Power Optimal Control in Multi-Hop Wireless Networks with Finite Buffers

Dongyue Xue, *Student Member*, *IEEE*, and Eylem Ekici, *Senior Member*, *IEEE*

***Abstract*—In this paper, we propose two cross-layer algorithms, namely, Power-optimal Scheduling Algorithm (PSA) and Throughput-optimal Scheduling Algorithm (TSA), to minimize energy consumption and to maximize throughput, respectively, in multi-hop wireless networks. Our algorithms guarantee a flow-based minimum data rate and jointly integrate congestion control, power allocation, routing and link rate scheduling. Different from traditional algorithms which assume infinite buffers, the proposed algorithms deterministically upper-bound the flow-based packet queue length and thus can be employed in multi-hop networks with finite buffers. In addition, the algorithms achieve a power expenditure/throughput "ϵ-close" to the optimal value, with a tradeoff of order $O(\frac{1}{\epsilon})$ in the buffer size. The average end-to-end delay upper-bound can also be derived from the finite buffer property. Finally, numerical results are presented to show the performance of the two algorithms with different system parameters.**

***Index Terms*—Flow control, power allocation, network scheduling, multi-hop wireless networks, finite buffer**

I. Introduction

With expanding wireless applications and increasing demand for wireless data rates, it is significant to develop power control algorithms that take maximum advantage of available capacity while satisfying certain Quality of Service (QoS) requirements such as minimum data rate and end-to-end delay constraints.

Recently, back-pressure algorithm [1] and its extensions have been widely employed in developing optimal scheduling in wireless networks. Throughput/utility-optimal routing and scheduling algorithms have been developed in [4]- [6], with suboptimal algorithms and distributed algorithms proposed in [7]- [12]. Optimal power allocation algorithm is further analyzed in [2], with additional congestion controllers considered in [3]. The above referenced works do not deal with delay-related issues in depth. Delay analysis of back-pressure-based algorithms and delay-related works can be found in [19]- [23]. However, these works do not provide explicit queue backlog (or buffer size) guarantees. The throughput-optimal algorithm in [18] guarantees finite buffer size but the link capacity is assumed to be fixed and power allocation is not considered.

Finite buffer property is an important factor for QoS sensitive wireless network applications. Not only are buffer sizes closely related to end-to-end delay on a per-flow basis, limiting buffer sizes is also essential to mitigate playout buffer overflow problems in multimedia applications. In addition, algorithms with finite buffer property can also find their application in resource-limited wireless networks such as wireless sensor networks.

In this paper, we develop two cross-layer scheduling algorithms that aim to optimize power allocation in multi-hop wireless networks with finite buffers. Specifically, we propose a Power-optimal Scheduling Algorithm (***PSA***) to minimize energy expenditure, and a Throughput-optimal Scheduling Algorithm (***TSA***) to maximize throughput, respectively. Since the objectives of energy consumption minimization and throughput optimization may cause unfairness to individual flows in the network, we place additional per-flow minimum data rate constraints. In addition, we impose maximum link energy consumption constraints in the ***PSA*** algorithm to limit energy expenditure. We introduce virtual queues to guarantee the data rate constraints and the energy consumption constraints. Both ***PSA*** and ***TSA*** are composed of a regulator, a congestion controller, a power allocator, and a link rate scheduler. The regulator regulates the virtual queue dynamics, the congestion controller is employed to admit packets from transport layer, while the power allocator determines the power allocation for links in the network, and the link rate scheduler schedules transmission rates for individual flows. Furthermore, we consider adaptive routing scenario, i.e., the routes of each flow are *not determined a priori*, which is more general than fixed-routing scenario. Note that the proposed ***PSA*** and ***TSA*** are centralized algorithms which are developed to serve as the benchmarks for real deployments. Specifically, we assume a centralized setting, such that power is allocated over all the links in the network simultaneously via a centralized component, e.g., access point or base station. The interference models employed in this work, such as the node-exclusive interference model, are generally used to handle cases involving hidden terminals in centralized algorithms. Thus, problems such as hidden terminals, which exist in a random access setting, do not exist in the context considered in this work.

To the best of our knowledge, our power allocation algorithms are the first of their kind to achieve an energy consumption/throughput performance *at least* ϵ-close to the optimal, with a tradeoff of in the buffer size for individual flows at nodes. The buffer size upper-bound is deterministic, which leads to bounded average end-to-end delay by Little's Law. In comparison, the buffer size is assumed infinitely large in the power allocation algorithm proposed in [3]. In [16] [17], which aim to achieve optimal throughput-utility, the

The authors are with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, Ohio 43210, USA (e-mail: xued, ekici@ece.osu.edu).

This work was supported in part by NSF grant CCF-0914912. Preliminary results [30] of this work were presented in part in IEEE ICC 2011, Kyoto, Japan, June 2011.

ingress buffer size (the buffer size at source nodes) is assumed infinitely large although the internal buffer sizes are finite. With the assumption of infinite buffer sizes, the algorithms proposed in the above related works may have to experience additional packet loss in finite-buffer networks such as sensor networks where resources are limited and packet queue lengths are constrained to finite (constant or varying) buffer sizes. On the contrary, during our algorithm design, we take the buffer size as a specific requirement which ensures feasible implementations of the algorithms in finite-buffer multi-hop wireless networks.

The rest of the paper is organized as follows. Section II provides the network model for multi-hop wireless networks. In Section III and IV, we propose the optimal power allocation and scheduling algorithms ***PSA*** and ***TSA***, respectively, and analyze their performances. The numerical results of the proposed algorithm are provided in Section V. Finally, we conclude our work in Section VI.

II. Network Model

A. Network Elements

We consider a multi-hop wireless network consisting of N nodes and K flows. In the network topology, we denote by $\mathcal{F}$ the set of flows, $\mathcal{N}$ the set of nodes, $\mathcal{L}$ the set of directed links in the network, and $(m,n) \in \mathcal{L}$ a link from node m to node n. In the network, flows follow routes that are determined adaptively. Additionally, we denote the source node and the destination node of a flow $c \in \mathcal{F}$ as $b(c)$ and $d(c)$, respectively.

A generic cross-layer power control algorithm consists of a congestion controller, a power allocator and link rate scheduler across transport layer and network layer. Packets are generated by specific applications at the transport layer, admitted to the network layer at the source node (by a congestion controller)[1] and transferred from source node to destination node in the network layer (by a power allocator and link rate scheduler). We consider a centralized schedule-based channel, where the channel access is characterized by a power allocation vector and determined by the power allocator.

We consider a time-slotted system with integer-valued time $t \in \{0,1,2,...\}$.[2] The centralized power scheduling over the channel is characterized by $\mathbf{P}(t)$, where $\mathbf{P}(t) = (P_{mn}(t))_{(m,n)\in\mathcal{L}}$ represents the power allocation vector for time slot t according to a generic power allocator. We constrain $\mathbf{P}(t)$ to be in Π, i.e., $\mathbf{P}(t) \in \Pi$, where Π is the compact and convex set of feasible power vectors. We also assume that $P_{mn}(t) \le P_M$, $\forall (m,n) \in \mathcal{L}$, $\forall \mathbf{P}(t) \in \Pi$, where P_M is the power upper-bound. In addition, we denote $\mu_{mn}(\mathbf{P}(t))$ as the link rate function for link (m,n) corresponding to the power assignment $\mathbf{P}(t)$, and denote $\mu(\mathbf{P}(t)) = (\mu_{mn}(\mathbf{P}(t)))_{(m,n)\in\mathcal{L}}$.

[1]We note that the backlogged source of a flow can be considered an application waiting for packet generation and admission (e.g., a variable rate multimedia encoder). The term "congestion controller" is different than in various TCP versions and corresponds to the packet admission rate to the network as in [3].

[2]Note also that a number of time synchronization methods have been proposed in the literature, e.g., [27] [28] [29], which can be utilized to ensure synchronized operation in the time-slotted system we use in our work.

A wide range of underlying interference models can be characterized by the link rate functions, where the interference caused by simultaneous relaying or transmission in shared wireless channel environment is addressed. In the following, we take the node-exclusive interference model and SNIR model for instance.

The node-exclusive interference model [13], which models wireless networks such as Bluetooth networks [14] and FH-CDMA networks [15], can be characterized by $\mu(\mathbf{P}(t))$ satisfying, $\forall (m_1,n_1) \neq (m_2,n_2) \in \mathcal{L}$ such that $\{m_1,n_1\} \cap \{m_2,n_2\} \neq \emptyset$,

$$\mu_{m_1n_1}(\mathbf{P}(t))\mu_{m_2n_2}(\mathbf{P}(t)) = 0, \ \forall \mathbf{P}(t). \tag{1}$$

In the node-exclusive interference model [13], a link transmission is only interfered by other simultaneous transmission within a one-hop distance, and equation (1) constrains that a node can involve in at most one communication in one time slot, i.e., simultaneous transmission or relaying within any one-hop distance is avoided.

