Delay-Guaranteed Cross-Layer Scheduling in Multihop Wireless Networks

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

***Abstract*—In this paper, we propose a cross-layer scheduling algorithm that achieves a throughput "ϵ-close" to the optimal throughput in multihop wireless networks with a tradeoff of $O(\frac{1}{\epsilon})$ in average end-to-end delay guarantees. The algorithm guarantees finite buffer sizes and aims to solve a joint congestion control, routing, and scheduling problem in a multihop wireless network while satisfying per-flow average end-to-end delay constraints and minimum data rate requirements. This problem has been solved for both backlogged as well as arbitrary arrival rate systems. Moreover, we discuss the design of a class of low-complexity suboptimal algorithms, effects of delayed feedback on the optimal algorithm, and extensions of the proposed algorithm to different interference models with arbitrary link capacities.**

***Index Terms*—Cross-layer designs, delay guarantees, finite buffer, flow control and scheduling, multihop wireless networks, throughput optimization.**

I. Introduction

CROSS-LAYER design of congestion control, routing and scheduling algorithms with quality-of-service (QoS) guarantees is one of the most challenging topics in wireless networking. The back-pressure algorithm first proposed in [1] and its extensions have been widely employed in developing throughput optimal dynamic resource allocation and scheduling algorithms for wireless systems. Back-pressure-based scheduling algorithms have also been employed in wireless networks with time-varying channels [2], [3]. Congestion controllers at the transport layer have assisted the cross-layer design of scheduling algorithms in [4]–[6], so that the admitted arrival rate is guaranteed to lie within the network capacity region. Low-complexity distributed algorithms have been proposed in [7]–[10]. Algorithms adapted to clustered networks have been proposed in [11] to reduce the number of queues maintained in the network. However, delay-related investigations are not included in these works.

In this paper, we propose a cross-layer algorithm to achieve guaranteed throughput while satisfying network QoS requirements and guaranteeing that all actual queue backlogs are deterministically upper-bounded. Specifically, we construct two virtual queues, i.e., a *virtual queue at transport layer* and a *virtual delay queue*, to guarantee average end-to-end delay bounds. Moreover, we construct a *virtual service queue* to guarantee the minimum data rate required by individual network flows. Our cross-layer design includes a congestion controller for the input rate to the virtual queue at transport layer, as well as a joint policy for packet admission, routing, and resource scheduling. We show that our algorithm can achieve a throughput arbitrarily close to the optimal. In addition, the algorithm exhibits a tradeoff of $O(\frac{1}{\epsilon})$ in the delay bound, where ϵ denotes the distance from the optimal throughput.

Our main algorithm is further extended: 1) to a set of low-complexity suboptimal algorithms; (2) from a model with constantly-backlogged sources to a model with sources of arbitrary input rates at transport layer; 3) to an algorithm employing delayed queue information; and 4) from a node-exclusive model with constant link capacities to a model with arbitrary link capacities and interference models over fading channels. These extensions add to the versatility of the algorithm and show the applicability of the proposed virtual-queue-based algorithm to more general models.

The rest of the paper is organized as follows. Section II discusses the related work. In Section III, the network model is presented, followed by the introduction of queue dynamics for packet queues and virtual queues. In Section IV, the optimal cross-layer control and scheduling algorithm is described, and its performance analyzed. In Section V, we provide a class of feasible suboptimal algorithms, consider sources with arbitrary arrival rates at the transport layer, employ delayed queue information in the scheduling algorithm, and extend the model to arbitrary link capacities and interference models over fading channels. We present numerical results in Section VI. Finally, we conclude our work in Section VII.

II. Related Work

Delay issues in single-hop wireless networks have been addressed in [12]–[15]. Especially, the scheduling algorithm in [13] provides a throughput utility that is inversely proportional to the delay guarantee. Authors of [14] have obtained delay bounds for two classes of scheduling policies. However, these works are not readily extendable to multihop wireless networks, where additional arrivals from neighboring nodes and routing must be considered. Delay analysis for multihop networks with fixed routing is provided in [16]. Delay-related scheduling in multihop wireless networks has been proposed in [17]–[21]. However, none of the above-mentioned works provides explicit end-to-end delay guarantees.

Manuscript received January 25, 2011; revised December 13, 2011 and August 23, 2012; accepted November 16, 2012; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Sarkar. Date of publication December 20, 2012; date of current version December 13, 2013. This work was supported in part by the NSF under Grant CCF-0914912.

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2012.2230404

There are several works aiming to address end-to-end delay or buffer occupancy guarantees in multihop wireless networks. Worst-case delay is guaranteed in [22] with a packet-dropping mechanism. However, dropped packets are not compensated or retransmitted with the algorithm of [22], which may lead to restrictions in its practical implementations. A low-complexity cross-layer fixed-routing algorithm is developed in [23] to guarantee order-optimal average end-to-end delay, but only for half of the capacity region (note that it may achieve more than half in a given network). A scheduling algorithm for finite-buffer multihop wireless networks with fixed routing is proposed in [24] and is extended to adaptive routing with congestion controller in [25]. Specifically, the algorithm in [25] guarantees $O(\frac{1}{\epsilon})$-scaling in buffer size with an ϵ-loss in throughput utility, but this is achieved at the expense of the buffer occupancy of the source nodes, where an infinite buffer size in the network layer is assumed in each source node. This leads to large average end-to-end delay since the network stability is achieved based on queue backlogs at these source nodes.

Compared to the above works, the algorithm presented in this paper develops and incorporates novel virtual queue structures. Different from traditional back-pressure-based algorithms, where the network stability is achieved at the expense of large packet queue backlogs, in our algorithm, "the burden" of actual packet queue backlogs is shared by our proposed virtual queues in an attempt to guarantee specific delay performances and finite buffer sizes. Specifically, we design a congestion controller for a virtual input rate and assign weights in the scheduling policy as a product of actual packet queue backlog and the weighted backlog of a designed virtual queue, which will be introduced in detail in Section IV. As such, the network stabilization is achieved with the help of virtual queue structures that do not contribute to delay in the network. Since all packet queues in the network, including those in source nodes, have finite sizes, all average end-to-end delays are bounded independent of length or multiplicity of paths.

III. Network Model

A. Network Elements

We consider a time-slotted multihop wireless network consisting of N nodes and K flows. Denote by $(m,n) \in \mathcal{L}$ a link from node m to node n, where $\mathcal{L}$ is the set of directed links in the network. Denoting the set of flows by $\mathcal{F}$ and the set of nodes by $\mathcal{N}$, we formulate the network topology $G = (\mathcal{N}, \mathcal{L})$. Note that we consider adaptive routing scenario, i.e., the routes of each flow are not determined *a priori*, which is more general than a fixed-routing scenario. In addition, we denote the source node and the destination node of a flow $c \in \mathcal{F}$ as $b(c)$ and $d(c)$, respectively.

We assume that the source node for flow c is always backlogged at the transport layer.[1] Let the scheduling parameter $\mu^c_{mn}(t)$ denote the link rate assignment of flow c for link (m,n) at time-slot t according to scheduling decisions and let $\mu^c_{s(c)b(c)}(t)$ denote the admitted rate of flow c from the packet generator of flow c to the source node, where $s(c)$ denotes the packet generator of flow c. It is clear that in any time-slot t, $\mu^c_{s(c)n}(t) = 0 \ \forall \ n \neq b(c)$. For simplicity of analysis, we assume only one packet can be transmitted over a link in one slot, so $(\mu^c_{mn}(t))$ takes values in $\{0,1\} \ \forall \ (m,n) \in \mathcal{L}$. We also assume that $\mu^c_{s(c)b(c)}(t)$ is bounded above by a constant $\mu_M \geq 1$

$$0 \leq \mu^c_{s(c)b(c)}(t) \leq \mu_M \qquad \forall \, c \in \mathcal{F}, \forall \, t \tag{1}$$

i.e., a source node can receive at most μ_M packets from the transport layer in any time-slot. To simplify the analysis, we prevent looping back to the source, i.e., we impose the following constraints:

$$\sum_{m \in \mathcal{N}} \left(\mu^c_{mb(c)}(t) \right) = 0 \qquad \forall \, c \in \mathcal{F}, \forall \, t. \tag{2}$$

We employ the node-exclusive model in our analysis, i.e., each node can communicate with at most one other node in a time-slot. Note that our model is extended to arbitrary interference models with arbitrary link capacities and fading channels in Section V-D.

We now specify the QoS requirements associated with each flow. The network imposes an average end-to-end delay threshold ρ_c for each flow c. The end-to-end delay period of a packet starts when the packet is admitted to the source node from the transport layer and ends when it reaches its destination. Note that the delay threshold is a time-averaged upper bound, not a deterministic one. In addition, each flow c requires a minimum data rate of a_c packets per time-slot.

