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

Dongyue Xue, Eylem Ekici

Department of Electrical and Computer Engineering
Ohio State University, USA
Email: {xued, ekici}@ece.osu.edu

***Abstract*—In this paper, we develop a cross-layer algorithm to minimize energy consumption in multi-hop wireless networks with finite buffers. Our algorithm guarantees a flow-based minimum data rate and a deterministic buffer size upper-bound for individual flows at network nodes. The algorithm jointly integrates congestion control, power allocation, routing and link rate scheduling. In addition, the algorithm achieves a power expenditure "ϵ-close" to the optimal value, with a tradeoff of order $O(\frac{1}{\epsilon})$ in the buffer size.**

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]-[7], with suboptimal algorithms and distributed algorithms proposed in [8]-[11]. 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 [14]-[17]. However, these works do not provide queue backlog (or buffer size) guarantees.

In this paper, we develop a cross-layer optimal algorithm that aims to optimize power allocation in multi-hop wireless networks with finite buffers. Since the objective of minimizing energy consumption may cause unfairness to individual flows in the network, we place additional per-flow minimum data rate constraints. Our algorithm is composed of a regulator, a congestion controller, a power allocator, and a link rate scheduler. The regulator and congestion controller are used 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.

To the best of our knowledge, our algorithm is the first of its kind to achieve an energy consumption performance arbitrarily close to the optimal, with a tradeoff of $O(\frac{1}{\epsilon})$ in the buffer size for individual flows at nodes, where ϵ characterizes the difference between the achieved power and the optimal theoretical value. The buffer size upper-bound is deterministic, which leads to bounded average end-to-end delay by Little's Theorem. In comparison, the buffer size is assumed infinitely large in the power allocation algorithm proposed in [3]. In [12][13], which aim to achieve optimal throughput-utility, the ingress buffer size (the buffer size at source nodes) is assumed infinitely large although the internal buffer sizes are finite.

The rest of the paper is organized as follows. Section II provides the network model for multi-hop wireless networks. In Section III, we propose the optimal power allocation and scheduling algorithm and analyze its performance. Finally, we conclude our work in Section IV.

II. NETWORK MODEL

A. Network Elements

We consider a multi-hop wireless network consisting of N nodes and K flows. Time is slotted with integer values $t \in \{0, 1, 2, ...\}$. 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.

Let $\mathbf{P}(t) = (P_{mn}(t))_{(m,n)\in\mathcal{L}}$ represent the power allocation vector for time slot t according to a power allocation algorithm. We constrain $\mathbf{P}(t)$ to be in Π, i.e., $\mathbf{P}(t) \in \Pi$, where Π is the compact set of feasible power vectors. We also assume that $P_{mn}(t) \leq P_M, \forall (m, n) \in \mathcal{L}, \forall \mathbf{P}(t) \in \Pi$, 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 $\mu(\mathbf{P}(t)) = (\mu_{mn}(\mathbf{P}(t)))_{(m,n)\in\mathcal{L}}$. For convenience of analysis, we define:

$$\begin{aligned} l_n &\triangleq \max_{\mathbf{P}\in\Pi} \sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}(\mathbf{P}), \\ l_M &\triangleq \max_{n\in\mathcal{N}} l_n, \\ f_M &\triangleq \max_{n\in\mathcal{N}} \max_{\mathbf{P}\in\Pi} \sum_{j:(j,n)\in\mathcal{L}} \mu_{jn}(\mathbf{P}), \end{aligned} \tag{1}$$

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.

We assume that the source node for flow c is always backlogged at the transport layer. For a feasible scheduling decision in time slot t, we let the scheduling parameter $\mu_{mn}^c(t)$ be the link rate assignment for flow c for link (m,n). Thus, given $\mathbf{P}(t)$, we must have $\sum_{c\in\mathcal{F}} \mu_{mn}^c(t) \le \mu_{mn}(\mathbf{P}(t))$, $\forall (m,n) \in \mathcal{L}$.

