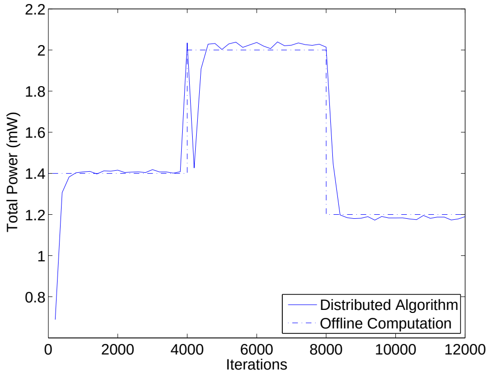


Fig. 2. Power consumption from distributed algorithm and offline computation. $\alpha = 0.1$.

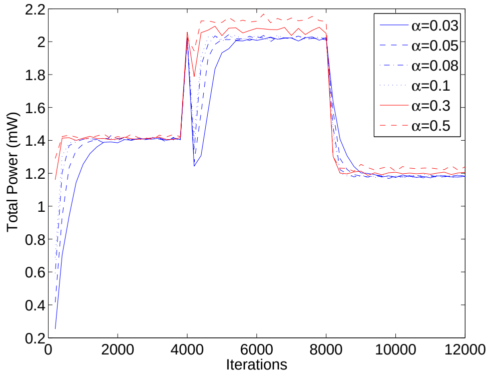


Fig. 3. Power consumption from distributed algorithm as the value of α varies from 0.03 to 0.5.

For each setting, offline computation is carried out to find the minimum value Θ^* of Problem (B), which is given by the dashed line in Figure 2. The power consumption from the proposed distributed algorithm is shown as the solid line in the same figure, where we have chosen $\alpha = 0.1$. This simulation result shows that our proposed solution is capable of achieving the near-optimal power-consumption Θ^* in a distributed manner, and automatically tracking the near-optimal operating point once the system parameters change.

To illustrate the tradeoff between convergence speed and power-efficiency as the stepsize α varies, we also simulate the distributed algorithm for different values of α. As we can see from Fig. 3, when α increases from 0.03 to 0.5, the algorithm converges faster, although the power-consumption increases slightly when α is large. Nonetheless, for the range of α that we simulated, the power-consumption levels are all close to the value Θ^*.

We also plot the evolution of dual variables at selected links when $\alpha = 0.03$ and $\alpha = 0.1$ (see Fig. 4 and Fig. 5, respectively). We can observe a similar tradeoff between convergence speed and accuracy for the dual variables.

As we discussed in Section V-A, in general the primal variables $R_e^d(m)$, $t_e(m)$ and $f_e^d(m)$ do not converge. In Fig. 6-Fig. 8, we plot their time-averaged values (over a moving window of 200 iterations). We can infer the change in the routing and scheduling decisions from Fig. 8:

- In the initial state, flow 1 concentrates on the minimum energy path, namely, link $(1,7)$ (see the solid line in

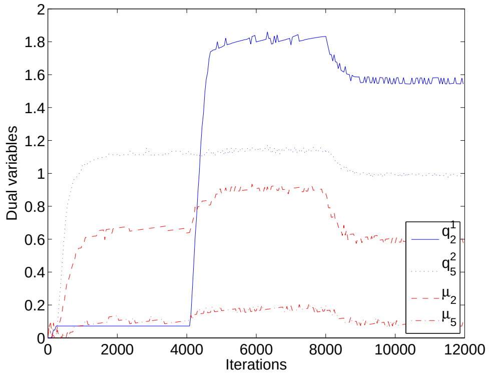


Fig. 4. Dual variables produced by the distributed algorithm. $\alpha = 0.03$.

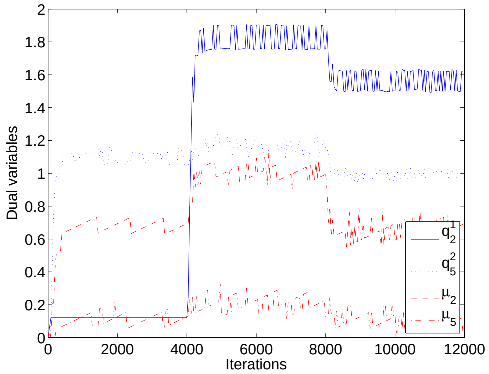


Fig. 5. Dual variables produced by the distributed algorithm. $\alpha = 0.1$.

