

# Constant-Time Distributed Scheduling Policies for Ad Hoc Wireless Networks

Xiaojun Lin, *Member, IEEE*, and Shahzada B. Rasool, *Student Member, IEEE*

**Abstract**—We propose a new class of distributed scheduling policies for ad hoc wireless networks that can achieve provable capacity regions. Previously known scheduling policies that guarantee comparable capacity regions are either centralized or require computation time that increases with the size of the network. In contrast, the unique feature of the proposed distributed scheduling policies is that they are constant-time policies, i.e., given a fixed approximation ratio and a bounded maximum node-degree of the network, the time needed for computing a new schedule is independent of the network size. Hence, they can be easily deployed in large networks.

**Index Terms**—Constant-time scheduling algorithms, ad hoc wireless networks, distributed algorithms, efficiency ratio.

## I. INTRODUCTION

In this paper, we study the link scheduling problem in ad hoc wireless networks. In wireless networks, the radio transmissions at different links can interfere with each other. Hence, in order to achieve the optimal capacity, it is usually more efficient to only use a subset of the radio links at each time [2]. Determining which subset of links should be active at each time becomes the link scheduling problem, which is mainly at the MAC layer in the OSI reference model.

Good scheduling policies are those that can achieve large capacity regions and can be easily computed. Consider a wireless network with  $L$  links, and let  $\lambda_l$  be the data rate offered to link  $l$ . Let  $\vec{\lambda} = [\lambda_1, \dots, \lambda_L]$ . The *capacity region* under a particular scheduling policy is the set of data-rate vectors  $\vec{\lambda}$  that the scheduling policy can support while keeping the queues at all links finite. A scheduling policy is said to be *throughput-optimal* if it can achieve the largest possible capacity region. Known throughput-optimal policies require solving a global optimization problem at each time [3]–[6]. Such scheduling policies are inappropriate for ad hoc networks because the distributive nature of these networks requires simple and decentralized scheduling solutions. Recently, a number of distributed scheduling policies have been proposed in the literature [7]–[12]. Since the capacity region under a distributed policy is typically smaller than the optimal one achieved by the throughput-optimal policy, we define the *efficiency ratio* of such a sub-optimal scheduling policy as the largest number  $\gamma$  such that, given any network topology, for

any  $\vec{\lambda}$  that can be supported by the throughput-optimal policy, this policy can support  $\gamma\vec{\lambda}$ . In other words, the scheduling policy with efficiency ratio  $\gamma$  can achieve at least  $\gamma$  fraction of the optimal capacity region. Related works have studied a number of distributed scheduling policies that are shown to achieve provable efficiency ratios. For example, by extending the result from the switching literature [13]–[15], the authors of [8] have shown that the maximal matching policy can achieve an efficiency ratio of no less than  $1/2$  under the node-exclusive interference model. Similar maximal scheduling policies are also studied under the bidirectional-equal-power model and the two-hop interference model [9]–[12], where different bounds on the efficiency ratio are derived. Finally, there also exists a class of randomized “pick-and-compare” policies, which could be implemented in a distributed fashion and could achieve an efficiency ratio of 1 [16], [17].

The problem with these existing distributed scheduling policies, however, is that the time needed to compute a schedule still increases with the size of the network. For example, one of the best known distributed algorithms in graph theory can compute maximal matching on a graph in  $O(\log L)$  rounds, where  $L$  is the total number of links in the graph [18]. Hence, its computation time increases as the size of the network grows.

In this paper, we propose a new class of distributed scheduling policies. A unique feature of these new policies is that, under appropriate assumptions, the time needed to compute a schedule can be made independent of the size of the network. Hence, they are more scalable and easier to implement in large networks. We provide such distributed scheduling policies for two types of interference models, i.e., the node-exclusive interference model and the two-hop interference model. Our policies require each link to learn the queue lengths of its one-hop neighboring links (in the case of the node-exclusive interference model) or that of its two-hop neighboring links (in the case of the two-hop interference model). Once they learn the queue-length information from neighboring links, in both cases our proposed policies only require *one* round of the computation of average length  $M/2$  to compute a new schedule, where  $M$  is a parameter related to an *approximation ratio* that determines how close one wants to approach the maximum possible efficiency ratio of this class of algorithms. Further, assuming that transmitting one piece of queue-length information takes constant time, one can design distributed algorithms for information exchange such that the amount of time required to exchange the queue-length information can be bounded by a function of the maximum node-degree of the network (see detailed discussions in Section III.) Therefore,

This work has been partially supported by the NSF grant CNS-0626703, and a grant from the Purdue Research Foundation. An earlier version of this paper has appeared in the Proceedings of 45th IEEE Conference on Decision and Control, 2006 [1].

The authors are with Center for Wireless Systems and Applications (CWSA) and School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, U.S.A.

Corresponding author: Xiaojun Lin. Email: linx@ecn.purdue.edu.

given a fixed approximation ratio and a bounded maximum node-degree, the total time for our proposed policies to compute a new schedule can be made independent of the size of the network. For this reason, we refer to the policies proposed in the paper as *constant-time* scheduling policies. We will show that our proposed scheduling policies can achieve comparable efficiency ratios as some of the non-constant-time policies in the literature.

We believe that the results in the paper offer new insights for the design of simple and efficient scheduling policies for ad hoc networks. The rest of the paper is organized as follows. In Section II, we outline the network model and review related results. In Sections III-IV, we will propose the constant-time distributed scheduling policies for the node-exclusive interference model and the two-hop interference model, respectively, and derive their efficiency ratios. We discuss implementation issues in Section V, and present simulation results in Section VI. Then we conclude.

## II. THE SYSTEM MODEL

Consider a wireless network with  $N$  nodes and  $L$  directed links. Each link corresponds to a pair of transmitter node and receiver node. Let  $b(l)$  and  $e(l)$  denote the transmitter node and the receiver node, respectively, of link  $l$ . Two nodes are one-hop neighbors of each other if they are the end-points of a common link. For each node  $i$ , let  $E(i)$  denote the set of links that connect to the one-hop neighbors of node  $i$ , i.e.,  $E(i)$  is the set of links that node  $i$  either acts as a transmitter or as a receiver. Two links are *one-hop* neighbors of each other if they share a common node. Two links are *two-hop* neighbors of each other if they have a common one-hop neighboring link. For each link  $l$ , let  $N^1(l)$  denote the set of one-hop neighbors of link  $l$  (including link  $l$  itself), i.e.,  $N^1(l) = E(b(l)) \cup E(e(l))$ . Further, let  $N^2(l)$  denote the set of two-hop neighbors of link  $l$ , i.e.,  $N^2(l) = \bigcup_{k \in N^1(l)} N^1(k)$ .

We first assume a single-hop traffic model, i.e., each packet only needs to traverse one of the  $L$  links and then leave the system. (We will discuss the extension to the multi-hop case in Section V.) We assume that time is divided into slots of unit length. Let  $A_l(t)$  denote the number of packets that arrive at link  $l$  at time slot  $t$ . We assume that packets are of unit length. Throughout the paper, we assume that the packet arrival processes  $A_l(t)$ ,  $l = 1, 2, \dots, L$ , are independent across links and *i.i.d.* in time, although the results of the paper could also be extended to more general arrival processes [19], [20].

We will study two types of interference models that govern the radio transmission. In both models, we say that two links interfere with each other if they cannot transmit data together. Under the *node-exclusive interference model*, each link  $l$  interferes with all of its one-hop neighboring links. Under the *two-hop interference model*, each link  $l$  interferes with all of its two-hop neighboring links. In both models, if the above interference constraints are satisfied, an active link  $l$  can transfer  $c_l$  packets within the time slot. We further assume that the system has carrier-sensing capabilities. In particular, under the one-hop interference model, we assume that all the one-hop neighboring links of link  $l$  can sense the transmission

at link  $l$ . Under the two-hop interference model, we assume that all one-hop neighboring nodes of node  $i$  can sense the transmission from node  $i$ .

*Remark:* The node-exclusive interference model can be viewed as a generalization of the bipartite graph model for modeling high-speed packet switches [13]–[15]. It has been used in [8], [21]–[23] to model wireless networks. While this is a somewhat simplified model, the main results can often be readily generalized to other more complex interference models, e.g., the two-hop interference model. Note also that the latter model is very close to the interference model that IEEE 802.11 DCF (Distributed Coordination Function) deals with [9], [12].

At time slot  $t$ , let  $\mathcal{M}(t)$  denote the outcome of the scheduling policy, which is defined as the set of non-interfering links that are chosen to be active at time  $t$ . Let  $D_l(t)$  denote the number of packets that link  $l$  can serve at time slot  $t$ . Then  $D_l(t) = c_l$  if  $l \in \mathcal{M}(t)$ , and  $D_l(t) = 0$  otherwise. Let  $Q_l(t)$  denote the number of packets queued at link  $l$  at the beginning of time slot  $t$ , then the evolution of  $Q_l(t)$  is governed by

$$Q_l(t+1) = [Q_l(t) + A_l(t) - D_l(t)]^+, \quad (1)$$

where  $[\cdot]^+$  denote the projection to  $[0, +\infty)$ .