The Signal to Noise and Interference Ratio (SNIR) model, where the link transmission rates are dependent on the interference heard from surrounding simultaneous transmission, can be characterized by $\mu(\mathbf{P}(t))$ satisfying:

$$\mu_{mn}(\mathbf{P}(t)) = f_{mn}\left(\frac{G_{mn}(t)P_{mn}(t)}{B_n + \sum_{(i,j)\neq(m,n)} G_{in}(t)P_{ij}(t)}\right), \tag{2}$$

where B_n is the base noise at node $n \in \mathcal{N}$, $G_{mn}(t)$ the propagation gain from the transmitter to the receiver of link $(m,n) \in \mathcal{L}$ in time slot t, and $f_{mn}(\cdot)$ the function of SNIR characterizing the link rates associated with the underlying interference model. We assume the propagation gain process $(G_{mn}(t))$ is ergodic and takes values over a finite state space.

For convenience of analysis, we assume that the link rate functions are upper semi-continuous, and define:

$$\begin{aligned} l_n &\triangleq \max_{\mathbf{P}\in\Pi} \sum_{j:(j,n)\in\mathcal{L}} \mu_{jn}(\mathbf{P}), \\ f_M &\triangleq \max_{n\in\mathcal{N}} l_n, \\ l_M &\triangleq \max_{n\in\mathcal{N}} \max_{\mathbf{P}\in\Pi} \sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}(\mathbf{P}), \end{aligned} \tag{3}$$

i.e., l_M and f_M are the maximum departure rate from a node and the maximum *endogenous* arrival rate into a node, respectively.

For a feasible link rate scheduler in time slot t, we let the scheduling parameter $\mu^c_{mn}(t)$ be the link rate assignment for flow c for link (m,n). Thus, given $\mathbf{P}(t)$, we must have $\sum_{c\in\mathcal{F}} \mu^c_{mn}(t) \le \mu_{mn}(\mathbf{P}(t))$, $\forall (m,n) \in \mathcal{L}$.

We assume that the source node for flow c is always backlogged at the transport layer. For a congestion controller, let $\mu^c_{s(c)b(c)}(t)$ be the admitted rate of flow c from the transport layer of flow to the source node, where we can regard $s(c)$ as the source at the transport layer of flow c. It is clear that in any time slot t, $\mu^c_{s(c)n}(t) = 0 \ \forall n \neq b(c)$. We also assume that $\mu^c_{s(c)b(c)}(t)$ is upper-bounded by a constant $\mu_M > 0$:

$$\mu^c_{s(c)b(c)}(t) \le \mu_M, \ \forall c \in \mathcal{F}, \forall t, \tag{4}$$

i.e., at most μ_M packets can be admitted into a source node in any time slot. To simplify the analysis, we prevent looping back to the source, i.e., we impose the following constraints

$$\sum_{m\in\mathcal{N}}(\mu^c_{mb(c)}(t))=0\ \forall c\in\mathcal{F},\forall t. \tag{5}$$

In addition, we assume that the network requires each flow c should transmit at a minimum data rate of a_c packets per time slot.

B. Network Constraints and Approaches

Network stability and optimality are two necessary goals for the algorithm designs. We first introduce the notion of network stability in this subsection and note that power optimality and throughput optimality will be defined in Section III and Section IV where we propose ***PSA*** and ***TSA***, respectively. A given power control algorithm is said to stabilize the network if it stabilizes all actual packet queues. Hence, to represent network stability, we begin with a definition of queue stability with respect to a generic queue backlog $A(t)$. The queue is *stable* if

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{A(\tau)\}<\infty.$$

In addition, we present the following lemma on the stability of queues:

Lemma 1: If a queue $A(t)$ is stable with some generic service process $\mu(t)$ and some generic arrival process $R(t)$ such that the queue dynamics is of the form $A(t+1)=[A(t)-\mu(t)]^+ + R(t)$, where we define the operator $[x]^+$ as $[x]^+=\max\{x,0\}$, then:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu(\tau)\}\geq\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{R(\tau)\} \tag{6}$$

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu(\tau)\}\geq\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{R(\tau)\} \tag{7}$$

Proof: From the queue dynamics, we have:

$$A(t)\geq A(0)-\sum_{\tau=0}^{t-1}\mu(\tau)+\sum_{\tau=0}^{t-1}R(\tau),$$

which yields:

$$\frac{\mathbb{E}\{A(t)\}}{t}+\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu(\tau)\}\geq\frac{\mathbb{E}\{A(0)\}}{t}+\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{R(\tau)\}, \tag{8}$$

$$\frac{\mathbb{E}\{A(t)\}}{t}-\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{R(\tau)\}\geq\frac{\mathbb{E}\{A(0)\}}{t}-\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu(\tau)\}. \tag{9}$$

By taking the limsup of t on both sides of (8) and the fact that $\limsup_{t\to\infty}\frac{\mathbb{E}\{A(t)\}}{t}=0$ [3], we can prove (6). Similarly, we can prove (7) by taking liminf of t on both sides of (9). ■

Now, let $U^c_n(t)$ be the actual queue backlog of flow c packets at node n. Then, the network is *stable* if queues $U^c_n(t)$ are stable, $\forall n\in\mathcal{N}$, $\forall c\in\mathcal{F}$.

For convenience of analysis, we define $\mathcal{L}^c\triangleq\mathcal{L}\cup\{(s(c),b(c))\}$, where the pair $(s(c),b(c))$ can be considered as a virtual link from transport layer to the source node. We now model queue dynamics and network constraints in the multi-hop network. For flow c, if $n=d(c)$, i.e., n is the destination node of flow c, then we have $U^c_n(t)=0\ \forall t$; Otherwise, the queue dynamics are:

$$\begin{aligned}U^c_n(t+1)\leq&[U^c_n(t)-\sum_{i:(n,i)\in\mathcal{L}}\mu^c_{ni}(t)]^+\\&+\sum_{j:(j,n)\in\mathcal{L}^c}\mu^c_{jn}(t),\ \text{if } n\in\mathcal{N}\backslash d(c).\end{aligned} \tag{10}$$

Note that in (10), we ensure that the number of packets transmitted for flow c from node n does not exceed its corresponding queue backlog, since a feasible scheduling algorithm may be independent of the information on queue backlogs. The terms $\sum_{i:(n,i)\in\mathcal{L}}\mu^c_{ni}(t)$ and $\sum_{j:(j,n)\in\mathcal{L}^c}\mu^c_{jn}(t)$ represent, respectively, the scheduled departure rate from node n and the scheduled arrival rate into node n for flow c. Note that (10) is an inequality since the arrival rates from corresponding neighbor nodes may be less than $\sum_j\mu^c_{jn}(t)$ if some neighbor node does not have enough packets to transmit. From (4)(5), we also have

$$\sum_{j:(j,b(c))\in\mathcal{L}^c}\mu^c_{jb(c)}(t)\leq\mu_M, \tag{11}$$

if it is guaranteed that no packets will be looped back to the source.

We utilize several types of virtual queues in our two proposed algorithms introduced in Section III and Section IV. For each flow c, we construct a virtual queue $U^c_{s(c)}(t)$ at transport layer. We denote by $R_c(t)$ the virtual input rate to the queue at the end of time slot t, and denote by r_c the time-average of $R_c(t)$. We place an upper-bound μ_M on $R_c(t)$ and update the virtual queue as follows:

$$U^c_{s(c)}(t+1)=[U^c_{s(c)}(t)-\mu^c_{s(c)b(c)}(t)]^+ + R_c(t), \tag{12}$$

where we set $U^c_{s(c)}(0)=0$. Considering the admitted rate $\mu^c_{s(c)b(c)}(t)$ as the service rate, if the virtual queue $U^c_{s(c)}(t)$ is stable, then by Lemma 1 the time-average admitted rate μ_c of flow c satisfies:

$$\begin{aligned}\mu_c\triangleq&\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu^c_{s(c)b(c)}(\tau)\}\\\geq&r_c\triangleq\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{R_c(\tau)\}.\end{aligned} \tag{13}$$

To satisfy the minimum data rate constraints, we construct a virtual queue $Z_c(t)$ for flow c with queue dynamics:

$$Z_c(t+1)=[Z_c(t)-R_c(t)]^+ + a_c, \tag{14}$$

where we set $Z_c(0)=0$. If queue $Z_c(t)$ is stable, we have $r_c\geq a_c$. Additionally, if $U^c_{s(c)}(t)$ is stable, then according to (13), we have $\mu_c\geq a_c$, i.e., the minimum data rate for flow c is achieved: $\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\mu^c_{s(c)b(c)}(\tau)\}\geq a_c$.

