Delay-Guaranteed Cross-Layer Scheduling in Multi-Hop Wireless Networks

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 propose a cross-layer scheduling algorithm that achieves a throughput "ϵ-close" to the optimal throughput in multi-hop wireless networks with a tradeoff of $O(\frac{1}{\epsilon})$ in delay guarantees. The algorithm aims to solve a joint congestion control, routing, and scheduling problem in a multi-hop wireless network while satisfying per-flow average end-to-end delay guarantees 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, the effects of delayed feedback on the optimal algorithm, and the extensions of the proposed algorithm to different interference models with arbitrary link capacities.**

I. Introduction

Cross-layer design of congestion control, routing and scheduling algorithms with guaranteed quality of service (QoS) is one of the most challenging topics in wireless networking. Back-pressure algorithm first proposed in [1] and its extensions have been widely employed in developing throughput guaranteed dynamic resource allocation and scheduling algorithms for wireless systems. Back-pressure-based scheduling algorithms have been employed in wireless networks with time-vary channels [5][10][11]. Congestion controllers at the transport layer have assisted the cross-layer design of scheduling algorithms [2][3][12], so that the admitted arrival rate is guaranteed to lie within the network capacity region. Low-complexity distributed algorithms have been proposed in [8][9][29][30]. Algorithms adapted to clustered networks have been proposed in [4] to reduce the number of queues maintained in the network. However, delay-related investigations are not included in these works.

Delay issues in single-hop wireless networks have been addressed in [20]-[25]. Especially, the scheduling algorithm in [25] provides a throughput-utility that is inversely proportional to the delay guarantee. But these works are not readily extendable to multi-hop wireless networks, where we have to consider additional arrivals from neighboring nodes and routing. Delay analysis for fixed-routing multi-hop networks is provided in [16]. Delay-related scheduling in multi-hop wireless networks have been proposed in [17][18][19][26][27]. However, none of the above provide explicit end-to-end delay guarantees. A fixed-routing scheduling algorithm for finite-buffer multi-hop wireless networks is proposed in [13] and is extended to adaptive-routing with congestion controller in [14]. Specifically, the algorithm in [14] guarantees $O(\frac{1}{\epsilon})$-scaling in buffer size with a ϵ-loss in throughput-utility, but assumes a source node to have an infinite buffer in the network layer, which may lead to large end-to-end delay.

In this paper, we propose a cross-layer algorithm to achieve *guaranteed throughput* while satisfying network QoS requirements. Specifically, we construct two virtual queues, namely, virtual queue at transport layer and virtual delay queue, to *guarantee average end-to-end delay bounds*; and we construct a virtual service queue to *guarantee the minimum data rate required by individual network flows*. The 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 value. In addition, the algorithm exhibits a tradeoff of $O(\frac{1}{\epsilon})$ in the delay bound where ϵ denotes the loss in 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; (4) from a node-exclusive model with constant link capacities to a model with arbitrary link capacities and interference models over fading channels.

The rest of the paper is organized as follows. Section II provides the the network model and corresponding approaches for the considered multi-hop wireless networks. In Section III, we propose the optimal cross-layer control and scheduling algorithm and analyze its performance. In Section IV, we provide a class of feasible suboptimal algorithms, consider sources with arbitrary arrival rates at transport layer, employ delayed queueing 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 V. Finally, we conclude our work in Section VI.

An abbreviated version of this work has been submitted to IEEE INFOCOM 2011.

II. Network Model

A. Network Elements

We consider a time-slotted multi-hop 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 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. Let the scheduling parameter $\mu^c_{mn}(t)$ denote the link rate assignment of flow c for link (m,n) at time t according to scheduling decisions and let $\mu^c_{s(c)b(c)}(t)$ denote the admitted rate of flow c from the transport layer of flow to the source node, where $s(c)$ denotes the source at the transport layer of flow c. It is clear that in any time slot t, $\mu^c_{s(c)n}(t) = 0$ $\forall n \neq b(c)$. 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 $\mu_M \geq 1$:

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

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

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

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 IV.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-average upperbound, not a deterministic upper-bound. 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 $\mathcal{L}^c \triangleq \mathcal{L} \cup \{(s(c), b(c))\}$. We now model queue dynamics and network constraints in the multi-hop 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 we have $U^c_n(t) = 0$ $\forall t$; Otherwise, the queue dynamics is as follows:

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

where the operator $[x]^+$ is defined as $[x]^+ = \max\{x, 0\}$ so that the number of packets transmitted for flow c from a node does not exceed the backlog at node n, 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 $\sum_{j:(j,n)\in\mathcal{L}} \mu^c_{jn}(t)$ represent, respectively, the departure rate from node n and the endogenous 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 not have sufficient number of packets to transmit. Since we employ the node-exclusive model, we have

$$0 \leq \sum_{i:(n,i)\in\mathcal{L}} \mu^c_{ni}(t) + \sum_{j:(j,n)\in\mathcal{L}} \mu^c_{jn}(t) \leq 1, \forall n \in \mathcal{N}. \tag{4}$$

Assuming that each node can be the source node of at most one flow, we also have

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

if it is ensured that no packets will be looped back to the source. Note that our analysis can be extended to a model where a node can be the source node for multiple flows.

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 later assist the development of our algorithm:
(1) For each flow c at transport layer, we construct a virtual queue $U^c_{s(c)}(t)$ which will be employed in the algorithm proposed in the next section. 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:

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

(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:

$r_c \geq a_c$. 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:

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

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

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

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

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

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.

