Optimal Control of Wireless Networks with Finite Buffers

Long Bao Le, *Member, IEEE,* Eytan Modiano, *Senior Member, IEEE,* and Ness B. Shroff, *Fellow, IEEE*

***Abstract*—This paper considers network control for wireless networks with finite buffers. We investigate the performance of joint flow control, routing, and scheduling algorithms which achieve high network utility and deterministically bounded backlogs inside the network. Our algorithms guarantee that buffers inside the network never overflow. We study the tradeoff between buffer size and network utility and show that under the one hop interference model if internal buffers have size $(N-1)/(2\epsilon)$ then ϵ-optimal network utility can be achieved, where ϵ is a control parameter and N is the number of network nodes. The underlying scheduling/routing component of the considered control algorithms requires ingress queue length information (IQI) at all network nodes. However, we show that these algorithms can achieve the same utility performance with delayed ingress queue length information at the cost of a larger average backlog bound. We also show how to extend the results to other interference models and to wireless networks with time varying link quality. Numerical results reveal that the considered algorithms achieve nearly optimal network utility with a significant reduction in queue backlog compared to existing algorithms in the literature.**

***Index Terms*—Network control, wireless scheduling, flow control, routing, delay control, throughput region, finite buffer, utility maximization**

I. Introduction

The design of wireless networks that efficiently utilize network capacity and provide quality of service guarantees for end users is one of the most important problems in network theory and engineering. Since the seminal paper of Tassiulas and Ephremides [1] in which they proposed a joint routing and scheduling algorithm that achieves the maximum network throughput, significant efforts have been invested in developing more efficient network control algorithms [2]-[22]. Most existing works, however, focus on achieving a guaranteed fraction of the maximum throughput region with low communication and computation complexity.

In addition, most papers on network control assume that all buffers in the network are infinite, so buffer overflow never occur. In practice, network buffers are finite. Therefore, sizing buffers such that buffer overflow inside the network can be alleviated or completely avoided is an important engineering problem. Moreover, buffer sizing should be performed in such a way that network capacity is not wasted. In fact, networks with finite buffers may suffer from significant throughput loss if they are not designed appropriately (e.g., see [23] and references therein).

There have been some recent papers that analyze delay performance of cross-layer scheduling algorithms [24]-[28]. In particular, it was shown that the well-known maximum weight scheduling algorithm achieves order-optimal delay in the uplink-downlink of cellular networks [24] and in most practical large-scale multihop wireless networks [25]. Other works on delay analysis for different scheduling algorithms in wireless networks can be found in [27]-[28]. In [4], [26], [29], and [30], it was shown that by combining the principle of shortest-path routing and differential backlog routing, end-to-end delay performance can be improved. In [15], [32], the virtual queue technique was used to improve network delay performance. In [33], it was shown that it is possible to achieve a guaranteed stability region with bounded information delay if we allow a small reduction in achievable network utility. These existing works, however, do not consider the problem of providing backlog or delay performance guarantees.

In this paper, we employ flow controllers to deterministically bound queue backlogs inside the network. Specifically, we combine the Lyapunov optimization technique of [2], [3], [30] and the scheduling mechanism proposed in [6] to construct joint flow control, routing and scheduling algorithms for wireless networks with finite buffers. Note that in [2], [3], [30] the problem of network utility maximization is considered assuming that all buffers in the network are infinite. The authors of [6] proposed scheduling algorithms for networks with finite buffers. However, this work does not consider flow control or dynamic routing, and requires that the traffic arrival rates are strictly within the feasible throughput region. Moreover, [31] demonstrated that it is possible to achieve deterministically bounded queue backlogs for the throughput optimization problem (i.e., linear utility), without taking into account general utility maximization. Our current paper considers the general setting where traffic arrival rates can be either inside or outside the throughput region, internal buffers in the network are finite, and dynamic routing is used to achieve the largest possible network throughput. Our contributions can be summarized as follows.

- We consider control algorithms that achieve high network utility and deterministically bounded backlogs for all buffers inside the network. Moreover, these algorithms ensure that internal buffers never overflow.
- We demonstrate a tradeoff between buffer sizes and achievable network utility under the one hop interference model.
- We show that delayed ingress queue information does

Manuscript received March 03 2011; revised August 18 2011 and November 7, 2011; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor L. Andrew. This work was supported by ARO Muri grant number W911NF-08-1-0238, NSF grant numbers CNS-0626781, DTRA grant HDTRA1-07-1-0004, NSERC Postdoctoral fellowship, and NSERC Discovery Grant. The preliminary results of this paper was presented in part at IEEE INFOCOM'2010.

Long Bao Le is with INRS-EMT, University of Quebec, Montreal, QC, Canada (e-mail: long.le@emt.inrs.ca).

Eytan Modiano is with Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA, USA (e-mail: modiano@mit.edu).

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

not affect the utility of the control algorithms albeit at the cost of a larger backlog bound.

- We extend the obtained results to other interference models and to wireless networks with time varying link quality
- We show via simulations that the considered control algorithms perform very well in both the under and over-loaded traffic regimes. Specifically, they achieve nearly optimal utility performance with very low and bounded backlogs.

The remainder of this paper is organized as follows. The system model is described in section II. In section III, we analyze the performance of the control algorithm in the heavy traffic regime. Section IV focuses on the performance analysis of the control algorithm for arbitrary traffic arrival rates. Some extensions are presented in section V. Numerical results are presented in section VI followed by conclusion in section VII.

II. System Model

We consider a wireless network which is modeled as a graph $G = (\Gamma, E)$ where Γ is the set of nodes and E is the set of links. Let N and L be the number of nodes and links in the network, respectively. We assume a time-slotted wireless system where packet arrivals and transmissions occur at the beginning of time slots of unit length. There are multiple network flows in the network each of which corresponds to a particular source-destination pair.

Arrival traffic is stored in input reservoirs and flow controllers are employed at source nodes to determine the amount of traffic to admit from input reservoirs into the network in each time slot. Let n_c be the source node and d_c be the destination node of flow c. We will refer to the queue at the source node n_c, which "stores" admitted traffic of flow c as an ingress buffer. It is worth emphasizing that we do not need physical buffers to implement these ingress queues in practice. Specifically, the backlog values of these ingress buffers can be simply maintained by software counters while all data packets are physically stored in the input reservoirs.

All other buffers storing packets of flow c inside the network are called internal buffers. Let $R_{n_c}^{(c)}(t)$ be the amount of traffic of flow c injected from the input reservoir into the network at node n_c in time slot t. Note that a particular node can be a source node for several flows. Let C_n be the set of flows whose source node is n. Hence, for any flow $c \in C_n$ its source node n_c is node n. It is assumed that $\sum_{c \in C_n} R_n^{(c)} \leq R_n^{\mathsf{max}}$ where the parameter R_n^{max} can be used to control the burstiness of admitted traffic from node n into the network. Let $R_{\mathsf{max}} = \max_{\{n\}} \{R_n^{\mathsf{max}}\}$, which will be used in the analysis. Let C denote the total number of flows in the network.

Each internal node maintains multiple finite buffers (one per flow) while ingress buffers at all source nodes are assumed to be unlimited. This implies that the input reservoirs are unlimited in size because data packets at source nodes are physically stored in the input reservoirs. This assumption is justified by the fact that in many wireless networks (e.g., wireless sensor networks) buffer space is limited. However, buffers in ingress routers or devices are relatively large. Moreover, since input buffers only need to store traffic from a small number of end-users, they can be made large enough to accommodate all incoming traffic with high probability.

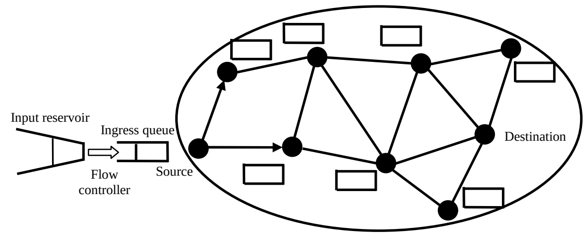

Fig. 1. Wireless network with finite internal buffers

Let l_c be the size of the internal buffer used to store packets of flow c at each network node. We denote the *queue length*[1] of flow c at node n at the beginning of time slot t by $Q_n^{(c)}(t)$. Note that data packets of any flow are delivered to the higher layer upon reaching the destination node, so $Q_{d_c}^{(c)}(t) = 0$. Assume that the capacity of any link is one packet per time slot. In addition, let $\mu_{nm}^{(c)}(t)$ be the number of packets of flow c transmitted over link (n,m) in time slot t. Therefore, $\mu_{nm}^{(c)}(t) = 1$ if we transmit a packet of flow c over link (n,m) and $\mu_{nm}^{(c)}(t) = 0$, otherwise. In the following, we will use $\mu_{nm}^{(c)}(t)$ or $\mu_l^{(c)}(t)$ to denote the number of packet transmitted over link (n,m) or link l, respectively (i.e., network links can be represented by the corresponding transmitting and receiving nodes or just by a single letter). Let Ω_n^{in} and Ω_n^{out} be the set of incoming and outgoing links at node n. The network model is illustrated in Fig. 1. For notational convenience, when there is no ambiguity, we omit the time index t in related variables.

We assume that the traffic of any flow is not routed back to its source node. Therefore, we have $\mu_{mn_c}^{(c)} = 0, \forall m, c$. It is clear that this restriction does not impact the achievable throughput of the considered control algorithms. We assume that a node can communicate with (i.e., transmit or receive) at most one neighboring node. Any link can be activated as long as no node is involved in more than one transmission or reception (i.e., node exclusive interference constraints).[2]

We further assume that node n will not transmit data of flow c along any link (n,m) whenever $Q_n^{(c)} < 1$ (i.e., a node will not transmit traffic of any flow if the corresponding queue does not have enough data to fill the link capacity). Under this assumption, the queue evolutions can be written as

$$Q_n^{(c)}(t+1) = Q_n^{(c)}(t) - \sum_{l \in \Omega_n^{\mathsf{out}}} \mu_l^{(c)}(t) + \sum_{l \in \Omega_n^{\mathsf{in}}} \mu_l^{(c)}(t) + R_n^{(c)}(t), \quad (1)$$

where $R_n^{(c)}(t) = 0, \ \forall t$ and $n \neq n_c$. Note that $\sum_{l \in \Omega_n^{\mathsf{out}}} \mu_l^{(c)}(t) = 1$ only if $Q_n^{(c)}(t) \geq 1$ and one of the outgoing links of node n is activated for flow c. Let $\overline{r}_n^{(c)}(t)$ be the time average rate of admitted traffic for flow c at the

[1] To be precise, $Q_n^{(c)}(t)$ is a virtual queue length because it can take real numbers for source buffers.

[2] This assumption is made for simplicity of the derivations. Extensions to other interference models (e.g., k-hop interference model) are discussed later in section V. Note that the k-hop interference model implies that no two links within k hops of each other can be simultaneously active. The node exclusive interference model corresponds to the 1-hop interference model.

corresponding source node n_c up to time t, that is

$$\overline{r}_{n_c}^{(c)}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} E\left\{R_{n_c}^{(c)}(\tau)\right\}. \tag{2}$$

The long-term time-average admitted rate for flow c is defined as

$$\overline{r}_{n_c}^{(c)} \triangleq \lim_{t\to\infty} \overline{r}_{n_c}^{(c)}(t). \tag{3}$$

Now, we recall the definitions of network stability and the maximum achievable throughput region (or throughput region for brevity) [1], which will be used in our analysis. A queue for a particular flow c at node n is called stable if its average backlog is bounded. In addition, the network is called stable if all individual queues in the network are stable. The maximum achievable throughput region Λ of a wireless network with unlimited input and internal buffers is the set of all traffic arrival rate vectors for which there exists a network control algorithm to stabilize all individual queues in the network.