The queue evolutions and relationships are illustrated in Figure 1. In the figure, for simplicity we do not represent the actual packet queue evolutions for nodes other than source nodes, since the dynamics for actual queues (10)

are dependent on specific network topologies. The decision variable $R_c(t)$, to be determined by the $R_c(t)$ regulator in the proposed algorithms, is both the input rate to the virtual queue $U^c_{s(c)}(t)$ and the service rate of the virtual queue $Z_c(t)$. The decision variable $\mu^c_{s(c)b(c)}(t)$, to be determined by the congestion controller in the proposed algorithms, is both the service rate of the virtual queue $Z_c(t)$ and the input rate to the actual queue at the source $U^c_{b(c)}(t)$. Thus, the decision of $R_c(t)$, together with that of $\mu^c_{s(c)b(c)}(t)$, regulates the queue evolutions and stability of the virtual queues ($Z_c(t)$ and $U^c_{s(c)}(t)$). In the proposed algorithms, after determining the power assignment $\mathbf{P}(t)$ (by a power allocator), we determine (through a link rate scheduler) the decision variables $(\mu^c_{mn}(t))$, which, together with the decision of $\mu^c_{s(c)b(c)}(t)$, regulate the queue evolution and stability of the actual packet queues. We also note that physical packets are only involved in the actual packet queues and the corresponding queue evolutions. Thus, in the following sections, the finite buffer properties refer only to the actual packet queues. By imposing a finite buffer size to actual queues, we can monitor the average flow-based end-to-end delay upper-bounds for the multi-hop network simply by employing Little's Law. In the proposed algorithms, the virtual queues $Z_c(t)$ will be used to satisfy the minimum data rate requirements; the virtual queues $U^c_{s(c)}(t)$ are employed as a weight for the differential backlogs across each link, in an attempt to guarantee the finite buffer property and the optimality; the virtual queues $X_{mn}(t)$ will be defined in Section IV and are specially employed for ***TSA*** algorithms to meet average link energy consumption constraints.

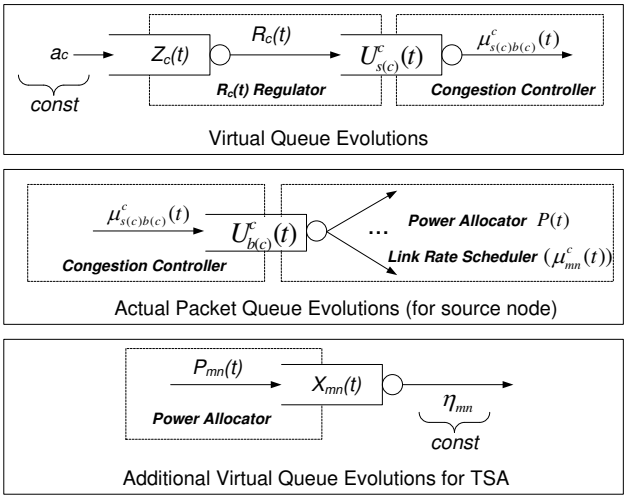

Fig. 1. Queue relationship diagram with decision variables $R_c(t)$, $\mu^c_{s(c)b(c)}(t)$, $(\mu^c_{mn}(t))$ and $\mathbf{P}(t)$, where virtual queues $X_{mn}(t)$ are specially employed for ***TSA***.

In Section III and Section IV, we introduce ***PSA*** and ***TSA***, respectively. We note that the objectives of ***PSA*** (minimizing energy expenditure) and ***TSA*** (maximizing throughput) cannot be achieved by a single solution, since energy expenditure and throughput tradeoff between each other cannot be optimized at the same time. While the two algorithms have different objectives and solutions, they share some common features:

- Both algorithms employ virtual queues $Z_c(t)$ to guarantee the minimum data rate constraints and $U^c_{s(c)}(t)$ as weight for the differential backlogs when schedule link rates.
- Finite buffer property is satisfied in both algorithms with an ϵ-versus-$\frac{1}{\epsilon}$ tradeoff. Specifically, ***PSA/TSA*** achieves an energy expenditure/throughput "ϵ-close" to the optimality with a tradeoff in the uniform finite buffer size of order $O(\frac{1}{\epsilon})$.

To assist the development of the following sections, we can define the capacity region Λ of the multi-hop network, similar as in [2] [3], as the closure of all stabilizable rate vectors considering all power control algorithms choosing $\mathbf{P}(t) \in \Pi$. Without loss of generality, we assume that the minimum data rate vector $(a_c)_{c\in\mathcal{F}}$ is within Λ.

III. Power-Optimal Scheduling Algorithm (PSA) for Multi-Hop Wireless Networks

In this section, we propose a power-optimal scheduling algorithm ***PSA*** for the introduced multi-hop wireless network so that ***PSA*** stabilizes the network and satisfies the minimum data rate constraint.

We let P^*_ϵ denote the minimum sum power for stabilizing rates $(a_c + \epsilon)$, where ϵ is a positive number which can be chosen arbitrarily small. According to [3] [5] we have $\lim_{\epsilon\to 0^+} P^*_\epsilon = P^*$, where P^* is the minimum sum power for stabilizing rates (a_c). Note that P^*_ϵ can be considered as the ϵ-optimal sum power for the multi-hop wireless network.

Given $\epsilon > 0$, ***PSA*** is designed to achieve a sum power arbitrarily close to P^*_ϵ, with a tradeoff with buffer size which will be later given in Theorem 1 and further explained in Remark 1.

Let $q_M \geq \max\{f_M, \mu_M\}$ be a control parameter standing for buffer size. The optimal algorithm ***PSA*** operates on a time-slot basis consisting of four parts: $R_c(t)$ regulator, a congestion controller, a power allocator, and a link rate scheduler.

1) $R_c(t)$ **Regulator**:

$$\min_{0\leq R_c(t)\leq \mu_M} R_c(t)\left(\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t) - Z_c(t)\right). \qquad (15)$$

The $R_c(t)$ regulator controls the virtual queue evolutions of $U^c_{s(c)}(t)$ and $Z_c(t)$. Since $R_c(t)$ is the arrival process for virtual queue $U^c_{s(c)}(t)$ and the service process for virtual queue $Z_c(t)$, we assign $R_c(t) = 0$ when $U^c_{s(c)}(t)$ is more congested than $Z_c(t)$ and assign $R_c(t) = \mu_M$ otherwise. Specifically, when $\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t) - Z_c(t) > 0$, $R_c(t)$ is set to zero; Otherwise, $R_c(t) = \mu_M$.

2) Congestion Controller:

$$\max_{0\leq \mu^c_{s(c)b(c)}(t)\leq \mu_M} \mu^c_{s(c)b(c)}(t)(q_M - \mu_M - U^c_{b(c)}(t)). \qquad (16)$$

The congestion controller aims to upper-bound by q_M the actual packet queue at source node. Specifically, when $q_M - \mu_M - U^c_{b(c)}(t) \leq 0$, $\mu^c_{s(c)b(c)}(t)$ is set to zero; Otherwise, $\mu^c_{s(c)b(c)}(t) = \mu_M$, where we recall that $\mu^c_{s(c)b(c)}(t)$ is the admitted number of packets from transport layer into the source node in time slot t.

3) Power Allocator:

$$\max_{\mathbf{P}(t)\in\Pi} \sum_{(m,n)\in\mathcal{L}} (\mu_{mn}(\mathbf{P}(t))w_{mn}(t) - VP_{mn}(t)), \quad (17)$$

where $V > 0$ is a control parameter and $w_{mn}(t)$ is defined as follows:

$$w_{mn}(t) \triangleq [\max_{c\in\mathcal{F}} \frac{U^c_{s(c)}(t)}{q_M}(U^c_m(t) - U^c_n(t) - l_n)]^+. \quad (18)$$

Note that when $w_{mn}(t) = 0$, without loss of optimality we allocate $\mathbf{P}(t)$ such that $\mu_{mn}(\mathbf{P}(t)) = 0$ to maximize (17). Different from the traditional back-pressure algorithm, for link $(m,n) \in \mathcal{L}$ and flow $c \in \mathcal{F}$, we add a weight of l_n to the differential backlog $(U^c_m(t) - U^c_n(t))$ which is further multiplied by $\frac{U^c_{s(c)}(t)}{q_M}$. This new type of back-pressure ensures the finite buffer property and the optimality, proven later in Proposition 1 and Theorem 1, respectively. In (17), we can consider $\mu_{mn}(\mathbf{P}(t))w_{mn}(t)$ as the reward and $P_{mn}(t)$ as the cost weighted by V, induced from link (m,n) by allocating $\mathbf{P}(t)$.

Under the node-exclusive interference model (1), the power allocator is equivalent to the well-known maximal weight matching optimization problem [1], which can be solved in a centralized way. Under the SNIR interference model (2), with a high SNIR assumption where $f_{mn}(\cdot)$ can be approximated as $\log(\cdot)$ in equation (2), the optimization can be converted to a nonlinear convex optimization via a log transform [24].

4) Link Rate Scheduler:

$$\mu^c_{mn}(t) = \begin{cases} \mu_{mn}(\mathbf{P}(t)), & \text{if } c = c^*_{mn}(t), \\ 0, & \text{otherwise}, \end{cases} \quad (19)$$

where $\mathbf{P}(t)$ is determined by the power allocator and $c^*_{mn}(t)$ is defined as follows:

$$c^*_{mn}(t) \triangleq \arg\max_{c\in\mathcal{F}} \frac{U^c_{s(c)}(t)}{q_M}(U^c_m(t) - U^c_n(t) - l_n).$$

For a link $(m,n) \in \mathcal{L}$, the link rate scheduler greedily schedules the available link resource $\mu_{mn}(\mathbf{P}(t))$ to the flow $c^*_{mn}(t)$ with the largest back-pressure (18).