From the above description, we know that the network is *stable* (i.e., each queue at all nodes is stable) and the average end-to-end delay constraint and minimum data rate requirement are achieved if queues $U^c_n(t)$ and the three virtual queues are stable for any node and flow, i.e.,

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

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

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

Now we define the capacity region of the considered multi-hop 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 the following sections, we let $(r^*_{\epsilon,c})$ denote the solutions to the following optimization problem:

$$\max_{(r_c):(r_c+\epsilon)\in\Lambda}\sum_{c\in\mathcal{F}} r_c \qquad (11)$$

$$\text{s.t. } r_c \leq \mu_M, \quad \forall c, \qquad (12)$$

$$r_c \geq a_c, \ \forall c. \qquad (13)$$

where ϵ is a positive number which can be chosen arbitrarily small. For simplicity of analysis, we assume that (a_c) is in the interior of Λ and $(\mu_M)_{K\times 1}$ is large enough to be in the exterior of Λ. Without loss of generality, we assume that there exists $\epsilon' > 0$ such that $r^*_{\epsilon,c} \geq a_c + \epsilon' \ \forall c \in \mathcal{F}$. Then, we can omit constraint (12) and according to [6] we have

$$\lim_{\epsilon\to 0}\sum_{c\in\mathcal{F}} r^*_{\epsilon,c} = \sum_{c\in\mathcal{F}} r^*_c, \qquad (14)$$

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

$$\max_{(r_c):(r_c)\in\Lambda}\sum_{c\in\mathcal{F}} r_c. \qquad (15)$$

III. Control Scheduling Algorithm for Multi-Hop Wireless Networks

Now we propose a control and scheduling algorithm ***ALG*** for the introduced multi-hop 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 which 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 the following subsections.

A. Algorithm Description and Analysis

Let $q_M \geq \mu_M$ be a control parameter for queue length. We first propose a congestion controller for the input rate of virtual queues at transport layer:

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

$$\min_{0\leq R_c(t)\leq\mu_M} R_c(t)\Big(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M} - X_c(t)\rho_c - Z_c(t) - V\Big) \qquad (16)$$

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, $R_c(t) = \mu_M$.

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 II including (1)(2)(4), the network solves the following optimization problem:

$$\max_{(\mu^c_{mn}(t))}\sum_{m,n}\mu_{mn}^{c^*_{mn}(t)}(t)w_{mn}(t) \qquad (17)$$

$$\text{s.t.} \quad \mu^c_{mn}(t) = 0 \quad \forall c \neq c^*_{mn}(t),\ \forall(m,n)\in\mathcal{L}^c,$$

$$\mu^c_{mn}(t) = 0 \quad \text{if } n = s(c),\ \forall c\in\mathcal{F},$$

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

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

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

and

$$w_{mn}^c(t) = \begin{cases} \frac{U_{s(c)}^c(t)}{q_M}[U_m^c(t) - U_n^c(t)], \text{ if } (m,n) \in \mathcal{L}, \\ \frac{U_{s(c)}^c(t)}{q_M}[q_M - \mu_M - U_{b(c)}^c(t)], \\ \qquad\qquad \text{if } (m,n) = (s(c), b(c)), \\ 0, \qquad\qquad \text{otherwise.} \end{cases} \tag{18}$$

Note that $\mathcal{L} \cup \{(s(c), b(c)) : c \in \mathcal{F}\}$ forms the (m,n) pairs in $(\mu_{mn}^c(t))$ over which the optimization (17) 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 that when $w_{mn}(t) = 0$, we must have $\mu_{mn}^c(t) = 0\ \forall c$ to maximize (17). Note also that similar approaches have been utilized in [13][14], while we employ the virtual queue $U_{s(c)}^c(t)$ at transport layer as a weight on the queue backlog differences in (18).

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:

$$U_n^c(t) \le q_M, \quad \forall t, \forall n \in \mathcal{N}, \forall c \in \mathcal{F}. \tag{19}$$

Proof: We use induction in the proof. When $t = 0$, we have $U_n^c(0) = 0\ \forall n, c$. Now assume in time slot t we have $U_n^c(t) \le q_M\ \forall n, c$. In the induction step, we consider two cases as follows:

(1) We first consider the case when $n = b(c)$ for some flow c, i.e., when n is a source node. If $U_{b(c)}^c(t) \le q_M - \mu_M$, then according to the queueing dynamics (3) and (5) we have $U_{b(c)}^c(t+1) \le q_M$; Otherwise, we have $U_{b(c)}^c(t) > q_M - \mu_M$ and according to the weight assignment (18), we have $w_{s(c)b(c)}^c(t) < 0$ which leads to $\mu_{s(c)b(c)}^c(t) = 0$, so $U_{b(c)}^c(t+1) \le U_{b(c)}^c(t) \le q_M$ by (2)(3).

(2) In the second case, we have $n \ne b(c)$ for any c, i.e., n is not the source node of any flow c. If $U_n^c(t) < q_M$, then, since we employ node-exclusive model, we have $U_n^c(t) \le q_M$ by (3)(4); Otherwise, we have $U_n^c(t) = q_M$, and according to the scheduling algorithm (18) we have $w_{mn}^c(t) = 0\ \forall m \in \mathcal{N}$, which induces $\mu_{mn}^c(t) = 0\ \forall m \in \mathcal{N}$, so $U_n^c(t+1) \le q_M$ by the queueing dynamics (3). ■

Now we present our main results in Theorem 1.