Let $\mu_{s(c)b(c)}^c(t)$ 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, $\mu_{s(c)n}^c(t) = 0\ \forall n \ne b(c)$. We also assume that $\mu_{s(c)b(c)}^c(t)$ is upper-bounded by a constant $\mu_M > 0$:

$$\mu_{s(c)b(c)}^c(t) \le \mu_M,\ \forall c \in \mathcal{F}, \forall t, \tag{2}$$

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

$$\sum_{m\in\mathcal{N}} (\mu_{mb(c)}^c(t)) = 0\ \forall c \in \mathcal{F}, \forall t. \tag{3}$$

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

To represent network stability, we begin with a definition of queue stability with respect to queue backlog $A(t)$. The queue is *stable* if

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

Let $U_n^c(t)$ be queue backlog of flow c packets at node n. Then, the network is *stable* if queues $U_n^c(t)$ 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 $n = d(c)$ then we have $U_n^c(t) = 0\ \forall t$; Otherwise, the queue dynamics are:

$$\begin{aligned} U_n^c(t+1) \le & [U_n^c(t) - \sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}^c(t)]^+ \\ & + \sum_{j:(j,n)\in\mathcal{L}^c} \mu_{jn}^c(t),\ \text{if } n \in \mathcal{N}\backslash d(c), \end{aligned} \tag{4}$$

where we define the operator $[x]^+$ as $[x]^+ = \max\{x, 0\}$. Note that in (4), 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 $\sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}^c(t)$ and $\sum_{j:(j,n)\in\mathcal{L}^c} \mu_{jn}^c(t)$ represent, respectively, the scheduled departure rate from node n and the scheduled arrival rate into node n for flow c. Note that (4) is an inequality since the arrival rates from corresponding neighbor nodes may be less than $\sum_j \mu_{jn}^c(t)$ if some neighbor node does not have enough packets to transmit. From (2)(3), we also have

$$\sum_{j:(j,b(c))\in\mathcal{L}^c} \mu_{jb(c)}^c(t) \le \mu_M, \tag{5}$$

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

For each flow c at transport layer, we construct a virtual queue $U_{s(c)}^c(t)$ at transport layer to later assist the development of our proposed algorithm in the next section. 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 $R_c(t)$. We update the virtual queue as follows:

$$U_{s(c)}^c(t+1) = [U_{s(c)}^c(t) - \mu_{s(c)b(c)}^c(t)]^+ + R_c(t), \tag{6}$$

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

$$\mu_c \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mu_{s(c)b(c)}^c(\tau) \ge r_c \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R_c(\tau). \tag{7}$$

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

$$Z_c(t+1) = [Z_c(t) - R_c(t)]^+ + a_c, \tag{8}$$

where we set $Z_c(0) = 0$. If queue $Z_c(t)$ is stable, we have $r_c \ge a_c$. Additionally, if $U_{s(c)}^c(t)$ is stable, then according to (7), we have $\mu_c \ge a_c$, i.e., the minimum data rate for flow c is achieved.

Similar as in [2][3], we can define the capacity region Λ of the multi-hop network as the closure of all stabilizable rate vectors satisfying the minimum data rate constraints, considering all power allocation algorithm choosing $\mathbf{P}(t) \in \Pi$. To assist the analysis in the following section, 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 [5] we have $\lim_{\epsilon\to 0^+} P_\epsilon^* = P^*$, where P^* is the minimum sum power for stabilizing rates (a_c), and we assume the minimum rate requirement (a_c) is strictly inside Λ without loss of generality.

III. Power Allocation Algorithm for Multi-Hop Wireless Networks

In this section, we propose a power allocation and scheduling algorithm ***ALG*** for the introduced multi-hop wireless network so that ***ALG*** stabilizes the network and satisfies the minimum data rate constraint. Given $\epsilon > 0$, ***ALG*** can achieve a sum power arbitrarily close to P_ϵ^*, with a tradeoff with buffer size which will be later given in Theorem 1.