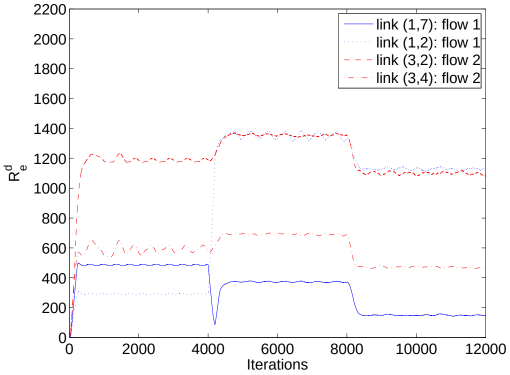


Fig. 6. Instantaneous data rates (R_e^d) for different flows on four links. $\alpha = 0.1$

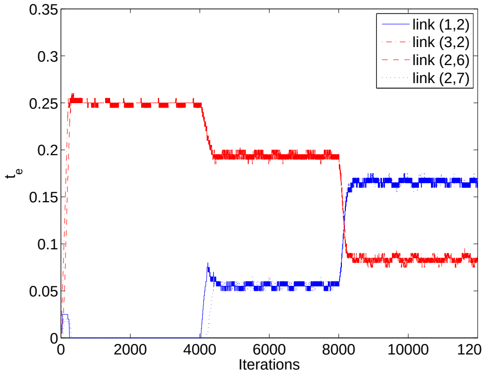


Fig. 7. Activation time (t_e) for four links. $\alpha = 0.1$

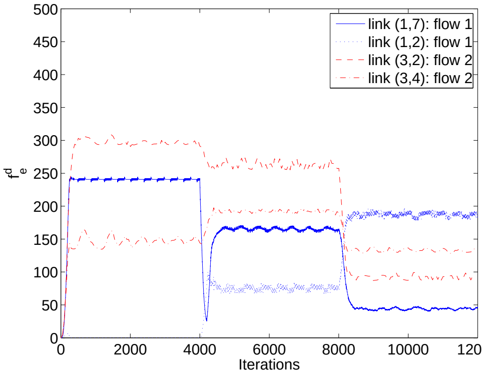


Fig. 8. Average data rates (f_e^d) for different flows on four links, $\alpha = 0.1$

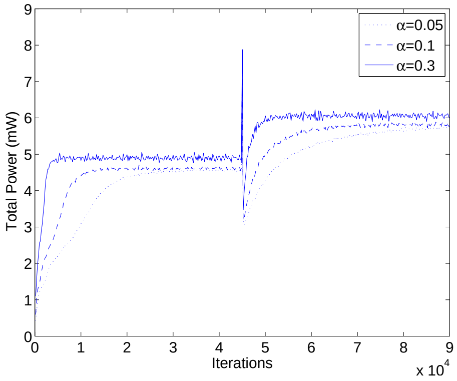


Fig. 9. Power consumption from distributed algorithm for a bigger network. The value of α varies from 0.05 to 0.3.

Fig 8).

- At iteration $t = 4000$, the channel gain $\sigma_{(1,7)}$ reduces by 75%, and part of flow 1 is shifted to path $1-2-7$ (the dotted line). Since the scheduling capacity of node 2 is saturated, a larger percentage of flow 2 is then routed through path $3-4-5-6$ in the optimal solution (the dash-dotted line).
- At iteration $t = 8000$, the traffic that the network has to support between node 3 and node 6 reduces (flow 2 is reduced to 250 kbps). As a consequence, part of the scheduling capacity of node 2 is freed, and more of flow 1 takes path $1-2-7$ to reduce the overall power consumption (the dotted line).

We then simulate the distributed algorithm on a bigger network (see the bottom topology in Fig. 1). The channel gains and rate-power functions are chosen as before. There are four flows from source S_i to destination D_i, $i = 1, 2, 3, 4$, respectively. The data rate of each flow is 250kbps. In the middle of the simulation, the noise density in the shaded area is increased by four times. In Fig. 9, we plotted the power-consumption for three values of α. Again, the algorithm converges close to the optimal operating point, and the power-efficiency improves when α is smaller.