Theorem 1: Given that

$$q_M > \frac{2N - 1 + \mu_M^2}{2\epsilon} + \mu_M \text{ and } \rho_c > \frac{N q_M}{r_{\epsilon,c}^*}\ \forall c \in \mathcal{F}, \tag{20}$$

ALG ensures that the virtual queues have a time-averaged bound:

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{c\in\mathcal{F}} \mathbb{E}\{U_{s(c)}^c(\tau) + X_c(\tau) + Z_c(\tau)\} \le \frac{B'}{\delta}, \tag{21}$$

where δ and B' are constant positive numbers which will be given in the next subsection.

In addition, ***ALG*** can achieve a throughput

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(\tau)\} \ge \sum_{c\in\mathcal{F}} r_{\epsilon,c}^* - \frac{B}{V}, \tag{22}$$

where B is a constant positive number independent of V which will be given in the next subsection.

Remark 1: The results (19) and (21) indicate that ***ALG*** stabilizes the network and satisfies the average end-to-end delay constraint and the minimum data rate requirement. The inequality (22) gives the lower-bound of the throughput that ***ALG*** can achieve. Given some $\epsilon > 0$, since B is independent of V, (22) 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 (20) in Theorem 1, i.e., the average end-to-end delay bound must be of order $O(\frac{1}{\epsilon})$, where we recall that $a_c + \epsilon' \le r_{\epsilon,c}^* \le \mu_M$, $\forall c \in \mathcal{F}$.

B. Proof of Theorem 1

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

Lemma 1: For nonnegative numbers $A_1, A_2, A_3, Q \in \mathbb{R}$ such that $Q \le [A_1 - A_2]^+ + A_3$, we have $Q^2 \le A_1^2 + A_2^2 + A_3^2 - 2A_1(A_2 - A_3)$.

The proof of Lemma 1 is trivial and omitted. We will later use Lemma 1 to simplify virtual queue dynamics.

Lemma 2: For any feasible rate vector $(\theta_c) \in \Lambda$ with $\theta_c \ge a_c\ \forall c \in \mathcal{F}$, there exists a stationary randomized algorithm STAT that stabilizes the network with input rate vector $(\mu_{s(c)b(c)}^{STAT}(t))$ 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:

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

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 [2] and [3], so we omit the proof of Lemma 2 for brevity.

Remark 2: According to the STAT algorithm in Lemma 2, 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 the update equation (6), it is easy to show that the virtual queues under STAT are bounded above 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 $(r_{\epsilon,c}^*)$ or $(r_{\epsilon,c}^* + \epsilon)$ or $(r_{\epsilon,c}^* - \frac{1}{2}\epsilon')$, where we recall $(r_{\epsilon,c}^* + \epsilon) \in \Lambda$ and $r_{\epsilon,c}^* \ge a_c + \epsilon'\ \forall c \in \mathcal{F}$.

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

$$\begin{aligned} L(\mathbf{Q}(t)) = &\frac{1}{2}\{\sum_{c\in\mathcal{F}} \frac{q_M-\mu_M}{q_M} U_{s(c)}^c(t)^2 + \sum_{c\in\mathcal{F}} X_c(t)^2 \\ &+ \sum_{c\in\mathcal{F}} Z_c(t)^2 + \sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{1}{q_M} U_n^c(t)^2 U_{s(c)}^c(t)\}. \end{aligned} \tag{23}$$

It is obvious that $L(\mathbf{Q}(0)) = 0$. We denote the Lyapunov drift by

$$\Delta(t) = \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)\}. \tag{24}$$

Note that the last term of the Lyapunov function (23) takes the same form as that in [13][14][1]. From the queue dynamics (3)(6), we have:

$$\begin{aligned} &\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{1}{q_M} U_n^c(t+1)^2 U_{s(c)}^c(t+1) \\ \leq &\sum_{c\in\mathcal{F}} \frac{1}{q_M}(R_c(t) + U_{s(c)}^c(t)) \sum_{n\in\mathcal{N}} U_n^c(t+1)^2 \\ \leq &\mu_M q_M NK + \sum_{c\in\mathcal{F}} \frac{1}{q_M} U_{s(c)}^c(t) \sum_{n\in\mathcal{N}} \{U_n^c(t)^2 \\ &+ (\sum_{i:(n,i)\in\mathcal{L}} \mu_{ni}^c(t))^2 + (\sum_{j:(j,n)\in\mathcal{L}^c} \mu_{jn}^c(t))^2 \\ &- 2U_n^c(t)(\sum_i \mu_{ni}^c(t) - \sum_j \mu_{jn}^c(t))\}, \end{aligned} \tag{25}$$

where we recall that $R_c(t) \leq \mu_M$ and we employ Lemma 1 to deduce the second inequality.

From (25), we have

$$\begin{aligned} &\frac{1}{2}(\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{1}{q_M}(U_n^c(t+1)^2 U_{s(c)}^c(t+1) \\ &\qquad - U_n^c(t)^2 U_{s(c)}^c(t))) \\ \leq &\frac{1}{2}\sum_{c\in\mathcal{F}} \frac{(2N-1+\mu_M^2)U_{s(c)}^c(t)}{q_M} + \frac{1}{2}NKq_M\mu_M \\ -&\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{U_n^c(t)U_{s(c)}^c(t)}{q_M} \\ &(\sum_{j:(n,j)\in\mathcal{L}} \mu_{nj}^c(t) - \sum_{i:(i,n)\in\mathcal{L}^c} \mu_{in}^c(t)), \end{aligned} \tag{26}$$