B. Network Constraints and Approaches

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 the backlogged packet generator to the source node. We now model queue dynamics and network constraints in the multihop network. Let $U^c_n(t)$ be the backlog of the total amount of flow c packets waiting for transmission at node n. For a flow c, if $n = d(c)$, then $U^c_n(t) = 0 \ \forall \ t$; otherwise, the queue dynamics is as follows:

$$U^c_n(t+1) \leq \left[U^c_n(t) - \sum_{i:(n,i) \in \mathcal{L}} \mu^c_{ni}(t) \right]^+ + \sum_{j:(j,n) \in \mathcal{L}^c} \mu^c_{jn}(t), \quad \text{if } n \in \mathcal{N} \backslash d(c) \tag{3}$$

where the operator $[x]^+$ is defined as $[x]^+ = \max\{x, 0\}$. Note that in (3), we ensure that the actual number of packets transmitted for flow c from node n does not exceed its queue backlog since a feasible scheduling algorithm may not depend on 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 by the scheduling algorithm with respect to flow c. Note that (3) is an inequality since the arrival rates from neighbor nodes may be less than $\sum_j \mu^c_{jn}(t)$ if some neighbor node does

[1] Note that the constantly backlogged source is not necessarily a physical queue. It can be an application waiting for packet generation and admission, e.g., a variable rate multimedia encoder, and its end-to-end delay is the packet end-to-end delay in the network/data link layer. Thus, we can consider the backlogged source as a constantly backlogged packet generator.

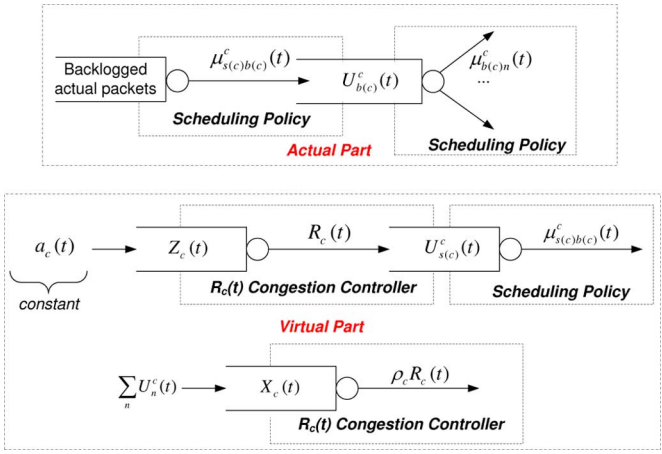

Fig. 1. Queue relationship diagram where for clarity we only show the three types of virtual queues and the actual queue at the source node.

TABLE I
UPDATE RULES OF EACH TYPE OF QUEUE, WHERE $R_c(t)$ IS DETERMINED BY A CONGESTION CONTROLLER AND $(\mu^c_{mn}(t))_{(m,n)\in\mathcal{L}^c}$ IS DETERMINED BY A SCHEDULING POLICY

	Instantaneous arrival rate	Instantaneous departure rate
$U^c_n(t)$	$\sum_{j:(j,n)\in\mathcal{L}^c} \mu^c_{jn}(t)$	$\sum_{i:(n,i)\in\mathcal{L}} \mu^c_{ni}(t)$
$U^c_{s(c)}(t)$	$R_c(t)$	$\mu^c_{s(c)b(c)}(t)$
$Z_c(t)$	a_c	$R_c(t)$
$X_c(t)$	$\sum_{n\in\mathcal{N}} U^c_n(t)$	$\rho_c R_c(t)$

not have sufficient number of packets to transmit. Since we employ the node-exclusive model, we have

$$0 \le \sum_{c\in\mathcal{F}} \left[\sum_{i:(n,i)\in\mathcal{L}} \mu^c_{ni}(t) + \sum_{j:(j,n)\in\mathcal{L}} \mu^c_{jn}(t) \right] \le 1 \qquad \forall\, n \in \mathcal{N}. \tag{4}$$

From (1) and (2), we also have

$$\sum_{j:(j,n)\in\mathcal{L}^c} \mu^c_{jn}(t) \le \mu_M, \qquad \text{if } n = b(c) \tag{5}$$

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

Now we construct three kinds of virtual queues, namely, virtual queue $U^c_{s(c)}(t)$ at transport layer, virtual service queue $Z_c(t)$ at sources, and virtual delay queue $X_c(t)$, to assist the development of our algorithm. We will show later that the minimum data rate requirement is achieved when $U^c_{s(c)}(t)$ and $Z_c(t)$ are stable; the average end-to-end delay constraint is satisfied if $U^c_{s(c)}(t)$ and $X_c(t)$ are stable. With update rules of each type of queue summarized in Table I and a general queue relationship illustrated in Fig. 1, we formally introduce the three virtual queue structures in the following.

1) To maintain the functionality of the traditional back-pressure algorithms while ensuring finite buffers, we construct a virtual queue $U^c_{s(c)}(t)$ for each flow c at the transport layer that acts as a weight to the "back-pressure" of actual queues, which will be further explained in Section IV. We denote the virtual input rate to the queue as $R_c(t)$ at the end of time-slot t, and we upper-bound $R_c(t)$ by μ_M. Let r_c denote the time-average of $R_c(t)$. We update the virtual queue as follows:

$$U^c_{s(c)}(t+1) = [U^c_{s(c)}(t) - \mu^c_{s(c)b(c)}(t)]^+ + R_c(t) \tag{6}$$

where the initial $U^c_{s(c)}(0) = 0$. 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 the time-average admitted rate μ_c of flow c satisfies

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

2) To satisfy the minimum data rate constraints, we construct a virtual queue $Z_c(t)$ associated with flow c as follows:

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

where the initial $Z_c(0) = 0$. Considering a_c as the arrival rate and $R_c(t)$ as the service rate, if queue $Z_c(t)$ is stable, we have $r_c \ge a_c$. Additionally, if $U^c_{s(c)}(t)$ is stable, then according to (7), the minimum data rate for flow c is achieved.

3) To satisfy the end-to-end delay constraints, we construct a virtual delay queue $X_c(t)$ for any given flow c as follows:

$$X_c(t+1) = [X_c(t) - \rho_c R_c(t)]^+ + \sum_{n\in\mathcal{N}} U^c_n(t) \tag{9}$$

where the initial $X_c(0) = 0$. Considering the packets kept in the network in time-slot t, i.e., $\sum_{n\in\mathcal{N}} U^c_n(t)$, as the arrival rate and $\rho_c R_c(t)$ as the service rate, and according to queueing theory, if queue $X_c(t)$ is stable, we have

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n\in\mathcal{N}} U^c_n(\tau) \le \rho_c \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R_c(\tau) = \rho_c r_c.$$

Furthermore, if $U^c_{s(c)}(t)$ is stable, then according to (7), we have

$$\frac{1}{\mu_c} \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n\in\mathcal{N}} U^c_n(\tau) \le \rho_c. \tag{10}$$

In addition, by Little's Theorem, (10) ensures that the average end-to-end delay of flow c is less than or equal to the threshold ρ_c with probability (w.p.) 1.

If queues $U^c_n(t)$ and the three virtual queues are stable for all nodes and flows, we know that the network is *stable* (i.e., all queues at all nodes are stable) and the average end-to-end delay constraint and minimum data rate requirement are achieved. Specifically, the network is stable and constraints are met if

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{X_c(\tau)\} < \infty \qquad \forall\, c$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{U^c_n(\tau)\} < \infty \qquad \forall\, n \in \mathcal{N} \cup \{s(c) : c \in \mathcal{F}\}$$

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Z_c(\tau)\} < \infty \qquad \forall\, c.$$

We note that the decision variable $R_c(t)$ represents the arrival rate to the virtual queue $U^c_{s(c)}(t)$ and serves as the service rate in virtual queues $Z_c(t)$ and $X_c(t)$. The algorithm proposed in Section IV provides an $R_c(t)$ congestion controller that determines the value of $R_c(t)$ for each time-slot t. The decision variable $\mu^c_{s(c)b(c)}(t)$ represents both the physical packet admission rate (i.e., the arrival rate to the actual source queue $U^c_{b(c)}(t)$) and the virtual service rate of $U^c_{s(c)}(t)$. Our proposed algorithm provides a scheduling policy that determines the value of $\mu^c_{s(c)b(c)}(t)$ joint with the assignment of the link service rate $\mu^c_{mn}(t)$, $(m,n) \in \mathcal{L}$. The average end-to-end packet delay in the network layer is only affected by the number of packets in the actual packet queues. Even if a virtual queue has a large number of virtual packets, they are not propagated in the network through an actual packet queue.