We say that the system is *stable* if the queue lengths at all links remain finite [4], i.e.

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{\{\sum_{l=1}^L Q_l(t) > \eta\}} \rightarrow 0, \text{ almost surely as } \eta \rightarrow \infty. \quad (2)$$

Let  $\lambda_l$  be the mean packet arrival rate at link  $l$ , i.e.,  $\lambda_l = \mathbf{E}[A_l(t)]$ . Let  $\vec{\lambda} = [\lambda_1, \dots, \lambda_L]$ . As we defined in the Introduction, the *capacity region* under a particular scheduling policy is the set of  $\vec{\lambda}$  such that the system remains stable. The optimal capacity region  $\Omega$  is the union of the capacity regions of all scheduling policies. A scheduling policy is *throughput-optimal* if it can achieve the optimal capacity region  $\Omega$ . The *efficiency ratio* of a (possibly sub-optimal) scheduling policy is the largest number  $\gamma$  such that the scheduling policy can stabilize the system under any load  $\vec{\lambda} \in \gamma\Omega$ . By definition, a throughput-optimal scheduling policy has an efficiency ratio of 1.

### A. Related Results

1) *Scheduling Policies for the Node-Exclusive Interference Model:* One of the known throughput-optimal scheduling policies under the node-exclusive interference model computes the set  $\mathcal{M}(t)$  of non-interfering links at time-slot  $t$  such that  $\mathcal{M}(t)$  maximizes the sum of the queue-weighted-rates

$$\sum_{l \in \mathcal{M}(t)} Q_l(t) c_l. \quad (3)$$

This scheduling policy is a direct application of the more general result in [3]–[6]. The resulted schedule corresponds a **Maximum-Weighted-Matching (MWM)** of the underlying graph, where the weight of each link is  $Q_l(t) c_l$ . (Note that a *matching* is a subset of the links such that no two links share the same node. The *weight* of a matching is the total

weight over all links belonging to the matching. A *maximum-weighted-matching (MWM)* is the matching with the maximum weight.) An  $O(N^3)$ -complexity centralized algorithm for MWM can be found in [24], where  $N$  is the number of nodes. On the other hand, the following much simpler algorithm can be used to compute a suboptimal schedule that corresponds to a **Greedy Maximal Matching (GMM)** [8], [15], [25], [26]: Start from an empty schedule; From all possible links, pick the link with the largest weight  $Q_l(t)c_l$ ; Add this link to the schedule; Remove all links that are incident with either the transmitter node or the receiver node of link  $l$ ; Pick the link with the largest weight  $Q_l(t)c_l$  from the *remaining* links, and add to the schedule; Continue until there are no links left. The above centralized GMM algorithm has only  $O(L \log L)$ -complexity (where  $L$  is the number of links), and is much easier to implement than MWM. Using the technique in Theorem 10 of [15], we can show that the GMM policy achieves an efficiency ratio no less than  $1/2$ . There also exist distributed algorithms that can compute the GMM schedule in  $O(L)$  rounds [27].

The optimal capacity region  $\Omega$  under the node-exclusive interference model is known to be bounded by [21]:

$$\frac{2}{3}\Psi_0 \subseteq \Omega \subseteq \Psi_0, \quad (4)$$

where

$$\Psi_0 = \left\{ \vec{\lambda} \left| \sum_{l \in E(i)} \frac{\lambda_l}{c_l} \leq 1, \text{ for all nodes } i \right. \right\}. \quad (5)$$

The following **Maximal-Matching (MM)** policy can be shown to achieve a capacity region of  $\Psi_0/2$ , and thus also has an efficiency ratio of at least  $1/2$  [8]. The MM policy simply picks a set  $\mathcal{M}(t)$  of non-interfering links such that no more links can be added to  $\mathcal{M}(t)$  without violating the node-exclusive interference constraint. To be precise, a Maximal Matching  $\mathcal{M}(t)$  is a set of non-interfering links such that: (a)  $Q_l(t) \geq c_l$  for all  $l \in \mathcal{M}$ , and (b) for each link  $l$  in the network, either  $Q_l(t) < c_l$  or some links in  $E(b(l)) \cup E(e(l))$  is included in  $\mathcal{M}$ . The distributed algorithm in [18] can compute a maximal matching in  $O(\log L)$  rounds, where  $L$  is the total number of links in the network.

There also exists a class of randomized ‘‘pick-and-compare’’ scheduling policies, which can be shown to be throughput-optimal and could be implemented in a distributed fashion [16], [17]. Their complexity is known to be at least  $O(L)$ .

2) *Scheduling Policies for the Two-Hop Interference Model:* Under the two-hop interference model, the optimal capacity region  $\Omega$  is bounded by  $\Omega \subseteq \Psi'_0$ , where

$$\Psi'_0 = \left\{ \vec{\lambda} \left| \sum_{k \in N^1(l)} \frac{\lambda_l}{c_l} \leq 1, \text{ for all } l \right. \right\}. \quad (6)$$

The policy that maximizes (3) among all set  $\mathcal{M}(t)$  of non-interfering links is still throughput-optimal. However, finding such a set  $\mathcal{M}(t)$  is generally an NP-Complete problem [28], [29]. Greedy Maximal Scheduling policy and Maximal Scheduling policy under the two-hop interference model can

be defined analogously to the GMM and MM policies, respectively, under the node-exclusive interference model. The efficiency ratio of these policies can be shown to be  $1/\widehat{N}^1$ , where  $\widehat{N}^1 \triangleq \max_l |N^1(l)|$  is the maximum number of one-hop neighboring links of any link [12]. This efficiency ratio can be tightened to  $1/\widetilde{N}^1$ , where  $\widetilde{N}^1$  is the maximum number of two-hop neighbors of each link that do not interfere with each other [9]–[11]. Neither of the two policies are constant-time scheduling policies.

### III. A CONSTANT-TIME DISTRIBUTED SCHEDULING POLICY FOR THE NODE-EXCLUSIVE INTERFERENCE MODEL

None of the distributed scheduling policies in Section II-A can compute a schedule in constant time (i.e., in a time that is independent of the network size). In this section, we propose a new distributed scheduling policy for the node-exclusive interference model that only needs  $O(1)$  time to compute a new schedule, and we will show that it achieves an efficiency ratio at least close to  $1/3$ . The new policy operates as follows:

#### Constant-Time Distributed Scheduling Policy *GP*:

At each time slot  $t$ :

- Each link  $l$  computes a probability  $p_l(t)$  based on its own queue-length and that of its one-hop neighboring links as follows:  $p_l(t) = 0$  if  $Q_l(t) = 0$ . Otherwise,

$$p_l(t) = \frac{\beta_l Q_l^\alpha(t)}{\max\left[\sum_{k \in E(b(l))} \beta_k Q_k^\alpha(t), \sum_{k \in E(e(l))} \beta_k Q_k^\alpha(t)\right]}, \quad (7)$$

where  $\alpha$  is a system-wide positive constant, and  $\beta_l$  is a positive constant for each link  $l$ .

- Each link  $l$  attempts transmission with probability  $p_l(t)$ , and does not attempt transmission with probability  $1 - p_l(t)$ . For those links that attempt transmission, each of them randomly and independently chooses a backoff time uniformly from  $\{0, 1, \dots, M - 1\}$ , where  $M$  is a system-wide positive integer constant. We assume that all backoff timers start at the beginning of the time slot. When a link’s backoff timer expires, the transmission at the link starts, provided that it has not overheard (i.e., through carrier-sensing) any other transmission from its one-hop neighboring links. Hence, the link  $l$  whose backoff timer expires ahead of all of its interfering links will win, and will be able to successfully transfer packets in the time-slot. It is possible that two or more links’ backoff timers expire at the same time, in which case collision occurs and none of the interfering links can transfer packets in time-slot  $t$ .

Note that the constant-time distributed scheduling policy in our earlier work [1] corresponds to the special case  $\alpha = 1$  and  $\beta_l = 1/c_l$ . In this paper, we allow  $\beta_l$  to take any positive value. This wider choice of  $\beta_l$  is motivated by the idea of providing preferential service to a subset of links. One would expect that a link with a larger value of  $\beta_l$  will have a better chance of being picked for transmission, and hence its backlog will be smaller. The result of this paper shows that such wider

choice of  $\beta_l$  will not degrade the overall capacity region of the system.

We make the following remarks before we derive the efficiency ratio of Policy *GP*.