Finally, in Fig. 10 we plot the sum of the virtual queues $\nu_e(m)$ (for the bigger network) due to the maximal matching scheduling component. We can see that the virtual queues are

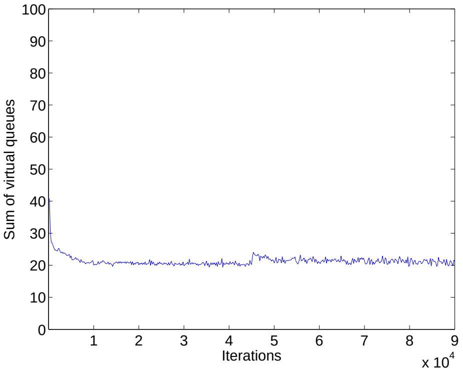


Fig. 10. The sum of the virtual queues over all links of the bigger network.

in fact very small (the sum is around 30) even though $\beta = 1/2 - 10^{-4}$ is close to $1/2$. The average scheduling delay (not plotted) is less than 5 time-slots over all links, with the maximum scheduling delay less than 25 time-slots during the entire simulation. Hence, we observe that the scheduling delay due to the maximal matching scheduling component tends to be much smaller than the bound given in Lemma 5.

VII. Conclusion

In this paper, we propose a joint power control, link scheduling, and routing algorithm to minimize the power consumption in multi-hop wireless networks. The known cross-layer solution to this problem is centralized, and with high computational complexity. In contrast, our algorithm is distributed, and with low computational complexity. We establish the power-efficiency ratio of our solution, and show that the performance bound of our solution, achieved in a distributed manner, is provably tighter than a centralized solution in the literature.

As in related works on cross-layer control and optimization of wireless networks [12], our solution borrows extensively the techniques from convex optimization and duality theory. However, we often observe that straightforward applications of optimization theory may not produce a control protocol that is directly usable in real systems. For example, for the problem that we studied in this paper, duality theory leads to a solution in (18)-(20) where the interference constraints could in fact be violated. Hence, additional modification of the solution is needed. One of the main contributions of the paper is to design an easy-to-implement scheduling component that accounts for the interference constraints, and to carefully quantify the stability and power-efficiency of the resulting protocol. Our simulation results verify that the proposed distributed solution can compute and track the near-optimal operating point whenever the system parameters change. For future work, we plan to extend the results to more general interference models, e.g., the bi-directional equal-power model used in [23]–[26].

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments on our paper, and Mr. Joohwan Kim for running some of the simulations reported in this paper.

Appendix

A. Proof of Theorem 4

The proof technique here is similar to [7]. First, note that the subgradient of the dual objective function is bounded because $0 \leq t_e(m) \leq 1$ and $f_e^d(m) \leq a_e t_e(m) \leq a_e$. Given any $\varepsilon > 0$, define the set $\Phi_{\varepsilon/2} = \{(\vec{\mu}, \vec{q}) | d((\vec{\mu}, \vec{q}), \Phi) < \varepsilon/2\}$. Using the results of [29, Lemma 8.2.1 and Proposition 8.2.2, p471-473], if the stepsize α is sufficiently small, whenever the vector of dual variables $(\vec{\mu}(m), \vec{q}(m))$ is outside the set $\Phi_{\varepsilon/2}$, it will move closer to Φ, and hence will eventually enter the set $\Phi_{\varepsilon/2}$. Further, once the vector of dual variables is in the set $\Phi_{\varepsilon/2}$, at the next iteration it can move away from $\Phi_{\varepsilon/2}$ by at most a distance proportional to α. Hence, by (possibly) further decreasing the stepsize α, we can ensure that the dual variables will never leave the set Φ_{ε}. This proves the first part of the Theorem. Note that it implies that the sequence $\{(\vec{\mu}(m), \vec{q}(m))\}_m$ is bounded.