Note that in the power allocator and the link rate scheduler, we constrain that there is no looping back to source.

To analyze the performance of the algorithm, we first introduce the following proposition of finite buffer property.

Proposition 1: Employing ***PSA***, if $q_M \geq \max\{f_M, \mu_M\}$, then each queue backlog in the network has a deterministic worst-case bound:

$$U^c_n(t) \leq q_M, \quad \forall t, \forall n \in \mathcal{N}, \forall c \in \mathcal{F}. \quad (20)$$

Proof: We prove Proposition 1 by induction on time slot. When $t = 0$, we have $U^c_n(0) = 0\ \forall n, c$. Now suppose in time slot t we have $U^c_n(t) \leq q_M\ \forall n, c$. In the induction step, for any given node n and flow c, we consider two cases as follows:
(1) We first consider the case when n is the source node, i.e., when $n = b(c)$. If $U^c_{b(c)}(t) \leq q_M - \mu_M$, then according to the queue dynamics (10)(11), $U^c_{b(c)}(t+1) \leq q_M$; Otherwise, $U^c_{b(c)}(t) > q_M - \mu_M$ and according to the congestion controller (16), we have $\mu^c_{s(c)b(c)}(t) = 0$, so $U^c_{b(c)}(t+1) \leq U^c_{b(c)}(t) \leq q_M$ by (5)(10).
(2) In the second case, n is not the source node of flow c. If $U^c_n(t) \leq q_M - l_n$, then $U^c_n(t) \leq q_M$ by (10); Otherwise, $U^c_n(t) > q_M - l_n$, and according to the link rate scheduler (19) we have $\mu^c_{mn}(t) = 0\ \forall m \in \mathcal{N}$, so $U^c_n(t+1) \leq U^c_n(t) \leq q_M$ by the queue dynamics (10).

Since the above analysis holds for any given n and c, the induction step holds, i.e., $U^c_n(t+1) \leq q_M\ \forall n, c$, which completes the proof. ■

Now we present our main results of the ***PSA*** algorithm in Theorem 1 which is further explained with Remark 1 and Remark 2.

Theorem 1: Given $\epsilon > 0$, if

$$q_M > \frac{Nl^2_M + (N-1)f^2_M + \mu^2_M + Nl_M f_M}{\epsilon} + \mu_M, \quad (21)$$

then ***PSA*** can achieve a time-average power

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \sum_{(m,n)\in\mathcal{L}} \mathbb{E}\{P_{mn}(\tau)\} \leq P^*_\epsilon + \frac{B_1}{V}, \quad (22)$$

where $B_1 \triangleq \frac{1}{2}\mu_M q_M NK + \frac{3q_M - 2\mu_M}{2q_M} K\mu^2_M + \frac{1}{2}\sum_{c\in\mathcal{F}} a^2_c$.

In addition, ***PSA*** ensures that the virtual queues have a time-averaged upper-bound:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \sum_{c\in\mathcal{F}} \mathbb{E}\{U^c_{s(c)}(\tau) + Z_c(\tau)\} \leq \frac{B_1 + VP^*_\epsilon}{\delta}, \quad (23)$$

where δ is a positive constant satisfying $\delta \leq \frac{\epsilon(q_M - \mu_M)}{2q_M} - \frac{Nl^2_M + (N-1)f^2_M + \mu^2_M + Nl_M f_M}{2q_M}$.

Proof: The proof for Theorem 1 is provided in Appendix A. ■

Remark 1: The results (20) and (23) indicate that ***PSA*** stabilizes the network and satisfies the minimum data rate requirement. Specifically, q_M in (20) can be employed as the uniform buffer size at each node for a single flow. The inequality (22) gives the upper-bound of the power ***PSA*** can achieve. Since the constant B_1 is independent of V, (22) also ensures that ***PSA*** can achieve a power arbitrarily close to P^*_ϵ. When ϵ tends to 0, ***PSA*** can achieve a power arbitrarily close to the optimal value P^* with the tradeoff in buffer size q_M which is of order $O(\frac{1}{\epsilon})$ as shown in (21). In comparison, in [3], the tradeoff in the *average buffer occupancy* is of order $O(\frac{V}{\epsilon})$, where a large value of V is required to achieve close to the optimal value. In [17] which aims to achieve optimal throughput-utility, the internal buffer size is of order $O(\frac{1}{\epsilon})$, but the buffer size at source nodes is assumed infinitely large, which will result in a large average end-to-end delay. Note also that given buffer size q_M, the average end-to-end delay for flow $c \in \mathcal{F}$ can be upper-bounded by $\frac{Nq_M}{a_c}$ by Little's Law.

Remark 2: Note that in ***PSA***, the $R_c(t)$ regulator, the congestion controller and the link rate scheduler can operate locally at transport layer, source nodes and links, respectively. To reduce the complexity of the optimization of power allocator (17), distributed implementation can be developed in much the same way as in [2]. In addition, delayed queue backlogs can be employed similar to the analysis in [17], and our results

can be extended to the case where flows have arbitrary arrival rate at transport layer as in [4].

IV. Throughput-Optimal Algorithm (TSA) for Multi-Hop Wireless Networks

In this section, we propose a throughput-optimal scheduling algorithm ***TSA*** for the introduced multi-hop wireless network so that ***TSA*** maximizes the network throughput while satisfying the minimum data rate constraint. In addition, each link must meet the average link energy consumption constraint $(\eta_{mn})_{(m,n)\in\mathcal{L}}$, i.e.,

$$p_{mn} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{P_{mn}(\tau)\} \leq \eta_{mn}, \forall (m,n)\in\mathcal{L}. \tag{24}$$

Given $\epsilon > 0$, we define the ϵ-optimal throughput μ_ϵ^* as follows

$$\begin{aligned} \mu_\epsilon^* = \max_{(\mu_c)\in\Lambda} & \sum_{c\in\mathcal{F}}\mu_c, \\ s.t. \quad & p_{mn} \leq \eta_{mn} - \epsilon, \\ & \mu_c \geq a_c + \epsilon. \end{aligned}$$

Accordingly, we define the optimal throughput μ^* satisfying both the minimum data rate constraints and the link energy consumption constraints as

$$\begin{aligned} \mu^* = \max_{(\mu_c)\in\Lambda} & \sum_{c\in\mathcal{F}}\mu_c, \\ s.t. \quad & p_{mn} \leq \eta_{mn}, \\ & \mu_c \geq a_c. \end{aligned}$$

Then we have $\lim_{\epsilon\to 0+}\mu_\epsilon^* = \mu^*$, since

$$\mu_{\epsilon_M}^*\frac{\epsilon}{\epsilon_M} + \mu^*(1-\frac{\epsilon}{\epsilon_M}) \leq \mu_\epsilon^* \leq \mu^*,$$

where

$$\epsilon_M = \min\{\min_{(m,n)\in\mathcal{L}}\eta_{mn}, \arg\max_{\epsilon>0}\{(a_c+\epsilon)\in\Lambda\}\}$$

and the first inequality is derived from the fact that the ϵ-optimal throughput μ_ϵ^* is greater than or equal to the throughput achieved by the randomized algorithm that adopts a $\mu_{\epsilon_M}^*$-throughput achieving algorithm with probability $\frac{\epsilon}{\epsilon_M}$ and a μ^*-throughput achieving algorithm with probability $1-\frac{\epsilon}{\epsilon_M}$.

Similar to the virtual queue $Z_c(t)$ defined in Section II-B, for each link $(m,n)\in\mathcal{L}$, we define the virtual queue $X_{mn}(t)$ associated with each link energy consumption constraint (24), with queue dynamics as follows:

$$X_{mn}(t+1) = [X_{mn}(t) - \eta_{mn}]^+ + P_{mn}(t).$$

Thus, if the virtual queue $X_{mn}(t)$ is stable, by Lemma 1 the energy consumption constraint associated with link (m,n) is satisfied.

Given $\epsilon > 0$, ***TSA*** is designed to achieve a throughput arbitrarily close to μ_ϵ^*, with a tradeoff in buffer size which will be later shown in Theorem 2 and further explained in Remark 3.

With q_M standing for the buffer size, the algorithm ***TSA*** also operates on a time-slot basis consisting of four parts: $R_c(t)$ regulator, a congestion controller, a power allocator, and a link rate scheduler.

1) $R_c(t)$ **Regulator**:

$$\min_{0\leq R_c(t)\leq\mu_M} R_c(t)(\frac{q_M-\mu_M}{q_M}U_{s(c)}^c(t) - Z_c(t) - V), \tag{25}$$

where V is the same control parameter as that in ***PSA*** algorithm. Specifically, when $\frac{q_M-\mu_M}{q_M}U_{s(c)}^c(t) - Z_c(t) - V > 0$, $R_c(t)$ is set to zero; Otherwise, $R_c(t) = \mu_M$.