Now we define the capacity region of the considered multihop network. An arrival rate vector (z_c) is called *admissible* if there exists some scheduling algorithm (without congestion control) under which the node queue backlogs (not including virtual queues) are stable. We denote Λ to be the capacity region consisting of all admissible (z_c), i.e., Λ consists of all feasible rates stabilizable by some scheduling algorithm without considering QoS requirements (i.e., delay constraints and minimum data rate constraints). To assist the analysis in Sections IV–VII, we let $(r^*_{\epsilon,c})$ denote a solution to the following optimization problem:

$$\begin{aligned} \max_{(r_c):(r_c+\epsilon)\in\Lambda} \quad & \sum_{c\in\mathcal{F}} r_c \\ \text{s.t.} \quad & r_c \geq a_c \qquad \forall\, c \in \mathcal{F} \end{aligned}$$

where ϵ is a positive number that can be chosen arbitrarily small. For simplicity of analysis, we assume that (a_c) is in the interior of Λ, and without loss of generality, we assume that there exists $\epsilon' > 0$ such that $r^*_{\epsilon,c} \geq a_c + \epsilon'\ \forall\, c \in \mathcal{F}$. According to [26], we have

$$\lim_{\epsilon\to 0} \sum_{c\in\mathcal{F}} r^*_{\epsilon,c} = \sum_{c\in\mathcal{F}} r^*_c$$

where (r^*_c) is a solution to the following optimization:

$$\begin{aligned} \max_{(r_c):(r_c)\in\Lambda} \quad & \sum_{c\in\mathcal{F}} r_c \\ \text{s.t.} \quad & r_c \geq a_c \qquad \forall\, c \in \mathcal{F}. \end{aligned}$$

IV. Control Scheduling Algorithm for Multihop Wireless Networks

Now we propose a control and scheduling algorithm ***ALG*** for the introduced multihop model so that ***ALG*** stabilizes the network and satisfies the delay constraint and minimum data rate constraint. Given ϵ, the proposed ***ALG*** can achieve a throughput arbitrarily close to $\sum_{c\in\mathcal{F}} r^*_{\epsilon,c}$, under certain conditions related to delay constraints that will be later given in Theorem 1.

The optimal algorithm ***ALG*** consists of two parts: a congestion controller of $R_c(t)$ and a joint packet admission, routing, and scheduling policy. We propose and analyze the algorithm in Sections IV-A and IV-B.

A. Algorithm Description and Analysis

Let $q_M \geq \mu_M$ be a control parameter of the algorithm, which we will later show to be a deterministic upper bound on the actual queues. We first propose a congestion controller for the input rate of virtual queues at transport layer.

1) *Congestion Controller of $R_c(t)$:*

$$\min_{0\leq R_c(t)\leq \mu_M} R_c(t)\left(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M} - X_c(t)\rho_c - Z_c(t) - V\right) \tag{11}$$

where $V > 0$ is a control parameter. Specifically, when $\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t) - X_c(t)\rho_c - Z_c(t) - V > 0$, $R_c(t)$ is set to zero; otherwise, $R_c(t) = \mu_M$. It will be later shown in Theorem 1 that when V is large, ***ALG*** approaches optimal throughput. We will further analyze the effect of parameter V on the convergence behavior of the virtual queues with numerical results provided in Section VI-A.

After performing the congestion control, we perform the following joint policy for packet admission, routing, and scheduling (abbreviated as *scheduling policy*).

2) *Scheduling Policy:* In each time-slot, with the constraints of the underlying interference model as described in Section III including (1), (2), and (4), the network solves the following optimization problem:

$$\begin{aligned} \max_{(\mu^c_{mn}(t))} \quad & \sum_{m,n} \mu^{c^*_{mn}(t)}_{mn}(t) w_{mn}(t) \\ \text{s.t.} \quad & \mu^c_{mn}(t) = 0 \qquad \forall\, c \neq c^*_{mn}(t),\ \forall\,(m,n) \in \mathcal{L}^c \\ & \mu^c_{mn}(t) = 0 \qquad \text{if } n = s(c),\ \forall\, c \in \mathcal{F} \end{aligned} \tag{12}$$

where $c^*_{mn}(t)$ and $w_{mn}(t)$ are defined as follows:

$$c^*_{mn}(t) = \arg\max_{c\in\mathcal{F}} w^c_{mn}(t)$$

$$w_{mn}(t) = \left[\max_{c\in\mathcal{F}} w^c_{mn}(t)\right]^+$$

with weight assignment as follows:

$$w^c_{mn}(t) = \begin{cases} \frac{U^c_{s(c)}(t)}{q_M}\left[U^c_m(t) - U^c_n(t)\right], & \text{if } (m,n)\in\mathcal{L} \text{ and } n \neq b(c) \\ \frac{U^c_{s(c)}(t)}{q_M}\left[q_M - \mu_M - U^c_{b(c)}(t)\right], & \text{if } (m,n) = (s(c), b(c)) \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

In addition, when $w_{mn}(t) = 0$, without loss of optimality, we set $\mu^c_{mn}(t) = 0\ \forall\, c \in \mathcal{F}$ to maximize (12).

Note that $\mathcal{L} \cup \{(s(c), b(c)) : c \in \mathcal{F}\}$ forms the (m,n) pairs in $(\mu^c_{mn}(t))$ over which the optimization (12) is performed. Thus, the optimization is a typical Maximum Weight Matching (MWM) problem. We first decouple flow scheduling from the MWM. Specifically, for each pair (m,n), the flow $c^*_{mn}(t)$ is fixed as the candidate for transmission. We then assign the weight as $w_{mn}(t)$. Note also that although similar product form of the weight assignment (13) has been utilized in [24] and [25], no virtual queues are involved there. Meanwhile in ***ALG***, we assign weights as a product of weighted

virtual queue backlog ($\frac{U^c_{s(c)}(t)}{q_M}$) and the actual back-pressure in an aim to shift the burden of the actual queue backlog to the virtual backlog.

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

Proposition 1: Employing ***ALG***, each queue backlog in the network has a deterministic worst-case bound

$$U^c_n(t) \le q_M \qquad \forall\, t, \forall\, n \in \mathcal{N}, \forall\, c \in \mathcal{F}. \tag{14}$$

Proof: We use mathematical induction on time-slot in the proof. When $t = 0$, $U^c_n(0) = 0 \le q_M\ \forall\, n, c$. In the induction hypothesis, we suppose in time-slot t we have $U^c_n(t) \le q_M\ \forall\, n, c$. In the induction step, for any given $n \in \mathcal{N}$ and $c \in \mathcal{F}$, we consider two cases as follows.

1) We first consider the case when $n = b(c)$, i.e., when n is the source node of flow c. Since $U^c_n(t) \le q_M$ from the induction hypothesis, we further consider two subcases.
   - In the first subcase, $U^c_{b(c)}(t) \le q_M - \mu_M$. Then, according to the queue dynamics (3) and the inequality (5), $U^c_{b(c)}(t+1) \le U^c_{b(c)}(t) + \mu_M \le q_M$.
   - In the second subcase, $q_M - \mu_M < U^c_{b(c)}(t) \le q_M$. According to the weight assignment (13), we have $w^c_{s(c)b(c)}(t) < 0$, which leads to $\mu^c_{s(c)b(c)}(t) = 0$. Hence, $U^c_{b(c)}(t+1) \le U^c_{b(c)}(t) \le q_M$ by (2) and (3).
2) In the second case, $n \ne b(c)$, i.e., n is not the source node of flow c. Similar to the first case, we further consider the following two subcases.
   - In the first subcase, $U^c_n(t) < q_M$. Then, since we employ node-exclusive model, $U^c_n(t+1) \le U^c_n(t) + 1 \le q_M$ by (3) and (4).
   - In the second subcase, $U^c_n(t) = q_M$. According to the weight assignment (13), we have $w^c_{mn}(t) \le 0\ \forall\, m : (m,n) \in \mathcal{L}$. Now, for any given node $m : (m,n) \in \mathcal{L}$, we have the following.
     i) If $c \ne c^*_{mn}(t)$, then by (12), $\mu^c_{mn}(t) = 0$.
     ii) Otherwise, $c = c^*_{mn}(t)$, which induces $w_{mn}(t) = [w^c_{mn}(t)]^+ = 0$, and by the scheduling policy, $\mu^c_{mn}(t) = 0$.
     Hence, $\mu^c_{mn}(t) = 0\ \forall\, m : (m,n) \in \mathcal{L}$, and $U^c_n(t+1) \le U^c_n(t) = q_M$ by the queue dynamics (3).

   The above analysis holds for any given $n \in \mathcal{N}$ and $c \in \mathcal{F}$. Therefore, 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 that

$$q_M > \frac{2N - 1 + \mu_M^2}{2\epsilon} + \mu_M \quad \text{and} \quad \rho_c > \frac{N q_M}{r^*_{\epsilon,c}} \qquad \forall\, c \in \mathcal{F} \tag{15}$$

ALG can achieve a throughput

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{c\in\mathcal{F}} \mathbb{E}\left\{\mu^c_{s(c)b(c)}(\tau)\right\} \ge \sum_{c\in\mathcal{F}} r^*_{\epsilon,c} - \frac{B}{V} \tag{16}$$

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

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

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{c\in\mathcal{F}} \mathbb{E}\left\{U^c_{s(c)}(\tau) + X_c(\tau) + Z_c(\tau)\right\} \le \frac{B'}{\delta} \tag{17}$$

where $B' \triangleq B + VB_R$, with B_R and δ constant positive numbers given in Section IV-B.