Note that when internal buffers are finite, stability corresponds to maintaining bounded backlogs in ingress buffers (since internal buffers are finite by design, the issue of stability does not arise there). In this context, it is necessary to devise a control algorithm that achieves high throughput and/or utility without any overflow at internal buffers.

III. Network Optimization in the Heavy Traffic Regime

We start by considering the case where all sources are constantly backlogged. We seek a balance between optimizing the total network utility and bounding total queue backlog inside the network. Specifically, we want to solve the following optimization problem

$$\text{maximize} \quad \sum_c g^{(c)}\left(\overline{r}_{n_c}^{(c)}\right) \tag{4}$$

$$\text{subject to} \quad \left(\overline{r}_{n_c}^{(c)}\right) \in \Lambda, \tag{5}$$

$$Q_n^{(c)} \leq l_c,\ \forall c, \text{ and } n \neq n_c, \tag{6}$$

where $g^{(c)}(.)$ are assumed to be non-negative, increasing and strictly concave utility functions, $\overline{r}_{n_c}^{(c)}$ is the time average admitted rate for flow c at node n_c, $\left(\overline{r}_{n_c}^{(c)}\right) = \left(\overline{r}_{n_1}^{(1)}, \overline{r}_{n_2}^{(2)}, \cdots, \overline{r}_{n_C}^{(C)}\right)^T$ is the time average admitted rate vector, $(.)^T$ denotes the vector transposition, and l_c is the internal buffer size. Here, utility functions express the level of satisfaction of users with respect to admitted rates. Constraint (6) ensures that the backlogs in internal buffers are finite and bounded by l_c at all times.

To quantify the performance of the control algorithms, we need some more definitions. First, let us define the ϵ-stripped throughput region as follows:

$$\Lambda_\epsilon \triangleq \left\{\left(r_{n_c}^{(c)}\right) \mid \left(r_{n_c}^{(c)}+\epsilon\right) \in \Lambda\right\}, \tag{7}$$

where $\left(r_{n_c}^{(c)}\right) \triangleq \left(r_{n_1}^{(1)}, r_{n_2}^{(2)}, \cdots, r_{n_C}^{(C)}\right)^T$. Also, let $(r_{n_c}^{*(c)}(\epsilon))$ be an ϵ-optimal solution of a general network optimization problem, which is defined as the solution of the following optimization problem

$$\text{maximize} \quad \textstyle\sum_c g^{(c)}\left(r_{n_c}^{(c)}\right) \tag{8}$$

$$\text{subject to} \quad \left(r_{n_c}^{(c)}\right) \in \Lambda_\epsilon, \tag{9}$$

$$r_{n_c}^{(c)} \leq \lambda_{n_c}^{(c)}, \tag{10}$$

where $\left(\lambda_{n_c}^{(c)}\right) = \left(\lambda_{n_1}^{(1)}, \lambda_{n_2}^{(2)}, \cdots, \lambda_{n_C}^{(C)}\right)^T$ is the average traffic arrival rate vector. We will quantify the performance of the considered control algorithms in terms of $(r_{n_c}^{*(c)}(\epsilon))$[3]. Note that $(r_{n_c}^{*(c)}(\epsilon))$ tends to the optimal solution $(r_{n_c}^{*(c)})$ as $\epsilon \to 0$ where $(r_{n_c}^{*(c)})$ is the optimal solution of the optimization problem (8)-(10) where Λ_ϵ is replaced by Λ (i.e., the original throughput region). Also, for constantly backlogged sources the constraints on traffic arrival rates (10) are not needed.

In [3], [30], the corresponding problem without the buffer limit is considered. As a result, queue backlogs inside the network at internal buffers can be very large (although with bounded expectations). The large backlog accumulation inside the network is not desirable because it can lead to increased delays, buffer overflows, and throughput reduction due to retransmission of lost packets. Now, consider the following algorithm that provides a feasible solution for the problem formulated in (4)-(6).

Algorithm 1: Constantly Backlogged Sources

1) Flow Control: Each node n_c injects an amount of traffic into the network which is equal to $R_{n_c}^{(c)}(t) = x_{n_c}^{(c)}$ where $x_{n_c}^{(c)}$ is the solution of the following optimization problem

$$\begin{array}{ll} \text{maximize}_{\left\{x_{n_c}^{(c)}\right\}} & \sum_c \left[V g^{(c)}\left(x_{n_c}^{(c)}\right) - 2Q_{n_c}^{(c)} x_{n_c}^{(c)}\right] \\ \text{subject to} & 0 \leq \sum_{d\in C_{n_c}} x_{n_c}^{(d)} \leq R_{n_c}^{\max},\ \forall c \end{array}, \tag{11}$$

where V is a controlled parameter.

2) Routing/Scheduling: Each link (n,m) calculates the differential backlog for flow c as follows:

$$dB_{n,m}^{(c)} \triangleq \begin{cases} \frac{Q_{n_c}^{(c)}}{l_c}\left[l_c - Q_m^{(c)}\right], & \text{if } n = n_c \\ \frac{Q_{n_c}^{(c)}}{l_c}\left[Q_n^{(c)} - Q_m^{(c)}\right], & \text{if } n,m \neq n_c \end{cases}. \tag{12}$$

Then, link (n,m) calculates the maximum differential backlog as follows:

$$W_{(n,m)}^* \triangleq \max_c \left\{dB_{n,m}^{(c)}\right\}. \tag{13}$$

Let $\vec{S}$ be the schedule where its l-th component $S_l = 1$ if link l is scheduled and $S_l = 0$ otherwise. The schedule $\vec{S}^*$ is chosen in each time slot t as follows:

$$\vec{S}^* = \underset{\vec{S}\in\Phi}{\operatorname{argmax}}\, \vec{S}^T \vec{W}^*, \tag{14}$$

where Φ is the set of all feasible schedules as determined by the underlying wireless interference model. We will not schedule any link l with $W_l^* \leq 0$. For each scheduled link, one packet of the flow that achieves the maximum differential backlog in (12) is transmitted if the corresponding buffer has at least one packet waiting for transmission.

Before investigating the performance of this algorithm, we would like to make some comments. First, it can be observed

[3]Note that the ϵ-optimality notion in this paper may not be the same with that used in optimization theory literature.

that the flow controller of this algorithm admits less traffic of flow c if the corresponding ingress buffer is more congested (i.e., large $Q_{n_c}^{(c)}$). Also, a larger value V results in higher achievable utility albeit at the cost of larger queue backlogs. Moreover, the routing component of this algorithm (12) is an adaptation of the differential backlog routing [1] to the case of finite buffers [6]. The scheduling rule (14) is the well-known max-weight scheduling algorithm.

In fact, the flow controller of this algorithm is the same as that proposed in [30], [3] while the differential backlog in (12) is the same as that in [6]. However, [30], [3] assume that all network buffers are infinite while [6] assumes that traffic arrival rates are inside the corresponding throughput region and static routing. In contrast, we consider dynamic routing to achieve the largest possible network throughput. Also, we consider both heavy traffic and arbitrary arrival rates and employ flow controllers to control the utility-backlog tradeoff.

The differential backlog in (12) ensures that internal buffers never overflow. In fact, the differential backlog of any source link (n_c, m), given by $Q_{n_c}^{(c)}/l_c\left[l_c - Q_m^{(c)}\right]$, bounds the backlogs of all internal buffers by l_c. The differential backlogs of all other links (n, m), given by $Q_{n_c}^{(c)}/l_c\left[Q_n^{(c)} - Q_m^{(c)}\right]$, is essentially the multiplication of the standard differential backlog $\left[Q_m^{(c)} - Q_m^{(c)}\right]$ from [1] and the normalized ingress queue backlog $Q_{n_c}^{(c)}/l_c$. Incorporating ingress queue backlog into the differential backlog in (12) prioritizes the flow whose ingress queue is more congested. This helps stabilize ingress queues because the scheduling component of algorithm 1 still has the "max-weight" structure as that proposed in [1].

Assume that for each flow c, each wireless node maintains a buffer space of at least l_c packets. Assume that all buffers in the network are empty initially. Then, the queue backlogs in all internal buffers for flow c are always smaller than or equal to l_c. That is

$$Q_n^{(c)}(t) \leq l_c, \ \forall c, t \text{ and } n \neq n_c. \quad (15)$$

To see this, note that all internal buffers in the network always have an integer number of packets. This is because the scheduler of Algorithm 1 does not transmit a fraction of packet from any ingress buffer to the corresponding internal buffer. Because at most one packet can be transmitted from any queue in one time slot, buffer overflow can only occur at a particular internal buffer if the number of packets in that buffer is l_c and it receives one more packet from one of its neighboring nodes. We will show that this could not happen by considering the following cases.

- Consider any link (n_c, m) for a particular flow c. It can be observed that if $Q_m^{(c)} = l_c$ then the buffer at node m for flow c will never receive any more packets from node n_c. This is because the differential backlog of link (n_c, m) in this case is $dB_{n_c,m}^{(c)} = Q_{n_c}^{(c)}/l_c\left[l_c - Q_m^{(c)}\right] = 0$. Therefore, link (n_c, m) is not scheduled for flow c.
- Consider any link (n, m) for $n \neq n_c$. Suppose the first buffer overflow event in the network occurs at node m for flow c. Right before the time slot where this first buffer overflow occured, we must have $Q_m^{(c)} = l_c$. However, consider the differential backlog of any link (n, m) in this previous time slot, we have $dB_{n,m}^{(c)} = Q_{n_c}^{(c)}/l_c\left[Q_n^{(c)} - Q_m^{(c)}\right] = Q_{n_c}^{(c)}/l_c\left[Q_n^{(c)} - l_c\right] \leq 0$. This means any such link (n, m) will not be scheduled for flow c. Therefore, overflow could not occur at node m.

Therefore, we have deterministically bounded backlogs inside the network at all times. Now, we are ready to state one of the main results of this paper.

Theorem 1: Given $\epsilon > 0$, if the internal buffer size satisfies

$$l_c \geq \frac{N-1}{2\epsilon}. \quad (16)$$

Then, we have the following bounds for utility and ingress queue backlog

$$\liminf_{t\to\infty} \sum_c g^{(c)}\left(\overline{r}_{n_c}^{(c)}(t)\right) \geq \sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - \frac{D_1}{V}, \quad (17)$$

$$\overline{\sum_c Q_{n_c}^{(c)}} \leq \frac{D_1 + VG_{\mathsf{max}}}{2\lambda_{\mathsf{max}} - \frac{(N-1)}{l_c}}, \quad (18)$$

where recall that $(r_{n_c}^{*(c)}(\epsilon))$ is an ϵ-optimal solution, which is an optimal solution of the optimization problem (8)-(10), $D_1 \triangleq C(R_{\mathsf{max}}^2 + 2) + C(N-1)R_{\mathsf{max}}l_c$ is a finite number, C is the number of flows in the network, R_{max} is the maximum amount of traffic injected into the network from any node, V is a design parameter which controls the utility and backlog bound tradeoff, λ_{max} is the largest number such that $\vec{\lambda}_{\mathsf{max}} \in \Lambda$ where $\vec{\lambda} = (\lambda, \lambda, \cdots, \lambda)^T$ is a column vector with all elements equal to λ,

$$G_{\mathsf{max}} \triangleq \max_{\left\{\sum_{c\in C_n} r_n^{(c)} \leq R_n^{\mathsf{max}}\right\}} \left\{\sum_c g^{(c)}\left(r_{n_c}^{(c)}\right)\right\}, \quad (19)$$

$$\overline{\sum_c Q_{n_c}^{(c)}} \triangleq \limsup_{M\to\infty} \frac{1}{M} \sum_{\tau=0}^{M-1} \sum_c E\left\{Q_{n_c}^{(c)}(\tau)\right\}. \quad (20)$$

Proof: The proof is given in Appendix A. ■

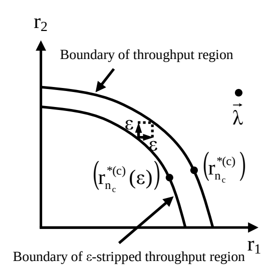

Fig. 2. Geometric illustration of optimal rate and lower-bound of achievable rate in the heavy load regime

Here, we would like to make some comments. First, as $V \to \infty$, the total utility achieved by the time-average admitted rate $\left(\overline{r}_{n_c}^{(c)}\right)$ is lower-bounded by that due to $\left(r_{n_c}^{*(c)}(\epsilon)\right)$ from (8)-(10). This lower-bound is parameterized by the design parameter ϵ, which in turn determines the queue backlog bound in all internal buffers as given by (16). Specifically, suppose we choose $l_c = (N-1)/(2\epsilon)$ as suggested by (16). Then, when ϵ is larger, the buffer size l_c is smaller but the utility lower-bound achieved by $\left(r_{n_c}^{*(c)}(\epsilon)\right)$ is also smaller as given in (17). This achievable admitted rate vector $\left(r_{n_c}^{*(c)}(\epsilon)\right)$

is illustrated in Fig. 2. Second, the inequality (18) gives the upper-bound for the total ingress queue backlog, which is controlled by another design parameter V. In particular, when V increases, the utility lower bound in (17) also increases but the upper-bound on the total ingress queue backlog becomes larger.