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

From the queue length dynamics (6) and by employing Lemma 1, we have:

$$\begin{aligned} &\frac{1}{2}\sum_{c\in\mathcal{F}} \frac{q_M-\mu_M}{q_M}(U_{s(c)}^c(t+1)^2 - U_{s(c)}^c(t)^2) \\ \leq &\frac{1}{2}\sum_{c\in\mathcal{F}} \frac{q_M-\mu_M}{q_M}(\mu_{s(c)b(c)}^c(t)^2 + R_c(t)^2 \\ &- 2U_{s(c)}^c(t)(\mu_{s(c)b(c)}^c(t) - R_c(t))) \\ \leq &K\frac{q_M-\mu_M}{q_M}\mu_M^2 \\ &- \frac{q_M-\mu_M}{q_M}\sum_{c\in\mathcal{F}} U_{s(c)}^c(t)(\mu_{s(c)b(c)}^c(t) - R_c(t)). \end{aligned} \tag{27}$$

From the virtual queue dynamics (9), we have:

$$\begin{aligned} &\frac{1}{2}\sum_{c\in\mathcal{F}}(X_c(t+1)^2 - X_c(t)^2) \\ \leq &\frac{1}{2}\sum_{c\in\mathcal{F}}(\rho_c^2 R_c(t)^2 + (\sum_{n\in\mathcal{N}} U_n^c(t))^2 \\ &- 2X_c(t)(\rho_c R_c(t) - \sum_{n\in\mathcal{N}} U_n^c(t))) \\ \leq &\frac{1}{2}\mu_M^2 \sum_{c\in\mathcal{F}} \rho_c^2 + \frac{1}{2}KN^2q_M^2 \\ &- \sum_{c\in\mathcal{F}} X_c(t)\rho_c R_c(t) + Nq_M \sum_{c\in\mathcal{F}} X_c(t). \end{aligned} \tag{28}$$

From the virtual queue dynamics (8), we have:

$$\begin{aligned} &\frac{1}{2}\sum_{c\in\mathcal{F}}(Z_c(t+1)^2 - Z_c(t)^2) \\ \leq &\frac{1}{2}\sum_{c\in\mathcal{F}}(R_c(t)^2 + a_c^2 - 2Z_c(t)(R_c(t) - a_c)) \\ \leq &\frac{1}{2}K\mu_M^2 + \frac{1}{2}\sum_{c\in\mathcal{F}} a_c^2 - \sum_{c\in\mathcal{F}} Z_c(t)R_c(t) + \sum_{c\in\mathcal{F}} a_c Z_c(t). \end{aligned} \tag{29}$$

Substituting (26)(27)(28)(29) into the Lyapunov drift (24) and subtracting $V\sum_c \mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}$ from both sides, we then have:

$$\begin{aligned} &\Delta(t) - V\sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)|\mathbf{Q}(t)\} \\ \leq &B + \sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)(\frac{(q_M-\mu_M)U_{s(c)}^c(t)}{q_M} \\ &\qquad - X_c(t)\rho_c - Z_c(t) - V)|\mathbf{Q}(t)\} \\ &+Nq_M\sum_{c\in\mathcal{F}} X_c(t) + \sum_{c\in\mathcal{F}} a_c Z_c(t) \\ &+\frac{1}{2}\sum_{c\in\mathcal{F}} \frac{(2N-1+\mu_M^2)U_{s(c)}^c(t)}{q_M} \\ &-\mathbb{E}\{\frac{q_M-\mu_M}{q_M}\sum_{c\in\mathcal{F}} U_{s(c)}^c(t)\mu_{s(c)b(c)}^c(t) \\ &+\sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{U_n^c(t)U_{s(c)}^c(t)}{q_M} \\ &(\sum_{j:(n,j)\in\mathcal{L}} \mu_{nj}^c(t) - \sum_{i:(i,n)\in\mathcal{L}^c} \mu_{in}^c(t))|\mathbf{Q}(t)\}, \end{aligned} \tag{30}$$

[1]Note that, however, the queue dynamics (3)(6) takes different form from those in [13][14].

where $B \triangleq \frac{1}{2}NKq_M\mu_M + K\frac{q_M-\mu_M}{q_M}\mu_M^2 + \frac{1}{2}\mu_M^2\sum_{c\in\mathcal{F}}\rho_c^2 + \frac{1}{2}KN^2q_M^2 + \frac{1}{2}K\mu_M^2 + \frac{1}{2}K\sum_{c\in\mathcal{F}}a_c^2$. We can rewrite the last term of RHS of (30) by simple algebra as

$$\begin{aligned}&-\mathbb{E}\{\sum_{c\in\mathcal{F}}\sum_{(m,n)\in\mathcal{L}}\mu_{mn}^c(t)\frac{U_{s(c)}^c(t)}{q_M}(U_m^c(t)-U_n^c(t))\\&+\sum_{c\in\mathcal{F}}\mu_{s(c)b(c)}^c(t)\frac{U_{s(c)}^c(t)}{q_M}(q_M-\mu_M-U_{b(c)}^c(t))|\mathbf{Q}(t)\}.\end{aligned} \tag{31}$$

Then, the second term and the last term of the RHS of (30) are minimized by the congestion controller (16) and the scheduling policy (17), respectively, over a set of feasible algorithms including the stationary randomized algorithm STAT introduced in Lemma 2 and Remark 2, which require the input rate to be less than μ_M. We can substitute into the second term of RHS of (30) 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:

$$\begin{aligned}&\Delta(t) - V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}\\ \le&B - V\sum_{c\in\mathcal{F}}r_{\epsilon,c}^*\\&-\sum_{c\in\mathcal{F}}\frac{U_{s(c)}^c(t)}{q_M}(\epsilon(q_M-\mu_M)-\frac{2N-1+\mu_M^2}{2})\\&-\sum_{c\in\mathcal{F}}(r_{\epsilon,c}^*-a_c)Z_c(t)-\sum_{c\in\mathcal{F}}(\rho_c r_{\epsilon,c}^*-Nq_M)X_c(t).\end{aligned} \tag{32}$$