Remark 1 (Network Stability): The inequalities (14) from Proposition 1 and (17) from Theorem 1 indicate that ***ALG*** stabilizes the actual and virtual queues. As an immediate result, ***ALG*** stabilizes the network and satisfies the average end-to-end delay constraint and the minimum data rate requirement. In addition, Proposition 1 states that the actual queues are *deterministically* bounded by q_M, which ensures finite buffer sizes for all queues in the network, including those in source nodes.

Remark 2 (Optimal Throughput and Delay Analysis): Since $(U^c_{s(c)}(t))$ are stable, the inequality (16) gives a lower bound on the throughput that ***ALG*** can achieve. Given some $\epsilon > 0$, since B is independent of V, (16) also ensures that ***ALG*** can achieve a throughput arbitrarily close to $\sum_{c\in\mathcal{F}} r^*_{\epsilon,c}$. When ϵ tends to 0, ***ALG*** can achieve a throughput arbitrarily close to the optimal value $\sum_{c\in\mathcal{F}} r^*_c$ with the tradeoff in queue backlog upper bound q_M and the delay constraints (ρ_c), both of which are lower-bounded by the reciprocal terms of ϵ as shown in (15) in Theorem 1. In other words, the average end-to-end delay bound is of order $O(\frac{1}{\epsilon})$. We note that similar tradeoff between delay and throughput (though without finite buffer and end-to-end delay guarantees) has also been observed in earlier works such as [4] and [36].

In ***ALG***, the control parameter V, which is typically chosen to be large, does not affect the actual queue backlog upper bound or the average end-to-end delay bound. However, a larger V increases the upper bound of the virtual queue backlogs [from (17)] and results in a slower convergence time of the virtual queues. Note that with a finite V, all virtual queues are stable nevertheless from (17). More details on the effect of V are presented in Section VI-A with numerical results. In comparison, in the algorithm proposed in [25], the authors show that the internal buffer size is deterministically bounded with order $O(\frac{1}{\epsilon})$, but at the expense of the buffer occupancy at source nodes, which is of order $O(V)$, where V has to be large enough for their algorithm to approach $\sum_{c\in\mathcal{F}} r^*_{\epsilon,c}$. This design assumes an infinite buffer size at source nodes and typically results in congestion at the source nodes as shown in the simulation results in [25], which further induces an unguaranteed and large average end-to-end delay. Moreover, one can expect that there are no buffer-size guarantees for single-hop flows by employing the algorithm in [25]. In contrast, in our proposed ***ALG***, we shift "the burden of V" from actual queues to virtual queues and ensure that the average end-to-end delay constraints are satisfied with finite buffer sizes for all actual packet queues.

Remark 3 (Implementation Issues): To update the virtual queue $X_c(t)$ and perform the $R_c(t)$ congestion controller at the transport layer, the queue backlog information of flow c is crucial. This information can be collected back to the source

node by piggybacking it on ACK from each node. In order to account for such delay of queue backlog information, the $R_c(t)$ congestion controller (11) of the algorithm can employ delayed queue backlog of $X_c(t)$. Similarly, delayed queue backlog information of $U^c_{s(c)}(t)$ can be employed at the weight assignment (13) of the scheduling policy. The modified algorithm and its validity are further discussed in Section V-C. By employing delayed queue backlog information, we can extend the algorithm to distributed implementation in much the same way as in [7] and [10] to achieve a fraction of the optimal throughput. In order to achieve a throughput arbitrarily close to the optimal value with distributed implementation, we can employ random access techniques [31], [32] in the scheduling policy with fugacities [33][2] chosen as $\exp\left\{\frac{\alpha \bar{U}^c_{s(c)}(t)[U^c_m(t)-U^c_n(t)]^+}{q_M}\right\}$ for each link $(m,n)\in\mathcal{L}$, where $\bar{U}^c_{s(c)}(t)$ is a local estimate (e.g., delayed information) of $U^c_{s(c)}(t)$ and α a positive weight. It can be shown that the distributed algorithm can still achieve an average end-to-end delay of order $O(\frac{1}{\epsilon})$ with the time-scale separation assumption [30], [31].[3] A variation of such distributed implementation in single-hop networks can be found in our recent work [34].

We prove Theorem 1 in Section IV-B.

B. Proof of Theorem 1

Before we proceed, we present the following lemma that 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 algorithm STAT that stabilizes the network with input rate vector $(\mu^{\rm STAT}_{s(c)b(c)}(t))$ and scheduling parameters $(\mu^{c,\rm STAT}_{mn}(t))$ independent of queue backlogs, such that the expected admitted rates are

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

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

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

Note that it is not necessary for the randomized algorithm STAT to satisfy the average end-to-end delay constraints. Similar formulations of STAT and their proofs have been given in [4] and [5], so we omit the proof of Lemma 1 for brevity.

Remark 4: Note that (θ_c) can take values as $(r^*_{\epsilon,c})$ or $(r^*_{\epsilon,c}+\epsilon)$ or $(r^*_{\epsilon,c}-\frac{1}{2}\epsilon')$, where we recall $(r^*_{\epsilon,c}+\epsilon)\in\Lambda$ and $r^*_{\epsilon,c}\geq a_c+\epsilon'\ \forall\, c\in\mathcal{F}$. According to the STAT algorithm in Lemma 1, we assign the input rates of the virtual queues at transport layer as $R^{\rm STAT}_c(t)=\mu^{c,\rm STAT}_{s(c)b(c)}(t)$. Thus, we also have $\mathbb{E}\{R^{\rm STAT}_c(t)\}=\theta_c$, with the time-average of $R^{\rm STAT}_c(t)$ satisfying: $r^{\rm STAT}_c=\theta_c$.

[2]Fugacity [33] determines the transmission probability of a link in the CSMA framework (also referred to as Glauber dynamics) introduced in [31] and [32].

[3]Note that the random access works cited above either do not provide delay guarantees or are not readily extended to multihop settings.

In addition, according to the aforementioned assignment, the virtual queues $U^c_{s(c)}(t)$ under STAT are bounded above by μ_M.

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

$$L(\mathbf{Q}(t))=\frac{1}{2}\left\{\sum_{c\in\mathcal{F}}\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t)^2+\sum_{c\in\mathcal{F}}X_c(t)^2+\sum_{c\in\mathcal{F}}Z_c(t)^2+\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{1}{q_M}U^c_n(t)^2U^c_{s(c)}(t)\right\}. \quad (18)$$

We note that the structure of the Lyapunov function is different from the standard quadratic one (e.g., [4]), in that there is a virtual queue backlog multiplied to the quadratic term of the actual queue backlog, i.e., $\frac{U^c_{s(c)}(t)}{q_M}U^c_n(t)^2$. Such structure leads to the structure of the scheduler (13) in ***ALG***, as we shall prove in the following.

We denote the Lyapunov drift by

$$\Delta(t)=\mathbb{E}\{L(\mathbf{Q}(t+1))-L(\mathbf{Q}(t))|\mathbf{Q}(t)\}. \quad (19)$$

From the queue dynamics (3) and (6), we have

$$\begin{aligned}&\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{1}{q_M}U^c_n(t+1)^2U^c_{s(c)}(t+1)\\&\leq\sum_{c\in\mathcal{F}}\frac{1}{q_M}\left(R_c(t)+U^c_{s(c)}(t)\right)\sum_{n\in\mathcal{N}}U^c_n(t+1)^2\\&\leq\mu_Mq_MNK+\sum_{c\in\mathcal{F}}\frac{1}{q_M}U^c_{s(c)}(t)\sum_{n\in\mathcal{N}}\left\{U^c_n(t)^2\right.\\&\quad+\left(\sum_{i:(n,i)\in\mathcal{L}}\mu^c_{ni}(t)\right)^2+\left(\sum_{j:(j,n)\in\mathcal{L}^c}\mu^c_{jn}(t)\right)^2\\&\quad\left.-2U^c_n(t)\left(\sum_i\mu^c_{ni}(t)-\sum_j\mu^c_{jn}(t)\right)\right\}\end{aligned} \quad (20)$$

where we recall that $R_c(t)\leq\mu_M$.

From (20), we have

$$\begin{aligned}&\frac{1}{2}\left(\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{1}{q_M}\left(U^c_n(t+1)^2U^c_{s(c)}(t+1)-U^c_n(t)^2U^c_{s(c)}(t)\right)\right)\\&\leq\frac{1}{2}\sum_{c\in\mathcal{F}}\frac{\left(2N-1+\mu_M^2\right)U^c_{s(c)}(t)}{q_M}+\frac{1}{2}NKq_M\mu_M\\&\quad-\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{U^c_n(t)U^c_{s(c)}(t)}{q_M}\left(\sum_{j:(n,j)\in\mathcal{L}}\mu^c_{nj}(t)-\sum_{i:(i,n)\in\mathcal{L}^c}\mu^c_{in}(t)\right)\end{aligned} \quad (21)$$

where we employ the fact deduced from (4) and (5) that $\sum_i\mu^c_{ni}(t)\leq 1$ and $\sum_j\mu^c_{jn}(t)\leq 1$ when $n\neq b(c)$ and $\sum_j\mu^c_{jn}(t)\leq\mu_M$ when $n=b(c)$. Note that we use the summation index i and j interchangeably for convenience of analysis.