To show (31), we consider the following Lyapunov function:

$$V(\vec{\mu}(m), \vec{q}(m)) = \frac{1}{2} \sum_{v \in V} \sum_{d \in D} [q_v^d(m)]^2 + \frac{1}{2} \sum_{v \in V} \mu_v^2(m).$$

The subgradient of $D(\vec{\mu}, \vec{q})$ at time-slot m can be written as

$$\begin{aligned} \Delta\mu_v(m) &= \sum_{e \in N(v)} t_e(m) - \beta, \\ \Delta q_v^d(m) &= \sum_{e \in N_i(v)} f_e^d(m) + T_{vd} - \sum_{e \in N_o(v)} f_e^d(m). \end{aligned}$$

Using the above notation along with (19) and (20), the one-step drift of the Lyapunov function can be calculated as follows:

$$\begin{aligned} & V(\vec{\mu}(m+1), \vec{q}(m+1)) - V(\vec{\mu}(m), \vec{q}(m)) \\ \leq\ & \frac{1}{2} \sum_{v \in V} \left\{ [\mu_v(m) + \alpha \Delta\mu_v(m)]^2 - \mu_v^2(m) \right\} \\ & + \frac{1}{2} \sum_{v \in V} \sum_{d \in D} \left\{ \left[q_v^d(m) + \alpha \Delta q_v^d(m)\right]^2 - \left[q_v^d(m)\right]^2 \right\} \\ \leq\ & \alpha \sum_{v \in V} \mu_v(m) \Delta\mu_v(m) + \alpha \sum_{v \in V} \sum_{d \in D} q_v^d(m) \Delta q_v^d(m) \\ & \quad + \frac{\alpha^2}{2} W(m), \end{aligned} \tag{32}$$

where

$$W(m) = \sum_{v \in V} (\Delta\mu_v(m))^2 + \sum_{v \in V} \sum_{d \in D} (\Delta q_v^d(m))^2.$$

Since $t_e(m)$ is bounded by 1, and $\sum_{d \in D} f_e^d(m)$ is bounded by a_e, we can bound $W(m)$ by

$$\begin{aligned} & W(m) \\ \leq\ & \left(\sum_{v \in V} |N(v)| \right)^2 + |V| + 2 \left(\sum_{v \in V} \sum_{e \in N(v)} \sum_{d \in D} f_e^d(m) \right)^2 \\ & \quad + 2 \sum_{v \in V} \sum_{d \in D} T_{vd}^2 \\ \leq\ & 4|E|^2 + |V| + 8 \left(\sum_{e \in E} a_e \right)^2 + 2 \sum_{v \in V} \sum_{d \in D} T_{vd}^2 \triangleq \bar{W}. \end{aligned}$$

Adding $\alpha \sum_{e\in E} \Theta_e(\vec{f}_e(m), t_e(m))$ to both sides of (32), we have,

$$\begin{aligned}
& V(\vec{\mu}(m+1), \vec{q}(m+1)) - V(\vec{\mu}(m), \vec{q}(m)) \\
& +\alpha \sum_{e\in E} \Theta_e(\vec{f}_e(m), t_e(m)) \\
\leq\ & \alpha \left[\sum_{e\in E} \Theta_e(\vec{f}_e(m), t_e(m)) + \sum_{v\in V} \mu_v(m)\Delta\mu_v(m) \right. \\
& \left. + \sum_{v\in V}\sum_{d\in D} q_v^d(m)\Delta q_v^d(m) \right] + \alpha^2 \bar{W}/2 \\
=\ & \alpha D(\vec{\mu}(m), \vec{q}(m)) + \alpha^2 \bar{W}/2 \\
& \text{(from (16) and (18))} \\
\leq\ & \alpha\Theta^* + \alpha^2 \bar{W}/2,
\end{aligned}$$

where in the last step we have used Theorem 3.

Summing the above inequality over $m = 0, 1, 2, \ldots, M$, and dividing both sides by M, we have

$$\begin{aligned}
& \frac{V(\vec{\mu}(M+1), \vec{q}(M+1)) - V(\vec{\mu}(0), \vec{q}(0))}{M} \\
& + \frac{\alpha}{M} \sum_{m=1}^{M} \sum_{e\in E} \Theta_e(\vec{f}_e(m), t_e(m)) \\
\leq\ & \alpha\Theta^* + \alpha^2 \bar{W}/2. \qquad (33)
\end{aligned}$$

Let

$$M_1(\alpha) = \frac{2V(\vec{\mu}(0), \vec{q}(0))}{\alpha^2 \bar{W}}.$$

Then for all $M \geq M_1(\alpha)$, we will have

$$\frac{1}{M} \sum_{m=1}^{M} \sum_{e\in E} \Theta_e(\vec{f}_e(m), t_e(m)) \leq \Theta^* + \alpha\bar{W}.$$

The result of Theorem 4 then follows.