When (20) holds, we can find $\epsilon_1>0$ such that $\epsilon_1 \le \rho_c r_{\epsilon,c}^* - Nq_M$ $\forall c\in\mathcal{F}$ 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:

$$\begin{aligned}&\Delta(t) - V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}\\ \le&B-\delta\sum_{c\in\mathcal{F}}(X_c(t)+U_{s(c)}^c(t)+Z_c(t))-V\sum_{c\in\mathcal{F}}r_{\epsilon,c}^*,\end{aligned} \tag{33}$$

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

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

$$\begin{aligned}&\frac{1}{t}\mathbb{E}\{L(\mathbf{Q}(t))\} - \frac{V}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(\tau)\}\\ \le&B-V\sum_{c\in\mathcal{F}}r_{\epsilon,c}^*\\&-\frac{\delta}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{X_c(\tau)+U_{s(c)}^c(\tau)+Z_c(\tau)\}.\end{aligned} \tag{34}$$

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) and $\mathbb{E}\{L(\mathbf{Q}(t))\}$ is nonnegative, by taking limsup of t on both sides of (34), we have:

$$\begin{aligned}&\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{X_c(\tau)+U_{s(c)}^c(\tau)+Z_c(\tau)\}\\ \le&\frac{B}{\delta}+\frac{V}{\delta}[\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(\tau)\}-\sum_{c\in\mathcal{F}}r_{\epsilon,c}^*]\\ \le&\frac{B'}{\delta},\end{aligned} \tag{35}$$

where $B' \triangleq B + VB_R$. Thus, we have proved (21).

By taking liminf of t on both sides of (34), we have

$$\begin{aligned}&\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(\tau)\}\\ \ge&\frac{\delta}{V}\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{X_c(\tau)+U_{s(c)}^c(\tau)+Z_c(\tau)\}\\&-\frac{B}{V}+\sum_{c\in\mathcal{F}}r_{\epsilon,c}^*,\end{aligned} \tag{36}$$

which proves (22) since the first term of the RHS of (36) is nonnegative.

IV. Further Discussions

A. Suboptimal Algorithms

Solving MWM optimization problem can be NP-hard depending on the underlying interference model as indicated in [7]. 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 $(\mu_{mn}^{c,SUB}(t))$. For convenience, we also denote the scheduling parameters of ***ALG*** by $(\mu_{mn}^{c,OPT}(t))$. Algorithms are called *suboptimal* if the scheduling parameters $(\mu_{mn}^{c,SUB}(t))$ satisfy the following:

$$\sum_{m,n}\mu_{mn}^{c_{mn}^*(t),SUB}(t)w_{mn}(t) \ge \gamma\sum_{m,n}\mu_{mn}^{c_{mn}^*(t),OPT}(t)w_{mn}(t), \tag{37}$$

where $\gamma \in (0,1)$ is constant and we recall that $c_{mn}^*(t)$ and $w_{mn}(t)$ are defined in Section III.A. In addition, the congestion controller of suboptimal algorithms is the same as that of ***ALG*** (16).

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

$$\begin{aligned}q_M > \frac{2N-1+\mu_M^2}{2\gamma\epsilon}+\mu_M \text{ and } \rho_c > \frac{Nq_M}{\gamma r_{\epsilon,c}^*}\ \forall c\in\mathcal{F},\\ \exists\epsilon_2>0 \text{ s.t. } \gamma r_{\epsilon,c}^* \ge a_c+\epsilon_2\ \forall c\in\mathcal{F},\end{aligned} \tag{38}$$

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

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{U_{s(c)}^c(\tau)+X_c(\tau)+Z_c(\tau)\} \le \frac{\bar{B}}{\delta}, \tag{39}$$

where $\bar{B} \triangleq B + \gamma V B_R$. In addition, a suboptimal algorithm can achieve a throughput

$$\liminf_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(\tau)\} \geq \gamma \sum_{c\in\mathcal{F}} r^*_{\epsilon,c} - \frac{B}{V}. \quad (40)$$

Proof: The proof is provided in Appendix A. ■

Remark 3: 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 [15] 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 [28] with $\gamma = \frac{1}{\Delta}$, where Δ is the maximum 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 (16) 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 subsection, 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 μ_M $\forall c \in \mathcal{F}$.[2] Let $L_c(t)$ denote the backlog of flow c data at the transport layer which is updated as follows:

$$L_c(t+1) = \min\{[L_c(t) + A_c(t) - \mu^c_{s(c)b(c)}(t)]^+, L_M\}, \quad (41)$$

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.

Following the idea introduced in [2], we construct a virtual queue $Y_c(t)$ and an auxiliary variable $v_c(t)$ for each input rate $R_c(t)$, with queue dynamics for $Y_c(t)$ as follows

$$Y_c(t+1) = [Y_c(t) - R_c(t)]^+ + v_c(t), \quad (42)$$

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 (16) 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:

$$\min_{0\leq v_c(t)\leq \mu_M} v_c(t)(\eta Y_c(t) - V), \quad (43)$$

$$\min_{R_c(t)} R_c(t)\left(\frac{q_M - \mu_M}{q_M} U^c_{s(c)}(t) - \eta Y_c(t) - X_c(t)\rho_c - Z_c(t)\right) \quad (44)$$

$$\text{s.t.} \qquad 0 \leq R_c(t) \leq \min\{L_c(t) + A_c(t), \mu_M\}$$

where $\eta > 0$ is a weight associated with the virtual queue $Y_c(t)$. Note that (43) and (44) 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 III.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. And we present a theorem below for the performance of the algorithm:

Theorem 3: Given that

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

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

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

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

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

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

Proof: The proof is provided in Appendix B. ■

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 (16) is performed at the transport layer and link weight assignment in (18) 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 (16) and employ delayed queue backlog of $(U^c_{s(c)}(t))$ for links in $\mathcal{L}$ in (18). Specifically, we rewrite (16) in ***ALG*** as:

$$\min R_c(t)\left(\frac{(q_M - \mu_M)U^c_{s(c)}(t)}{q_M} - X_c(t-T)\rho_c - Z_c(t) - V\right), \quad (45)$$

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

where T is an integer number that is larger than the maximum propagation delay from a source to a node, and we rewrite (18) as:

$$w_{mn}^c(t)=\begin{cases}\dfrac{U_{s(c)}^c(t-T)}{q_M}[U_m^c(t)-U_n^c(t)], \\ \qquad\qquad \text{if } (m,n)\in\mathcal{L}, \\ \dfrac{U_{s(c)}^c(t)}{q_M}[q_M-\mu_M-U_{b(c)}^c(t)], \\ \qquad\qquad \text{if } (m,n)=(s(c),b(c)), \\ 0, \qquad\qquad \text{otherwise.}\end{cases} \tag{46}$$

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

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

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

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

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

$$\liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(\tau)\}\ge\sum_{c\in\mathcal{F}}r_{\epsilon,c}^*-\frac{B_3}{V}.$$

Proof: The proof is provided in Appendix C. ■

On employing delayed queue backlogs, we can extend the centralized optimization problem (17) to distributed implementations in much the same way as [8][29][30].

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

Recall that in the model description in Section II, the link capacity is assumed constant (one packet per slot) and node-exclusive model is employed. In this subsection, 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 $(\mu_{mn}^c(t))_{(m,n)\in\mathcal{L}}\in I(t)$, 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 $\sum_{c\in\mathcal{F}}\mu_{mn}^c(t)\le l_{mn}$, where l_{mn} is the arbitrarily chosen link capacity for a link $(m,n)\in\mathcal{L}$. We define $l_n\triangleq\max_{(\mu_{mn}^c(t))\in I(t)}\sum_{c\in\mathcal{F}}\sum_{m:(m,n)\in\mathcal{L}}\mu_{mn}^c(t)$. Note that it is clear that $l_n\le\sum_{m:(m,n)\in\mathcal{L}}l_{mn}$. Then we can update the optimization (17) and weight assignment (18), respectively, as follows:

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

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

$$w_{mn}^c(t)=\begin{cases}\dfrac{U_{s(c)}^c(t)}{q_M}[U_m^c(t)-U_n^c(t)-l_n], \text{ if } (m,n)\in\mathcal{L}, \\ \dfrac{U_{s(c)}^c(t)}{q_M}[q_M-\mu_M-U_{b(c)}^c(t)], \\ \qquad\qquad \text{if } (m,n)=(s(c),b(c)), \\ 0, \qquad\qquad \text{otherwise.}\end{cases}$$

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.

V. Numerical Results

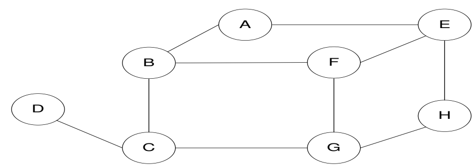

Fig. 1. Network topology for simulations

In this section, we present the simulation results for the proposed optimal algorithm ***ALG***. Simulations are run in Matlab 2009A with results averaged over 10^5 time slots. In the network topology illustrated in Figure 1, 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 rate for the three flows are all set to 0.1. We denote by *BP* the back-pressure scheduling algorithm in [1] with a congestion controller in [2], and denote by *Finite Buffer* the cross-layer algorithm developed in [14] with buffer size equal to the queue length limit q_M. Note that it is shown in simulation results in [14] 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 set the control parameter $V=1000$, where in simulations we find that a higher V cannot further improve the throughput.

We first illustrate in Table I the throughput optimality of ***ALG*** when the sources are constantly backlogged. We loosen the delay constraint as $\rho_c=30q_M$. As we increase the control parameter q_M, the ***ALG*** achieves a throughput approaching the throughput of BP algorithm which 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 in Figure 2 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 (20) is actually quite loose, and thus our algorithm can achieve better delay performance than stated in (20). Figure 2 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 that of the optimal BP algorithm when arrival rates are small (≤ 0.3). Specifically, when the arrival rate is 0.3, ***ALG***

TABLE I
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

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***.

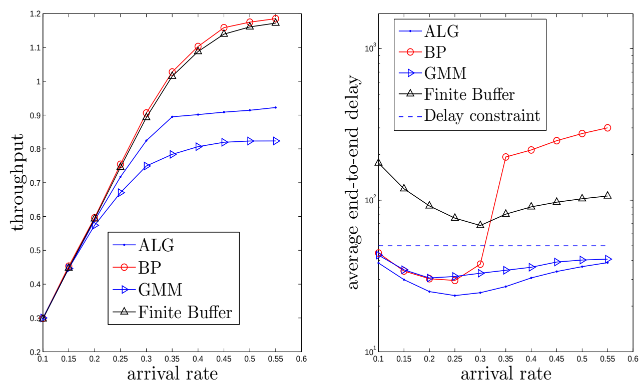

Fig. 2. Throughput and delay tradeoff under Alg. with performances compared to Finite Buffer algorithm and BP algorithm, with varying arrival rates at the transport layer.