2) Congestion Controller: The congestion controller is the same as (16) in the ***PSA*** algorithm in Section III.

3) Power Allocator:

$$\max_{\mathbf{P}(t)\in\Pi}\sum_{(m,n)\in\mathcal{L}}(\mu_{mn}(\mathbf{P}(t))w_{mn}(t) - X_{mn}(t)P_{mn}(t)), \tag{26}$$

where we recall that $w_{mn}(t)$ is defined in (18). In (26), we can consider $\mu_{mn}(\mathbf{P}(t))w_{mn}(t)$ as the reward and $X_{mn}(t)P_{mn}(t)$ the cost induced from link (m,n) by allocating $\mathbf{P}(t)$. Compared to the power allocator in ***PSA***, now the "cost" is weighted by a time-varying virtual queue $X_{mn}(t)$ standing for the link power constraint.

4) Link Rate Scheduler: The link rate scheduler is the same as (19) in the ***PSA*** algorithm in Section III.

It is not difficult to check that Proposition 1 on finite buffer property still holds under the ***TSA*** algorithm. Now we present our main results of the ***TSA*** algorithm in Theorem 2.

Theorem 2: Given $\epsilon > 0$, if the buffer size q_M satisfies (21), ***TSA*** can achieve a throughput

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{\mu_{s(c)b(c)}^c(\tau)\} \geq \mu_{\epsilon/2}^* - \frac{B_2}{V}, \tag{27}$$

where $B_2 \triangleq B_1 + \frac{1}{2}\sum_{(m,n)\in\mathcal{L}}\eta_{mn}^2 + \frac{1}{2}|\mathcal{L}|P_M^2$. In addition, ***TSA*** ensures that the virtual queues are upper-bounded:

$$\begin{aligned} \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{\sum_{c\in\mathcal{F}}(U_{s(c)}^c(\tau) + Z_c(\tau)) \\ + \sum_{(m,n)\in\mathcal{L}}X_{mn}(t)\} \\ \leq \frac{B_2 + V(\mu^* - \mu_{\epsilon/2}^*)}{\delta}, \end{aligned} \tag{28}$$

where we recall δ is a positive constant defined in Theorem 1.

Proof: The proof for Theorem 2 is provided in Appendix B. ■

Remark 3: The inequality (28) indicates that ***TSA*** satisfies the minimum data rate requirements and the link consumption constraints. Since the constant B_2 is independent of V, (27) ensures that ***TSA*** can achieve a throughput arbitrarily close to μ^* when ϵ is small enough and V is large enough, with the tradeoff in buffer size q_M of order $O(\frac{1}{\epsilon})$ as shown in (21). By Little's Law, the average end-to-end delay for flow $c\in\mathcal{F}$ is upper-bounded by $\frac{Nq_M}{a_c}$.

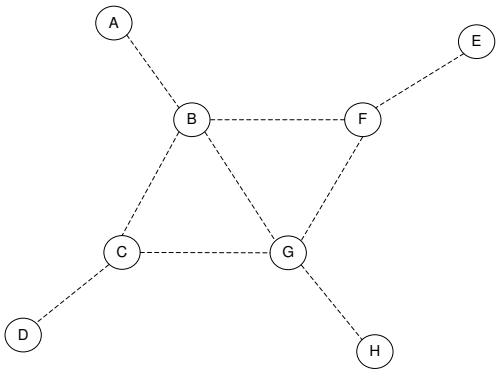

Fig. 2. Network topology for simulations

V. Numerical Results

In this section, a simulation-based performance evaluation of our proposed algorithms is presented. Consider the network topology in Figure 2 where each link is bidirectional. For simplicity, we assume the noise for any one-hop transmission is constant and normalized to 1. In addition, we set the maximum number of admissible packets as $\mu_M = 4$.

In our simulation setup in Figure 2, there are two flows with source-destination pairs (A, H) and (D, E) and the required minimum data rate a_c for the two flows are identical for simplicity. Simulations are run in Matlab 2009A over 10^5 time slots. Virtual queue dynamics are illustrated in Section V-A with performance for ***PSA*** and ***TSA*** introduced in Section V-B and Section V-C, respectively.

In Section V-A, Section V-B and Section V-C, we employ node-exclusive model (1) in our simulation. With the employment of quadrature amplitude modulation (QAM) schemes [2], the link rate function $\mu_{m_1n_1}(\mathbf{P}(t))$, $(m_1, n_1) \in \mathcal{L}$, is defined (in unit of packet/ time slot) as follows:

$$\mu_{m_1n_1}(\mathbf{P}(t)) = \begin{cases} 1, & \text{when } 0.25 \leq P_{m_1n_1}(t) < 0.5, \\ 2, & \text{when } 0.5 \leq P_{m_1n_1}(t) < 1.25, \\ 4, & \text{when } P_{m_1n_1}(t) \geq 1.25, \end{cases}$$

if $P_{m_2n_2}(t) = 0$, $\forall (m_2, n_2) \in \mathcal{L}$ s.t. $\{m_1, n_1\} \cap \{m_2, n_2\} \neq \emptyset$. Otherwise, $\mu_{m_1n_1}(\mathbf{P}(t)) = 0$. In Section V-D, we present simulation results for ***PSA*** under SNIR interference model (2) with time-varying propagation gains.

A. Control Parameter V and Virtual Queue Evolutions

We first present the virtual queue evolutions with different choices of parameter V for the two algorithms, where we recall that V is the control parameter in the power allocator (17) of ***PSA*** and the $R_c(t)$ regulator (25) of ***TSA***, which tradeoffs between the optimality and the upper-bounds of virtual queues as shown in Theorem 1 and Theorem 2, respectively. Figure 3 shows the virtual queue evolutions of per-flow $Z_c(t)$ and $U^c_{s(c)}(t)$ for ***PSA*** and Figure 4 the virtual queue evolutions of per-flow $Z_c(t)$, $U^c_{s(c)}(t)$ and per-link $X_{mn}(t)$.

By increasing V, we can approach the optimality, which is both illustrated in Theorem 1 and 2 and exemplified in the next subsection with ***PSA***. However, the convergence time becomes large for all vitual queues with a large V as shown in both Figure 3 and 4. For instance, when $V = 500$, it can take up to 10000 time slots for $Z_c(t)$ to converge for ***TSA*** algorithm. However, we will show in the next subsection that V does not have to be large to approach the optimal value.

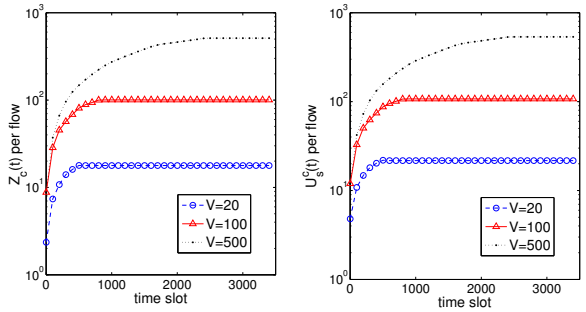

Fig. 3. Virtual queue dynamics for PSA algorithm

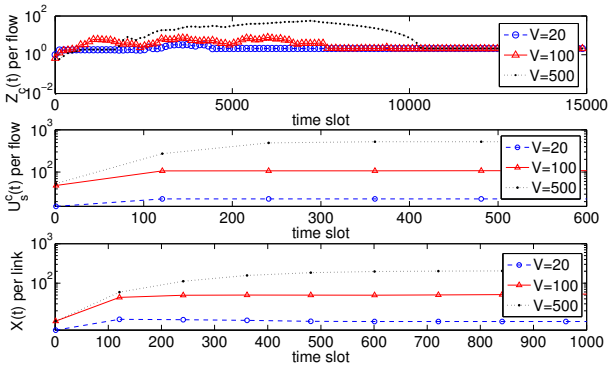

Fig. 4. Virtual queue dynamics for TSA algorithm

B. Performance of PSA Algorithm under Node-Exclusive Interference Model

In Figure 5, with fixed buffer size $q_M = 100$, we illustrate the energy consumption by increasing minimum data rate requirement a_c for different values of V. By fixing the control parameter V, the energy consumption increases for ***PSA*** as a_c increases; By fixing a_c, the energy consumption decreases as V increases (at the expense of convergence time as explained in Section V-A). Figure 5 also shows that $V = 100$ is sufficiently large such that almost no power gain is observed by further increasing the value of V. We also note that through simulation, V does not have a direct effect on delay/congestion performance of the algorithm. We will show in the next subsection that the value of q_M has a large impact on the congestion level of the network.