Let $q_M \ge \max\{l_M, \mu_M\}$ be a control parameter for buffer size. The optimal algorithm ***ALG*** consists of four parts: $R_c(t)$ regulator, a congestion controller, a power allocator, and a link rate scheduler. We propose and analyze the algorithm in the following subsections.

A. Algorithm Description and Analysis

1) $R_c(t)$ Regulator:

$$\min_{0\le R_c(t)\le \mu_M} R_c(t)\left(\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t)-Z_c(t)\right). \tag{9}$$

$R_c(t)$ regulator determines the input rate to the virtual queue $U^c_{s(c)}(t)$ in each time slot. 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:

$$\max_{0\le \mu^c_{s(c)b(c)}(t)\le \mu_M} \mu^c_{s(c)b(c)}(t)(q_M-\mu_M-U^c_{b(c)}(t)). \tag{10}$$

Specifically, when $q_M-\mu_M-U^c_{b(c)}(t)\le 0$, $\mu^c_{s(c)b(c)}(t)$ 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:

$$\max_{\mathbf{P}(t)\in\Pi} \sum_{(m,n)\in\mathcal{L}} (\mu_{mn}(\mathbf{P}(t))w_{mn}(t)-VP_{mn}(t)), \tag{11}$$

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

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

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 (11). Different from the traditional back-pressure algorithm, we add a weight of l_n to the differential backlog $(U^c_m(t)-U^c_n(t))$ for link $(m,n)\in\mathcal{L}$ and flow $c\in\mathcal{F}$. In (11), we can consider $\mu_{mn}(\mathbf{P}(t))w_{mn}(t)$ as the reward and $VP_{mn}(t)$ the cost induced from link (m,n) by allocating $\mathbf{P}(t)$.

4) Link Rate Scheduler:

$$\mu^c_{mn}(t)=\begin{cases}\mu_{mn}(\mathbf{P}(t)), & \text{if } c=c^*_{mn}(t),\\ 0, & \text{otherwise},\end{cases} \tag{12}$$

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

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

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.

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

$$U^c_n(t)\le q_M, \quad \forall t,\forall n\in\mathcal{N},\forall c\in\mathcal{F}. \tag{13}$$

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 $U^c_n(t)\le q_M\ \forall n,c$. 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 $U^c_{b(c)}(t)\le q_M-\mu_M$, then according to the queue dynamics (4)(5), $U^c_{b(c)}(t+1)\le q_M$; Otherwise, $U^c_{b(c)}(t)>q_M-\mu_M$ and according to the congestion controller (10), we have $\mu^c_{s(c)b(c)}(t)=0$, so $U^c_{b(c)}(t+1)\le U^c_{b(c)}(t)\le q_M$ by (3)(4).
(2) In the second case, n is not the source node of flow c. If $U^c_n(t)\le q_M-l_M$, then $U^c_n(t)\le q_M$ by (4); Otherwise, $U^c_n(t)>q_M-l_M$, and according to the link rate scheduler (12) we have $\mu^c_{mn}(t)=0\ \forall m\in\mathcal{N}$, so $U^c_n(t+1)\le U^c_n(t)\le q_M$ by the queue dynamics (4).

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

Now we present our main results in Theorem 1.

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

$$q_M>\frac{Nl_M^2+(N-1)f_M^2+\mu_M^2+Nl_Mf_M}{\epsilon}+\mu_M, \tag{14}$$

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

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(\tau)\}\le P^*_\epsilon+\frac{B}{V}, \tag{15}$$

where $B\triangleq\frac{1}{2}\mu_Mq_MNK+\frac{3q_M-2\mu_M}{2q_M}K\mu_M^2+\frac{1}{2}\sum_{c\in\mathcal{F}}a_c^2$.