**Random Backoff:** Note that the random backoff procedure in the second step of the policy is typical in random access protocols (e.g., IEEE 802.11 and Ethernet) to reduce excessive contention. In practical implementations, the actual backoff time depends both on the constant  $M$  and on how long each unit of backoff time lasts. In practice, due to propagation and processing delays, the length of each unit of backoff time cannot be arbitrarily small. For example, in IEEE 802.11, each unit of backoff time lasts  $20\mu s$ . Therefore, in order to compute the schedule in constant time, we need to provide an upper bound on  $M$ . In Section III-A, we will see how the efficiency ratio of Policy *GP* depends on  $M$ .

**Attempt Probability:** The choice of the attempt probability  $p_l$  is also essential to obtain constant-time scheduling policies with an efficiency ratio independent of the network topology. Otherwise, if  $p_l$  is lower bounded by a constant, we can show below that the throughput of the system may drop to zero. To see this, consider the simple example of  $L$  nodes transmitting to a common receiver. Hence, the  $L$  links interfere with each other. Given a fixed value of  $M$ , the probability that any one of the  $L$  links can successfully transfer data in a given time slot is bounded from above by

$$\sum_{l=1}^L \frac{p_l}{M} \prod_{k \neq l} \left(1 - \frac{p_k}{M}\right) + \prod_{l=1}^L \left(1 - \frac{p_l}{M}\right),$$

where the first term is the probability that one link wins with backoff time equal to 1, and the second term is the probability that no links have backoff times equal to 1. If  $p_l$  is bounded from below by a constant  $p$ , then the above bound will go to zero as  $L \rightarrow \infty$ . Hence, the total system throughput will drop to zero for any fixed value of  $M$ . On the other hand, since in (7) we set the attempt probability inversely proportional to the sum of the queue-length at the interfering links, we reduce the chance of contention in the neighborhood. As we will see in Section III-A, a fixed value of  $M$  will then be sufficient to guarantee a fixed efficiency ratio for *arbitrary* network topologies.

**Overhead of Queue-length Exchange:** To compute the attempt probability, Policy *GP* requires each link to learn the queue-length of its one-hop neighboring links, which also consumes time and communication overhead. Assume that the number of one-hop neighbors of each link is at most  $\widehat{N}^1$ , which can be obtained as a function of the maximum node-degree of the network if the degree of the nodes in the network is bounded. Further, assume that transmitting one piece of queue-length information takes constant time. It is then possible to design a distributed algorithm for information exchange such that the amount of time required to exchange the queue-length information is bounded by  $(\widehat{N}^1)^2 + 1$ . Specifically, using results from graph-coloring, we can color the links with at most  $(\widehat{N}^1)^2 + 1$  colors such that any two links within two-hop of each other are labeled with different colors [30, Chapter 7]. We can then assign each color to a mini-slot, and have

each link broadcast its queue-length within the mini-slot that corresponds to its color. Since the links with the same color are at least three-hop apart, their neighbors will be able to receive the broadcast information without collision. This mechanism allows all links to broadcast their queue-length to all of their respective one-hop neighbors in at most  $(\widehat{N}^1)^2 + 1$  mini-slots. Note that the graph coloring and the mini-slot assignment only need to be computed once at the beginning of the system operation, and they are independent of the traffic load. The graph coloring can be computed by a distributed algorithm such as the one in [30, Chapter 7]. After that, each link only needs to remember which mini-slot it should use to broadcast queue-length information. Hence, the above scheme can be implemented in a distributed fashion provided that the clocks at all links are synchronized. As a result, given a fixed  $M$  and a bounded maximum node-degree, the total time required for Policy *GP* to compute a new schedule can be made independent of the size of the network.

Finally, we note that the special case of Policy *GP* with  $\alpha = 1$  and  $\beta_l = 1/c_l$  can also be viewed as an extension of the Longest-Queue-Driven (LQD) scheduling algorithm from the switching literature [15]. However, there are two key differences: (a) in the switching literature, the network topology is a bipartite graph, while ad hoc network topology is non-bipartite; (b) in the switching literature, the transmitting nodes (i.e., input ports) and receiving node (i.e., output ports) are determined a priori, while in ad hoc networks a node can alternate its role as transmitter or receiver from time-slot to time-slot. The proposed policy *GP* has carefully accounted for these differences through the random backoff phase in the second part of the policy.

#### A. The Efficiency Ratio of Policy *GP*

We next show that the efficiency ratio of the above policy *GP* is at least close to  $1/3$ . Recall that the optimal capacity region  $\Omega$  under the node-exclusive interference model is upper bounded by  $\Psi_0$  in (5). For a vector  $\vec{Q} = [Q_1, \dots, Q_L]$ , define function  $\pi_l(\vec{Q})$  for each link  $l$  as

$$\pi_l(\vec{Q}) = \begin{cases} 0, & \text{if } Q_l = 0 \\ \frac{\beta_l Q_l^\alpha}{\max\left\{\sum_{k \in \mathcal{E}(b(l))} \beta_k Q_k^\alpha, \sum_{k \in \mathcal{E}(e(l))} \beta_k Q_k^\alpha\right\}}, & \text{otherwise.} \end{cases}$$

Note that this function is simply the relationship used to determine the attempt probability at each time slot in (7), i.e.,  $p_l(t) = \pi_l(\vec{Q}(t))$ . For any two vectors  $\vec{x} = [x_1, \dots, x_d]$  and  $\vec{y} = [y_1, \dots, y_d]$  with the same dimension  $d$ , we say that  $\vec{x}$  is *longer* than  $\vec{y}$  if  $\sum_{i=1}^d x_i^2 > \sum_{i=1}^d y_i^2$ . For any  $\vec{Q} \neq 0$ , let  $\vec{\mu}(\vec{Q}) = [\mu_1, \dots, \mu_L]$  be the *longest* element in  $\Psi_0$  such that for some non-negative real number  $\delta$ ,

$$\mu_l = \delta c_l \beta_l Q_l^\alpha, \text{ for all } l. \quad (8)$$

Note that such a longest element  $\vec{\mu}(\vec{Q})$  always exists and is unique because vectors that satisfy (8) are on the same line. According to (8),  $\mu_l(\vec{Q}) = 0$  if  $Q_l = 0$ . Further, if  $\vec{Q} = 0$ , we define  $\vec{\mu}(\vec{Q}) = 0$ .

Let  $\psi_l(\vec{Q})$  be the expected amount of available service provided by policy *GP* to link  $l$  at a given time slot, conditioned on the queue length vector being  $\vec{Q}$ , i.e.,  $\psi_l(\vec{Q}) =$

$\mathbf{E}[D_l(t)|\vec{Q}(t) = \vec{Q}]$ , where the expectation is taken with respect to the randomness in the random-attempt and backoff procedures of Policy *GP*. We will see soon that the relationship between the expected available service  $\psi_l(\vec{Q})$  and the value of  $\mu_l(\vec{Q})$  plays a key role in determining the stability of the system. Let  $d_l^0(\vec{Q}) = \pi_l(\vec{Q})c_l$ , which can be interpreted as the expected amount of available service at link  $l$  if *there was no collision*, conditioned on the queue length vector being  $\vec{Q}$ . Let  $\vec{d}^0(\vec{Q}) = [d_1^0(\vec{Q}), \dots, d_L^0(\vec{Q})]$ . The following lemma relates  $\vec{d}^0(\vec{Q})$  to  $\vec{\mu}(\vec{Q})$ .

**Lemma 1:** If  $\vec{Q} \neq 0$ , then for all  $l$ , we have  $d_l^0(\vec{Q}) \geq \mu_l(\vec{Q})$ .

**Proof:** By definition of  $\mu_l(\vec{Q})$ , there exists some  $\delta > 0$  such that  $\mu_l(\vec{Q}) = \delta c_l \beta_l Q_l^\alpha$  for all  $l$ . Substituting into (5), we have

$$\delta \sum_{k \in E(i)} \beta_k Q_k^\alpha \leq 1, \text{ for all node } i.$$

Since  $\vec{\mu}(\vec{Q})$  forms the longest vector in  $\Psi_0$  that satisfies the above inequality, we have

$$\delta = \frac{1}{\max_i \sum_{k \in E(i)} \beta_k Q_k^\alpha}.$$

Combining with (7), we have

$$\delta \leq \frac{\pi_l(\vec{Q})}{\beta_l Q_l^\alpha}, \text{ for all } l.$$

Hence,

$$d_l^0(\vec{Q}) = \pi_l(\vec{Q})c_l \geq \delta c_l \beta_l Q_l^\alpha = \mu_l(\vec{Q}), \text{ for all } l.$$

Lemma 1 shows that, if links that attempt transmission were to win every time, then the expected amount of service available to each link would be component-wise no less than  $\vec{\mu}(\vec{Q})$ . However, due to the random-backoff procedure in the second part of Policy *GP*, only a subset of those links that attempt transmission will win. We next show that, if a link attempts transmission, the conditional probability that it wins is no less than  $\frac{1}{3} - \frac{1}{M}$ . In fact, we will prove a more general result as follows. Fix a particular link 0. Label its interfering links as 1, 2, ...,  $K$ .