C. Performance of TSA Algorithm under Node-Exclusive Interference Model

In Figure 6, with fixed minimum data rate constraint $a_c = 0.6$ and fixed control parameter $V = 100$, we plot the throughput, congestion level (represented by the total number of packets in the network) and energy consumption of ***TSA***

TABLE I
PERFORMANCE OF ***PSA*** WHEN MINIMUM DATA RATE IS SET AS $a_c = 0.2$

	PSA ($q_M = 20$)	***PSA*** ($q_M = 50$)	***PSA*** ($q_M = 100$)	***PSA*** ($q_M = 200$)	EECA
Sum power	3.5336	3.2915	3.2696	3.2180	3.2145
No. packets in queues	102.59	283.78	584.50	1.1854×10^3	1.2665×10^3

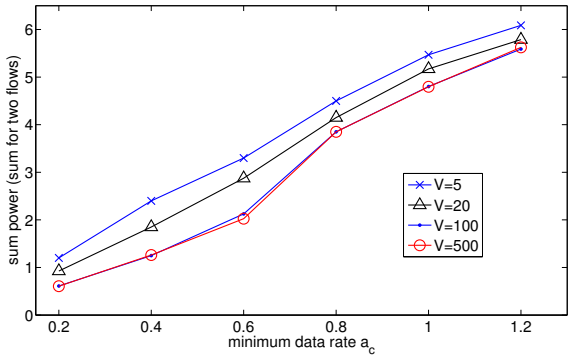

Fig. 5. Performance for PSA with fixed buffer size $q_M = 100$

against buffer size q_M with different values of link energy consumption constraint, where we uniformly set $\eta_{mn} = \eta$, $\forall (m,n) \in \mathcal{L}$, for simplicity. By fixing buffer size q_M and increasing the maximum allowed link power η, ***TSA*** improves the throughput at the expense of power expenditure where we note the congestion level (represented by the total number of packets in the network in Figure 6) is hardly affected. By fixing η and increasing buffer size q_M, ***TSA*** improves the throughput at the expense of power expenditure and an almost linear increase in the congestion level. Given η, Figure 6 also shows that throughput and energy consumption behave similarly and converge when q_M is large enough. Specifically, Figure 6 illustrates that $q_M = 100$ is sufficiently large such that there is hardly any gain in throughput with a larger q_M while we note that the congestion level increases almost linearly with q_M.

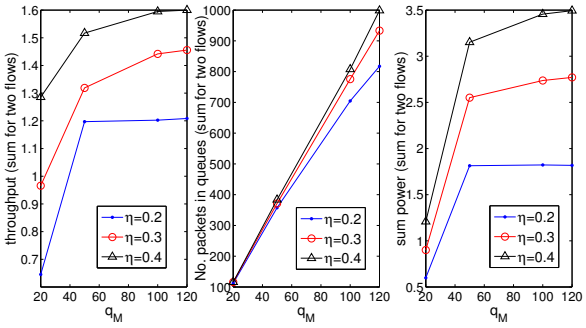

Fig. 6. Performance for TSA with fixed minimum data rate requirement $a_c = 0.6$ and control parameter $V = 100$

D. Performance of **PSA** *Under SNIR Interference Model*

In this section, with the same system setup and the same topology in Figure 2, we employ ***PSA*** under the SNIR interference model (2). We assume the distance between the two communication nodes of each link is normalized to 1. We setup the simulation parameters in (2) with reference to [2] [24] [25] [26]. Specifically, we set $f_{mn}(\cdot) = \log(\cdot)$ and the i.i.d. time-varying propagation gains in (2), $\forall t$, as

$$G_{mn}(t) = \begin{cases} d_{mn}^{-4}, & \text{w.p. } \dfrac{2}{3}, \\ \dfrac{1}{3} d_{mn}^{-4}, & \text{w.p. } \dfrac{1}{3}, \end{cases}$$

where d_{mn} denotes the Euclidean distance between any two nodes m and n. We set the power upper-bound $P_M = 10$ with respect to the normalized base noise (i.e., $B_n = 1$), and assume the transmission power of each communication link can be tuned to a number of levels in the set $\{0, \frac{1}{4}P_M, \frac{1}{2}P_M, \frac{3}{4}P_M, P_M\}$.

In Table I, we present the performance of our proposed finite-buffer algorithm ***PSA***, and compare to the result of EECA algorithm proposed in [3] which is proved to achieve optimal power while stabilizing any rate vector inside the capacity region. Since we also require minimum rates for individual flows, an additional virtual queue is employed in the congestion controller in EECA according to [4]. We set the control parameter $V = 10$ in ***PSA***, since a larger V cannot further minimize energy consumption but may increase buffer occupancy while a smaller V will lead to worse power performance. Table I illustrates that, with the consumed energy in ***PSA*** slightly larger than that of EECA, the queue backlog in the network is far smaller than that of EECA, which results in better end-to-end delay performance at a small energy cost. As we increase q_M, ***PSA*** approaches the optimal sum power which is the simulation value by EECA. In addition, the minimum data rate requirement and finite buffer property are achieved under ***PSA*** in the simulation.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed two cross-layer algorithms to minimize energy consumption and maximize throughput, respectively, for multi-hop wireless networks with finite buffers. Our work aims at a better understanding of the fundamental properties and performance limits of dynamic power allocation and scheduling in multi-hop wireless networks. We showed a tradeoff between $O(\frac{1}{\epsilon})$ in the finite buffer size and the ϵ-characterized proximity to the optimal power/throughput. Our future work will involve investigation on short-lived flows and distributed implementations of back-pressure-based algorithms with random access techniques.

REFERENCES

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[2] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks", in *IEEE Journal on Selected Area in Communications (JSAC)*, vol.23, no.1, pp. 89-103, March 2005.
[3] M. Neely, "Energy Optimal Control for Time Varying Wireless Networks", in *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915-2934, July 2006.
[4] L. Georgiadis, M. Neely and L.Tassiulas, "Resource allocation and cross-Layer control in wireless networks", in *Foundations and Trends in Networking*, pp. 1-149, 2006.
[5] M. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varing channels", Ph.D. dissertation, Mass. Inst. Technol. (MIT), Cambridge, MA, 2003.
[6] A. Eryilmaz, R. Srikant and J. Perkins, "Stable scheduling policies for fading wireless channels", in *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411-424, April 2005.
[7] X. Wu, R. Srikant and J. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks", in *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 595-605, June 2007.
[8] P. Chaporkar, K. Kar and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks", in *Proc. 43rd Annual Allerton Conference on Communicastions, Control and Computing*, pp. 1-11, 2005.
[9] C. Joo, X. Lin and N. Shroff, "Greedy maximal matching: performance limits for arbitrary network graphs under the node-exclusive interference model", in *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2734-2744, December 2009.
[10] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks", in *IEEE/ACM Trans. on Networking*, vol. 14, no. 2, pp. 302-315, April 2006.
[11] A. Eryilmaz, A. Ozdaglar, D. Shah and E. Modiano, "Distributed cross-layer algorithms for the optimal control of multi-hop wireless networks", in *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 638-651, April 2010.
[12] X. Lin and S. Rasool,"A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks", in *Proc. IEEE INFOCOM07*, pp. 1118-1126, May 2007.
[13] G. Sharma, N. Shroff, and R. Mazumdar, "Maximum weighted matching with interference constraints", in *Proc. IEEE Pervasive Computing and Communications Workshops (PERCOMW'06)*, pp. 1-5, March 2006.
[14] B. Miller and C. Bisdikian, "Bluetooth revealed: the insider's guide to an open specification for global wireless communications", Prentice Hall, 2000.
[15] B. Hajek and G. Sasaki, "Link scheduling in polynomial time", in *IEEE Transactions on Information Theory*, vol.34, no.5, pp. 910-917, Sep. 1988.
[16] P. Giaccone, E. Leonardi, and D. Shah, "Throughtput region of finite-buffered networks", in *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 251-263, Feb. 2007.
[17] L. Le, E. Modiano, and N. Shroff, "Optimal control of wireless networks with finite buffers", in *Proc. IEEE INFOCOM2010*, pp. 1-9, April 2010.
[18] D. Xue and E. Ekici, "Delay-guaranteed cross-layer scheduling in multi-hop wireless networks", Technical Report, 2011. Available: http://arxiv.org/abs/1009.4954.
[19] G. Gupta and N. Shroff, "Delay analysis for multi-hop wireless networks", in *Proc. IEEE INFOCOM09*, pp. 2356-2364, April 2009.
[20] V. Venkataramanan, X. Lin, L. Ying and S. Shakkottai, "On scheduling for minimizing end-to-end buffer usage over multihop wireless networks", in *Proc. IEEE INFOCOM2010*, pp. 1-9, March 2010.
[21] L. Bui, R. Srikant and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing", in *IEEE INFOCOM 2009 Mini-Conference*, pp. 2936-2940, April 2009.
[22] G. Gupta and N. Shroff, "Delay analysis for wireless networks with single hop traffic and general interference constraints", in *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 393-405, April 2010.
[23] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over mult-hop wireless networks", in *Proc. IEEE INFOCOM09*, pp. 1674-1682, April 2009.
[24] M. Chiang, "Balancing transport and physical layers in wireless multi-hop networks: jointly optimal congestion control and power control", in *IEEE JSAC*, vol. 23, no. 1, pp. 104-116, Jan. 2005.
[25] A. Reddy, S. Shakkottai, and L. Ying, "Distributed power control in wireless ad hoc networks using message passing: throughput optimality and network utility maximization", in *42nd Annual Conference on Information Sciences and Systems (CISS) 2008*, pp. 770-775, March 2008.
[26] Y. Shi, Y. Hou, S. Kompella, and H. Sherali, "Maximizing capacity in multi-hop cognitive radio networks under the SINR model", in *IEEE Transactions on Moble Computing*, vol. 10, no. 7, pp. 954-967, July 2011.
[27] J. Elson, G. Lewis, and D. Estrin, "Fine-grained network time synchronization using reference broadcast", in *Proc. The Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 147-163, 2002.
[28] Y. Tsai and G. Zhang, "Time and Frequency Synchronization for 3GPP Long Term Evolution Systems", in *Proc. IEEE Vehicular Technology Conference, 65th VTC2007-Spring*, pp. 1727-1731, April 2007.
[29] K. Romer, "Time synchronization in ad hoc networks", in *Proc. ACM MobiHoc'01*, pp. 173-182, October 2001.
[30] Dongyue Xue, Eylem Ekici, "Optimal Power Allocation in Multi-Hop Wireless Networks with Finite Buffers", in *IEEE International Conference on Communications (ICC)*, pp. 1-5, June 2011.