B. Proof of Lemma 5

We first show that $\nu_e(m)$ in (28) is bounded for all e and m. From Theorem 4, we know that $\mu_v(m)$ is bounded for all nodes v and all time-slots m. Let this bound be Q_1. Using (19), we then have, for any τ_1,

$$\begin{aligned}
\frac{Q_1}{\alpha} &\geq \frac{\mu_v(m+\tau_1) - \mu_v(m)}{\alpha} \\
&\geq \sum_{s=m}^{m+\tau_1-1} \left[\sum_{e\in N(v)} t_e(s) - \beta \right].
\end{aligned}$$

Hence,

$$\frac{1}{\tau_1} \sum_{s=m}^{m+\tau_1-1} \sum_{e\in N(v)} t_e(s) \leq \beta + \frac{Q_1}{\alpha\tau_1}.$$

Let $\epsilon = \frac{1}{2} - \beta$. Since $\beta < 1/2$, by choosing $\tau_1 \geq \frac{2Q_1}{\alpha\epsilon}$, we can have

$$\frac{1}{\tau_1} \sum_{s=m}^{m+\tau_1-1} \sum_{e\in N(v)} t_e(s) \leq \frac{1}{2}(1-\epsilon) \qquad (34)$$

for all time-slot m.

Next, define the Lyapunov function

$$U(\vec{\nu}(m)) = \frac{1}{2} \sum_{v\in V} \left(\sum_{e\in N(v)} \nu_e(m) \right)^2.$$

Note that this Lyapunov function is standard in proving the stability of maximal-matching [10]. Let $\Delta\nu_e(m) = \nu_e(m+1) - \nu_e(m)$. According to Equation (28) that governs the evolution of $\vec{\nu}$, we can compute the τ_1-step drift of $U(\cdot)$ as

$$\begin{aligned}
& U(\vec{\nu}(m+\tau_1)) - U(\vec{\nu}(m)) \\
\leq\ & \sum_{e\in E} \nu_e(m) \sum_{s=m}^{m+\tau_1-1} \left[\sum_{l\in N(x(e))} t_l(s) + \sum_{l\in N(r(e))} t_l(s) \right. \\
& \left. - \sum_{l\in N(x(e))} \mathbf{1}_{\{l\in\mathcal{M}(s)\}} - \sum_{l\in N(r(e))} \mathbf{1}_{\{l\in\mathcal{M}(s)\}} \right] + M_1(m),
\end{aligned}$$

where

$$\begin{aligned}
M_1(m) &\leq \frac{1}{2} \sum_{v\in V} \left[\sum_{k=0}^{\tau_1-1} \sum_{e\in N(v)} |\Delta\nu_e(m+k)| \right]^2 \\
&\leq \frac{{\tau_1}^2}{2} \sum_{v\in V} |N(v)|^2 \triangleq \bar{M}_1.
\end{aligned}$$

Note that for any link e with $\nu_e(m) \geq \tau_1 + 1$, we must have $\nu_e(s) \geq 1$ for $s = m, m+1, ..., m+\tau_1$. Hence, according to the definition of maximal-matching,

$$\sum_{l\in N(x(e))} \mathbf{1}_{\{l\in\mathcal{M}(s)\}} + \sum_{l\in N(r(e))} \mathbf{1}_{\{l\in\mathcal{M}(s)\}} \geq 1, \qquad (35)$$

for $s = m, m+1, ...m+\tau_1$. Therefore, by letting $M_2(m) = M_1(m) + \tau_1 \sum_{e\in E} \nu_e(m) \mathbf{1}_{\{\nu_e(m)\leq\tau_1\}}$, we have

$$\begin{aligned}
& U(\vec{\nu}(m+\tau_1)) - U(\vec{\nu}(m)) \\
\leq\ & \sum_{e\in E} \nu_e(m) \sum_{s=m}^{m+\tau_1-1} \left[\sum_{l\in N(x(e))} t_l(s) + \sum_{l\in N(r(e))} t_l(s) - 1 \right] \\
& +M_2(m), \\
\leq\ & -\epsilon\tau_1 \sum_{e\in E} \nu_e(m) + M_2(m),
\end{aligned}$$

where in the last step we have used (34). Note that $M_2(m)$ is bounded by

$$M_2(m) \leq \bar{M}_1 + {\tau_1}^2 |E| \leq {\tau_1}^2 \sum_{v\in V} |N(v)|^2 \triangleq \bar{M}_2.$$