By squaring both sides of the queue dynamics (6), (8), and (9) and by employing (21), we obtain the following inequality on the Lyapunov drift (19):

$$\begin{aligned}
&\Delta(t) - V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}\\
&\le B + \sum_{c\in\mathcal{F}}\mathbb{E}\left\{R_c(t)\left(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M}\right.\right.\\
&\qquad\left.\left.-X_c(t)\rho_c - Z_c(t) - V\right)|\mathbf{Q}(t)\right\}\\
&+Nq_M\sum_{c\in\mathcal{F}}X_c(t)+\sum_{c\in\mathcal{F}}a_cZ_c(t)+\frac{1}{2}\sum_{c\in\mathcal{F}}\frac{(2N-1+\mu_M^2)U^c_{s(c)}(t)}{q_M}\\
&\quad-\mathbb{E}\left\{\frac{q_M-\mu_M}{q_M}\sum_{c\in\mathcal{F}}U^c_{s(c)}(t)\mu^c_{s(c)b(c)}(t)\right.\\
&\qquad+\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}}\frac{U^c_n(t)U^c_{s(c)}(t)}{q_M}\\
&\qquad\left.\times\left(\sum_{j:(n,j)\in\mathcal{L}}\mu^c_{nj}(t)-\sum_{i:(i,n)\in\mathcal{L}^c}\mu^c_{in}(t)\right)|\mathbf{Q}(t)\right\}.
\end{aligned}\tag{22}$$

We can rewrite the last term of the right-hand side (RHS) of (22) by simple algebra as

$$\begin{aligned}
&-\mathbb{E}\left\{\sum_{c\in\mathcal{F}}\sum_{(m,n)\in\mathcal{L}}\mu^c_{mn}(t)\frac{U^c_{s(c)}(t)}{q_M}\left(U^c_m(t)-U^c_n(t)\right)\right.\\
&\quad\left.+\sum_{c\in\mathcal{F}}\mu^c_{s(c)b(c)}(t)\frac{U^c_{s(c)}(t)}{q_M}\left(q_M-\mu_M-U^c_{b(c)}(t)\right)|\mathbf{Q}(t)\right\}.
\end{aligned}\tag{23}$$

Then, the second term and the last term of the RHS of (22) are minimized by the congestion controller (11) and the scheduling policy (12), respectively, over a set of feasible algorithms including the stationary randomized algorithm STAT introduced in Lemma 1 and Remark 4. We can substitute into the second term of RHS of (22) a stationary randomized algorithm with admitted arrival rate vector $(r^*_{\epsilon,c})$ and into the last term with a stationary randomized algorithm with admitted arrival rate vector $(r^*_{\epsilon,c}+\epsilon)$. Thus, we have

$$\begin{aligned}
&\Delta(t)-V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}\\
&\le B-V\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}-\sum_{c\in\mathcal{F}}\frac{U^c_{s(c)}(t)}{q_M}\left(\epsilon(q_M-\mu_M)-\frac{2N-1+\mu_M^2}{2}\right)\\
&\quad-\sum_{c\in\mathcal{F}}\left(r^*_{\epsilon,c}-a_c\right)Z_c(t)-\sum_{c\in\mathcal{F}}\left(\rho_cr^*_{\epsilon,c}-Nq_M\right)X_c(t).
\end{aligned}\tag{24}$$

When (15) holds, we can find $\epsilon_1>0$ such that $\epsilon_1\le\rho_cr^*_{\epsilon,c}-Nq_M\ \forall\, c\in\mathcal{F}$ and $\epsilon_1\le\frac{\epsilon(q_M-\mu_M)-\frac{2N-1+\mu_M^2}{2}}{q_M}$. Recall that ϵ' is defined such that $r^*_{\epsilon,c}\ge a_c+\epsilon'\ \forall\, c\in\mathcal{F}$. Thus, we have

$$\begin{aligned}
&\Delta(t) - V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}\\
&\quad\le B-\delta\sum_{c\in\mathcal{F}}(X_c(t)+U^c_{s(c)}(t)+Z_c(t))-V\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}
\end{aligned}\tag{25}$$

where $\delta\triangleq\min\{\epsilon_1,\epsilon'\}$.

We take the expectation with respect to the distribution of $\mathbf{Q}$ on both sides of (25) and take the time average on $\tau=0,\ldots,t-1$, which leads to

$$\begin{aligned}
&\frac{1}{t}\mathbb{E}\{L(\mathbf{Q}(t))\}-\frac{V}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(\tau)\}\\
&\quad\le B-V\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}-\frac{\delta}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{X_c(\tau)+U^c_{s(c)}(\tau)+Z_c(\tau)\}.
\end{aligned}\tag{26}$$

Since $\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_c\mathbb{E}\{R_c(\tau)\}$ is bounded above (say, by a constant B_R with $B_R\le K\mu_M$) and $\mathbb{E}\{L(\mathbf{Q}(t))\}$ is nonnegative, by taking limsup of t on both sides of (26), we have

$$\begin{aligned}
&\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{X_c(\tau)+U^c_{s(c)}(\tau)+Z_c(\tau)\}\\
&\quad\le\frac{B}{\delta}+\frac{V}{\delta}\left[\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(\tau)\}-\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}\right]\\
&\quad\le\frac{B'}{\delta}
\end{aligned}\tag{27}$$

which proves (17). Similarly, by taking liminf of t on both sides of (26) and employing (7), we can prove (16).

V. Further Discussions

A. Suboptimal Algorithms

Solving MWM optimization problem can be NP-hard depending on the underlying interference model as indicated in [27]. In this section, we introduce a group of suboptimal algorithms that aim to achieve at least a γ fraction of the optimal throughput. We denote the scheduling parameters of suboptimal algorithms by $(\mu^{c,\mathrm{SUB}}_{mn}(t))$. For convenience, we also denote the scheduling parameters of ***ALG*** by $(\mu^{c,\mathrm{OPT}}_{mn}(t))$. Algorithms are called *suboptimal* if the scheduling parameters $(\mu^{c,\mathrm{SUB}}_{mn}(t))$ satisfy the following:

$$\sum_{m,n}\mu^{c^*_{mn}(t),\mathrm{SUB}}_{mn}(t)w_{mn}(t)\ge\gamma\sum_{m,n}\mu^{c^*_{mn}(t),\mathrm{OPT}}_{mn}(t)w_{mn}(t)\tag{28}$$

where $\gamma\in(0,1)$ is constant and we recall that $c^*_{mn}(t)$ and $w_{mn}(t)$ are defined in Section IV-A. In addition, the congestion controller of suboptimal algorithms is the same as that of ***ALG*** (11).

Following the same analysis of ***ALG***, Proposition 1 holds for suboptimal algorithms, i.e., the queue backlogs are bounded above by q_M, and we derive the following theorem.

Theorem 2: Given that

$$q_M > \frac{2N-1+\mu_M^2}{2\gamma\epsilon}+\mu_M \quad \text{and} \quad \rho_c > \frac{Nq_M}{\gamma r^*_{\epsilon,c}} \qquad \forall\, c\in\mathcal{F}$$
$$\exists \epsilon_2>0 \qquad \text{s.t. } \gamma r^*_{\epsilon,c} \geq a_c+\epsilon_2 \qquad \forall\, c\in\mathcal{F} \tag{29}$$

a suboptimal algorithm ensures that the virtual queues have a time-averaged bound

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\left\{U^c_{s(c)}(\tau)+X_c(\tau)+Z_c(\tau)\right\}\leq\frac{\bar{B}}{\bar{\delta}} \tag{30}$$

where $\bar{B}\triangleq B+\gamma V B_R$ and $\bar{\delta}$ is a positive constant associated with condition (29). In addition, a suboptimal algorithm can achieve a throughput

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\left\{\mu^c_{s(c)b(c)}(\tau)\right\}\geq\gamma\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}-\frac{B}{V}. \tag{31}$$

The proof of Theorem 2 follows that of Theorem 1, where we have employed (28) to (22) and (23), and is available in [37].

Remark 5: From Theorem 2, given an arbitrarily small ϵ, we show that a suboptimal algorithm can *at least* achieve a throughput arbitrarily close to a fraction γ of the optimal results $\sum_{c\in\mathcal{F}} r^*_{\epsilon,c}$. Suboptimal algorithms include the well-known Greedy Maximal Matching (GMM) algorithm [28] with $\gamma=\frac{1}{2}$ as well as the solutions to the maximum weighted independent set (MWIS) optimization problem such as GWMAX and GWMIN proposed in [29] with $\gamma=\frac{1}{\Delta}$, where Δ is the maximum node degree of the network topology G. The delay bound and throughput tradeoff in Theorem 1 still hold in Theorem 2.