APPENDIX A
PROOF OF THEOREM 1

Before we proceed, we present the following lemma which will assist us in proving Theorem 1.

Lemma 2: For any feasible rate vector $(\theta_c) \in \Lambda$ with $\theta_c \geq a_c \ \forall c \in \mathcal{F}$, there exists a stationary randomized power allocation and scheduling algorithm STAT that stabilizes the network with input rate vector $(\mu^{c,STAT}_{s(c)b(c)}(t))$, power allocations $(P^{STAT}_{mn}(t))$ and scheduling parameters $(\mu^{c,STAT}_{mn}(t))$ independent of queue backlogs, such that the expected admitted rates are:

$$\mathbb{E}\{\mu^{c,STAT}_{s(c)b(c)}(t)\} = \theta_c, \forall t, \forall c \in \mathcal{F}.$$

In addition, $\forall t$, $\forall n \in \mathcal{N}, \forall c \in \mathcal{F}$, the flow balance constraints are satisfied:

$$\mathbb{E}\{\sum_{i:(n,i)\in\mathcal{L}} \mu^{c,STAT}_{ni}(t) - \sum_{j:(j,n)\in\mathcal{L}^c} \mu^{c,STAT}_{jn}(t)\} = 0.$$

Further, if there exists $\epsilon > 0$ such that $\theta_c = a_c + \epsilon \ \forall c \in \mathcal{F}$, then STAT can be developed to satisfy $\sum_{(m,n)\in\mathcal{L}} \mathbb{E}\{P^{STAT}_{mn}(t)\} = P^*_\epsilon$.

Note that it is not necessary for the randomized algorithm STAT to satisfy the buffer size constraint (21). Similar formulations of STAT and their proofs have been given in [2]- [4], so we omit the proof of Lemma 2 for brevity.

Remark 4: Given STAT algorithm in Lemma 2, we assign the input rates of the virtual queues $U^c_{s(c)}(t)$ at transport layer as $R^{STAT}_c(t) = \mu^{c,STAT}_{s(c)b(c)}(t)$. Thus, we also have $\mathbb{E}\{R^{STAT}_c(t)\} = \theta_c$. According to queue dynamics (12), it is easy to show that the virtual queues under STAT are upper-bounded by μ_M and the time-average of $R^{STAT}_c(t)$ satisfies: $r^{STAT}_c = \theta_c$. Note that (θ_c) can take values as $(a_c + \frac{\epsilon}{2})$ or $(a_c + \epsilon)$, where $\epsilon > 0$ such that $(a_c + \epsilon)$ is strictly inside Λ.

To prove Theorem 1, we let $\mathbf{Q}_1(t) = ((U^c_n(t)), (U^c_{s(c)}(t)), (Z_c(t)))$ and define the Lyapunov function $L_1(\mathbf{Q}_1(t))$ as follows:

$$L_1(\mathbf{Q}_1(t)) \triangleq \frac{1}{2}\{\frac{q_M - \mu_M}{q_M}\sum_{c\in\mathcal{F}} U^c_{s(c)}(t)^2 + \sum_{c\in\mathcal{F}} Z_c(t)^2 + \sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{1}{q_M} U^c_n(t)^2 U^c_{s(c)}(t)\},$$

with $L_1(\mathbf{Q}_1(0)) = 0$. We denote the Lyapunov drift by

$$\Delta_1(t) = \mathbb{E}\{L_1(\mathbf{Q}_1(t+1)) - L_1(\mathbf{Q}_1(t))|\mathbf{Q}_1(t)\}. \quad (29)$$

Note that the last term of the Lyapunov function $L_1(\mathbf{Q}_1(t))$ takes the same form as that in [16] [17]. From the queue

dynamics (10)(12), we have:

$$\begin{aligned}
&\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{1}{q_M}U_n^c(t+1)^2U_{s(c)}^c(t+1)\\
\leq&\sum_{c\in\mathcal{F}}\frac{1}{q_M}(R_c(t)+U_{s(c)}^c(t))\sum_{n\in\mathcal{N}}U_n^c(t+1)^2\\
\leq&\mu_M q_M NK+\sum_{c\in\mathcal{F}}\frac{1}{q_M}U_{s(c)}^c(t)\sum_{n\in\mathcal{N}}\{U_n^c(t)^2\\
&+(\sum_{i:(n,i)\in\mathcal{L}}\mu_{ni}^c(t))^2+(\sum_{j:(j,n)\in\mathcal{L}^c}\mu_{jn}^c(t))^2\\
&-2U_n^c(t)(\sum_i\mu_{ni}^c(t)-\sum_j\mu_{jn}^c(t))\},
\end{aligned}\tag{30}$$

where we recall that $R_c(t)\leq\mu_M$ and we square both sides of (10) to deduce the second inequality.

From (30), we have

$$\begin{aligned}
&\frac{1}{2}(\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{1}{q_M}(U_n^c(t+1)^2U_{s(c)}^c(t+1)-U_n^c(t)^2U_{s(c)}^c(t)))\\
\leq&\frac{1}{2}\sum_{c\in\mathcal{F}}\frac{(Nl_M^2+(N-1)f_M^2+\mu_M^2)U_{s(c)}^c(t)}{q_M}\\
+&\frac{1}{2}NKq_M\mu_M-\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{U_n^c(t)U_{s(c)}^c(t)}{q_M}\times\\
&(\sum_{i:(n,i)\in\mathcal{L}}\mu_{ni}^c(t)-\sum_{j:(j,n)\in\mathcal{L}^c}\mu_{jn}^c(t)),
\end{aligned}\tag{31}$$

where we employ (3)(11).

By squaring both sides of the queue dynamics (12)(14) and employing (31), we obtain from the Lyapunov drift (29):

$$\begin{aligned}
&\Delta_1(t)+V\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(t)|\mathbf{Q}_1(t)\}\\
\leq&B_1+\sum_{c\in\mathcal{F}}a_cZ_c(t)\\
&+\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)(\frac{q_M-\mu_M}{q_M}U_{s(c)}^c(t)-Z_c(t))|\mathbf{Q}_1(t)\}\\
&+\frac{1}{2}\sum_{c\in\mathcal{F}}\frac{(Nl_M^2+(N-1)f_M^2+\mu_M^2)U_{s(c)}^c(t)}{q_M}\\
&-\mathbb{E}\{\sum_{c\in\mathcal{F}}U_{s(c)}^c(t)\mu_{s(c)b(c)}^c(t)\frac{q_M-\mu_M}{q_M}\\
&-\sum_{c\in\mathcal{F}}\sum_{(m,n)\in\mathcal{L}}\mu_{mn}^c(t)\frac{U_{s(c)}^c(t)}{q_M}l_n-V\sum_{(m,n)\in\mathcal{L}}P_{mn}(t)\\
&+\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{U_n^c(t)U_{s(c)}^c(t)}{q_M}\times\\
&(\sum_{i:(n,i)\in\mathcal{L}}\mu_{ni}^c(t)-\sum_{j:(j,n)\in\mathcal{L}^c}\mu_{jn}^c(t))|\mathbf{Q}_1(t)\}.
\end{aligned}\tag{32}$$

We can rewrite the last term of RHS of (32) (the last four lines of (32)) by simple algebra as

$$\begin{aligned}
&-\sum_{c\in\mathcal{F}}\mathbb{E}\{\mu_{s(c)b(c)}^c(t)\frac{U_{s(c)}^c(t)}{q_M}(q_M-\mu_M-U_{b(c)}^c(t))|\mathbf{Q}_1(t)\}\\
&-\mathbb{E}\{\sum_{(m,n)\in\mathcal{L}}[\sum_{c\in\mathcal{F}}\mu_{mn}^c(t)\frac{U_{s(c)}^c(t)}{q_M}\\
&\times(U_m^c(t)-U_n^c(t)-l_n)-VP_{mn}(t)]|\mathbf{Q}_1(t)\}.
\end{aligned}$$