In other words, whenever $\sum_{e\in E} \nu_e(m)$ is greater than $2\bar{M}_2/(\epsilon\tau_1)$, the value of $U(\vec{\nu}(m))$ must decrease in τ_1 steps. This implies that $U(\vec{\nu}(m))$ must be bounded for all m, and hence all $\nu_e(m)$ must also be bounded for all links e and all time-slots m. Specifically, we have,

$$\sum_{l\in N(x(e))} \nu_l(m) + \sum_{l\in N(r(e))} \nu_l(m) \leq \frac{4\bar{M}_2}{\epsilon\tau_1} \triangleq \bar{Q}_2$$

for all e and m.

Finally, let K be the smallest integer that is greater than or equal to $\bar{Q}_2/(\epsilon\tau_1)$. Suppose $\delta_e(m) - m > K\tau_1$ for some e and m. We must then have

$$\nu_e(s) \geq 1, \text{ for } s = m, m+1, ..., m+K\tau_1. \tag{36}$$

According to the definition of maximal-matching, it implies that (35) holds for all $m \leq s \leq m + K\tau_1$. Using (34) again, we can show that, because $\epsilon K\tau_1 \geq \bar{Q}_2$, in $K\tau_1$ time-slots the virtual queues at link e and at all links next to e will be empty. This contradicts to (36). Hence, the delay $\delta_e(m) - m$ must be bounded by $K\tau_1$ for all e and m. Letting $\bar{T}_0 = K\tau_1$, the result of the lemma then follows with

$$\bar{T}_0 \;=\; K\tau_1 \leq \left(\frac{\bar{Q}_2}{\epsilon\tau_1} + 1\right)\tau_1 \leq \frac{16Q_1}{\alpha\epsilon^3}\sum_{v\in V}|N(v)|^2.$$

C. Proof of Theorem 6

We first show that the real queues are bounded. By Lemma 5, the delay in activating the links is bounded by a number $\bar{T}_0$. Consider Equation (27). Recall that $Q_v^d(m)/\alpha$ provides an upper bound on the real queue maintained at node v for packets destined to node d. Hence, it suffices to show that $Q_v^d(m)$ is bounded for all nodes v, destinations d and time-slots m. For any time-slot m, assuming $Q_v^d(0) = 0$ for all v and d, we then have

$$\frac{Q_v^d(m)}{\alpha} \leq \sup_{0\leq\tau_2\leq m}\sum_{s=m-\tau_2}^{m-1}\left[T_v^d + \sum_{e\in N_i(v)} f_e^d(\delta_e^{-1}(s)) - \sum_{e\in N_o(v)} f_e^d(\delta_e^{-1}(s))\right]. \tag{37}$$

Since $\delta_e(m) - m \leq \bar{T}_0$, we have, for all $0 \leq \tau_2 \leq m$,

$$\sum_{s=m-\tau_2}^{m-1}\sum_{e\in N_i(v)} f_e^d(\delta_e^{-1}(s)) \leq \sum_{s=m-\tau_2-\bar{T}_0}^{m-1}\sum_{e\in N_i(v)} f_e^d(s),$$

where we have adopted the convention that $f_e^d(m) = 0$ for all $m < 0$. Similarly, we have, for all $0 \leq \tau_2 \leq m$,

$$\begin{aligned}\sum_{s=m-\tau_2}^{m-1}\sum_{e\in N_o(v)} f_e^d(\delta_e^{-1}(s)) &\geq \sum_{s=m-\tau_2}^{m-\bar{T}_0-1}\sum_{e\in N_o(v)} f_e^d(s)\\ &\geq \sum_{s=m-\tau_2-\bar{T}_0}^{m-1}\sum_{e\in N_o(v)} f_e^d(s) - M_3(m),\end{aligned}$$

where $M_3(m) \leq 2\bar{T}_0\sum_{e\in N_o(v)} a_e \triangleq \bar{M}_3$. We then have, for all $0 \leq \tau_2 \leq m$,

$$\begin{aligned}&\sum_{s=m-\tau_2}^{m-1}\left[T_v^d + \sum_{e\in N_i(v)} f_e^d(\delta_e^{-1}(s)) - \sum_{e\in N_o(v)} f_e^d(\delta_e^{-1}(s))\right]\\ &\leq \sum_{s=m-\tau_2-\bar{T}_0}^{m-1}\left[T_v^d + \sum_{e\in N_i(v)} f_e^d(s) - \sum_{e\in N_o(v)} f_e^d(s)\right] + \bar{M}_3 + T_v^d\bar{T}_0.\end{aligned}$$