B. Arbitrary Arrival Rates at Transport Layer

Note that in the previous model description, we assumed that the flow sources are constantly backlogged, that is, the congestion controller (11) can always guarantee $R_c(t)=\mu_M$ when $\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t)-X_c(t)\rho_c-Z_c(t)-V\leq 0$. In this section, we present an optimal algorithm when the flows have arbitrary arrival rates at the transport layer.

Let $A_c(t)$ denote the arrival rate of flow c packets at the beginning of the time-slot t at the transport layer. We assume that $A_c(t)$ is i.i.d. with respect to t with mean λ_c. For simplicity of analysis, we assume (λ_c) to be in the exterior of the capacity region Λ so that a congestion controller is needed, and we assume that $A_c(t)$ is bounded above by $\mu_M\ \forall\, c\in\mathcal{F}$.[4] Let $L_c(t)$ denote the backlog of flow c data at the transport layer, which is updated as follows:

$$L_c(t+1)=\min\left\{\left[L_c(t)+A_c(t)-\mu^c_{s(c)b(c)}(t)\right]^+, L_M\right\} \tag{32}$$

where $L_M\geq 0$ is the buffer size for flow c at the transport layer. Note that we have $L_M=0$ and $L_c(t)=0$ if there is no buffer for flow c at the transport layer.

[4]Note that our analysis also works for the case when $A_c(t)$ is bounded above by some constant $A_M\ \forall\, c\in\mathcal{F}$, where $A_M\geq\mu_M$.

Following the idea introduced in [4], we construct a virtual queue $Y_c(t)$ and an auxiliary variable $v_c(t)$ for each virtual input rate $R_c(t)$, with queue dynamics for $Y_c(t)$ as follows:

$$Y_c(t+1)=[Y_c(t)-R_c(t)]^+ + v_c(t) \tag{33}$$

where initially we have $Y_c(0)=0$. The intuition is that $v_c(t)$ serves as the function of $R_c(t)$ in congestion controller (11) and we note that when $Y_c(t)$ is stable, we have $r_c\geq v_c$, where v_c is the time-average rate for $v_c(t)$, recalling that r_c is the time average rate for $R_c(t)$. Thus, when $Y_c(t)$ and $U^c_{s(c)}(t)$ are stable, if we can ensure the value $\sum_c v_c$ is arbitrarily close to the optimal value $\sum_c r^*_{\epsilon,c}$, then so is the throughput $\sum_c \mu_c$ since $\mu_c\geq r_c\geq v_c$.

Now we provide the optimal algorithm for arbitrary arrival rates at the transport layer.

1) *Congestion Controller:*

$$\min_{0\leq v_c(t)\leq\mu_M} \quad v_c(t)(\eta Y_c(t)-V) \tag{34}$$

$$\min_{R_c(t)} \quad R_c(t)\Big(\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t) - \eta Y_c(t)-X_c(t)\rho_c-Z_c(t)\Big)$$
$$\text{s.t.} \quad 0\leq R_c(t)\leq\min\{L_c(t)+A_c(t),\mu_M\} \tag{35}$$

where $\eta>0$ is a weight associated with the virtual queue $Y_c(t)$. Note that (34) and (35) can be solved independently. Specifically, when $\eta Y_c(t)-V\geq 0$, $v_c(t)$ is set to zero; otherwise, $v_c(t)=\mu_M$. When $\frac{q_M-\mu_M}{q_M}U^c_{s(c)}(t)-\eta Y_c(t)-X_c(t)\rho_c-Z_c(t)\geq 0$, $R_c(t)$ is set to zero; otherwise, $R_c(t)=\min\{L_c(t)+A_c(t),\mu_M\}$.

2) *Scheduling Policy:* The scheduling algorithm is the same as that of ***ALG*** provided in Section IV-B, except for the updated constraints: $0\leq\mu^c_{s(c)b(c)}(t)\leq\min\{L_c(t)+A_c(t),\mu_M\}$.
Since the scheduling policy is not changed, Proposition 1 still holds. We present the following theorem for the performance of the algorithm.

Theorem 3: Given that

$$q_M > \frac{2N-1+\mu_M^2}{2\epsilon}+\mu_M \quad \text{and} \quad \rho_c > \frac{Nq_M}{r^*_{\epsilon,c}} \qquad \forall\, c\in\mathcal{F}$$

the algorithm ensures that the virtual queues have a time-averaged bound

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\Big\{U^c_{s(c)}(\tau)+X_c(\tau)+Z_c(\tau)+Y_c(\tau)\Big\}\leq\frac{B_2}{\delta'}$$

where $B_2\triangleq B+K\eta\mu_M^2+VB_R$ and δ' is a constant positive number. In addition, the algorithm can achieve a throughput

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\left\{\mu^c_{s(c)b(c)}(\tau)\right\}\geq\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}-\frac{B_1}{V}$$

where $B_1\triangleq B+K\eta\mu_M^2$.

Proof: The proof is provided in Appendix A. ∎

Theorem 3 shows that optimality is preserved and $O(\frac{1}{\epsilon})$ delay scaling is kept.

C. Employing Delayed Queue Backlog Information

Recall that in ***ALG***, congestion controller (11) is performed at the transport layer and link weight assignment in (13) is performed locally at each link. Thus, in order to account for the propagation delay of queue information, we employ delayed queue backlog of $(X_c(t))$ in (11) and employ delayed queue backlog of $(U^c_{s(c)}(t))$ for links in $\mathcal{L}$ in (13). Specifically, we rewrite (11) in ***ALG*** as

$$\min R_c(t)\left(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M}-X_c(t-T)\rho_c-Z_c(t)-V\right) \tag{36}$$

where T is an integer number that is larger than the maximum propagation delay from a source to a node, and we rewrite (13) as

$$w^c_{mn}(t) = \begin{cases} \frac{U^c_{s(c)}(t-T)}{q_M}\left[U^c_m(t)-U^c_n(t)\right], & \text{if } (m,n)\in\mathcal{L}, \\ \frac{U^c_{s(c)}(t)}{q_M}\left[q_M-\mu_M-U^c_{b(c)}(t)\right], & \text{if } (m,n)=(s(c),b(c)) \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

Proposition 1 still holds, and we present a theorem for the scheduling algorithm using delayed queue backlog information, which maintains the throughput optimality and $O(\frac{1}{\epsilon})$ scaling in delay bound.

Theorem 4: Given that

$$q_M > \frac{2N-1+\mu_M^2}{2\epsilon}+\mu_M \quad \text{and} \quad \rho_c > \frac{Nq_M}{r^*_{\epsilon,c}} \qquad \forall\, c\in\mathcal{F}$$

the algorithm ensures that the virtual queues have a time-averaged bound

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\left\{U^c_{s(c)}(\tau)+X_c(\tau)+Z_c(\tau)\right\}\le\frac{B_4}{\delta}$$

where $B_4 \triangleq B_3 + VB_R$ and $B_3 \triangleq B + KN\mu_M T + Nq_M T\mu_M\rho_c + K\rho_c^2\mu_M^2 T$. In addition, the algorithm can achieve a throughput

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\left\{\mu^c_{s(c)b(c)}(\tau)\right\}\ge\sum_{c\in\mathcal{F}}r^*_{\epsilon,c}-\frac{B_3}{V}.$$

The proof of Theorem 4 follows the analysis in Section IV-B and is given in [37]. On employing delayed queue backlogs, we can extend the centralized optimization problem (12) to distributed implementations with methods introduced in Remark 3.

D. Arbitrary Link Capacities and Arbitrary Interference Models With Fading Channels

Recall that in the model description in Section III, the link capacity is assumed constant (one packet per slot) and node-exclusive model is employed. In this section, we extend the model to arbitrary link capacities and arbitrary interference models with fading channels of finite channel states. Thus, instead of (4), we have $(\mu^c_{mn}(t))_{(m,n)\in\mathcal{L}} \in I(t)$, where $I(t)$ is the feasible activation set for time-slot t determined by the underlying interference model and current channel states, with link capacity constraints $\sum_{c\in\mathcal{F}}\mu^c_{mn}(t)\le l_{mn}$, where l_{mn} is the arbitrarily chosen link capacity for a link $(m,n)\in\mathcal{L}$. We define $l_n \triangleq \max_{(\mu^c_{mn}(t))\in I(t)}\sum_{c\in\mathcal{F}}\sum_{m:(m,n)\in\mathcal{L}}\mu^c_{mn}(t)$. Note that it is clear that $l_n \le \sum_{m:(m,n)\in\mathcal{L}} l_{mn}$. Then, we can update the optimization (12) and weight assignment (13), respectively, as follows:

$$\max_{(\mu^c_{mn}(t))}\sum_{m,n}\mu^{c^*_{mn}(t)}_{mn}(t)w_{mn}(t)$$

$$\text{s.t. } (\mu^c_{mn}(t))_{(m,n)\in\mathcal{L}}\in I(t) \text{ and } \mu^c_{s(c)b(c)}(t)\le\mu_M \quad \forall\, c\in\mathcal{L}.$$

$$w^c_{mn}(t) = \begin{cases} \frac{U^c_{s(c)}(t)}{q_M}[U^c_m(t)-U^c_n(t)-l_n], & \text{if } (m,n)\in\mathcal{L} \\ \frac{U^c_{s(c)}(t)}{q_M}[q_M-\mu_M-U^c_{b(c)}(t)], & \text{if } (m,n)=(s(c),b(c)) \\ 0, & \text{otherwise.} \end{cases}$$

It is not difficult to check that Proposition 1 still holds with $q_M \ge \max\{\max_{n\in\mathcal{N}} l_n, \mu_M\}$ and Theorem 1 holds with a different definition of constant B. The above modified algorithm can be further extended to solve power allocation problems, where we refer interested readers to our recent work [35].