VI. Conclusions and Future works

In this paper, we proposed a cross-layer framework which approaches the optimal throughput arbitrarily close for a general multi-hop 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 multi-hop 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.

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] 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.

[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. Ying, R. Srikant and D. Towsley, "Cluster-based back-pressure routing algorithm", in *Proc. IEEE INFOCOM08*, April 2008.

[5] M. Neely and E. Modiano, "Dynamic power allocation and routing for time varying wireless networks", in *IEEE Journal on Selected Area in Communications*, vol.23, no.1, pp. 89-103, March 2005.

[6] 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.

[7] G. Sharma, R. Mazumdar and N. Shroff, "On the complexity of scheduling in wireless neworks", in *Proc. of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom'06)*, 2006, pp. 227-238.

[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, 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.

[11] M. Lotfinezhad, B. Liang and E. Sousa, "On stability region and delay performance of linear-memory randomized scheduling for time-varing networks", in *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1860-1873, December 2009.

[12] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and MAC for stability and fairness in wireless networks", in *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1514-1524, August 2006.

[13] 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.

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

[15] C. Joo, X. Lin and N. Shroff, "Greedy maximal matching: performance limits for arbitrary network graphs under the node-exclusive interference model", in *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2734-2744, December 2009.

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

[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 INFOCOM2010*, March 2010.

[18] J. Ryu, L. Ying and S. Shakkottai, "Back-pressure routing for intermittently connected networks", in *IEEE INFOCOM2010 Mini-Conference*, March 2010.

[19] 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.

[20] G. Gupta and N. Shroff, "Delay analysis for wireless networks with single hop traffic and general interference constraints", in *IEEE/ACM Transactions on Networking*, 2010.

[21] I. Hou and P. Kumar, "Utility maximization for delay constrainted QoS in wireless", in *Proc. IEEE INFOCOM 2010*, March 2010.

[22] I. Hou and P. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels", in *Proc. IEEE INFOCOM 2010*, March 2010.

[23] I. Hou and P. Kumar, "A theory of QoS for wireless", in *Proc. IEEE INFOCOM09*, April 2009.

[24] I. Hou and P. Kumar, "Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels", in *Proc. ACM MobiHoc*, pp. 175-184, 2009.

[25] M. Neely, "Delay-based network utility maximization", in *Proc. IEEE INFOCOM 2010*, March 2010.

[26] 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.

[27] K. Jung and D. Shah, "Low delay scheduling in wireless networks", in *IEEE International Symposium on Information Theory (ISIT)*, June 2007.

[28] S. Sakai, M. Togasaki and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem", in *Discrete Applied Mathematics*, vol. 126, pp. 313-322, 2003.

[29] 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*.

[30] 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.

Appendix A
Proof of Theorem 2

Proof: Let $\Delta^{SUB}(t)$ denote the corresponding Lyapunov drift of a suboptimal algorithm which takes the same form as (24). By analyzing (30)(31) which also hold for suboptimal algorithms, we note that the second term of RHS of (30) is always non-positive ensured by the congestion controller (16). Employing (37) to (30)(31), we derive the following

$$\begin{aligned}
&\Delta^{SUB}(t) - V\sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)|\mathbf{Q}(t)\} \\
\leq& B + \gamma \sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M} \\
&\qquad - X_c(t)\rho_c - Z_c(t) - V)|\mathbf{Q}(t)\} \\
&+ Nq_M \sum_{c\in\mathcal{F}} X_c(t) + \sum_{c\in\mathcal{F}} a_c Z_c(t) \\
&+ \frac{1}{2}\sum_{c\in\mathcal{F}} \frac{(2N-1+\mu_M^2)U^c_{s(c)}(t)}{q_M} \\
&- \gamma\mathbb{E}\{\frac{q_M-\mu_M}{q_M}\sum_{c\in\mathcal{F}} U^c_{s(c)}(t)\mu^{c,SUB}_{s(c)b(c)}(t) \\
&+ \sum_{c\in\mathcal{F}}\sum_{n\in\mathcal{N}} \frac{U^c_n(t)U^c_{s(c)}(t)}{q_M} \\
&\quad (\sum_j \mu^{c,SUB}_{nj}(t) - \sum_i \mu^{c,SUB}_{in}(t))|\mathbf{Q}(t)\},
\end{aligned} \tag{47}$$

Following the steps in proving (32), we have from (47)

$$\begin{aligned}
&\Delta(t)^{SUB} - V\sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)|\mathbf{Q}(t)\} \\
\leq& B - V\gamma\sum_{c\in\mathcal{F}} r^*_{\epsilon,c} \\
&- \sum_{c\in\mathcal{F}} \frac{U^c_{s(c)}(t)}{q_M}(\gamma\epsilon(q_M-\mu_M) - \frac{2N-1+\mu_M^2}{2}) \\
&- \sum_{c\in\mathcal{F}} (\gamma r^*_{\epsilon,c} - a_c)Z_c(t) - \sum_{c\in\mathcal{F}}(\gamma\rho_c r^*_{\epsilon,c} - Nq_M)X_c(t).
\end{aligned} \tag{48}$$

Employing the conditions (38) and following the steps in proving (35) and (36), we can prove Theorem 2. ∎