**Lemma 2:** Let  $x_k$  denote the probability that the  $k$ -th interfering link attempts transmission,  $k = 1, 2, \dots, K$ . Assume that all links follow the random backoff procedure in the second part of Policy *GP*. If  $\sum_{k=1}^K x_k \leq H$ , where  $H \geq 0$ , then the conditional probability that link 0 wins, conditioned on it attempts transmission, is no less than  $\frac{1}{H+1} - \frac{1}{M}$ .

**Proof:** Condition the following derivation on the event that link 0 attempts transmission. Let  $Y$  be the random variable that denote the backoff time of link 0. Conditioned on  $Y = y$ , the probability that link 0 wins is no less than the probability that all  $K$  interfering links either do not attempting transmission, or have backoff time greater than  $y$ . Note that each interfering link attempts transmission and chooses its backoff time independently. Let  $S$  denote the event that link

0 wins. We thus have,

$$\begin{aligned} \mathbf{P}[S|Y = y] &\geq \prod_{k=1}^K \left[ \frac{(M-1-y)x_k}{M} + (1-x_k) \right] \\ &= \prod_{k=1}^K \left[ 1 - \frac{y+1}{M} x_k \right]. \end{aligned}$$

Since  $Y$  is uniformly distributed among  $\{0, \dots, M-1\}$ , we have

$$\begin{aligned} \mathbf{P}[S] &= \sum_{y=0}^{M-1} \frac{\mathbf{P}[S|Y = y]}{M} \\ &\geq \sum_{y=0}^{M-1} \frac{1}{M} \prod_{k=1}^K \left[ 1 - \frac{y+1}{M} x_k \right]. \end{aligned}$$

Since  $\prod_{k=1}^K (1 - ux_k)$  is decreasing in  $u$ , we have,

$$\begin{aligned} \mathbf{P}[S] &\geq \int_{\frac{1}{M}}^1 \prod_{k=1}^K (1 - ux_k) du \\ &\geq \int_0^1 \prod_{k=1}^K (1 - ux_k) du - \frac{1}{M}. \end{aligned}$$

By comparing the derivatives, we can show that

$$\prod_{k=1}^K (1 - ux_k) \geq (1 - u)^H.$$

Hence,

$$\mathbf{P}[S] \geq \int_0^1 (1 - u)^H du - \frac{1}{M} = \frac{1}{H+1} - \frac{1}{M}.$$

**Remark:** A special case of Lemma 2 that corresponds to  $H = 1$  and  $M = \infty$  was shown in Theorem 5 of [15]. Here we have provided a more general result using a much different proof technique.

Under Policy *GP*, we infer from (7) that, for any link  $l$ , the attempt probabilities of its one-hop neighboring links must satisfy  $\sum_{k \in E(b(l))} p_k(t) \leq 1$  and  $\sum_{k \in E(e(l))} p_k(t) \leq 1$ . Hence, the sum of the attempt probabilities over its interfering links is no greater than 2. We thus obtain the following corollary to Lemma 2.

**Corollary 3:** Under Policy *GP*, the conditional probability that link  $l$  wins, conditioned on it attempts transmission, is no less than  $\frac{1}{3} - \frac{1}{M}$ .

Let

$$\frac{1}{S} = \frac{1}{3} - \frac{1}{M}.$$

Using Lemma 1 and Corollary 3, we thus conclude that, conditioned on the queue-length vector being  $\vec{Q}$ , the expected available service  $\psi_l(\vec{Q})$  at link  $l$  under Policy *GP* satisfies

$$\psi_l(\vec{Q}) \geq \left( \frac{1}{3} - \frac{1}{M} \right) d_l^0(\vec{Q}) \geq \frac{1}{S} \mu_l(\vec{Q}) \text{ for all } l. \quad (9)$$

Note that by definition  $\psi_l(\vec{Q}) = \mathbf{E}[D_l(t)|\vec{Q}(t) = \vec{Q}]$ . The following proposition will connect Inequality (9) with the efficiency ratio of the scheduling policy.

*Proposition 4:* If for any  $\vec{Q} \neq 0$ ,  $\mathbf{E}[D_l(t)|\vec{Q}(t) = \vec{Q}] \geq \mu_l(\vec{Q})/S$  for all  $l$ , where  $\mu_l(\vec{Q})$  is defined in (8) and  $S$  is a positive constant, then the system is stable for any offered load vector  $\vec{\lambda}$  strictly inside  $\Psi_0/S$ , i.e., for any offered load vector  $\vec{\lambda}$  such that  $(1 + \epsilon)\vec{\lambda} \in \Psi_0/S$  for some positive number  $\epsilon$ .

*Remark:* This result and the following proof technique are inspired by the work in [31]. The authors of [31] show that  $D_l(t) = \mu_l(\vec{Q}(t))$  corresponds to a throughput-optimal policy, which is a special case of Proposition 4 with  $S = 1$ . The throughput-optimality in [31] is shown for all  $\alpha > 1$ . In this paper, we use a simple fluid-limit argument and prove Proposition 4 for all  $\alpha > 0$ .

We will use a Lyapunov function of the following form to show Proposition 4, which is slightly different from the one used in [31]:

$$V(\vec{Q}) = \max_l Q_l \left( \frac{c_l \beta_l}{\lambda_l} \right)^{1/\alpha}. \quad (10)$$

The following lemma shows that if the maximum in (10) is attained by some link  $k$ , then  $\mu_k(\vec{Q}) \geq (1 + \epsilon)S\lambda_k$ .

*Lemma 5:* Assume that  $(1 + \epsilon)\vec{\lambda} \in \Psi_0/S$  for some positive constants  $\epsilon$  and  $S$ . Given  $\vec{Q} \neq 0$ , if for some  $k$  the maximum of (10) is attained, i.e.,

$$Q_k \left( \frac{c_k \beta_k}{\lambda_k} \right)^{1/\alpha} = \max_l Q_l \left( \frac{c_l \beta_l}{\lambda_l} \right)^{1/\alpha}, \quad (11)$$

then  $\mu_k(\vec{Q}) \geq (1 + \epsilon)S\lambda_k$ .

*Proof:* We prove by contradiction. Assume that  $\mu_k(\vec{Q}) < (1 + \epsilon)S\lambda_k$ , where  $k$  is defined in (11). Then for all other links  $l$ , either  $\mu_l(\vec{Q}) = 0$ , or, if  $\mu_l(\vec{Q}) > 0$ , we have (by the definition of  $\mu_l(\vec{Q})$ ),

$$\frac{c_l \beta_l Q_l^\alpha}{\mu_l(\vec{Q})} = \frac{c_k \beta_k Q_k^\alpha}{\mu_k(\vec{Q})}.$$

Using (11), in either case we will have,

$$\begin{aligned} \mu_l(\vec{Q}) &= \frac{c_l \beta_l Q_l^\alpha}{c_k \beta_k Q_k^\alpha} \mu_k(\vec{Q}) \\ &= \frac{c_l \beta_l Q_l^\alpha}{\lambda_l} \frac{\lambda_k}{c_k \beta_k Q_k^\alpha} \mu_k(\vec{Q}) \frac{\lambda_l}{\lambda_k} \\ &\leq \mu_k(\vec{Q}) \frac{\lambda_l}{\lambda_k} \quad (\text{using (11)}) \\ &< (1 + \epsilon)S\lambda_l, \quad \text{for all } l. \end{aligned}$$

Since  $(1 + \epsilon)S\vec{\lambda} \in \Psi_0$ , this implies that  $\vec{\mu}(\vec{Q})$  cannot be the longest vector in  $\Psi_0$  that satisfies (8), which contradicts with the definition of  $\vec{\mu}(\vec{Q})$ . Thus, the result of the lemma must hold. ■

We can now prove Proposition 4.