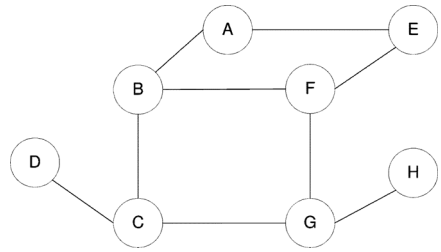

Fig. 2. Network topology for simulations.

TABLE II
THROUGHPUT (SUM OVER THREE FLOWS) AND DELAY (AVERAGED OVER THREE FLOWS) PERFORMANCE OF ***ALG*** WITH DIFFERENT V'S FOR BACKLOGGED SOURCES

V	200	1000	10000
Throughput	0.9202	0.9368	0.9379
End-to-end delay	46.01	45.76	46.51

VI. NUMERICAL RESULTS

In this section, we present the simulation results for the optimal algorithm ***ALG*** proposed in Section IV-A. Simulations are run in MATLAB 2009A for 3×10^5 time-slots with results averaged over the last 10^5 time-slots. In the network topology illustrated in Fig. 2, there are three source–destination pairs (A,G), (D,E), and (F,H) with same Poisson arrival rates and $\mu_M = 2$. The required minimum data rates for the three flows are all set to 0.1.

A. Effects of Parameter V

We show the algorithm performance with backlogged sources in Table II by varying the control parameter V, where we set $q_M = 5$ and $\rho_c = 30q_M$ for each flow c. With an increasing V, we observe that the throughput is slightly increased while the average end-to-end delay is not affected.

However, a large value of V increases the upper bound of the virtual queue backlogs (as stated in Remark 2 in Section IV-A)

TABLE III
THROUGHPUT PERFORMANCE OF ***ALG*** WHEN SOURCES ARE BACKLOGGED AT THE TRANSPORT LAYER

	ALG ($\rho_c = 150$)	***ALG*** ($\rho_c = 300$)	***ALG*** ($\rho_c = 3000$)	***ALG*** ($\rho_c = 30000$)	BP
Throughput (sum for three flows)	0.9368	1.1834	1.2007	1.2305	1.2315
End-to-end delay (averaged over three flows)	45.76	131.47	1.514×10^3	1.3687×10^4	3.753×10^4

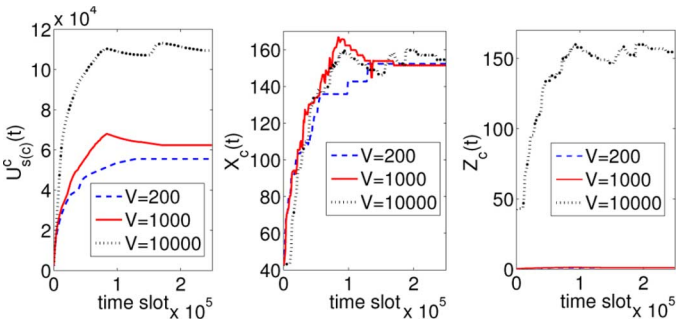

Fig. 3. Virtual queue evolutions with backlogged sources.

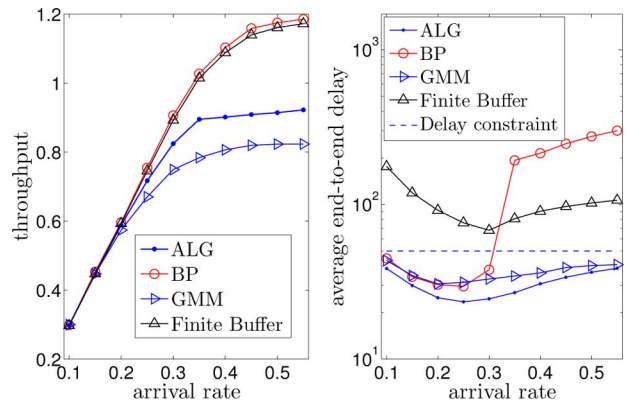

Fig. 4. Throughput and delay tradeoff under ***ALG*** with performances compared to Finite Buffer algorithm and BP algorithm, with varying arrival rates at the transport layer.

and has a negative effect on the convergence rates of virtual queues, as is illustrated in Fig. 3. Specifically, for $V = 10\,000$, it takes more than 2×10^5 time-slots for $U^c_{s(c)}(t)$ and $Z_c(t)$ to converge. Thus, in Section VI-B, we choose $V = 1000$ since a larger V does not lead to a noticeable increase in the throughput while resulting in a smaller convergence rate of virtual queues.

B. Throughput Optimality and Delay Tradeoff

In this section, we illustrate the throughput and delay performance of the proposed algorithm. For comparison, we denote by *BP* the back-pressure scheduling algorithm in [1] with a congestion controller in [4], and denote by *Finite Buffer* the cross-layer algorithm developed in [25] with buffer size equal to the queue length limit q_M. Note that it is shown in simulation results in [25] that Finite Buffer algorithm ensures much smaller internal queue length (of nodes excluding the source node) than BP algorithm (and queue length is related to delay performance).

We first illustrate in Table III the throughput optimality of ***ALG*** when the sources are constantly backlogged. We loosen the delay constraint as $\rho_c = 30q_M$. With fixed $V = 1000$, as we increase the control parameter q_M, the ***ALG*** achieves a throughput approaching the throughput of BP algorithm that is known to be optimal. We also note that this approximation in throughput results in worse average end-to-end delay performance, which complies with Remark 1.

We then illustrate the throughput and delay tradeoff for both the ***ALG*** and its corresponding suboptimal GMM algorithm (discussed in Remark 5 in Section V-A) in Fig. 4 for the case of arbitrary arrival rates at transport layer with $L_M = 0$, where we set $q_M = 5$ and $\rho_c = 50$ for each flow c. Note that this pair of q_M and ρ_c shows that the bound in (15) is actually quite loose, and thus our algorithm can achieve better delay performance than stated in (15). Fig. 4 shows that the average end-to-end delay under ***ALG*** is well below the constraint ($\rho_c = 50$) and lower than that under BP and Finite Buffer algorithms. The throughput of ***ALG*** is close to (although lower than) that of the optimal BP algorithm when arrival rates are small (≤ 0.3). Specifically, when the arrival rate is 0.3, ***ALG*** achieves a throughput 10% more than the GMM algorithm and 9.0% less than BP algorithm, with an average end-to-end delay 35.2% less than the BP algorithm. In the large-input-rate-region (> 0.3), we also observe that the delay in both the BP and Finite Buffer algorithm violates the delay constraints. In addition, in the above illustrated scenarios with backlogged and arbitrary arrival rates, the minimum arrival rates and average end-to-end delay requirements are satisfied for individual flows under ***ALG***. As a side note, the average end-to-end delay in all of the four algorithms in Fig. 4 first decreases, which can be explained by the intuition that all the algorithms are based on back-pressure of links (i.e., queue backlog difference of links) and the queue backlog difference tends to be larger for each hop with a larger arrival rate. When arrival rate further increases, congestion level becomes higher since more packets are admitted into the network.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a cross-layer framework that approaches the optimal throughput arbitrarily close for a general multihop wireless network. We show a tradeoff between the throughput and average end-to-end delay bound while satisfying the minimum data rate requirements for individual flows.

Our work aims at a better understanding of the fundamental properties and performance limits of QoS-constrained multihop wireless networks. While we show an $O(\frac{1}{\epsilon})$ delay bound with ϵ-loss in throughput, how small the actual delay can become still remains elusive, which is dependent on specific network topologies. In our future work, we will investigate the capacity region under end-to-end delay constraints applied to different network topologies. Our future work will also involve power management in the scheduling policies.