Remark 1: Without flow control, [6] has shown how to choose the internal buffer size to stabilize the input buffers. However, this chosen buffer size depends on the relative distance between the traffic arrival rate vector and the boundary of the throughput region. In practice, it is not easy to determine this distance parameter. Moreover, input queues will become unstable when the traffic arrival rate is outside the throughput region. In contrast, by appropriately designing a flow controller, our cross-layer algorithm can always stabilize the ingress queues. In addition, the internal buffer size in our design depends on a tunable design parameter ϵ, which only impacts the utility lower bound and does not depend on the traffic itself.

Remark 2: Given ϵ, if we choose the smallest possible buffer size suggested by Theorem 1 (i.e., $l_c = (N-1)/(2\epsilon)$), the backlog bound becomes

$$\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_{c} E\left\{Q_{n_c}^{(c)}(\tau)\right\} \leq \frac{D_1 + VG_{\mathsf{max}}}{2\left(\lambda_{\mathsf{max}} - \epsilon\right)}. \tag{21}$$

Taking $\epsilon = 0$, we have a backlog bound of same form as that derived in [3].

IV. Network Optimization with Arbitrary Arrival Rates

Now, we consider the more general case where traffic arrival rates $\left(\lambda_{n_c}^{(c)}\right)$ can be inside or outside the maximum throughput region. In this case, the long-term time average admitted rates should be constrained to be smaller than the average traffic arrival rates. For this case, our objective is to solve the following stochastic optimization problem

$$\begin{aligned} \text{maximize} \quad & \sum_c g^{(c)}\left(\overline{r}_{n_c}^{(c)}\right) && (22)\\ \text{subject to} \quad & \left(\overline{r}_{n_c}^{(c)}\right) \in \Lambda, && (23)\\ & 0 \leq \overline{r}_{n_c}^{(c)} \leq \lambda_{n_c}^{(c)}, && (24)\\ & Q_n^{(c)} \leq l_c,\ \forall c, \text{ and } n \neq n_c. && (25)\end{aligned}$$

In this network setting, because the traffic arrival rates may be inside the throughput region, we rely on newly-introduced variables $Y_n^{(c)}(t)$ that capture "virtual queues backlogs" to perform flow control. These virtual queues track the difference between the instantaneous admitted traffic rates $R_n^{(c)}(t)$ and the "potential" admitted rates $z_{n_c}^{(c)}(t)$. In particular, we consider the following control algorithm.

Algorithm 2: Sources with Arbitrary Arrival Rates

1) Flow Control: Each node n_c injects an amount of traffic of flow c into the network $R_{n_c}^{(c)}(t)$, which is the solution of the following optimization problem

$$\begin{aligned} \text{maximize}_{\left\{R_{n_c}^{(c)}(t)\right\}} \quad & \textstyle\sum_{c\in C_{n_c}} \left[Y_{n_c}^{(c)}(t) - Q_{n_c}^{(c)}(t)\right] R_{n_c}^{(c)}(t)\\ \text{subject to} \quad & \textstyle\sum_{d\in C_{n_c}} R_{n_c}^{(d)}(t) \leq R_{n_c}^{\mathsf{max}},\\ & R_{n_c}^{(c)}(t) \leq A_{n_c}^{(c)}(t) + L_{n_c}^{(c)}(t), \end{aligned} \tag{26}$$

where $L_{n_c}^{(c)}(t)$ is the backlog of the input reservoir, $A_{n_c}^{(c)}(t)$ is the number of arriving packets in time slot t, and $Y_{n_c}^{(c)}(t)$ represents "backlog" in the virtual queue which has the following queue-like evolution

$$Y_{n_c}^{(c)}(t+1) = \max\left\{Y_{n_c}^{(c)}(t) - R_{n_c}^{(c)}(t), 0\right\} + z_{n_c}^{(c)}(t), \tag{27}$$

where $z_{n_c}^{(c)}(t)$ are auxiliary variables which are calculated from the following optimization problem

$$\begin{aligned} \text{maximize}_{\left\{z_{n_c}^{(c)}(t)\right\}} \quad & \textstyle\sum_c \left\{Vg\left(z_{n_c}^{(c)}(t)\right) - 2Y_{n_c}^{(c)}(t) z_{n_c}^{(c)}(t)\right\}\\ \text{subject to} \quad & \textstyle\sum_{d\in C_{n_c}} z_{n_c}^{(d)}(t) \leq R_{n_c}^{\mathsf{max}}. \end{aligned} \tag{28}$$

We then update the virtual queue variables $Y_{n_c}^{(c)}$ according to (27) in every time slot. We update $L_{n_c}^{(c)}(t)$ by decreasing it by $R_{n_c}^{(c)}(t)$ and then increasing it by the number of arriving packets for flow c in time slot t.

2) Scheduling and routing are performed as in Algorithm 1.

The flow controller in this algorithm is the same as that in [2]. Its operation can be interpreted intuitively as follows. The auxiliary variables $z_{n_c}^{(c)}(t)$ plays the role of $R_{n_c}^{(c)}(t)$ in algorithm 1 for the heavy traffic regime. In fact, the optimization problem (28) from which we calculate $z_{n_c}^{(c)}(t)$ is similar to the one in (11) where $Q_{n_c}^{(c)}$ is replaced by $Y_{n_c}^{(c)}$. Hence, $z_{n_c}^{(c)}$ represents the potential rate that would have been admitted if sources were backlogged. The virtual queue $Y_{n_c}^{(c)}(t)$ captures the difference between the potential admitted rate $z_{n_c}^{(c)}(t)$ and the actual admitted rate $R_{n_c}^{(c)}(t)$ based on which the flow controller determines the amount of admitted traffic $R_{n_c}^{(c)}(t)$ from (26). Here, the "virtual differential backlogs" $\left[Y_{n_c}^{(c)} - Q_{n_c}^{(c)}\right]$ determines from which flow to inject data into the network.

In addition, it can be observed that (27) captures the dynamics of a virtual queue with service process $R_{n_c}^{(c)}(t)$ and arrival process $z_{n_c}^{(c)}(t)$. Hence, if these virtual queues are stable, then the time-average rates of $R_{n_c}^{(c)}(t)$ are equal to the time-average of $z_{n_c}^{(c)}(t)$. That means we must have $\overline{r}_{n_c}^{(c)} = \overline{z}_{n_c}^{(c)}$ where $\overline{z}_{n_c}^{(c)}$ denotes the time average of $z_{n_c}^{(c)}(t)$.

It can be verified that the amount of admitted traffic $R_{n_c}^{(c)}(t)$ calculated from (26) only takes integer values assuming that $R_{n_c}^{\mathsf{max}}$ are integer. Specifically, the solution of (26) can be calculated as follows. First, we sort the quantities $\left[Y_n^{(c)} - Q_n^{(c)}\right]$ for all flows c whose source nodes are node n. Then, starting from the flow with the largest positive value in the sorted list, we admit the largest possible amount of traffic considering the constraints in (26). Because $A_{n_c}^{(c)}$, $L_{n_c}^{(c)}$, and $R_{n_c}^{\mathsf{max}}$ are all integer, $R_{n_c}^{(c)}(t)$ will take integer values. Because $R_{n_c}^{(c)}(t)$ are integer, each ingress buffer is either empty or contains an integer number of packets. Assume that packet arrivals are i.i.d over time slots [4], we state the performance of Algorithm 2 in the following theorem.

Theorem 2: Given $\epsilon > 0$, if the internal buffer size satisfies

$$l_c \geq \frac{N-1}{2\epsilon}. \tag{29}$$

Then, we have the following bounds for utility and ingress queue backlog

$$\liminf_{M\to\infty} \sum_c E\left\{g^{(c)}\left(\overline{r}_{n_c}^{(c)}(M)\right)\right\} \geq \sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - \frac{D_2}{V}, \tag{30}$$

$$\overline{\sum_c Q_{n_c}^{(c)}} \leq \frac{D_2 + VG_{\mathsf{max}}}{2\lambda_{\mathsf{max}} - \frac{(N-1)}{l_c}}, \tag{31}$$

where again $(r_{n_c}^{*(c)}(\epsilon))$ is the ϵ-optimal solution, which is an optimal solution of the optimization problem (8)-(10), and $D_2 \triangleq 3CR_{\mathsf{max}}^2 + C(N-1)R_{\mathsf{max}}l_c$ is a finite number.

Proof: The proof is given in Appendix B. ■

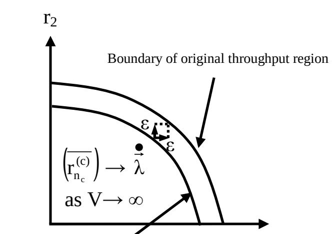

Fig. 3. Geometric illustration of optimal rate and lower-bound of achievable rate in the low load regime

Note that when $\left(\lambda_{n_c}^{(c)}\right) \in \Lambda_\epsilon$, we have $\left(r_{n_c}^{*(c)}(\epsilon)\right) = \left(\lambda_{n_c}^{(c)}\right)$ where $\left(\lambda_{n_c}^{(c)}\right)$ is the average traffic arrival rate vector. Therefore, we have $\left(\overline{r}_{n_c}^{(c)}(\epsilon)\right) \to \left(\lambda_{n_c}^{(c)}\right)$ as $V \to \infty$ in this case. That means we can achieve optimal utility in this case by letting $V \to \infty$. This observation is illustrated in Fig. 3. Otherwise, if $\left(\lambda_{n_c}^{(c)}\right) \neq \Lambda_\epsilon$ then we return to the heavy traffic case considered in the previous section. In this case, the lower-bound on the achieved utility depends on the chosen parameter ϵ.

V. Further Extensions

In this section, we show how to extend the results obtained in the previous sections in two important directions, namely the impact of delayed IQI and extensions to other interference models and to wireless networks with time varying link quality.

[4]While we assume i.i.d arrivals for simplicity, the results can be extended to more general arrivals that satisfy $E\left\{A_{n_c}^{(c)}(t)\right\} = \lambda_{n_c}^{(c)}$ for any time slot t. This condition is required so that (80) in Appendix B holds. Therefore, the results in Theorem 2 are valid for Markovian traffic.