From the proof of Theorem 4, we know that $q_v^d(m)$ is bounded for all v, d and m. Let this bound by Q_3. Then, using (20), we must have, for any $\tau_3 \geq 0$,

$$\begin{aligned}\frac{Q_3}{\alpha} &\geq \frac{q_v^d(m+\tau_3) - q_v^d(m)}{\alpha}\\ &\geq \sum_{s=m}^{m+\tau_3-1}\left[T_v^d + \sum_{e\in N_i(v)} f_e^d(s) - \sum_{e\in N_o(v)} f_e^d(s)\right].\end{aligned}$$

Substituting into (37), we must have

$$\begin{aligned}\frac{Q_v^d(m)}{\alpha} &\leq \sup_{0\leq\tau_2\leq m}\sum_{s=m-\tau_2}^{m-1}\left[T_v^d + \sum_{e\in N_i(v)} f_e^d(\delta_e^{-1}(s)) - \sum_{e\in N_o(v)} f_e^d(\delta_e^{-1}(s))\right]\\ &\leq \sup_{0\leq\tau_2\leq m}\sum_{s=m-\tau_2-\bar{T}_0}^{m-1}\left[T_v^d + \sum_{e\in N_i(v)} f_e^d(s) - \sum_{e\in N_o(v)} f_e^d(s)\right] + \bar{M}_3 + T_v^d\bar{T}_0\\ &\leq \frac{Q_3}{\alpha} + \bar{M}_3 + T_v^d\bar{T}_0.\end{aligned}$$

The first part of the theorem then follows. Further, since link activations are delayed, the true energy consumption in any interval $[1, m]$ is bounded from above by

$$\sum_{\tau=1}^{m}\sum_{e\in E}\Theta_e(\vec{f}_e(\tau), t_e(\tau)).$$

Hence, the second part of Theorem 6 follows from the second part of Theorem 4.

References

[1] L. Lin, X. Lin, and N. B. Shroff, “Low-complexity and distributed energy minimization in multi-hop wireless networks,” in *IEEE INFOCOM’07*, Anchorage, Alaska, May 2007.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] T. P. Ruggaber and J. W. Talley, “Detection and control of combined sewer overflow events using embedded sensor network technology,” in *World Water Congress*, Anchorage, Alaska, May 2005.

[4] J. Luo and J.-P. Hubaux, “A survey of inter-vehicle communication,” EPFL, Lausanne, Switzerland, Tech. Rep., 2004.

[5] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Comput. Netw. ISDN Syst.*, vol. 47, no. 4, pp. 445–487, 2005.

[6] R. L. Cruz and A. V. Santhanam, “Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks,” in *Proceedings of IEEE INFOCOM*, San Francisco, April 2003.

[7] M. J. Neely, “Energy Optimal Control for Time Varying Wireless Networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 2915–2934, July 2006.

[8] R. Bhatia and M. Kodialam, “On power efficient communication over multi-hop wireless networks: Joint routing, scheduling and power control,” in *IEEE INFOCOM’04*, Hong Kong, Mar. 2004.

[9] M. J. Neely and R. Urgaonkar, “Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity,” *Ad Hoc Networks (Elsevier)*, vol. 7, no. 5, pp. 862–881, July 2009.

[10] X. Lin and N. B. Shroff, “ The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Wireless Networks,” in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[11] ——, “The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, 2006.

[12] X. Lin, N. B. Shroff, and R. Srikant, “A Tutorial on Cross-Layer Optimization in Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452– 1463, Aug. 2006.

[13] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas, “Routing for network capacity maximization in energy-constrained ad-hoc networks,” *IEEE INFOCOM’03*, 2003.

[14] Q. Li, J. A. Aslam, and D. Rus, “Online power-aware routing in wireless ad-hoc networks,” in *Proc. the Seventh Annual International Conference on Mobile Computing and Networking (ACM Mobicom’01)*, 2001, pp. 97–107.