APPENDIX
PROOF OF THEOREM 3

Before we proceed to the proof, we extend the stationary randomized algorithm STAT introduced in Lemma 1 and Remark 4. Given (θ_c) introduced in Lemma 1 and given flow c at node n, recall that $(A_c(t))$ is i.i.d. with mean (λ_c) and $(\lambda_c) > (\theta_c)$ elementwise. The flow control for STAT can be given as: Admit $\mu^{c,\mathrm{STAT}}_{s(c)b(c)}(t) = A_c(t)$ w.p. $\frac{\theta_c}{\lambda_c}$; otherwise, $\mu^{c,\mathrm{STAT}}_{s(c)b(c)}(t) = 0$. Then, $\mathbb{E}\left\{\mu^{c,\mathrm{STAT}}_{s(c)b(c)}(t)\right\} = \theta_c$, $\forall\ t$. Now take $v_c^{\mathrm{STAT}}(t) = R_c^{\mathrm{STAT}}(t) = \mu^{c,\mathrm{STAT}}_{s(c)b(c)}(t)\ \forall\ c \in \mathcal{F}$. Then, we also have $\mathbb{E}\left\{v_c^{\mathrm{STAT}}(t)\right\} = \mathbb{E}\left\{R_c^{\mathrm{STAT}}(t)\right\} = \theta_c$. Note that $R_c^{\mathrm{STAT}}(t) \le A_c(t) \le \min\{L_c(t) + A_c(t), \mu_M\}$ and $v_c^{\mathrm{STAT}}(t) \le \mu_M$.

Now we present the proof.

Proof: We define the Lyapunov function as $L(\mathbf{Q}'(t)) = L(\mathbf{Q}(t)) + \frac{\eta}{2}\sum_{c\in\mathcal{F}} Y_c^2(t)$ and the Lyapunov drift as $\Delta'(t) = \mathbb{E}\{L(\mathbf{Q}'(t+1)) - L(\mathbf{Q}'(t))|\mathbf{Q}'(t)\}$, where $\mathbf{Q}'(t) = (\mathbf{Q}(t), (Y_c(t)))$. Following the steps in deriving (22) and (23) and squaring both sides of the virtual queue dynamics (33), we have

$$
\begin{aligned}
&\Delta'(t) - V\sum_{c\in\mathcal{F}} \mathbb{E}\{v_c(t)|\mathbf{Q}'(t)\} \\
&\quad\le B_1 + \sum_{c\in\mathcal{F}} \mathbb{E}\{v_c(t)(\eta Y_c(t) - V)|\mathbf{Q}'(t)\} \\
&\qquad + \sum_{c\in\mathcal{F}} \mathbb{E}\left\{ R_c(t)\left(\frac{(q_M - \mu_M)\, U^c_{s(c)}(t)}{q_M} \right.\right. \\
&\qquad\qquad \left.\left. -\eta Y_c(t) - X_c(t)\rho_c - Z_c(t)\right) |\mathbf{Q}'(t)\right\} \\
&\qquad + N q_M \sum_{c\in\mathcal{F}} X_c(t) + \sum_{c\in\mathcal{F}} a_c Z_c(t) \\
&\qquad + \frac{1}{2}\sum_{c\in\mathcal{F}} \frac{\left(2N - 1 + \mu_M^2\right) U^c_{s(c)}(t)}{q_M} \\
&\qquad - \mathbb{E}\left\{ \sum_{c\in\mathcal{F}} \sum_{(m,n)\in\mathcal{L}} \mu^c_{mn}(t) \frac{U^c_{s(c)}(t)}{q_M} \left(U^c_m(t) - U^c_n(t)\right) \right. \\
&\qquad \left. + \sum_{c\in\mathcal{F}} \mu^c_{s(c)b(c)}(t) \frac{U^c_{s(c)}(t)}{q_M} \left(q_M - \mu_M - U^c_{b(c)}(t)\right) |\mathbf{Q}'(t)\right\}.
\end{aligned}
\tag{38}
$$

The second term, third term, and the last term of the RHS of (38) are minimized by the congestion controller (34) and (35) and the scheduling policy (12), respectively, over a set of feasible algorithms including the stationary randomized algorithm STAT. Substitute into the second term of RHS of (38) a stationary randomized algorithm with admitted arrival rate vector $(r^*_{\epsilon,c} - \frac{1}{2}\epsilon')$, the third term a stationary randomized algorithm with admitted arrival rate vector $(r^*_{\epsilon,c})$, and the last term a stationary randomized algorithm with admitted arrival rate vector $(r^*_{\epsilon,c} + \epsilon)$. Then, following the steps in proving Theorem 1, we can prove Theorem 3. ■

REFERENCES

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

[2] M. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Mar. 2005.

[3] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 411–424, Apr. 2005.

[4] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-Layer control in wireless networks," *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–149, 2006.

[5] M. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[6] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[7] X. Wu, R. Srikant, and J. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 595–605, Jun. 2007.

[8] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proc. 43rd Annu. Allerton Conf. Commun., Control, Comput.*, 2005, pp. 1–11.

[9] A. Eryilmaz, A. Ozdaglar, D. Shah, and E. Modiano, "Distributed cross-layer algorithms for the optimal control of multi-hop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 638–651, Apr. 2010.

[10] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 1118–1126.

[11] L. Ying, R. Srikant, and D. Towsley, "Cluster-based back-pressure routing algorithm," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 484–492.

[12] G. Gupta and N. Shroff, "Delay analysis for wireless networks with single hop traffic and general interference constraints," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 393–405, Apr. 2010.

[13] M. Neely, "Delay-based network utility maximization," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[14] K. Kar, X. Luo, and S. Sarkar, "Delay guarantees for throughput-optimal wireless link scheduling," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2331–2339.

[15] H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-aware cross-layer design for network utility maximization in multi-hop networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 951–959, May 2011.

[16] G. Gupta and N. Shroff, "Delay analysis for multi-hop wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2356–2364.

[17] 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 INFOCOM*, Mar. 2010, pp. 1–9.

[18] J. Ryu, L. Ying, and S. Shakkottai, "Back-pressure routing for intermittently connected networks," in *Proc. IEEE INFOCOM Mini-Conf.*, Mar. 2010, pp. 1–5.

[19] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proc. IEEE INFOCOM Mini-Conf.*, Apr. 2009, pp. 2936–2940.

[20] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1674–1682.

[21] K. Jung and D. Shah, "Low delay scheduling in wireless networks," in *Proc. IEEE ISIT*, Jun. 2007, pp. 1396–1400.

[22] M. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1728–1736.

[23] P. Huang, X. Lin, and C. Wang, "A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2588–2596.

[24] P. Giaccone, E. Leonardi, and D. Shah, "Throughtput region of finite-buffered networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 2, pp. 251–263, Feb. 2007.

[25] L. Le, E. Modiano, and N. Shroff, "Optimal control of wireless networks with finite buffers," in *Proc. IEEE INFOCOM*, Apr. 2010, pp. 1–9.

[26] M. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," Ph.D. dissertation, MIT, Cambridge, MA, 2003.
[27] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless neworks," in *Proc. 12th Annu. MobiCom*, Sep. 2006, pp. 227–238.
[28] C. Joo, X. Lin, and N. Shroff, "Greedy maximal matching: Performance limits for arbitrary network graphs under the node-exclusive interference model," *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2734–2744, Dec. 2009.
[29] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem," *Discrete Appl. Math.*, vol. 126, pp. 313–322, 2003.
[30] L. Jiang and J. Walrand, "A distributed algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 960–972, Jun. 2010.
[31] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," in *Proc. IEEE INFOCOM Mini-Conf.*, Apr. 2010, pp. 1–5.
[32] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *Proc. ACM SIGMETRICS*, 2009, pp. 133–144.
[33] L. Jiang, M. Leconte, J. Ni, R. Srikant, and J. Walrand, "Fast mixing of parallel Glauber dynamics and low-delay CSMA scheduling," in *Proc. IEEE INFOCOM Mini-Conf.*, Apr. 2011, pp. 371–375.
[34] D. Xue, R. Murawski, and E. Ekici, "Distributed utility-optimal scheduling with finite buffers," in *Proc. WiOpt*, May 2012, pp. 278–285.
[35] D. Xue and E. Ekici, "Optimal power allocation in multi-hop wireless networks with finite buffers," in *Proc. IEEE ICC*, Jun. 2011, pp. 1–5.
[36] Y. Yi and M. Chiang, "Stochastic network utility maximization: A tribute to Kelly's paper published in this journal a decade ago," *Eur. Trans. Telecommun.*, vol. 19, no. 4, pp. 421–442, Apr. 2008.
[37] D. Xue and E. Ekici, "Delay-guaranteed cross-layer scheduling in multi-hop wireless networks," Ohio State University, Columbus, OH, 2012 [Online]. Available: http://ece.osu.edu/~xued/delay.pdf

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.

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 the B.S. and M.S. degrees in computer engineering from Bogazici University, Istanbul, Turkey, in 1997 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2002.

Currently, he is an Associate Professor with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus. 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.

Dr. Ekici is an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, *Computer Networks*, and *Mobile Computing and Communications Review*.