A. Delayed Ingress Queue Length Information

Assume that the same scheduling/routing algorithm as in Algorithm 1 is employed. However, only delayed IQI is available at all network nodes. Let T be the time delay of ingress queue length $Q_{n_c}^{(c)}(t)$ at all other network nodes (i.e, only $Q_{n_c}^{(c)}(t-T)$ is available at other nodes in time slot t). In the following, we investigate the performance of Algorithm 1 when delayed IQI is used for scheduling. In particular, each link (n,m) and flow c calculate the differential backlog in slot t as follows:

$$dB_{n,m}^{(c)}(t) \triangleq \begin{cases} \frac{Q_{n_c}^{(c)}(t)}{l_c}\left[l_c - Q_m^{(c)}(t)\right], & \text{if } n = n_c \\ \frac{Q_{n_c}^{(c)}(t-T)}{l_c}\left[Q_n^{(c)}(t) - Q_m^{(c)}(t)\right], & \text{if } n, m \neq n_c \end{cases}.$$

This differential backlog is used for routing/scheduling while the same congestion controller as in Algorithm 1 is employed. Note that the flow controller for flow c at node n_c only requires its local ingress queue length information $Q_{n_c}^{(c)}(t)$. Hence, this queue length information is available without delay. We have the following results for this cross-layer algorithm with delayed IQI.

Theorem 3: If the buffer size satisfies

$$l_c \geq \frac{N-1}{2\epsilon}. \tag{32}$$

Then, we have the following bounds for utility and ingress queue backlog

$$\liminf_{t\to\infty} \sum_c g^{(c)}\left(\overline{r}_{n_c}^{(c)}(t)\right) \geq \sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - \frac{D_3}{V}, \tag{33}$$

$$\overline{\sum_c Q_{n_c}^{(c)}} \leq \frac{D_3 + VG_{\mathsf{max}}}{2\lambda_{\mathsf{max}} - \frac{(N-1)}{l_c}}, \tag{34}$$

where again $(r_{n_c}^{*(c)}(\epsilon))$ is an ϵ-optimal solution, which is an optimal solution of the optimization problem (8)-(10), V is predetermined number, $D_3 \triangleq D_1 + C(N + 2R_{\mathsf{max}} + 1)T$, and D_1 is given in Theorem 1.

Proof: The proof is given in Appendix C. ■

This theorem says that the same network utility lower-bound as that in Algorithm 1 can be achieved under delay IQI at the cost of a larger average backlog bound as $V \to \infty$. However, the backlog upper-bound in (34) is larger than that derived in Theorem 1 as D_3 depends on T.

It has been shown in [6] that a slightly larger buffer size is required when scheduling is performed based on queue length vector (for both ingress and internal buffers) with estimation errors. In this theorem, we show that when only IQI is delayed while internal queue information is available without delay, we do not need a larger buffer size if an appropriate flow controller is available. In addition, there are some other existing works along this line in the literature [19], [20] where it has been shown that infrequent or delayed channel estimation may reduce the maximum throughput region. However, delayed queue backlog information does not decrease utility.

Remark 3: Although we have assumed that the IQI delay value is the same for all nodes, the same utility performance guarantee can be achieved even if routing and scheduling is performed based on IQI with different delay values at different nodes. This is because ingress backlogs can only be offset by finite values in any finite time interval. Also, these offsets are negligible if ingress backlogs become sufficiently large. Therefore, the same utility performance can be achieved even under the heterogeneous delay scenario. In fact, we can modify the proof in Appendix C to account for these heterogeneous delays by changing constant factors in (92), (94) accordingly.

B. Extensions to Other Interference Models

In this section, we extend the obtained results to a general interference model. For brevity, we only consider the case of constantly backlogged sources. Let K be the maximum number of links activated in any feasible schedule under the underlying interference model. Then, given $\epsilon > 0$, the required buffer size to achieve the backlog bound in Theorem 1 is

$$l_c \geq \frac{K}{\epsilon}. \tag{35}$$

In addition, given this chosen buffer size, the total average ingress backlog can be bounded as

$$\overline{\sum_c Q_{n_c}^{(c)}} \leq \frac{D_1 + VG_{\mathsf{max}}}{2\lambda_{\mathsf{max}} - \frac{2K}{l_c}}, \tag{36}$$

where D_1 is defined in Theorem 1.

Proof: We can prove these results by using the Lyapunov drift technique with the same Lyapunov function used in the proof of Theorem 1. However, in deriving the Lyapunov drift, we have a slightly different bound for (46). Specifically, we have

$$\sum_c \frac{Q_{n_c}^{(c)}}{l_c} \sum_{n:n \neq n_c} E\left\{\left[-\sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\mathsf{in}}} \mu_l^{(c)}\right]^2\right\} \leq \sum_c \frac{2KQ_{n_c}^{(c)}}{l_c} \tag{37}$$

where the inequality in (37) holds because we have

$$\sum_{n:n\neq n_c} E\left\{\left[-\sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\mathsf{in}}} \mu_l^{(c)}\right]^2\right\} \leq 2K, \ \forall c. \tag{38}$$

This is because for any particular flow c each activated link l contributes at most 2 to LHS of (38) and we can activate at most K links in the network simultaneously. Taking the same proof procedure as in Appendix 1 with this new bound (37), we can obtain the desired results. ■

It can be observed that the required lower bound on internal buffer sizes under a general interference model is smaller than that under the one hop interference model if $K < (N-1)/2$ (i.e., the number of activated links in any feasible schedule is small). This would be the case for stringent interference models such as the k-hop interference model with large k, and the SINR interference model with large required SINR thresholds [34]. This result is illustrated in the following example.

Example: Consider a line network with N links and the k-hop interference model. The maximum number of links that can be activated by any feasible schedule in this setting is $K = \max\{1, \lfloor N/(k+1) \rfloor\}$. Therefore, the required lower bound for buffer size to achieve the ϵ-optimal utility is $K = \max\{1, \lfloor N/(k+1) \rfloor\}/\epsilon$. For large N, this required buffer size decreases by a factor of $(k+1)/2$ compared to that derived under the one hop interference model. It can be observed that this result for line networks is quite conservative. In general, for wireless networks with richer connectivity, K would decrease much faster with k, which, therefore, requires much smaller buffer size. This would also intuitively holds for the SINR interference model with large required SINR thresholds [34].

C. Wireless Networks with Time Varying Link Quality

We have assumed deterministic interference and channel models in the previous sections. We now extend the obtained results to wireless networks having links with time varying quality [21], [35]. Again, assume that allowable schedules satisfy node exclusive interference constraints and let Φ denote the set of allowable schedules. In addition, let $\vec{Z}(t)$ denote the channel state vector, which determines the probability of successful transmissions in time slot t. Given a particular link activation set $\vec{S}(t) \in \Phi$ and channel state vector $\vec{Z}(t)$, we can define the following probability of successful transmission for link $l \in \vec{S}(t)$ as follows [35]:

$$\Pr\left\{\text{link } l \text{ succeeds}|\vec{S}(t), \vec{Z}(t)\right\} = P_l\left(\vec{S}(t), \vec{Z}(t)\right). \tag{39}$$

Note that the maximum achievable throughput region under this channel model is smaller than that assuming reliable transmissions considered in the previous sections. Under this channel and interference model, the same flow controller as in Algorithms 1 and 2 can be employed along with the corresponding modified differential backlog metric. However, link scheduling must take into account the time varying link reliability. Specifically, given the channel state vector $\vec{Z}(t)$ the schedule $\vec{S}^*$ is chosen in each time slot t as follows:

$$\vec{S}^* = \underset{\vec{S}\in\Phi}{\operatorname{argmax}} \sum_{l=1}^{L} S_l P_l\left(\vec{S}(t), \vec{Z}(t)\right) W_l^*(t). \tag{40}$$

For each scheduled link, one packet of the flow that achieves the maximum modified differential backlog is transmitted if the corresponding buffer has at least one packet waiting for transmission. It can be shown that the same performance as presented in Theorems 1 and 2 can be achieved for this modified algorithm.

VI. Numerical Results

We study the performance of the control algorithms using computer simulations. We consider a simple network of six nodes as shown in Fig. 4. All links are assumed to be bi-directional and the maximum weight scheduling algorithm is invoked in each time slot assuming the node exclusive (i.e., one-hop) interference model. We simulate four traffic flows whose (source, destination) pairs are (1, 6), (3, 4), (6, 2), (5, 2). We assume packet arrivals to all input reservoirs are Bernoulli processes. The utility function is chosen to be $g(r) = \ln(1+\beta r)$ with $\beta = 500$, $R_n^{\mathsf{max}} = 2$, $\forall n$, $V = 1000$ (it is observed that increasing V further does not improve the utility performance of the investigated algorithms).

TABLE I
OVERLOADED TRAFFIC FOR $\vec{\lambda} = (0.9, 0.9, 0.9, 0.9)$

		$\frac{N-1}{2l_c}$	$\overline{r}_1^{(1)}$	$\overline{r}_3^{(2)}$	$\overline{r}_6^{(3)}$	$\overline{r}_5^{(4)}$	r_{tot}	Q_{ingress}	Q_{internal}	$\overline{q}_{\text{max}}$	$\hat{\sigma}_{\text{max}}$
CLC2b [2]	$l_c = \infty$	0	0.2503	0.3004	0.3003	0.3001	1.1511	42×10^3	8.02×10^3	992.83	28.16
Proposed	$l_c = 2$	1.25	0.2788	0.2788	0.2789	0.2787	1.1152	7.18×10^3	16.38	1.49	0.50
	$l_c = 10$	0.25	0.2501	0.3002	0.3003	0.3001	1.1507	7.00×10^3	53.20	6.45	0.55
	$l_c = 50$	0.05	0.2501	0.3002	0.3004	0.3001	1.1507	7.00×10^3	237.19	30.43	0.56
	$l_c = 100$	0.025	0.2502	0.3002	0.3004	0.3000	1.1507	7.00×10^3	466.38	60.45	0.60

TABLE II
UNDERLOADED TRAFFIC FOR $\vec{\lambda} = (0.24, 0.29, 0.29, 0.29)$

		$\frac{N-1}{2l_c}$	$\overline{r}_1^{(1)}$	$\overline{r}_3^{(2)}$	$\overline{r}_6^{(3)}$	$\overline{r}_5^{(4)}$	r_{tot}	Q_{ingress}	Q_{internal}	$\overline{q}_{\text{max}}$	$\hat{\sigma}_{\text{max}}$
CLC2b [2]	$l_c = \infty$	0	0.2400	0.2899	0.2901	0.2897	1.1097	25.28	43.21	4.17	1.92
Proposed	$l_c = 2$	1.25	0.2400	0.2894	0.2896	0.2894	1.1084	6.72×10^3	16.31	1.43	0.51
	$l_c = 10$	0.25	0.2399	0.2900	0.2900	0.2900	1.1099	26.64	68.38	6.95	1.05
	$l_c = 50$	0.05	0.2397	0.2899	0.2899	0.2900	1.1095	22.11	296.59	34.59	2.84
	$l_c = 100$	0.025	0.2401	0.2901	0.2902	0.2898	1.1102	21.65	559.71	69.59	4.41

TABLE III
OVERLOADED TRAFFIC WITH DELAYED IQI FOR $\vec{\lambda} = (0.9, 0.9, 0.9, 0.9)$, $T = 20$

		$\frac{N-1}{2l_c}$	$\overline{r}_1^{(1)}$	$\overline{r}_3^{(2)}$	$\overline{r}_6^{(3)}$	$\overline{r}_5^{(4)}$	r_{tot}	Q_{ingress}	Q_{internal}	$\overline{q}_{\text{max}}$	$\hat{\sigma}_{\text{max}}$
Proposed	$l_c = 2$	1.25	0.2736	0.2737	0.2738	0.2736	1.0947	7.31×10^3	16.02	1.40	0.50
	$l_c = 10$	0.25	0.2502	0.3002	0.3003	0.3000	1.1507	7.00×10^3	56.97	6.45	0.55
	$l_c = 50$	0.05	0.2501	0.3002	0.3003	0.3001	1.1507	7.00×10^3	240.24	30.44	0.56
	$l_c = 100$	0.025	0.2502	0.3002	0.3003	0.3000	1.1507	7.00×10^3	468.96	60.45	0.60

For each set of simulation parameters, we obtained mean queue length and admitted rates for different traffic flows by averaging them over 10^7 time slots. We then repeated each simulation test 20 times. Results presented in section correspond to sample means over 20 simulation runs. The absolute deviations from the presented sample means of average queue length and admitted rates for each of the 20 simulation runs are less than 1% of the sample means.