Note that the third term of the RHS of (32) is minimized by the $R_c(t)$ regulator (15), and the last term of the RHS of (32) is minimized by the combined policy of congestion controller (16), power allocator (17) and link rate scheduler (19), over a set of feasible algorithms including the stationary randomized algorithm STAT introduced in Lemma 2 and Remark 4. We can substitute into the third term of RHS of (32) a stationary randomized algorithm with admitted rate vector $(a_c+\frac{\epsilon}{2})$ and into the last term with a stationary randomized algorithm with admitted rate vector $(a_c+\epsilon)$. Thus, we have:

$$\begin{aligned}
&\Delta_1(t)+V\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(t)|\mathbf{Q}_1(t)\}\\
\leq&B_1+VP_\epsilon^*-\frac{\epsilon}{2}\sum_{c\in\mathcal{F}}Z_c(t)-\sum_{c\in\mathcal{F}}U_{s(c)}^c(t)\times\\
&(\frac{\epsilon(q_M-\mu_M)}{2q_M}-\frac{Nl_M^2+(N-1)f_M^2+\mu_M^2+Nl_Mf_M}{2q_M}),
\end{aligned}$$

where we employ the fact $\sum_{(m,n)\in\mathcal{L}}\mu_{mn}^c(t)l_n\leq Nl_Mf_M$, $\forall c\in\mathcal{F}$. When (21) holds, we can find $\delta>0$ such that $\delta\leq\frac{\epsilon(q_M-\mu_M)}{2q_M}-\frac{Nl_M^2+(N-1)f_M^2+\mu_M^2+Nl_Mf_M}{2q_M}$. Thus, we have:

$$\begin{aligned}
&\Delta_1(t)+V\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(t)|\mathbf{Q}_1(t)\}\\
\leq&B_1-\delta\sum_{c\in\mathcal{F}}(U_{s(c)}^c(t)+Z_c(t))+VP_\epsilon^*.
\end{aligned}\tag{33}$$

We take the expectation with respect to the distribution of $\mathbf{Q}_1$ on both sides of (33) and take the time average on $\tau=0,...,t-1$, which leads to

$$\begin{aligned}
&\frac{1}{t}\mathbb{E}\{L_1(\mathbf{Q}_1(t))\}+\frac{V}{t}\sum_{\tau=0}^{t-1}\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(\tau)\}\\
\leq&B_1+VP_\epsilon^*-\frac{\delta}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{U_{s(c)}^c(\tau)+Z_c(\tau)\}.
\end{aligned}\tag{34}$$

By taking limsup of t on both sides of (34), we can prove (22) and (23), respectively.

APPENDIX B
PROOF OF THEOREM 2

Similar to the proof of Theorem 1 in Appendix A, we first construct a set of stationary randomized algorithms in the following lemma:

Lemma 3: Given some small enough $\epsilon>0$, there exists a stationary randomized power allocations and scheduling algorithm $STAT_\epsilon^*$ that stabilizes the network with input

rate vector $(\mu_{s(c)b(c)}^{c,STAT_\epsilon^*}(t))$, power allocation $(P_{mn}^{STAT_\epsilon^*}(t))$ and scheduling parameters $(\mu_{mn}^{c,STAT_\epsilon^*}(t))$ independent of queue backlogs, such that the expected admitted rates are:

$$\sum_{c\in\mathcal{F}}\mathbb{E}\{\mu_{s(c)b(c)}^{c,STAT_\epsilon^*}(t)\}=\mu_\epsilon^*,\forall t.$$

In addition, $\forall t$, $\forall n\in\mathcal{N},\forall c\in\mathcal{F}$, the flow balance constraints are satisfied:

$$\mathbb{E}\{\sum_{i:(n,i)\in\mathcal{L}}\mu_{ni}^{c,STAT_\epsilon^*}(t)-\sum_{j:(j,n)\in\mathcal{L}^c}\mu_{jn}^{c,STAT_\epsilon^*}(t)\}=0.$$

Further, the minimum data rate constraints and the energy consumption constraints are satisfied:

$$\mathbb{E}\{\mu_{s(c)b(c)}^{c,STAT_\epsilon^*}(t)\}\geq a_c+\epsilon,\forall t,\forall c\in\mathcal{F},$$
$$\mathbb{E}\{P_{mn}^{STAT_\epsilon^*}(t)\}\leq\eta_{mn}-\epsilon,(m,n)\in\mathcal{L}.$$

For brevity, we omit the proof of Lemma 3, and interested readers are referred to [2]- [4] for more details. In $STAT_\epsilon^*$, we take the virtual input rate to $U_{s(c)}^c(t)$ as $R_c^{STAT_\epsilon^*}(t)=\mu_{s(c)b(c)}^{c,STAT_\epsilon^*}$.

We let $\mathbf{Q}_2(t)=((U_n^c(t)),(U_{s(c)}^c(t)),(Z_c(t)),(X_{mn}(t)))$ and define the Lyapunov function $L_2(\mathbf{Q}_2(t))$ as follows:

$$L_2(\mathbf{Q}_2(t))\triangleq L_1(\mathbf{Q}_1(t))+\frac{1}{2}\sum_{(m,n)\in\mathcal{L}}X_{mn}(t)^2,$$

with $L_2(\mathbf{Q}_2(0))=0$. We denote the Lyapunov drift by

$$\Delta_2(t)=\mathbb{E}\{L_2(\mathbf{Q}_2(t+1))-L_2(\mathbf{Q}_2(t))|\mathbf{Q}_2(t)\}.$$

Following the analysis in deriving (32), we obtain the inequality on the Lyapunov drift as follows:

$$\begin{aligned}
&\Delta_2(t)-V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}_2(t)\}\\
\leq&B_2+\sum_{c\in\mathcal{F}}a_cZ_c(t)-\sum_{(m,n)\in\mathcal{L}}\eta_{mn}X_{mn}(t)\\
&+\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)(\frac{q_M-\mu_M}{q_M}U_{s(c)}^c(t)-Z_c(t)-V)|\mathbf{Q}_2(t)\}\\
&+\frac{1}{2}\sum_{c\in\mathcal{F}}\frac{(Nl_M^2+(N-1)f_M^2+\mu_M^2)U_{s(c)}^c(t)}{q_M}\\
&-\sum_{c\in\mathcal{F}}\mathbb{E}\{\mu_{s(c)b(c)}^c(t)\frac{U_{s(c)}^c(t)}{q_M}\\
&\qquad\times(q_M-\mu_M-U_{b(c)}^c(t))|\mathbf{Q}_2(t)\}\\
&-\mathbb{E}\{\sum_{(m,n)\in\mathcal{L}}[\sum_{c\in\mathcal{F}}\mu_{mn}^c(t)\frac{U_{s(c)}^c(t)}{q_M}\\
&\qquad\times(U_m^c(t)-U_n^c(t)-l_n)-X_{mn}(t)P_{mn}(t)]|\mathbf{Q}_2(t)\}.
\end{aligned}\tag{35}$$

Note that the fourth term of the RHS of (35) is minimized by the $R_c(t)$ regulator (25), the sixth term minimized by the congestion controller, and the last term minimized by the combined policy of power allocator (26) and link rate scheduler, over a set of feasible algorithms including the stationary randomized algorithms introduced in Lemma 3. We can substitute into the fourth term of RHS of (35) a stationary randomized algorithm $STAT_{\epsilon/2}^*$ and into the last two terms a stationary randomized algorithm $STAT_\epsilon^*$. Thus, we have:

$$\begin{aligned}
&\Delta_2(t)-V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}_2(t)\}\\
\leq&B_2-V\mu_{\epsilon/2}^*-\delta[\sum_{c\in\mathcal{F}}(U_{s(c)}^c(t)+Z_c(t))+\sum_{(m,n)\in\mathcal{L}}X_{mn}(t)].
\end{aligned}$$

By taking the expectation with respect to the distribution of $\mathbf{Q}_2$ on both sides of the above inequality and taking the time average on $\tau=0,...,t-1$, we can prove (28) with limsup of t and (27) with liminf of t, respectively.

Dongyue Xue (S'11) received the B.S. degree in Information Engineering from Shanghai Jiaotong University, Shanghai, China, in 2009, and is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at The Ohio State University, Columbus, OH. His research interests include cross-layer scheduling in wireless networks and dynamic resource allocation in cognitive radio networks.

Eylem Ekici (S'99-M'02-SM'11) received his BS and MS degrees in Computer Engineering from Bogazici University, Istanbul, Turkey, in 1997 and 1998, respectively, and the PhD degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, GA, in 2002. Currently, he is an Associate Professor with the Department of Electrical and Computer Engineering, The Ohio State University. His current research interests include wireless networks, vehicular communication systems, cognitive radio networks, nano communication systems, with an emphasis on resource management, and analysis of network architectures and protocols. He is an Associate Editor of IEEE/ACM Transactions on Networking, Computer Networks Journal (Elsevier) and ACM Mobile Computing and Communications Review.