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

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{U^c_{s(c)}(\tau)+Z_c(\tau)\}\le\frac{B'}{\delta}, \tag{16}$$

where $B'\triangleq B+VP^*_\epsilon$ and δ is a positive constant satisfying $\delta\le\frac{\epsilon(q_M-\mu_M)}{2q_M}-\frac{Nl_M^2+(N-1)f_M^2+\mu_M^2+Nl_Mf_M}{2q_M}$.

Remark 1: The results (13) and (16) indicate that ***ALG*** stabilizes the network and satisfies the minimum data rate requirement. Specifically, q_M in (13) can be employed as the buffer size at each node for a single flow. The inequality (15) gives the upper-bound of the power ***ALG*** can achieve. Since the constant B is independent of V, (15) also ensures that ***ALG*** can achieve a power arbitrarily close to P^*_ϵ. When ϵ tends to 0, ***ALG*** 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 (14). In comparison, in [3], the tradeoff in the *average buffer occupancy* is of order $O(\frac{V}{\epsilon})$, where large V is required to achieve close to the optimal value. In [13] 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.

Remark 2: Note that in ***ALG***, 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 (11), 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 [13], and our results can be extended to the case where flows have arbitrary arrival rate at transport layer as in [4].

B. Proof of Theorem 1

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

Lemma 1: 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 $(\mu_{s(c)b(c)}^{c,STAT}(t))$ and scheduling parameters $(\mu_{mn}^{c,STAT}(t))$ independent of queue backlogs, such that the expected admitted rates are:

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

In addition, $\forall t$, $\forall n \in \mathcal{N}, \forall c$, the flow constraint is satisfied:

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

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 $\sum_{(m,n)\in\mathcal{L}} \mathbb{E}\{P_{mn}(t)\} = P_\epsilon^*$.

Note that it is not necessary for the randomized algorithm STAT to satisfy the buffer size constraint (14). Similar formulations of STAT and their proofs have been given in [2] and [4], so we omit the proof of Lemma 1 for brevity.

Remark 3: Given STAT algorithm in Lemma 1, we assign the input rates of the virtual queues at transport layer as $R_c^{STAT}(t) = \mu_{s(c)b(c)}^{c,STAT}(t)$. Thus, we also have $\mathbb{E}\{R_c^{STAT}(t)\} = \theta_c$. According to queue dynamics (6), it is easy to show that the virtual queues under STAT are upper-bounded by μ_M and the time-average of $R_c^{STAT}(t)$ satisfies: $r_c^{STAT} = \theta_c$. 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 $\mathbf{Q}(t) = ((U_n^c(t)), (U_{s(c)}^c(t)), (Z_c(t)))$ and define the Lyapunov function $L(\mathbf{Q}(t))$ as follows:

$$\begin{aligned} L(\mathbf{Q}(t)) = &\frac{1}{2}\{\frac{q_M - \mu_M}{q_M} \sum_{c\in\mathcal{F}} U_{s(c)}^c(t)^2 \\ &+ \sum_{c\in\mathcal{F}} Z_c(t)^2 + \sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{1}{q_M} U_n^c(t)^2 U_{s(c)}^c(t)\}, \end{aligned} \tag{17}$$

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

$$\Delta(t) = \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\}. \tag{18}$$

Note that the last term of the Lyapunov function (17) takes the same form as that in [12][13]. From the queue dynamics (4)(6), following the proof provided in our technical report [18], we can obtain:

$$\begin{aligned} &\frac{1}{2}(\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{1}{q_M}(U_n^c(t+1)^2 U_{s(c)}^c(t+1) \\ &\qquad - U_n^c(t)^2 U_{s(c)}^c(t))) \\ \leq &\frac{1}{2}\sum_{c\in\mathcal{F}} \frac{(Nl_M^2 + (N-1)f_M^2 + \mu_M^2)U_{s(c)}^c(t)}{q_M} \\ &+\frac{1}{2}NKq_M\mu_M - \sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{U_n^c(t)U_{s(c)}^c(t)}{q_M} \\ &\qquad (\sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}^c(t) - \sum_{j:(j,n)\in\mathcal{L}^c} \mu_{jn}^c(t)). \end{aligned} \tag{19}$$