We compare the performance of Algorithm 2 with that of Algorithm CLC2b proposed in [2], which assumed infinite internal buffer size (i.e., $l_c = \infty$ for all flows). We show time average admitted rates for each of the flows, total admitted rate of all flows r_{tot}, and the total average ingress and internal queue length. Specifically, the total average ingress and internal queue length are calculated as $Q_{\text{ingress}} \triangleq \sum_c \overline{Q}_{n_c}^{(c)}$ and $Q_{\text{internal}} \triangleq \sum_c \sum_{n \neq n_c} \overline{Q}_n^{(c)}$ where $\overline{Q}_n^{(c)}$ is the average queue length for flow c at node n.

In addition, to illustrate the queue dynamics under our proposed algorithms and Algorithm CLC2b, we present the maximum average backlog $\overline{q}_{\text{max}}$ and the maximum sample standard deviation of queue backlogs $\hat{\sigma}_{\text{max}}$ over all internal queues where $\overline{q}_{\text{max}} \triangleq \max_{\{c, n \neq n_c\}} \overline{Q}_n^{(c)}$ and $\hat{\sigma}_{\text{max}} \triangleq \max_{\{c, n \neq n_c\}} \hat{\sigma}_n^{(c)}$ where $\hat{\sigma}_n^{(c)}$ is the sample standard deviation of queue backlog of flow c at node n. The minimum values of average backlogs and sample standard deviation of queue backlogs over internal buffers are very close to zero, which are not presented for brevity.

We present the performance of Algorithm 2 for the overloaded case in Table I for different values of internal buffer size. Recall that given the buffer size l_c, Theorem 2 implies that the lower bound of the network utility corresponds to a rate vector which lies inside the ϵ-stripped throughput region where ϵ is bounded above by $(N-1)/(2l_c)$ as $V \to \infty$. To illustrate this performance bound, we also show the corresponding values of $(N-1)/(2l_c)$ for different l_c values in Tables I, II, and III. Table I shows that this performance bound is quite conservative, and that the proposed algorithm achieves utilities that are much closer to optimal than this bound would indicate. In particular, while $l_c = 2$ results in a slightly smaller total admitted rate and deviation from optimal rates, $l_c = 10$ achieves time-average admitted rates that are very close to the optimal ones. For $l_c = 50$, the corresponding ϵ is $\epsilon = (N-1)/(2l_c) = 5/100 = 0.05$. However, the simulation results suggest that the actual loss in utility is negligible in this case.

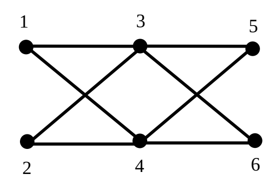

Fig. 4. Simulation network with primary interference model

Also, Table I shows that our proposed algorithm achieves significantly smaller average queue lengths in both ingress and internal buffers compared to those due to Algorithm CLC2b in [2]. For example, with $l_c = 10$, we can achieve admitted rates very close to the optimal values while the average ingress queue length is 6 times smaller, and the average total queue length in internal buffers is about 150 times smaller than those due to Algorithm CLC2b in [2]. It can also be observed that oversizing the internal buffers does not change the total average ingress queue length although it increases the total average internal queue length. Moreover, Algorithm 2 achieves much smaller $\overline{q}_{\mathsf{max}}$ and $\hat{\sigma}_{\mathsf{max}}$ compared to those due to Algorithm CLC2b. In particular, the values of $\hat{\sigma}_{\mathsf{max}}$ for different values of l_c are negligible, which suggests very stable queue backlogs achieved by Algorithm 2 in the highly loaded scenario.

We present numerical results for the underloaded traffic case in Table II. This table shows that Algorithm CLC2b [2] may achieve smaller average backlogs in ingress or internal buffers and our proposed algorithm has comparable performance with Algorithm CLC2b for $l_c = 10$. For $l_c = 2$ and $l_c = 10$, our proposed algorithm results in smaller $\hat{\sigma}_{\mathsf{max}}$ compared to Algorithm CLC2b. However, for $l_c = 50$ and $l_c = 100$, Algorithm CLC2b achieves smaller $\overline{q}_{\mathsf{max}}$ and $\hat{\sigma}_{\mathsf{max}}$ than those due to our proposed algorithm.

In addition, it is shown that under-sizing the internal buffers results in significant increase in average ingress queue length. Intuitively, this is because with finite buffers throughput is slightly reduced leading to greater backlogs in ingress buffers. Also, over-sizing internal buffers improves queue length of ingress buffers but increases total queue length in internal buffers. Finally, we show the performance of our algorithm with delayed IQI for the overloaded traffic case, and $T = 20$ in Table III. This table shows that delayed IQI does not impact the throughput performance while it only results in marginal increase in the total average queue length of internal buffers. It can also observed that $\overline{q}_{\mathsf{max}}$ and $\hat{\sigma}_{\mathsf{max}}$ presented in Table III are almost the same as those in Table I, which are achieved without IQI delay.

VII. Conclusion

In this paper, we proposed and analyzed the performance of control algorithms for wireless networks with finite buffers. Two different algorithms were investigated for both scenarios of heavy traffic and arbitrary traffic arrival rates. We also analyzed the performance of the control algorithms with delayed IQI. Numerical results confirm the superior performance of our proposed algorithms in terms of the backlog-utility tradeoff compared to existing algorithms in the literature. Our paper provides a unified framework for the problems of utility maximization, buffer sizing, and delay control in a very general setting. Developing decentralized control algorithms for wireless networks with finite buffers that achieve optimal buffer-utility tradeoff is an open problem which will be considered in our future work.

Appendix A
Proof of Theorem 1

Consider the following Lyapunov function

$$L(\vec{Q}) \triangleq \sum_c \left(Q_{n_c}^{(c)}(t)\right)^2 + \sum_c \sum_{n:n\neq n_c} \frac{1}{l_c}\left(Q_n^{(c)}(t)\right)^2 Q_{n_c}^{(c)}(t). \quad (41)$$

Note that this is the same Lyapunov function as that proposed in [6]. Now, consider the Lyapunov drift

$$\Delta(t) \triangleq E\left\{L(\vec{Q})(t+1) - L(\vec{Q})(t) \mid \vec{Q}(t)\right\}. \quad (42)$$

Recall that $\sum_{l\in\Omega_{n_c}^{\mathsf{in}}} \mu_l^{(c)}(t) = 0$, $\forall c, t$ because we do not allow traffic of any flow c to be routed back to the source node n_c. Hence, we have

$$Q_{n_c}^{(c)}(t+1) = Q_{n_c}^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{(c)}(t) + R_{n_c}^{(c)}(t). \quad (43)$$

Now, we find the drift for the first term in (41). Squaring both sides of (43) and performing some manipulations, we obtain

$$\begin{aligned} Q_{n_c}^{(c)}(t+1)^2 - Q_{n_c}^{(c)}(t)^2 &= \left[R_{n_c}^{(c)}(t) - \sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)}(t)\right]^2 \\ &\quad + 2Q_{n_c}^{(c)}(t)\left[R_n^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{(c)}(t)\right] \\ &\leq 2Q_{n_c}^{(c)}(t)\left[R_n^{(c)}(t) - \sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)}(t)\right] + R_{\max}^2 \quad (44) \end{aligned}$$

where (44) holds because we have

$$\left[R_{n_c}^{(c)}(t) - \sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)}(t)\right]^2 \leq R_{\max}^2.$$

Now, to find the drift for the second term in (41), we have

$$\begin{aligned} &\sum_c \frac{1}{l_c} Q_{n_c}^{(c)}(t+1) \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 \\ &= \sum_c \frac{1}{l_c}\left[Q_{n_c}^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{(c)}(t) + R_{n_c}^{(c)}(t)\right] \\ &\qquad \times \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 \\ &= \sum_c \frac{1}{l_c} Q_{n_c}^{(c)}(t) \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 \\ &\quad + \sum_c \frac{1}{l_c}\left[R_{n_c}^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{(c)}(t)\right] \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 \\ &= \sum_c \frac{1}{l_c} Q_{n_c}^{(c)}(t) \sum_{n:n\neq n_c} \left[Q_n^{(c)}(t) - \sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)}(t) \right. \\ &\qquad \left. + \sum_{l\in\Omega_l^{\mathsf{in}}} \mu_n^{(c)}(t)\right]^2 \\ &\quad + \sum_c \frac{1}{l_c}\left[R_{n_c}^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{(c)}(t)\right] \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 \\ &= \sum_c \frac{1}{l_c} Q_{n_c}^{(c)}(t) \sum_{n:n\neq n_c} Q_n^{(c)}(t)^2 \\ &\quad + 2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)} Q_{n_c}^{(c)}}{l_c}\left[-\sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\mathsf{in}}} \mu_l^{(c)}\right] \\ &\quad + \sum_c \sum_{n:n\neq n_c} \frac{Q_{n_c}^{(c)}}{l_c}\left[-\sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\mathsf{in}}} \mu_l^{(c)}\right]^2 \\ &\quad + \sum_c \frac{1}{l_c}\left[R_{n_c}^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{(c)}(t)\right] \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2. \quad (45) \end{aligned}$$

Note that we have

$$\sum_c \frac{Q_{n_c}^{(c)}}{l_c} \sum_{n:n\neq n_c} E\left\{\left[-\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)}\right]^2\right\} \leq \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c}, \quad (46)$$

$$\sum_c \frac{1}{l_c}\left[R_{n_c}^{(c)}(t) - \sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{(c)}(t)\right] \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 \leq C\frac{1}{l_c}R_{\max}(N-1)l_c^2 = CR_{\max}(N-1)l_c \quad (47)$$

where the inequality in (46) holds because we have

$$\left[-\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)}\right]^2 \leq 1, \ \forall n\neq n_c. \quad (48)$$

Using the results in (46) and (47) to (45), we have

$$\begin{aligned}&\sum_c \frac{1}{l_c}Q_{n_c}^{(c)}(t+1) \sum_{n:n\neq n_c} \left(Q_n^{(c)}(t+1)\right)^2 - \sum_c \frac{1}{l_c}Q_{n_c}^{(c)}(t) \sum_{n:n\neq n_c} Q_n^{(c)}(t)^2 \\ &\leq 2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)}Q_{n_c}^{(c)}}{l_c}\left[-\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)}\right] \\ &\quad + \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} + CR_{\max}(N-1)l_c. \quad (49)\end{aligned}$$