*Proof:* (of Proposition 4) We first prove the stability of the fluid model of the system, where the fluid model is defined as follows [14], [19]. For any integer  $t > 0$ , let  $E_l(t) = \sum_{s=0}^{t-1} A_l(s)$  denote the total number of arrivals to link  $l$  in time-slots 0 to  $t - 1$ . Let  $T_l(t) = \sum_{s=0}^{t-1} D_l(s)$  be the total amount of available service to link  $l$  in time-slots 0 to  $t - 1$ . Further, let  $E_l(0) =$

$T_l(0) = 0$ . Then the evolution of the queue length can be written as

$$Q_l(t) = Q_l(0) + E_l(t) - F_l(t), \quad (12)$$

where  $F_l(t) = \sum_{s=1}^t \min\{Q_l(s-1) + E_l(s) - E_l(s-1), T_l(s) - T_l(s-1)\}$ . We interpolate the values of  $E_l(t)$ ,  $T_l(t)$ ,  $Q_l(t)$  and  $F_l(t)$  to all non-negative real number  $t$  by linear interpolation between  $\lfloor t \rfloor$  and  $\lfloor t \rfloor + 1$  (where  $\lfloor t \rfloor$  denotes the largest integer no greater than  $t$ ). Then, using the techniques of Theorem 4.1 of [19], we can show that, for almost all sample paths and for any positive sequence  $x_n \rightarrow \infty$ , there exists a subsequence  $x_{n_j}$  with  $x_{n_j} \rightarrow \infty$  such that the following convergence holds uniformly over compact intervals of time  $t$ :

$$\begin{aligned} \frac{1}{x_{n_j}} E_l(x_{n_j} t) &\rightarrow \lambda_l t \quad \text{for all } l \\ \frac{1}{x_{n_j}} T_l(x_{n_j} t) &\rightarrow \nu_l(t) \quad \text{for all } l \\ \frac{1}{x_{n_j}} Q_l(x_{n_j} t) &\rightarrow q_l(t) \quad \text{for all } l \\ \frac{1}{x_{n_j}} F_l(x_{n_j} t) &\rightarrow f_l(t) \quad \text{for all } l, \end{aligned} \quad (13)$$

where  $\nu_l(t)$ ,  $q_l(t)$  and  $f_l(t)$  are continuous functions. Further, since the functions  $T_l(t)$ ,  $Q_l(t)$  and  $F_l(t)$  are Lipschitz continuous, so are the functions  $\nu_l(t)$ ,  $q_l(t)$  and  $f_l(t)$ . Hence, these limiting functions are differentiable for almost all  $t$ . Let  $\mathcal{T}$  denote the set of time instants where these limiting functions are differentiable. Let  $\vec{q}(t) = [q_1(t), q_2(t), \dots, q_L(t)]$ ,  $\vec{\nu}(t) = [\nu_1(t), \dots, \nu_L(t)]$ , and  $\vec{f}(t) = [f_1(t), f_2(t), \dots, f_L(t)]$ . Using the techniques of Theorem 4.1 of [19] again, we can show that the limiting functions must satisfy the following set of equations: for all  $l$  and for all  $t \in \mathcal{T}$ ,

$$\frac{d}{dt} q_l(t) = \lambda_l - \frac{d}{dt} f_l(t) \quad (14)$$

$$\frac{d}{dt} f_l(t) = \frac{d}{dt} \nu_l(t) \quad \text{if } q_l(t) > 0 \quad (15)$$

$$\frac{d}{dt} \nu_l(t) \geq \frac{1}{S} \mu_l(\vec{q}(t)) \quad \text{if } q_l(t) > 0. \quad (16)$$

To see this, note that (14) follows from the queue-evolution equation (12) by taking limits of the form in (13) as  $x_{n_j} \rightarrow \infty$ . To show (15), note that, if  $q_l(t) > 0$ , then there exists a positive  $\delta$  such that for all  $s \in [t, t + \delta]$ ,  $q_l(s) > 0$ . This implies that for all sufficiently large  $j$ , the backlog  $Q_l(\lfloor x_{n_j} s \rfloor)$  at link  $l$  is larger than its maximum capacity  $c_l$  for all  $s \in [t, t + \delta]$ . Therefore, the available service to link  $l$  will be fully utilized during the time interval  $[\lfloor x_{n_j} t \rfloor, \lfloor x_{n_j} (t + \delta) \rfloor]$ . We thus have,

$$F_l(\lfloor x_{n_j} s' \rfloor) - F_l(\lfloor x_{n_j} s \rfloor) = T_l(\lfloor x_{n_j} s' \rfloor) - T_l(\lfloor x_{n_j} s \rfloor)$$

for all  $t \leq s \leq s' \leq t + \delta$ . Dividing both sides by  $x_{n_j}$ , and taking limits as  $x_{n_j} \rightarrow \infty$ , we have

$$f_l(s') - f_l(s) = \nu_l(s') - \nu_l(s)$$

for all  $t \leq s \leq s' \leq t + \delta$ . Equation (15) then follows. Finally, (16) follows from the condition  $\mathbf{E}[D_l(t)|\vec{Q}(t)] \geq \mu_l(\vec{Q}(t))/S$ . To see this, fix a link  $l$  such that  $q_l(t) > 0$ . Note that  $\vec{q}(t)$  is Lipschitz-continuous with respect to  $t$ , and  $\mu_l(\vec{q})$  is continuous

with respect to  $\vec{q}$  when  $q_l > 0$ . Hence, for any  $\epsilon > 0$ , there exists a  $\delta > 0$  such that for all  $s \in [t, t + \delta]$ ,

$$\mu_l(\vec{q}(s)) \geq \mu_l(\vec{q}(t)) - \epsilon.$$

Further, since  $\vec{Q}(\lfloor x_{n_j} s \rfloor) \rightarrow \vec{q}(s)$  uniformly over compact intervals of time, and  $\mu(\vec{Q}) = \mu(a\vec{Q})$  for any  $a > 0$ , there exists an integer  $J > 0$  such that for all  $j > J$  and  $s \in [t, t + \delta]$ ,

$$\mu_l(\vec{Q}(\lfloor x_{n_j} s \rfloor)) \geq \mu_l(\vec{q}(t)) - 2\epsilon. \quad (17)$$

By the definition of the limits in (13), for any  $s \in [t, t + \delta]$ ,

$$\nu_l(s) - \nu_l(t) = \lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} \sum_{k=\lfloor x_{n_j} t \rfloor}^{\lfloor x_{n_j} s \rfloor} D_l(k). \quad (18)$$

Define the filtration  $\mathcal{F}_k, k = 1, 2, \dots$ , where  $\mathcal{F}_k$  is the  $\sigma$ -algebra generated by the random variables  $E_l(\lfloor x_{n_j} t \rfloor + k'), T_l(\lfloor x_{n_j} t \rfloor + k')$  and  $Q_l(\lfloor x_{n_j} t \rfloor + k')$  for all  $l$  and for  $k' = 1, 2, \dots, k$ . Let

$$X_k = D_l(\lfloor x_{n_j} t \rfloor + k) - \mathbf{E}[D_l(\lfloor x_{n_j} t \rfloor + k) | \vec{Q}(\lfloor x_{n_j} t \rfloor + k)]$$

and let  $Y_k = \sum_{h=1}^k X_h$ . Then  $Y_k, k = 1, 2, \dots$  is a martingale with respect to the filtration  $\mathcal{F}_k, k = 1, 2, \dots$  [32, p228]. Further,  $\mathbf{E}[X_k^2]$  is bounded by  $c_l^2$  for all  $k$ . Hence, using a strong law of large numbers for martingales [33], we have

$$\lim_{k \rightarrow \infty} \frac{Y_k}{k} = 0.$$

Substituting into (18), we have

$$\begin{aligned} \nu_l(s) - \nu_l(t) &= \lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} \sum_{k=\lfloor x_{n_j} t \rfloor}^{\lfloor x_{n_j} s \rfloor} D_l(k) \\ &= \lim_{j \rightarrow \infty} \frac{1}{x_{n_j}} \sum_{k=\lfloor x_{n_j} t \rfloor}^{\lfloor x_{n_j} s \rfloor} \mathbf{E}[D_l(k) | \vec{Q}(k)]. \end{aligned}$$

By (17) and the condition that  $\mathbf{E}[D_l(t) | \vec{Q}(t)] \geq \mu_l(\vec{Q}(t))/S$ , each term of  $\mathbf{E}[D_l(k) | \vec{Q}(k)]$  is no smaller than  $\mu_l(\vec{q}(t)) - 2\epsilon$  for  $j > J$  and for  $k$  between  $\lfloor x_{n_j} t \rfloor$  and  $\lfloor x_{n_j} s \rfloor$ . Hence,

$$\nu_l(s) - \nu_l(t) \geq \frac{s-t}{S} [\mu_l(\vec{q}(t)) - 2\epsilon]$$

for  $s \in [t, t + \delta]$ . Since we assume that  $\nu_l(t)$  is differentiable at  $t$  (i.e.,  $t \in T$ ), we have

$$\frac{d}{dt} \nu_l(t) \geq \frac{1}{S} [\mu_l(\vec{q}(t)) - 2\epsilon].$$

Finally, since this is true for all  $\epsilon > 0$ , Equation (16) then follows.

Any such limit  $[\vec{q}(t), \vec{\nu}(t), \vec{f}(t)]$  is called a *fluid limit* of the system. We say that a fluid limit model of the system is stable if there exists a constant  $T$  that depends only on the network topology, the arrival rates  $\lambda_l$  and the active link capacities  $c_l$ , such that for any fluid limit with  $\|\vec{q}(0)\| = 1$ , we have  $\|\vec{q}(t)\| = 0$  for all  $t \geq T$  [19].