Similarly, by squaring both sides of the queue dynamics (6), we have:

$$\begin{aligned} &\frac{1}{2}\sum_{c\in\mathcal{F}}(U_{s(c)}^c(t+1)^2 - U_{s(c)}^c(t)^2) \\ \leq &K\mu_M^2 - \sum_{c\in\mathcal{F}} U_{s(c)}^c(t)(\mu_{s(c)b(c)}^c(t) - R_c(t)). \end{aligned} \tag{20}$$

By squaring both sides of the queue dynamics (8), we have:

$$\begin{aligned} &\frac{1}{2}\sum_{c\in\mathcal{F}}(Z_c(t+1)^2 - Z_c(t)^2) \\ \leq &\frac{1}{2}K\mu_M^2 + \frac{1}{2}\sum_{c\in\mathcal{F}} a_c^2 - \sum_{c\in\mathcal{F}} Z_c(t)(R_c(t) - a_c). \end{aligned} \tag{21}$$

Substituting (19)(20)(21) into the Lyapunov drift (18) and adding $V\sum_{(m,n)\in\mathcal{L}} \mathbb{E}\{P_{mn}(t)|\mathbf{Q}(t)\}$ to both sides, we obtain:

$$\begin{aligned} &\Delta(t) + V\sum_{(m,n)\in\mathcal{L}} \mathbb{E}\{P_{mn}(t)|\mathbf{Q}(t)\} \\ \leq &B + \sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)(\frac{q_M - \mu_M}{q_M} U_{s(c)}^c(t) - Z_c(t))|\mathbf{Q}(t)\} \\ &+ \sum_{c\in\mathcal{F}} a_c Z_c(t) \\ &+ \frac{1}{2}\sum_{c\in\mathcal{F}} \frac{(Nl_M^2 + (N-1)f_M^2 + \mu_M^2)U_{s(c)}^c(t)}{q_M} \\ &- \mathbb{E}\{\sum_{c\in\mathcal{F}} U_{s(c)}^c(t)\mu_{s(c)b(c)}^c(t)\frac{q_M - \mu_M}{q_M} \\ &- \sum_{c\in\mathcal{F}}\sum_{(m,n)\in\mathcal{L}} \mu_{mn}^c(t)\frac{U_{s(c)}^c(t)}{q_M} l_n - V\sum_{(m,n)\in\mathcal{L}} P_{mn}(t) \\ &+ \sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{U_n^c(t)U_{s(c)}^c(t)}{q_M} \times \\ &\quad (\sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}^c(t) - \sum_{j:(j,n)\in\mathcal{L}^c} \mu_{jn}^c(t))|\mathbf{Q}(t)\}. \end{aligned} \tag{22}$$

We can rewrite the last term of RHS of (22) (the last four lines of (22)) by simple algebra as

$$-\sum_{c\in\mathcal{F}}\mathbb{E}\{\mu^c_{s(c)b(c)}(t)\frac{U^c_{s(c)}(t)}{q_M}(q_M-\mu_M-U^c_{b(c)}(t))|\mathbf{Q}(t)\}$$
$$-\mathbb{E}\{\sum_{(m,n)\in\mathcal{L}}[\sum_{c\in\mathcal{F}}\mu^c_{mn}(t)\frac{U^c_{s(c)}(t)}{q_M}\times$$
$$(U^c_m(t)-U^c_n(t)-l_n)-VP_{mn}(t)]|\mathbf{Q}(t)\}.$$