[15] L. Lin, N. B. Shroff, and R. Srikant, “Energy-aware routing in sensor networks: A large systems approach,” in *the Third Annual Conference on Wireless On demand Network Systems and Services (WONS)*, Les Menuires, France, Jan. 2006.

[16] C.-K. Toh, “Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks,” *IEEE communications Magazine*, vol. 39, no. 6, pp. 138–147, June 2001.

[17] Y. Xue, Y. Cui, and K. Nahrstedt, “A utility-based distributed maximum lifetime routing algorithm for wireless networks,” in *QSHINE*, Los Alamitos, CA, USA, 2005.

[18] B. Hajek and G. Sasaki, “Link Scheduling in Polynomial Time,” *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, September 1988.

[19] S. Sarkar and L. Tassiulas, “End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks,” in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003.

[20] Y. Yi and S. Shakkottai, “Hop-by-hop Congestion Control over a Wireless Multi-hop Network,” in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[21] L. Lin, X. Lin, and N. B. Shroff, “Low-complexity and distributed energy minimization in multi-hop wireless networks,” Purdue University, Tech. Rep., 2006, available at http://min.ecn.purdue.edu/∼linx/papers.html.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[23] P. Chaporkar, K. Kar, and S. Sarkar, “Throughput Guarantees in Maximal Scheduling in Wireless Networks,” in *Proceedings of 43d Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2005.

[24] ——, “Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks,” in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

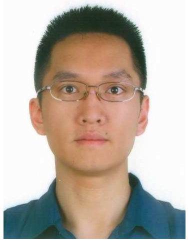
[25] X. Wu and R. Srikant, “Bounds on the Capacity Region of Multihop Wireless Networks Under Distributed Greedy Scheduling,” in *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006.

[26] X. Wu, R. Srikant, and J. R. Perkins, “Queue-Length Stability of Maximal Greedy Schedules in Wireless Network,” in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[27] A. Israeli and A. Itai, “A Fast and Simple Randomized Parallel Algorithm for Maximal Matching,” *Information Processing Letters*, vol. 22, no. 2, pp. 77–80, January 1986.

[28] A. Gupta, X. Lin, and R. Srikant, “Low-Complexity Distributed Scheduling Algorithms for Wireless Networks,” in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007.

[29] D. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Nashua, NH: Athena Scientific, 2003.



Longbi Lin received the B.S. (with honors) degree from Tsinghua University, Beijing, China, in 2000, and the M.S. and Ph.D. degrees from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2002 and 2006, respectively.

His research interests are resource allocation, optimization, routing, load balancing, network as a large system, energy-related issues and cross-layer design in wireless networks, and mobile ad hoc and sensor networks. He is currently working on modeling interest rate derivatives at Morgan Stanley & Co.

Xiaojun Lin (S’02 / M’05) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, Indiana, in 2000 and 2005, respectively. He is currently an Assistant Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin’s research interests are resource allocation, optimization, network pricing, routing, congestion control, network as a large system, cross-layer design in wireless networks, mobile ad hoc and sensor networks. He received the IEEE INFOCOM 2008 best paper award and 2005 best paper of the year award from *Journal of Communications and Networks*. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005. He received the NSF CAREER award in 2007. He was the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, and the TPC co-chair for ACM MobiHoc 2009.

Ness B. Shroff (S’91 / M’93 / SM’01 / F’07) received the Ph.D. degree from Columbia University, NY, in 1994.

He joined Purdue University, West Lafayette, IN, after receiving the Ph.D. degree. At Purdue, he became a Professor of the School of Electrical and Computer Engineering in 2003 and Director of the Center for Wireless Systems and Applications (CWSA) in 2004. In July 2007, he joined The Ohio State University, Columbus, as the Ohio Eminent Scholar of Networking and Communications and chaired Professor of ECE and CSE.

His research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, pricing, and security of these networks.

Dr. Shroff received the IEEE INFOCOM 2008 Best Paper Award, the IEEE INFOCOM 2006 Best Paper Award, the IEEE IWQoS 2006 Best Student Paper Award, the 2005 Best Paper of the Year Award for the Journal of Communications and Networking, the 2003 Best Paper of the Year Award for Computer Networks, and the NSF CAREER Award in 1996. His IEEE INFOCOM 2005 paper was also selected as one of two runner-up papers for the Best Paper Award.