We next use the Lyapunov function (10) on the fluid model. For all  $t \in T$  where the functions  $\vec{q}(t), \vec{\nu}(t)$  and  $\vec{f}(t)$  are differentiable, we have [34, p28],

$$\frac{D^+ V(\vec{q}(t))}{dt^+} \leq \max_{l \in R(t)} \frac{d}{dt} q_l(t) \left( \frac{c_l \beta_l}{\lambda_l} \right)^{1/\alpha},$$

where for any function  $f(t)$ ,  $\frac{D^+}{dt^+} f(t)$  is defined as  $\limsup_{u \downarrow 0} \frac{f(t+u) - f(t)}{u}$ , and

$$R(t) = \left\{ k | q_k(t) \left( \frac{c_k \beta_k}{\lambda_k} \right)^{1/\alpha} = \max_l q_l(t) \left( \frac{c_l \beta_l}{\lambda_l} \right)^{1/\alpha} \right\}.$$

If  $\vec{q}(t) \neq 0$ , then by Lemma 5,  $l \in R(t)$  implies  $\mu_l(\vec{q}(t)) \geq (1 + \epsilon) S \lambda_l$ . Further,  $q_l(t) > 0$  for  $l \in R(t)$ . Hence, using (14)-(16), we have,

$$\frac{d}{dt} q_l(t) = \lambda_l - \frac{d}{dt} \nu_l(t) \leq -\epsilon \lambda_l, \text{ for all } l \in R(t),$$

and thus,

$$\frac{D^+ V(\vec{q}(t))}{dt^+} \leq -\epsilon \min_l \lambda_l \left( \frac{c_l \beta_l}{\lambda_l} \right)^{1/\alpha}, \text{ if } \vec{q}(t) \neq 0.$$

Since the above property holds for almost all  $t$ , we can then conclude that the fluid limit model of the system is stable. By Theorem 4.2 of [19], the original system is positive Harris recurrent. Note that positive Harris recurrence implies that the stochastic process  $\vec{Q}(t), t \geq 0$  has a unique stationary distribution  $\Pi$ . Further, for every measurable function  $f(\vec{Q})$  with  $\Pi(|f|) < \infty$ , the following ergodic property holds [19, Section 3]

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T f(\vec{Q}(t)) = \Pi(f) \text{ almost surely.}$$

Taking  $f(\vec{Q}(t)) = \mathbf{1}_{\{\sum_{i=1}^L Q_i(t) > \eta\}}$ , and noting that

$\Pi \left( \mathbf{1}_{\{\sum_{i=1}^L Q_i(t) > \eta\}} \right) \rightarrow 0$  as  $\eta \rightarrow \infty$  (since  $\Pi$  is a finite measure), Equation (2) thus holds. ■

Combining with (9), we obtain the following immediate result:

*Corollary 6:* The policy  $GP$  can stabilize the network as long as the offered load vector  $\vec{\lambda}$  lies strictly inside  $(\frac{1}{3} - \frac{1}{M}) \Psi_0$ , i.e., as long as there exists a positive constant  $\epsilon$  such that

$$(1 + \epsilon) \vec{\lambda} \in \left( \frac{1}{3} - \frac{1}{M} \right) \Psi_0.$$

*Remarks:* Since the optimal capacity region  $\Omega$  of the system is a subset of  $\Psi_0$ , we conclude that the efficiency ratio of Policy  $GP$  is at least  $\frac{1}{3} - \frac{1}{M}$ . For any given  $\epsilon > 0$ , we can choose the maximum backoff time  $M = 1/\epsilon$ , which then ensures that the efficiency ratio of Policy  $GP$  is no less than  $1/3 - \epsilon$ . The parameter  $\epsilon$  can be viewed as an *approximation ratio* as to how close one want to approach  $1/3$ . Given  $\epsilon$ , the value of  $M$  is independent of the network topology. As we discussed earlier, one can design distributed algorithms for information exchange such that the amount of time required for all links to learn the queue-length information of their

one-hop neighbors can be bounded by a function of the maximum node-degree of the network. Hence, given a fixed approximation ratio  $\epsilon$  and a bounded maximum node-degree, Policy *GP* only takes constant time and can guarantee an efficiency ratio close to  $1/3$  for arbitrary network topologies.

*B. An Alternate Lyapunov Function for the Case When  $\alpha = 1$  and  $\beta_l = 1/c_l$*

For the case when  $\alpha = 1$  and  $\beta_l = 1/c_l$ , Policy *GP* reduces to Policy *P* in our prior work [1]. There, an alternate Lyapunov function was used to establish the efficiency ratio of Policy *P*. This alternate Lyapunov function does not depend on the arrival rates  $\lambda_l$ , and hence could be of independent interest [35], [36]. Due to space constraints, we omit the details here. Interested readers can refer to [1].

#### IV. A CONSTANT-TIME DISTRIBUTED SCHEDULING POLICY FOR THE TWO-HOP INTERFERENCE MODEL

We next extend the constant-time distributed policy in the previous section to the two-hop interference model. Under the two-hop interference model, the known distributed scheduling policies, i.e., the Maximal Scheduling Policy [9]–[12] and the distributed implementation of the Greedy Scheduling Policy [28], can both guarantee a worst-case efficiency ratio of  $1/\widehat{N}^1$ , where  $\widehat{N}^1 \triangleq \max_l |N^1(l)|$  is the maximum number of one-hop neighboring links for any link. However, they are again not constant-time policies. We now propose a constant-time distributed scheduling policy *GQ* that can guarantee a comparable worst-case efficiency ratio. (We note however that the worst-case efficiency ratio of Maximal Scheduling and Greedy Scheduling can be tightened to  $1/\widetilde{N}^1$ , where  $\widetilde{N}^1$  is the maximum number of two-hop neighbors of each link that do not interfere with each other [9]–[11]. The value of  $\widetilde{N}^1$  can be smaller than  $\widehat{N}^1$  for certain types of network topologies, e.g., with geometric graphs [9]. Thus, the efficiency ratios of Maximal Scheduling and Greedy Scheduling will be better than that of Policy *GQ* for those types of network topologies.)

Let  $W$  be a positive number between 1 and  $\widehat{N}^1$ . We will see soon that the parameter  $W$  puts an upper bound on the sum of the attempt probabilities in any two-hop neighborhood.

##### Constant-Time Distributed Scheduling Policy *GQ*:

At each time slot  $t$ :

- Each link  $l$  computes a probability  $p_l(t)$  based on its own queue-length and that of the interfering links as follows:  $p_l(t) = 0$  if  $Q_l(t) = 0$ . Otherwise,

$$p_l(t) = \frac{\beta_l Q_l^\alpha(t)}{\max_{k \in N^1(l)} \sum_{h \in N^1(k)} \beta_h Q_h^\alpha(t)} \times \min \left( 1, \frac{W}{\max_{k \in N^2(l)} |N^1(k)|} \right),$$

where  $\alpha$  is a system-wide positive constant, and  $\beta_l$  is a positive constant for each link  $l$ .

- Each link  $l$  attempts transmission with probability  $p_l(t)$ , and does not attempt transmission with probability  $1 - p_l(t)$ . For those links that attempt transmission, each of

them randomly chooses a backoff time uniformly from  $\{0, 1, \dots, M - 1\}$ . We assume that all backoff timers start at the beginning of the time slot. When the backoff timer of a link  $l$  expires, the transmitter node  $b(l)$  of link  $l$  will broadcast an RTS to all of its one-hop neighboring nodes, provided that node  $b(l)$  has not overheard any RTS from these one-hop neighboring nodes. Once the receiver node  $e(l)$  correctly receives the RTS, it will then respond with a CTS broadcasted to all of its neighboring nodes.<sup>1</sup> Through this RTS-CTS procedure, the link  $l$  that sends out an RTS before any of its two-hop neighboring links will win. This link  $l$  can then transfer packets at the rate of  $c_l$  during the rest of the time slot. It is possible that two or more links in a two-hop neighborhood send out RTS together, in which case collision occurs and none of the interfering links can transfer data in time-slot  $t$ .