Using the drifts for the two terms of (41) derived in (44) and (49), the Lyapunov drift can be written as

$$\begin{aligned}\Delta(t) \leq& D_4 + \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} \\ &+2\sum_c Q_{n_c}^{(c)} E\left\{-\sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{(c)} + R_{n_c}^{(c)} \mid \vec{Q}(t)\right\} \\ &+2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)}Q_{n_c}^{(c)}}{l_c} E\left\{-\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)} \mid \vec{Q}(t)\right\}\end{aligned}$$

where $D_4 = CR_{\max}^2 + CR_{\max}(N-1)l_c$ is a finite number. Subtracting $V\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}\right) \mid \vec{Q}(t)\right\}$ from both sides, we have

$$\begin{aligned}&\Delta(t) - V\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}\right) \mid \vec{Q}(t)\right\} \leq D_4 \\ &+\sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} - 2\sum_c Q_{n_c}^{(c)} E\left\{\sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{(c)} \mid \vec{Q}(t)\right\} \\ &-2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)}Q_{n_c}^{(c)}}{l_c} E\left\{\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} - \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)} \mid \vec{Q}(t)\right\} \\ &-\sum_c E\left\{Vg^{(c)}\left(R_{n_c}^{(c)}\right) - 2Q_{n_c}^{(c)}R_{n_c}^{(c)} \mid \vec{Q}(t)\right\}. \quad (50)\end{aligned}$$

Now, we define the following quantities

$$\begin{aligned}\Psi_1(\vec{Q}) \triangleq& 2\sum_c Q_{n_c}^{(c)} E\left\{\sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{(c)} \mid \vec{Q}(t)\right\} \\ &+2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)}Q_{n_c}^{(c)}}{l_c} E\left\{\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} - \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)} \mid \vec{Q}(t)\right\}, \quad (51)\end{aligned}$$

$$\Psi_2(\vec{Q}) \triangleq \sum_c E\left\{Vg^{(c)}\left(R_{n_c}^{(c)}\right) - 2Q_{n_c}^{(c)}R_{n_c}^{(c)} \mid \vec{Q}(t)\right\}. \quad (52)$$

Then, the inequality (50) can be rewritten as

$$\begin{aligned}\Delta(t) - V\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}\right) \mid \vec{Q}(t)\right\} \leq D_4 \\ +\sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} - \Psi_1(\vec{Q}) - \Psi_2(\vec{Q}). \quad (53)\end{aligned}$$

It can be verified that

$$\begin{aligned}\Psi_1(\vec{Q}) =& 2\sum_c \sum_{(n_c,m)\in\Omega_{n_c}^{\text{out}}} \frac{Q_{n_c}^{(c)}}{l_c} E\left\{\mu_{n_c m}^{(c)} \mid \vec{Q}(t)\right\}\left[l_c - Q_m^{(c)}\right] \\ &+2\sum_c \sum_{(n,m)\in E:n\neq n_c} \frac{Q_{n_c}^{(c)}}{l_c} E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t)\right\}\left[Q_n^{(c)} - Q_m^{(c)}\right]. \quad (54)\end{aligned}$$

Hence, the routing/scheduling and flow control components of Algorithm 1 maximize $\Psi_1(\vec{Q})$ and $\Psi_2(\vec{Q})$, respectively. Strictly speaking, the routing/scheduling component of Algorithm 1 maximizes $\Psi_1(\vec{Q})$ only for $Q_{n_c}^{(c)}(t) \geq 1$ because we do not transmit any traffic of flow c over any link (n_c, m) if $Q_{n_c}^{(c)}(t) < 1$. Given $r_{n_c}^{*(c)}(\epsilon) \in \Lambda_\epsilon$ defined before, we have $\left(r_{n_c}^{*(c)}(\epsilon) + \epsilon\right) \in \Lambda$. Hence, there exists a stationary and randomized policy that chooses a link rate vector $\left(\mu_l^{*(c)}(t)\right)$ that satisfies for all time slot t [4]:

$$r_{n_c}^{*(c)}(\epsilon) + \epsilon = E\left\{\sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{*(c)}(t)\right\}, \ \forall c, \quad (55)$$

$$E\left\{\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{*(c)}(t) - \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{*(c)}(t)\right\} = 0, \ \forall n\neq n_c. \quad (56)$$

Recall that the routing/scheduling and flow control components of Algorithm 1 maximize $\Psi_1(\vec{Q})$ and $\Psi_2(\vec{Q})$, respectively. Hence, we have

$$\Psi_1(\vec{Q}) \geq 2\sum_c Q_{n_c}^{(c)}\left[r_{n_c}^{*(c)}(\epsilon) + \epsilon\right] - 2C, \quad (57)$$

$$\Psi_2(\vec{Q}) \geq \sum_c \left\{Vg\left(r_{n_c}^{*(c)}(\epsilon)\right) - 2Q_{n_c}^{(c)}r_{n_c}^{*(c)}(\epsilon)\right\} \quad (58)$$

where the constant $2C$ in (57) accounts for the fact that we do not transmit data of flow c on any link (n_c, m) if $Q_{n_c}^{(c)}(t) < 1$. Specifically, when $Q_{n_c}^{(c)}(t) < 1$, we have

$$\sum_{(n_c,m)\in\Omega_{n_c}^{\text{out}}} \frac{Q_{n_c}^{(c)}}{l_c} E\left\{\mu_{n_c m}^{(c)} \mid \vec{Q}(t)\right\}\left[l_c - Q_m^{(c)}\right] < 1. \quad (59)$$

Using the results of (57) and (58) in (53), we have

$$\Delta(t) - V\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}\right) \mid \vec{Q}(t)\right\} \leq D_1$$

$$+\sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} - V\sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - 2\epsilon\sum_c Q_{n_c}^{(c)}$$

where D_1 is the constant given in Theorem 1. Taking the expectations over the distribution of $\vec{Q}$ and summing over $t \in \{1, 2, \cdots, M\}$, we have

$$\begin{aligned}
&E\left\{L(\vec{Q}(M))\right\} - E\left\{L(\vec{Q}(0))\right\} \\
&\qquad -V\sum_{\tau=0}^{M-1}\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(\tau)\right)\right\} \\
&\leq MD_1 + \sum_{\tau=0}^{M-1}\sum_c \frac{(N-1)Q_{n_c}^{(c)}(\tau)}{l_c} \\
&-VM\sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - 2\epsilon\sum_{\tau=0}^{M-1}\sum_c E\left\{Q_{n_c}^{(c)}(\tau)\right\}. \quad (60)
\end{aligned}$$

By arranging the terms of (60) appropriately and dividing both sides by VM, we have

$$\begin{aligned}
&\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(\tau)\right)\right\} \\
&\geq \sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - \frac{D_1 + E\left\{L(\vec{Q}(0))\right\}/M}{V} \\
&+\frac{1}{VM}\sum_{\tau=0}^{M-1}\sum_c\left(2\epsilon - \frac{(N-1)}{l_c}\right)E\left\{Q_{n_c}^{(c)}(\tau)\right\} \quad (61)
\end{aligned}$$

where we have used the fact that $L(\vec{Q}(M)) \geq 0$ to obtain (61). Given the value of ϵ that gives the desired utility reduction, and recall we choose the buffer size which satisfies

$$l_c \geq \frac{N-1}{2\epsilon}. \quad (62)$$

Hence, we have

$$\frac{1}{VM}\sum_{\tau=0}^{M-1}\sum_c\left(2\epsilon - \frac{(N-1)}{l_c}\right)E\left\{Q_{n_c}^{(c)}(\tau)\right\} \geq 0. \quad (63)$$

Now, due to Jensen's inequality, we have

$$\begin{aligned}
&\sum_c g^{(c)}\left(\frac{1}{M}\sum_{\tau=0}^{M-1}E\left\{R_{n_c}^{(c)}(\tau)\right\}\right) \\
&\qquad \geq \frac{1}{M}\sum_{\tau=0}^{M-1}\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(\tau)\right)\right\}. \quad (64)
\end{aligned}$$

Using the results of (63) and (64) to (61), we have

$$\begin{aligned}
&\sum_c g^{(c)}\left(\frac{1}{M}\sum_{\tau=0}^{M-1}E\left\{R_{n_c}^{(c)}(\tau)\right\}\right) \\
&\geq \sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - \frac{D_1 + E\left\{L(\vec{Q}(0))\right\}/M}{V}. \quad (65)
\end{aligned}$$

Taking the limit $M \to \infty$ in (65), we have

$$\liminf_{M\to\infty}\sum_c g^{(c)}\left(\overline{r}_{n_c}^{(c)}(M)\right) \geq \sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - \frac{D_1}{V}. \quad (66)$$

Therefore, we have proved the utility bound. To prove the backlog bound, we arrange the inequality (60) appropriately and divide both sides by M, we have

$$\begin{aligned}
&\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_c\left(2\epsilon - \frac{(N-1)}{l_c}\right)E\left\{Q_{n_c}^{(c)}(\tau)\right\} \\
&-\frac{E\left\{L(\vec{Q}(0))\right\}}{M} \leq D_1 + \frac{V}{M}\sum_{\tau=0}^{M-1}\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(\tau)\right)\right\} \quad (67)
\end{aligned}$$

where we have used the fact that $E\left\{L(\vec{Q}(M))\right\} \geq 0$ and $\sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) \geq 0$ to obtain (67). From the definition of G_{max}, we have

$$\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(\tau)\right)\right\} \leq G_{\mathsf{max}}. \quad (68)$$

Using this result to (67), we have

$$\begin{aligned}
&\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_c\left(2\epsilon - \frac{(N-1)}{l_c}\right)E\left\{Q_{n_c}^{(c)}(\tau)\right\} \\
&\qquad -\frac{E\left\{L(\vec{Q}(0))\right\}}{M} \leq D_1 + VG_{\mathsf{max}}. \quad (69)
\end{aligned}$$

Note that the above inequalities hold for any $0 < \epsilon \leq \lambda_{\mathsf{max}}$. Also, suppose we choose the same buffer sizes for different flows which satisfy (62). By choosing $\epsilon = \lambda_{\mathsf{max}}$ and taking the limit for $M \to \infty$ in (69), we have

$$\limsup_{M\to\infty}\frac{1}{M}\sum_{\tau=0}^{M-1}\sum_c E\left\{Q_{n_c}^{(c)}(\tau)\right\} \leq \frac{D_1 + VG_{\mathsf{max}}}{2\lambda_{\mathsf{max}} - \frac{(N-1)}{l_c}}. \quad (70)$$

Therefore, we have proved the backlog bound.