Appendix B
Proof of Theorem 3

Before we proceed to the proof, we extend the stationary randomized algorithm STAT introduced in Lemma 2 and Remark 2. Given (θ_c) introduced in Lemma 2 and given flow c at node n, recall that $(A_c(t))$ is i.i.d. with mean (λ_c) and $(\lambda_c) > (\theta_c)$ element-wise. The flow control for STAT can be given as: Admit $\mu^{c,STAT}_{s(c)b(c)}(t) = A_c(t)$ w.p. $\frac{\theta_c}{\lambda_c}$; otherwise, $\mu^{c,STAT}_{s(c)b(c)}(t) = 0$. Then $\mathbb{E}\{\mu^{c,STAT}_{s(c)b(c)}(t)\} = \theta_c$, $\forall t$. Now take $v^{STAT}_c(t) = R^{STAT}_c(t) = \mu^{c,STAT}_{s(c)b(c)}(t)\ \forall c \in \mathcal{F}$. Then we also have $\mathbb{E}\{v^{STAT}_c(t)\} = \mathbb{E}\{R^{STAT}_c(t)\} = \theta_c$. Note that $R^{STAT}_c(t) \leq A_c(t) \leq \min\{L_c(t) + A_c(t), \mu_M\}$ and $v^{STAT}_c(t) \leq \mu_M$.

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 $\mathbf{Q}'(t) = (\mathbf{Q}(t), (Y_c(t)))$. From the virtual queue dynamics (42) and Lemma 1, we have

$$\begin{aligned}
&\frac{\eta}{2}\sum_{c\in\mathcal{F}}(Y_c(t+1)^2 - Y_c(t)^2) \\
\leq& \frac{\eta}{2}\sum_{c\in\mathcal{F}}(R_c(t)^2 + v_c(t)^2 - 2Y_c(t)(R_c(t) - v_c(t))) \\
\leq& K\eta\mu_M^2 - \sum_{c\in\mathcal{F}} \eta Y_c(t)(R_c(t) - v_c(t)).
\end{aligned} \tag{49}$$

Following the steps in deriving (30)(31), we have

$$\begin{aligned}
&\Delta'(t) - V\sum_{c\in\mathcal{F}} \mathbb{E}\{v_c(t)|\mathbf{Q}'(t)\} \\
\leq& B_1 + \sum_{c\in\mathcal{F}} \mathbb{E}\{v_c(t)(\eta Y_c(t) - V)|\mathbf{Q}'(t)\} \\
&+ \sum_{c\in\mathcal{F}} \mathbb{E}\{R_c(t)(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M} \\
&\qquad - \eta Y_c(t) - X_c(t)\rho_c - Z_c(t))|\mathbf{Q}'(t)\} \\
&+ Nq_M\sum_{c\in\mathcal{F}} X_c(t) + \sum_{c\in\mathcal{F}} a_c Z_c(t) \\
&+ \frac{1}{2}\sum_{c\in\mathcal{F}} \frac{(2N-1+\mu_M^2)U^c_{s(c)}(t)}{q_M} \\
&- \mathbb{E}\{\sum_{c\in\mathcal{F}}\sum_{(m,n)\in\mathcal{L}} \mu^c_{mn}(t)\frac{U^c_{s(c)}(t)}{q_M}(U^c_m(t) - U^c_n(t)) \\
&+ \sum_{c\in\mathcal{F}} \mu^c_{s(c)b(c)}(t)\frac{U^c_{s(c)}(t)}{q_M}(q_M - \mu_M - U^c_{b(c)}(t))|\mathbf{Q}'(t)\},
\end{aligned} \tag{50}$$

The second term, third term and the last term of the RHS of (50) are minimized by the congestion controller (43), (44) and the scheduling policy (17), respectively, over a set of feasible algorithms including the stationary randomized algorithm STAT. Substitute into the second term of RHS of (50) 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. ■

APPENDIX C
PROOF OF THEOREM 4

Proof: According to queue dynamics (6)(9), we obtain

$$\begin{aligned}&U^c_{s(c)}(t)-\mu_M T\le U^c_{s(c)}(t-T)\le U^c_{s(c)}(t)+\mu_M T,\\&X_c(t)-Nq_M T\le X_c(t-T)\le X_c(t)+\rho_c\mu_M T.\end{aligned}\tag{51}$$

Employing the above inequalities to (30)(31), we have

$$\begin{aligned}&\Delta(t)-V\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)|\mathbf{Q}(t)\}\\&\le B+\sum_{c\in\mathcal{F}}\mathbb{E}\{R_c(t)(\frac{(q_M-\mu_M)U^c_{s(c)}(t)}{q_M}\\&\qquad\qquad -X_c(t-T)\rho_c-Z_c(t)-V)|\mathbf{Q}(t)\}\\&+Nq_M\sum_{c\in\mathcal{F}}X_c(t)+\sum_{c\in\mathcal{F}}a_cZ_c(t)+K\rho_c^2\mu_M^2T+\frac{1}{2}KN\mu_MT\\&+\frac{1}{2}\sum_{c\in\mathcal{F}}\frac{(2N-1+\mu_M^2)U^c_{s(c)}(t)}{q_M}\\&-\mathbb{E}\{\sum_{c\in\mathcal{F}}\sum_{(m,n)\in\mathcal{L}}\mu^c_{mn}(t)\frac{U^c_{s(c)}(t-T)}{q_M}(U^c_m(t)-U^c_n(t))\\&+\sum_{c\in\mathcal{F}}\mu^c_{s(c)b(c)}(t)\frac{U^c_{s(c)}(t)}{q_M}(q_M-\mu_M-U^c_{b(c)}(t))|\mathbf{Q}(t)\}.\end{aligned}$$

The second term and the last term of the RHS of the above inequality are minimized by the congestion controller (45) and the scheduling policy (17) with weight assignment (46), respectively, over a set of feasible algorithms including the stationary randomized algorithm STAT. Substitute into the second term of RHS 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, employing the inequalities (51) and following the steps in proving Theorem 1, we can prove Theorem 4. ■