We can use similar techniques as in Section III to show that policy *GQ* guarantees an efficiency ratio close to  $1/(\widehat{N}^1 + 1)$ . To see this, note that under the two-hop interference model, the optimal capacity region  $\Omega$  is upper bounded by  $\Psi'_0$  in (6). As in Section III, let  $d_l^0(\vec{Q})$  denote the expected amount of available service at link  $l$  if *there was no collision*, conditioned on the queue-length vector being  $\vec{Q}$ . Let  $\vec{d}^0(\vec{Q}(t)) = [d_1^0(\vec{Q}(t)), \dots, d_L^0(\vec{Q}(t))]$ . Using the technique of Lemma 1, we have

$$\vec{d}^0(\vec{Q}(t)) \succeq \frac{W}{\widehat{N}^1} \vec{\mu}(\vec{Q}(t)),$$

where  $\vec{\mu}(\vec{Q}(t))$  is the longest vector in  $\Psi'_0$  such that (8) holds. Further, for each link  $l$ , the sum of the attempt probabilities of its interfering links (i.e., its two-hop neighboring links) satisfies the following relationship: for all  $k \in N^1(l)$ ,

$$\begin{aligned} \sum_{h \in N^1(k)} p_h &\leq \sum_{h \in N^1(k)} \frac{\beta_h Q_h^\alpha(t)}{\max_{m \in N^1(h)} \sum_{n \in N^1(m)} \beta_n Q_n^\alpha(t)} \\ &\quad \times \min \left( 1, \frac{W}{\max_{m \in N^2(h)} |N^1(m)|} \right) \\ &\leq \sum_{h \in N^1(k)} \frac{\beta_h Q_h^\alpha(t)}{\sum_{n \in N^1(k)} \beta_n Q_n^\alpha(t)} \frac{W}{|N^1(l)|} \\ &\quad (\text{because } l \in N^2(h) \text{ and } k \in N^1(h)) \\ &\leq \frac{W}{|N^1(l)|}. \end{aligned}$$

We thus have,

$$\sum_{k \in N^2(l)} p_k \leq \sum_{k \in N^1(l)} \sum_{h \in N^1(k)} p_h \leq \sum_{k \in N^1(l)} \frac{W}{|N^1(l)|} = W.$$

Therefore, using Lemma 2 with  $H = W$ , and using the technique of Proposition 4, we can show the following main result.

*Proposition 7:* Under Policy *GQ*, the network is stable when  $\vec{\lambda}$  lies strictly inside the set  $\frac{W}{N^1} (\frac{1}{1+W} - \frac{1}{M}) \Psi'_0$ .

<sup>1</sup>We assume that the time required for this RTS-CTS procedure is less than one unit of backoff time.



*Remark:* For any fixed  $W$ , by choosing  $M \geq \frac{1}{(1+W)(2+W)}$ , the efficiency ratio of Policy  $GQ$  is at least  $\frac{W}{(2+W)\widehat{N}^1}$ . For each  $W$ , the value of  $M$  is independent of the network topology. Hence, once each link learns the queue-length information within its two-hop neighborhood,  $GQ$  only requires a constant time to compute a new schedule. Further, for  $W = \widehat{N}^1$ , the guaranteed efficiency ratio becomes  $\frac{1}{1+\widehat{N}^1} - \frac{1}{M}$ . By letting  $M \rightarrow \infty$ , the guaranteed efficiency ratio goes to  $\frac{1}{1+\widehat{N}^1}$ . The difference  $\frac{1}{M}$  can be viewed as an approximation ratio as to how close one wants to approach  $\frac{1}{1+\widehat{N}^1}$ .

Similar to the discussions in Section III, assuming that the number of two-hop neighbors of each link is at most  $\widehat{N}^2$ , and that transmitting one piece of queue-length information takes constant time, we can then design a distributed algorithms for information exchange such that the amount of time required to exchange the queue-length information can be bounded by  $O[(\widehat{N}^2)^2 + 1]$ . The quantity  $\widehat{N}^2$  can again be written as a function of the maximum node-degree if the degree of the nodes in the network is bounded. Therefore, given a fixed approximation ratio and a bounded maximum node-degree, the total time required for Policy  $GQ$  to compute a new schedule can be made independent of the size of the network.

## V. CONSTANT-TIME SCHEDULING POLICIES FOR MULTI-HOP NETWORKS AND SUBJECT TO FEEDBACK DELAYS

In this section, we extend the constant-time scheduling policies of the previous sections to multi-hop networks, and also address the overhead of communicating the queue lengths. We will focus on the node-exclusive interference model (and Policy  $GP$ ), while the same methodology can be applied to the two-hop interference model (and Policy  $GQ$ ) as well.

### A. Constant-Time Scheduling Policy for Multihop Wireless Networks

In Section II, we have assumed a single-hop traffic model, i.e., each packet only needs to traverse one of the  $L$  links and then leaves the system. We next extend policy  $GP$  to multi-hop networks with fixed routing. Assume that there are  $S$  end-users in the system. Each user  $s$  injects packets at the rate of  $x_s$  packets per time-slot. Assume that each user has a fixed path through the network, and let  $[H_s^l]$  denote the routing matrix, where  $H_s^l = 1$  if the path of user  $s$  traverse link  $l$ , and  $H_s^l = 0$  otherwise. Thus, the aggregate data rate on link  $l$ , denoted by  $\lambda_l$ , is given by  $\lambda_l = \sum_{s=1}^S H_s^l x_s$ . Redefine the capacity region of the network under a particular scheduling policy to be the set of  $\vec{x} = [x_1, \dots, x_S]$  such that the system can remain stable. Then the optimal capacity region  $\Omega_M$  is upper bounded by,

$$\Omega_M \subset \left\{ \vec{x} \left| \left[ \sum_{s=1}^S H_s^l x_s \right] \in \Psi_0 \right. \right\},$$

where  $\Psi_0$  is given in (5). If we assume that the “queues” are updated by

$$Q_l(t+1) = \left[ Q_l(t) + \sum_{s=1}^S H_s^l x_s - D_l(t) \right]^+, \quad (19)$$

then we can show as in Section III that the system is stable under Policy  $GP$ , i.e.,  $Q_l(t)$  satisfies (2), as long as

$$\left[ \sum_{s=1}^S H_s^l x_s \right] \in \left( \frac{1}{3} - \frac{1}{M} \right) \Psi_0.$$

Thus, we have shown that, under Policy  $GP$ , the capacity region of the system is at least  $(\frac{1}{3} - \frac{1}{M})$  fraction of the optimal capacity region  $\Omega_M$ . In other words, the efficiency ratio of Policy  $GP$  remains the same for multi-hop networks.

In the above argument, we have assumed in (19) that the “queues” are updated as if the data rate from each end-user  $s$  is applied instantaneously at all links  $l$  along the path of user  $s$ . In practice, the packets from each source have to traverse the path link-by-link. Hence, the equation (19) does not describe the dynamics of the *real* queue. The removal of this “user-rates-applied-simultaneously-to-all-links” assumption could invalidate our earlier argument for stability. In fact, examples have been created in prior works for wireline networks [20], [37], [38], where a queueing network appears to be stable under this “user-rates-applied-simultaneously-to-all-links” assumption, but is actually unstable when packets traverse the network link-by-link.

There are a number of approaches from the literature to address the above issue [11], [37]–[40]. One approach is to use the idea of a *regulator* at each link, which limits the burstiness of the traffic from upstream nodes [11], [39], [40]. The other approach is to assign appropriate priorities to packets in the queue when they are served [11], [38], [40], [41]. Both approaches may be applied to Policy  $GP$  so that it retains the same efficiency ratio for multi-hop networks.

### B. Overhead of Updating the Queue-lengths

Policy  $GP$  requires each link  $l$  to learn the queue-length of neighboring links in order to compute the attempt probability  $p_l(t)$ . We have discussed in Section III how one can design a distributed algorithm for exchanging the queue-length information among neighboring links such that the time needed for information exchange is bounded by a function of the maximum node-degree. Alternatively, it is well known that even if this type of scheduling policies act upon *delayed* queue-length information, as long as the delay is bounded, the efficiency ratio will still remain the same (see, e.g., [5]). In fact, we can show that, as long as in every time-slot each link exchanges its queue-length information with a success-probability that is bounded from below by a positive constant (which implies that the expected delay of queue-length information is bounded), then under the fluid-limit scaling (13), the fluid-limit model (14-16) will remain the same. (Details are omitted due to space constraints.) Hence, Policy  $GP$  will retain the same efficiency ratio. In Section VI, we will use simulation to study the performance of Policy  $GP$  when the queue length information is exchanged infrequently.

## VI. SIMULATION RESULTS

We have simulated the proposed scheduling policies using the network topology in Fig. 1. There are 16 nodes (represented by circles), and 24 links (represented by dashed lines).

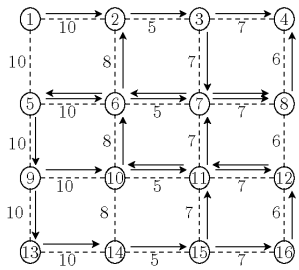
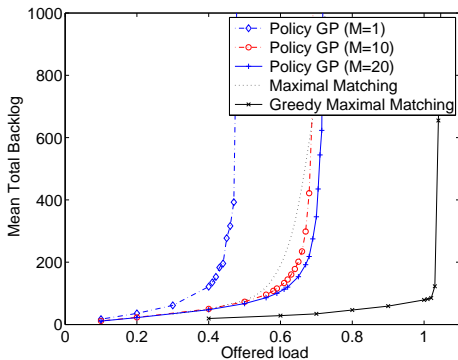


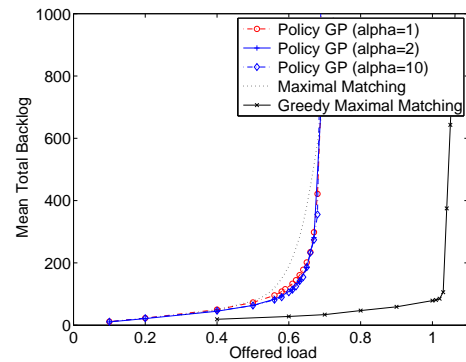
Fig. 1. Network topology


 Fig. 2. Performance comparison of Policy *GP* under the node-exclusive interference model. We have used  $\alpha = 1$  and  $\beta_l = 1/c_l$  for all links  $l$ .