Appendix B
Proof of Theorem 2

Define $\vec{\Theta}(t) \triangleq \left(\vec{Q}(t), \vec{Y}(t)\right)$ and consider the following Lyapunov function

$$\begin{aligned}
L(\vec{\Theta}) \triangleq &\sum_c\left(Q_{n_c}^{(c)}(t)\right)^2 + \sum_c\sum_{n:n\neq n_c}\frac{1}{l_c}\left(Q_n^{(c)}(t)\right)^2 Q_{n_c}^{(c)}(t) \\
&+\sum_c\left(Y_{n_c}^{(c)}(t)\right)^2. \quad (71)
\end{aligned}$$

Consider the following one-step Lyapunov drift

$$\Delta(t) \triangleq E\left\{L(\vec{\Theta})(t+1) - L(\vec{\Theta})(t) \mid \vec{\Theta}(t)\right\}. \quad (72)$$

Note that the first two terms in the Lyapunov function (71) are exactly the two terms in the Lyapunov function (41). Hence, the drifts for these two terms are available in (44) and (49). Now, we find the drift for the third term in (71). From (27), we have

$$\begin{aligned}
Y_{n_c}^{(c)}(t+1)^2 \leq &\left[Y_{n_c}^{(c)}(t) - R_{n_c}^{(c)}(t)\right]^2 + z_{n_c}^{(c)}(t)^2 \\
&+2z_{n_c}^{(c)}(t)\max\left\{Y_{n_c}^{(c)}(t) - R_{n_c}^{(c)}(t), 0\right\} \\
\leq Y_{n_c}^{(c)}(t)^2 &- 2Y_{n_c}^{(c)}(t)R_{n_c}^{(c)}(t) + 2z_{n_c}^{(c)}(t)Y_{n_c}^{(c)}(t) \\
&+z_{n_c}^{(c)}(t)^2 + R_{n_c}^{(c)}(t)^2. \quad (73)
\end{aligned}$$

Hence, we have

$$\begin{aligned}
&Y_{n_c}^{(c)}(t+1)^2 - Y_{n_c}^{(c)}(t)^2 \\
&\qquad \leq 2\left[-R_{n_c}^{(c)}(t) + z_{n_c}^{(c)}(t)\right]Y_{n_c}^{(c)}(t) + 2R_{\mathsf{max}}^2. \quad (74)
\end{aligned}$$

Using the results of (44) and (49), (74) in the Lyapunov drift (72), we have

$$\Delta(t) \leq D_2 + \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} + 2\sum_c Q_{n_c}^{(c)} E\left\{-\sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{(c)} + R_{n_c}^{(c)} \mid \vec{\Theta}(t)\right\} + 2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)}Q_{n_c}^{(c)}}{l_c} E\left\{-\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} + \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)} \mid \vec{\Theta}(t)\right\} + 2\sum_c Y_{n_c}^{(c)}(t) E\left\{z_{n_c}^{(c)}(t) - R_{n_c}^{(c)}(t) \mid \vec{\Theta}(t)\right\} \quad (75)$$

where D_2 is the constant given in Theorem 2. Subtracting $V\sum_c E\left\{g^{(c)}\left(z_{n_c}^{(c)}\right) \mid \vec{\Theta}(t)\right\}$ from both sides of this inequality, we have

$$\Delta(t) - V\sum_c E\left\{g^{(c)}\left(z_{n_c}^{(c)}\right) \mid \vec{\Theta}(t)\right\} \leq D_2 + \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} - 2\sum_c Q_{n_c}^{(c)} E\left\{\sum_{l\in\Omega_{n_c}^{\text{out}}} \mu_l^{(c)} \mid \vec{\Theta}(t)\right\} - 2\sum_c \sum_{n:n\neq n_c} \frac{Q_n^{(c)}Q_{n_c}^{(c)}}{l_c} E\left\{\sum_{l\in\Omega_n^{\text{out}}} \mu_l^{(c)} - \sum_{l\in\Omega_n^{\text{in}}} \mu_l^{(c)} \mid \vec{\Theta}(t)\right\} - \sum_c E\left\{Vg^{(c)}\left(z_{n_c}^{(c)}\right) - 2Y_{n_c}^{(c)} z_{n_c}^{(c)} \mid \vec{\Theta}(t)\right\} - 2\sum_c E\left\{Y_{n_c}^{(c)} R_{n_c}^{(c)} - Q_{n_c}^{(c)} R_{n_c}^{(c)} \mid \vec{\Theta}(t)\right\}. \quad (76)$$

Now, define the following quantities

$$\Psi_3(\vec{\Theta}) \triangleq \sum_c E\left\{Vg^{(c)}\left(z_{n_c}^{(c)}\right) - 2Y_{n_c}^{(c)} z_{n_c}^{(c)} \mid \vec{\Theta}(t)\right\} \quad (77)$$

$$\Psi_4(\vec{\Theta}) \triangleq 2\sum_c E\left\{Y_{n_c}^{(c)} R_{n_c}^{(c)} - Q_{n_c}^{(c)} R_{n_c}^{(c)} \mid \vec{\Theta}(t)\right\}. \quad (78)$$

Then, the inequality (76) can be rewritten as follows:

$$\Delta(t) - V\sum_c E\left\{g^{(c)}\left(z_{n_c}^{(c)}\right) \mid \vec{\Theta}(t)\right\} \leq D_2 + \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} - \Psi_1(\vec{\Theta}) - \Psi_3(\vec{\Theta}) - \Psi_4(\vec{\Theta}) \quad (79)$$

where $\Psi_1(\vec{\Theta})$ is defined in (51). It can be observed that the flow control component of Algorithm 2 maximizes $\Psi_3(\vec{\Theta})$ and $\Psi_4(\vec{\Theta})$ defined in (77) and (78), and the routing/scheduling component maximizes $\Psi_1(\vec{\Theta})$. Now, recall the definition of $(r_{n_c}^{*(c)}(\epsilon))$ in (8)-(10). The rates $r_{n_c}^{*(c)}(\epsilon)$ can be achieved by the following simple admission control rule for traffic arrival rates inside or outside the throughput region. In each time slot t, admit all arriving traffic $A_{n_c}^{(c)}(t)$ with probability $r_{n_c}^{*(c)}(\epsilon)/\lambda_{n_c}^{(c)}$. Under this admission control, we have $E\left\{R_{n_c}^{(c)} \mid \vec{\Theta}\right\} = r_{n_c}^{*(c)}(\epsilon)/\lambda_{n_c}^{(c)} E\left\{A_{n_c}^{(c)}(t)\right\} = r_{n_c}^{*(c)}(\epsilon)$ because arrivals are assumed to be i.i.d. over time slots. Let $(z_{n_c}^{*(c)}(\epsilon)) = (r_{n_c}^{*(c)}(\epsilon))$, $\forall c$. Then, from Algorithm 2, we have

$$\Psi_1(\vec{Q}) \geq 2\sum_c Q_{n_c}^{(c)}\left[r_{n_c}^{*(c)}(\epsilon) + \epsilon\right], \quad (80)$$

$$\Psi_3(\vec{\Theta}) \geq \sum_c \left\{Vg\left(r_{n_c}^{*(c)}(\epsilon)\right) - 2Y_{n_c}^{(c)} r_{n_c}^{*(c)}(\epsilon)\right\}, \quad (81)$$

$$\Psi_4(\vec{\Theta}) \geq 2\sum_c \left\{Y_{n_c}^{(c)} r_{n_c}^{*(c)}(\epsilon) - Q_{n_c}^{(c)} r_{n_c}^{*(c)}(\epsilon)\right\}. \quad (82)$$

Note that there is no constant $2C$ in (80) because $Q_{n_c}^{(c)}(t)$ only take integer numbers in this case (i.e., they are either zero or at least one). This is because the flow controller in Algorithm 2 only injects an integer number of packets into the network at all times. Using these results in (79), we have

$$\Delta(t) - V\sum_c E\left\{g^{(c)}\left(z_{n_c}^{(c)}\right) \mid \vec{\Theta}(t)\right\} \leq D_2 + \sum_c \frac{(N-1)Q_{n_c}^{(c)}}{l_c} - 2\epsilon\sum_c Q_{n_c}^{(c)} - \sum_c Vg^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right). \quad (83)$$

Using the same technique in the proof of Theorem 1, we can obtain the desired results.

Appendix C
Proof of Theorem 3

Consider the same Lyapunov function as in the proof of Theorem 1. However, we employ the conditional Lyapunov drift $\Delta(t)$ over $\vec{Q}(t,T) \triangleq (\vec{Q}(t-T), \vec{Q}(t-T+1), \ldots, \vec{Q}(t))$ in this proof. Then, we have

$$\Delta(t) - V\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(t)\right) \mid \vec{Q}(t,T)\right\} \leq D_4 + \sum_c \frac{(N-1)Q_{n_c}^{(c)}(t)}{l_c} - \Psi_1(\vec{Q}(t)) - \Psi_2(\vec{Q}(t)) \quad (84)$$

where $D_4 = CR_{\mathsf{max}}^2 + CR_{\mathsf{max}}(N-1)l_c$. Recall that we have defined

$$\Psi_1(\vec{Q}(t)) = 2\sum_c \sum_{(n_c,m)\in\Omega_{n_c}^{\text{out}}} Q_{n_c}^{(c)}(t) \times \frac{E\left\{\mu_{(n_c,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[l_c - Q_m^{(c)}(t)\right]}{l_c} + 2\sum_c \sum_{(n,m)\in E:n\neq n_c} \frac{Q_{n_c}^{(c)}(t)}{l_c} E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t) - Q_m^{(c)}(t)\right].$$

Now, let us define $\Psi_{1,d}(\vec{Q}(t))$ corresponding to $\Psi_1(\vec{Q}(t))$ with delayed IQI as follows:

$$\Psi_{1,d}(\vec{Q}(t)) = 2\sum_c \sum_{(n_c,m)\in\Omega_{n_c}^{\text{out}}} Q_{n_c}^{(c)}(t) \times \frac{E\left\{\mu_{(n_c,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[l_c - Q_m^{(c)}(t)\right]}{l_c} + 2\sum_c \sum_{(n,m)\in E:n\neq n_c} Q_{n_c}^{(c)}(t-T) \times \frac{E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t) - Q_m^{(c)}(t)\right]}{l_c}. \quad (85)$$

Now, we compare the second term of $\Psi_1(\vec{Q}(t))$ and $\Psi_{1,d}(\vec{Q}(t))$. First, note that

$$Q_{n_c}^{(c)}(t) \geq Q_{n_c}^{(c)}(t-T) - T \quad (86)$$

because at most one packet can be transmitted from a queue in one time slot. Hence,

$$\sum_c \sum_{(n,m)\in E: n\neq n_c} Q_{n_c}^{(c)}(t) \times \frac{E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c}$$
$$\geq \sum_c \sum_{(n,m)\in E: n\neq n_c} \left\{Q_{n_c}^{(c)}(t-T)-T\right\} \times \frac{E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c}$$
$$\geq \sum_c \sum_{(n,m)\in E: n\neq n_c} Q_{n_c}^{(c)}(t-T) \times \frac{E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c} - \frac{C(N-1)T}{2} \quad (87)$$

where (87) holds because

$$0 \leq \frac{E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c} \leq 1.$$

Hence

$$\sum_{(n,m)\in E: n\neq n_c} \frac{E\left\{\mu_{(n,m)}^{(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c} \leq \frac{(N-1)}{2}$$

where this inequality holds because we can activate at most $(N-1)/2$ links whose ends are not the source node n_c due to the node exclusive assumption. Using (87) in (85), we have

$$\Psi_1(\vec{Q}) \geq \Psi_{1,d}(\vec{Q}) - C(N-1)T \quad (88)$$

where this inequality holds because the first terms of $\Psi_1(\vec{Q}(t))$ and $\Psi_{1,d}(\vec{Q}(t))$ are the same. Given $r_{n_c}^{*(c)}(\epsilon)$, which is the optimal traffic admitted rate lying inside the ϵ-stripped throughput region, there exists a stationary and randomized policy that chooses a link rate vector $\left(\mu_l^{*(c)}(t)\right)$ that satisfies for all time slot t [4]:

$$r_{n_c}^{*(c)}(\epsilon)+\epsilon = E\left\{\sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{*(c)}(t)\right\}, \ \forall\, c, \quad (89)$$

$$E\left\{\sum_{l\in\Omega_n^{\mathsf{out}}} \mu_l^{*(c)}(t) - \sum_{l\in\Omega_n^{\mathsf{in}}} \mu_l^{*(c)}(t)\right\} = 0, \ \forall\, n\neq n_c. \quad (90)$$