Note that the second term of the RHS of (22) is minimized by the $R_c(t)$ regulator (9), and the last term of the RHS of (22) is minimized by the combined policy of congestion controller (10), power allocator (11) and link rate scheduler (12), over a set of feasible algorithms including the stationary randomized algorithm STAT introduced in Lemma 1 and Remark 3. We can substitute into the second term of RHS of (22) 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:

$$\Delta(t)+V\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(t)|\mathbf{Q}(t)\}$$
$$\le B+VP^*_\epsilon-\frac{\epsilon}{2}\sum_{c\in\mathcal{F}}Z_c(t)-\sum_{c\in\mathcal{F}}U^c_{s(c)}(t)\times$$
$$(\frac{\epsilon(q_M-\mu_M)}{2q_M}-\frac{Nl_M^2+(N-1)f_M^2+\mu_M^2+Nl_Mf_M}{2q_M}),$$

where we employ the fact $\sum_{(m,n)\in\mathcal{L}}\mu^c_{mn}(t)l_n\le Nl_Mf_M$, $\forall c\in\mathcal{F}$. When (14) holds, we can find $\delta>0$ such that $\delta\le\frac{\epsilon(q_M-\mu_M)}{2q_M}-\frac{Nl_M^2+(N-1)f_M^2+\mu_M^2+Nl_Mf_M}{2q_M}$. Thus, we have:

$$\begin{aligned}&\Delta(t)+V\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(t)|\mathbf{Q}(t)\}\\&\le B-\delta\sum_{c\in\mathcal{F}}(U^c_{s(c)}(t)+Z_c(t))+VP^*_\epsilon.\end{aligned}\tag{23}$$

We take the expectation with respect to the distribution of $\mathbf{Q}$ on both sides of (23) and take the time average on $\tau=0,...,t-1$, which leads to

$$\begin{aligned}&\frac{1}{t}\mathbb{E}\{L(\mathbf{Q}(t))\}+\frac{V}{t}\sum_{\tau=0}^{t-1}\sum_{(m,n)\in\mathcal{L}}\mathbb{E}\{P_{mn}(\tau)\}\\&\le B+VP^*_\epsilon-\frac{\delta}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{U^c_{s(c)}(\tau)+Z_c(\tau)\}.\end{aligned}\tag{24}$$

By taking limsup of t on both sides of (24), we have:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{U^c_{s(c)}(\tau)+Z_c(\tau)\}\le\frac{B'}{\delta}.$$

Thus, we have proved (16). Also by taking limsup of t on both sides of (24), we can prove (15).

IV. Conclusions and Further Discussions

In this paper, we proposed a cross-layer algorithm to minimize energy consumption 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. While we show a tradeoff between $O(\frac{1}{\epsilon})$ in the buffer size and the ϵ-characterized proximity to the optimal power, how the actual buffer occupancy evolves still remains elusive to us. Thus, our future work will involve queue backlog evolutions of back-pressure-based algorithms. In addition, we will investigate the impact of the number of flows on the performance of such algorithms.

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 and E. Modiano, “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] A. Eryilmaz and R. Srikant, “Joint congestion control, routing and MAC for stability and fairness in wireless networks”, in *IEEE JSAC*, vol. 24, no. 8, pp. 1514-1524, August 2006.

[8] 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.

[9] 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*, 2005.

[10] A. Eryilmaz, A. Ozdaglar, D. Shah and E. Modiano, “Distributed cross-layer algorithms for the optimal control of multi-hop wireless networks”, to appear in *IEEE/ACM Transactions on Networking*.

[11] 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*, May 2007.

[12] 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.

[13] L. Le, E. Modiano, and N. Shroff, “Optimal control of wireless networks with finite buffers”, in *Proc. IEEE INFOCOM2010*, April 2010.

[14] G. Gupta and N. Shroff, “Delay analysis for multi-hop wireless networks”, in *Proc. IEEE INFOCOM09*, April 2009.

[15] 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*, March 2010.

[16] 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*, April 2009.

[17] L. Ying, S. Shakkottai, and A. Reddy, “On combining shortest-path and back-pressure routing over mult-hop wireless networks”, in *Proc. IEEE INFOCOM09*, April 2009.

[18] D. Xue and E. Ekici, “Optimal Power Allocation in Multi-Hop Wireless Networks with Finite Buffers”, Technical Report, 2011. Available: http://www.ece.osu.edu/∼xued/ICC11.pdf