Then, due to the operation of our routing/scheduling policy we have

$$\Psi_{1,d}(\vec{Q}(t)) \geq 2\sum_c \sum_{(n_c,m)\in\Omega_{n_c}^{\mathsf{out}}} Q_{n_c}^{(c)}(t) \times \frac{E\left\{\mu_{(n_c,m)}^{*(c)} \mid \vec{Q}(t,T)\right\}\left[l_c-Q_m^{(c)}(t)\right]}{l_c} - 2C$$
$$+2\sum_c \sum_{(n,m)\in E: n\neq n_c} Q_{n_c}^{(c)}(t-T) \times \frac{E\left\{\mu_{(n,m)}^{*(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c}$$
$$\geq 2\sum_c \sum_{(n_c,m)\in\Omega_{n_c}^{\mathsf{out}}} Q_{n_c}^{(c)}(t-T) \times \frac{E\left\{\mu_{(n_c,m)}^{*(c)} \mid \vec{Q}(t,T)\right\}\left[l_c-Q_m^{(c)}(t)\right]}{l_c} - 2C - 2CT$$
$$+2\sum_c \sum_{(n,m)\in E: n\neq n_c} Q_{n_c}^{(c)}(t-T) \times \frac{E\left\{\mu_{(n,m)}^{*(c)} \mid \vec{Q}(t,T)\right\}\left[Q_n^{(c)}(t)-Q_m^{(c)}(t)\right]}{l_c}$$
$$= 2\sum_c Q_{n_c}^{(c)}(t-T)\left[r_{n_c}^{*(c)}(\epsilon)+\epsilon\right] - 2C - 2CT. \quad (91)$$

Using this result in (88), we have

$$\Psi_1(\vec{Q}) \geq 2\sum_c Q_{n_c}^{(c)}(t-T)\left[r_{n_c}^{*(c)}(\epsilon)+\epsilon\right] - 2C - C(N+1)T. \quad (92)$$

Note that we have

$$Q_{n_c}^{(c)}(t-T) \geq Q_{n_c}^{(c)}(t) - TR_{\mathsf{max}} \quad (93)$$

because the maximum amount of traffic injected into the network from any node is R_{max}. Using this result in (92) and using the fact that $r_{n_c}^{*(c)}(\epsilon)+\epsilon = E\left\{\sum_{l\in\Omega_{n_c}^{\mathsf{out}}} \mu_l^{*(c)}(t)\right\} \leq 1$, we have

$$\Psi_1(\vec{Q}) \geq 2\sum_c Q_{n_c}^{(c)}(t)\left[r_{n_c}^{*(c)}(\epsilon)+\epsilon\right] - 2C - C(N+2R_{\mathsf{max}}+1)T. \quad (94)$$

Also, due to the operation of our flow controller we have

$$\Psi_2(\vec{Q}) \geq \sum_c \left\{Vg\left(r_{n_c}^{*(c)}(\epsilon)\right) - 2Q_{n_c}^{(c)}(t) r_{n_c}^{*(c)}(\epsilon)\right\}. \quad (95)$$

Using the results in (95) and (94) to (84), we have

$$\Delta(t) - V\sum_c E\left\{g^{(c)}\left(R_{n_c}^{(c)}(t)\right) \mid \vec{Q}(t,T)\right\} = D_3$$
$$+\sum_c \frac{(N-1)Q_{n_c}^{(c)}(t)}{l_c} - V\sum_c g^{(c)}\left(r_{n_c}^{*(c)}(\epsilon)\right) - 2\epsilon\sum_c Q_{n_c}^{(c)}(t)$$

where $D_3 = D_1 + C(N+2R_{\mathsf{max}}+1)T$. Using the same technique as in the proof of Theorem 1, we can get the desired results.

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. Automatic Control*, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.

[2] L. Georgiadis, M. J. Neely, L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking,* vol. 1, no. 1, pp. 1-144, 2006.

[3] M. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Networking*, vol. 16, no. 2, April 2008.

[4] M. J. 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, Jan. 2005.

[5] H.-W. Lee, E. Modiano, and L. B. Le, "Distributed throughput maximization in wireless networks via random power allocation," *WiOpt 2009*.

[6] P. Giaccone, E. Leonardi and D. Shah, "Throughput region of finite-buffered networks," *IEEE Trans. Parallel Distributed Systems*, vol. 18, no. 2, pp. 251-262, Feb. 2007.

[7] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," *ACM SIGMETRICS 2006*.

[8] S. Sanghavi, L. Bui, and R. Srikant, “Distributed link scheduling with constant overhead,” *ACM SIGMETRICS 2007*, June 2007.
[9] X. Lin and N. B. Shroff, “The impact of imperfect scheduling on cross-layer congestion control in wireless networks,” *IEEE/ACM Trans. Networking,* vol. 14, no. 2, pp. 302–315, April 2006.
[10] P. Charporkar, K. Kar, and S. Sarkar, “Throughput guarantees through maximal scheduling in wireless networks,” *IEEE Trans. Inf. Theory*, vol. 54, no. 2, pp. 572-594, Feb. 2008.
[11] A. Eryilmaz, A. Ozdaglar, and E. Modiano, “Polynomial complexity algorithms for full utilization of multi-hop wireless networks,” *IEEE INFOCOM 2007*.
[12] C. Joo, X. Lin, and N. B. Shroff, “Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks,” *IEEE INFOCOM*, April 2008.
[13] G. Zussman, A. Brzezinski, and E. Modiano, “Multihop local pooling for distributed throughput maximization in wireless networks,” *IEEE INFOCOM*, April 2008.
[14] C. Joo and N. B. Shroff, “Performance of random access scheduling schemes in multi-hop wireless networks,” *IEEE INFOCOM 2007*.
[15] R. Li, L. Ying, A. Eryilmaz, and N. B. Shroff, “A unified approach to optimizing performance in networks serving heterogeneous flows,” *IEEE INFOCOM 2009*.
[16] A. Stolyar, “Large deviations of queues under QoS scheduling algorithms,” *Allerton 2006*, Sept. 2006.
[17] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, “Asynchronous congestion control in multi-hop wireless networks with maximal matching-based scheduling,” *IEEE/ACM Trans. Networking,* vol. 16, no. 4, pp. 826-839, Aug. 2008.
[18] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control,” *IEEE/ACM Trans. Networking,* vol. 15, no. 6, pp. 1333-1344, Dec. 2007.
[19] L. Ying and S. Shakkottai, “Scheduling in mobile wireless networks with topology and channel-state uncertainty,” *IEEE INFOCOM 2009*.
[20] K. Kar, X. Luo, S. Sarkar, “Throughput-optimal scheduling in multi-channel access point networks under infrequent channel measurements,” *IEEE Trans. Wireless Commun.,* vol. 7, no. 7, pp. 2619–2629, July 2008.
[21] L. Tassiulas and A. Ephremides, ”Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Trans. Inf. Theory,* vol. 39, no. 2, pp. 466–478, Mar. 1993.
[22] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Systems,* vol. 50, no. 4, pp. 401–457, Aug. 2005.
[23] A. Baron, R. Ginosar, and I. Keslassy, “The capacity allocation paradox,” *IEEE INFOCOM 2009*.
[24] M. Neely, “Order optimal delay for opportunistic scheduling in multi-user wireless uplinks and downlinks,” *IEEE/ACM Trans. Networking,* vol. 16, no. 5, pp. 1188–1199, Oct. 2008.
[25] L. B. Le, K. Jagannathan, and E. Modiano, “Delay analysis of maximum weight scheduling in wireless ad hoc networks,” *CISS'2009*, Mar. 2009.
[26] W. Khan, L. B. Le, and E. Modiano, “Autonomous routing algorithms for networks with wide-spread failures,” *IEEE MILCOM 2009,* Oct. 2009.
[27] M. Neely, “Delay analysis for maximal scheduling in wireless networks with bursty traffic,” *IEEE INFOCOM 2008*.
[28] G. R. Gupta and N. B. Shroff, “Delay analysis for multi-hop wireless networks,” *IEEE INFOCOM 2009*.
[29] L. Ying, S. Shakkottai, and A. Reddy, “On combining shortest-path and back-pressure routing over multihop wireless networks,” *IEEE INFOCOM 2009*.
[30] M. J. Neely, “Dynamic power allocation and routing for satellite and wireless networks with time varying channels.” PhD thesis, Massachusetts Institute of Technology, 2003.
[31] M. J. Neely, “Energy optimal control for time varying wireless networks,” *IEEE Trans. Inf. Theory,* vol. 52, no. 7, pp. 2915-2934, July 2006.
[32] L. Bui, R. Srikant, and A. Stolyar, “Novel architecture and algorithms for delay reduction in back-pressure scheduling and routing,” *IEEE INFOCOM 2009*.
[33] T. Lan, X. Lin, M. Chiang, and R. B. Lee, “Suboptimal resource allocation: Stability and benefits,” *IEEE/ACM Trans. Networking,* to appear.
[34] L. B. Le, E. Modiano, C. Joo, and N. B. Shroff, “Longest-queue-first scheduling under SINR interference model,” *ACM MobiHoc'10*, Sept. 2010.
[35] M. J. Neely, “Delay-based network utility maximization,” *IEEE INFOCOM 2010*.

PLACE PHOTO HERE

Long Bao Le (S'04-M'07) received the B.Eng. degree with highest distinction from Ho Chi Minh City University of Technology in 1999, the M.Eng. degree from Asian Institute of Technology (AIT) in 2002 and the PhD degree from University of Manitoba in 2007. He is currently an assistant professor at INRS-EMT, University of Quebec, Montreal, Quebec, Canada. Before that, he was a postdoctoral researcher first at University of Waterloo then at Massachusetts Institute of Technology. His current research interests include cognitive radio, wireless protocol engineering and resource allocation, stochastic control and cross-layer design for communication networks. He is a Member of the IEEE.

PLACE PHOTO HERE

Eytan Modiano (S'90-M'93-SM'00) received his B.S. degree in Electrical Engineering and Computer Science from the University of Connecticut at Storrs in 1986 and his M.S. and PhD degrees, both in Electrical Engineering, from the University of Maryland, College Park, MD, in 1989 and 1992 respectively. He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Post Doctoral Fellow during 1992-1993. Between 1993 and 1999 he was with MIT Lincoln Laboratory where he was a project leader for MIT Lincoln Laboratory's Next Generation Internet (NGI) project. Since 1999 he has been on the faculty at MIT; where he is a Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks. He is the co-recipient of the Sigmetrics 2006 Best paper award for the paper “Maximizing Throughput in Wireless Networks via Gossiping,” and the Wiopt 2005 best student paper award for the paper “Minimum Energy Transmission Scheduling Subject to Deadline Constraints.”

He is currently an Associate Editor for IEEE/ACM Transactions on Networking. He had served as Associate Editor for IEEE Transactions on Information Theory, and as guest editor for IEEE JSAC special issue on WDM network architectures; the Computer Networks Journal special issue on Broadband Internet Access; the Journal of Communications and Networks special issue on Wireless Ad-Hoc Networks; and for IEEE Journal of Lightwave Technology special issue on Optical Networks. He was the Technical Program co-chair for IEEE Wiopt 2006, IEEE Infocom 2007, and ACM MobiHoc 2007.

PLACE PHOTO HERE

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

He currently holds the Ohio Eminent Scholar chaired professorship in Networking and Communications in the departments of ECE and CSE at The Ohio State University, in Columbus, OH. He also currently serves as a Guest Chaired Professor of wireless communications with the Department of Electronic Engineering, Tsighnua University, Beijing, China. Previously, he was a Professor of ECE with Purdue University, West Lafayette, IN, and the Director of the Center for Wireless Systems and Applications (CWSA), a university-wide center on wireless systems and applications. His research interests span the areas of wireless and wireline communication networks, where he investigates fundamental problems in the design, performance, pricing, and security of these networks.

Dr. Shroff has received numerous awards for his networking research, including the National Science Foundation CAREER award, the Best Paper awards for IEEE INFOCOM 2006 and 2008, the Best Paper Award for IEEE IWQoS 2006, the Best Paper of the Year Award for Computer Networks, and the Best Paper of the Year Award for the Journal of Communications and Networks (his IEEE INFOCOM 2005 paper was one of two runner-up papers).