The capacity is labeled next to each link. The flows are represented by arrows. We simulate single-hop flows, and we let the rate of each flow be  $\lambda$ . Note that although the rates of the flows are the same, the link capacities and the flows have been chosen to avoid uniform patterns.

We first simulate Policy *GP* (for the node-exclusive interference model) under the setting that each link learns the current queue-length information of its one-hop neighbors in every time-slot. We first choose  $\alpha = 1$  and  $\beta_l = 1/c_l$  for all links  $l$ . This is equivalent to Policy *P* in [1]. In Fig. 2, we plot the mean total queue backlog summed over all links of the network, as the offered load  $\lambda$  increases. When  $\lambda$  approaches a certain limit, the average total backlog will increase to infinity. This limit can then be viewed as the boundary of the capacity region. We have plotted the curves for Policy *GP* with maximum backoff windows  $M = 1, M = 10$ , and  $M = 20$ . We can see that the performance of the scheduling policy is much worse when  $M = 1$ . Hence the random backoff procedure in the second step of the policy is essential. However, once  $M$  is above a reasonable number, the performance will be virtually the same (as we can see for  $M = 10$  and  $M = 20$ ). We have also plotted the performance of the Maximal Matching (MM) policy and the Greedy Maximal Matching (GMM) policy. Although the efficiency ratio that can be guaranteed in Proposition 4 for policy *GP* is slightly worse than that of MM, the simulation results indicate that their actual performance is roughly the same.

We next simulate policy *GP* with other values of  $\alpha$ . In Fig. 3, we plot the curves for Policy *GP* with  $\alpha = 1, \alpha = 2$  and  $\alpha = 10$ . Other parameters are chosen as follows for all


 Fig. 3. Performance comparison of Policy *GP* under the node-exclusive interference model as the value of  $\alpha$  varies. We have used  $M = 10$  and  $\beta_l = 1/c_l$  for all links  $l$ .

simulations:  $\beta_l = 1/c_l$  and  $M = 10$ . We observe that the performance of policy *GP* is relatively invariant to the choice of  $\alpha$ , which is not surprising given the fact that with any  $\alpha$  the policy can be shown to achieve the same efficiency ratio.

We also investigate the performance of Policy *GP* when queue-length information is exchanged less frequently (i.e., not in every time-slot). We use the following procedure to update the queue-length information. At the beginning of each time slot, we reserve a small mini-slot for queue-length updates. During this mini-slot, each node  $i$  will broadcast with probability  $\epsilon$  the current length of its out-going queues, along with the most-recent queue-length information of its incoming queues that it has received from its neighbors. At each of its neighboring nodes, if this broadcast message does not collide with the broadcast messages from other links, the neighboring node is considered to correctly receive the queue-length updates. Then, when each link  $l$  computes the attempt probability  $p_l$  (we assume that this computation is carried out at its transmitter), its transmitter will use the *current* queue-length of its own out-going links, and the most-recently received queue-length information of its neighboring links. This procedure ensures that the probability with which each link  $l$  can update the current queue-length information from its neighboring link  $k$  is bounded from below by a positive constant (which is a function of  $\epsilon$ ). Hence, the expected delay of queue-length update is bounded. As discussed in Section V-B, the efficiency ratio of Policy *GP* will remain the same.

We have simulated the performance of Policy *GP* using the above procedure for exchanging queue length. In Fig. 4, we plot the simulation results for the case when there is no feedback delay (i.e., assuming that each link knows the current queue-length information of all links in the network), and for the case when  $\epsilon = 0.1$  and  $\epsilon = 0.4$ , respectively. We observe that the performance of Policy *GP* is quite insensitive to the choice of  $\epsilon$ . This indicates that our algorithm is robust to the delay in exchanging queue length information.

Finally, we simulate policy *GQ* for the two-hop interference model, under the setting that each link learns the current queue-length information of its neighboring links in every time-slot. We plot the results in Fig. 5 for  $\alpha = 1$  and  $\beta_l = 1/c_l$ . Again, we observe that the performance of policy *GQ*

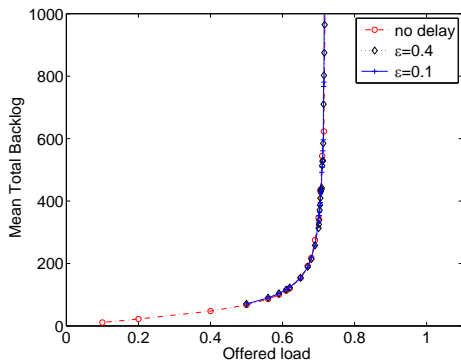


Fig. 4. The performance of Policy *GP* subject to feedback delays. We have used  $\alpha = 1$  and  $\beta_l = 1/c_l$  for all links, and  $M = 20$ .

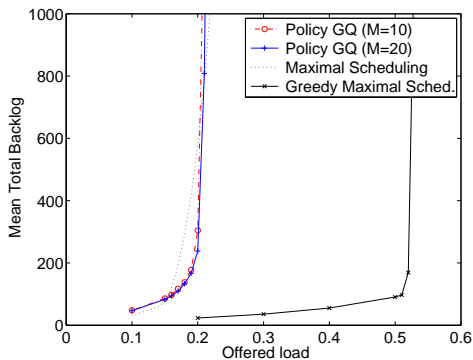


Fig. 5. Performance comparison under the two-hop interference model.

changes little when the maximum backoff window changes from  $M = 10$  to  $M = 20$ . Further, the performance is also comparable to the maximal scheduling policy.

## VII. CONCLUSION

In this paper, we have proposed a class of new distributed scheduling policies for ad hoc wireless networks. The unique feature of these new distributed scheduling policies is that they are constant-time policies, i.e., given a fixed approximation ratio and a bounded maximum node-degree of the network, the time needed for computing a new schedule is independent of the network size. Hence, they can be easily deployed in large networks. We have shown that these constant-time scheduling policies can guarantee efficiency ratios comparable to some other known distributed scheduling policies in the literature that are not constant-time.

We believe that these results offer new insights for the design of simple and efficient scheduling policies for ad hoc networks. For future work, we plan to generalize the main techniques and results to other types of interference models, e.g., the bi-directional equal-power model in [9]. We also note that Policy *GQ* for the two-hop interference model operates in a manner very similar to IEEE 802.11 DCF (Distributed Coordination Function). The main difference is: when there is excessive contention, IEEE 802.11 DCF will increase the backoff window exponentially; however, Policy *GQ* will reduce the attempt probability, and keep the backoff window

unchanged. It will be an interested direction for future work to explore the performance difference of these two approaches.

We can observe in both Fig. 2 and Fig. 5 that there is still a substantial performance gap when we compare Policy *GP* (and *GQ*, respectively) with the Greedy Maximal Matching policy (and the Greedy Maximal Scheduling policy, respectively). Note that the Greedy Maximal Matching policy and the Greedy Maximal Scheduling policy can both be implemented in a distributed fashion, although not in constant time. This opens the question as to whether one can develop simple, distributed, and constant-time scheduling policies that achieve even better performance than Policies *GP* and *GQ*. In fact, for the special case when  $\alpha = 1$  and  $\beta_l = 1/c_l$ , the more recent work in [35], [36] proposed a refined version of Policy *GP* that can guarantee an efficiency ratio close to  $1/2$ , and that empirically approximates the performance of Greedy Maximal Matching when the backoff window size is very large. However, it is not obvious how the idea of [35], [36] can be applied to the general class of Policies *GP* and *GQ*. In another paper [42], based on the idea of graph partitioning, the authors propose a class of scheduling policies that can achieve arbitrarily close to the optimal throughput with computation-time that also does not increase with the size of the network. (A different partitioning approach is proposed in [43], which may also be used to construct constant-time scheduling policies.) Compared with the policies in [42], [43], the policies that we study in this paper are much simpler, although with lower performance guarantees. In [44], the authors propose a distributed randomized algorithm for the node-exclusive interference model that achieves a similar goal as that of [42]. Thus, the scheduling policies in [35], [36], [42]–[44] and in this paper offer different tradeoffs in terms of the provable efficiency ratios, simplicity and overhead of operation, flexibility of tuning policy parameters, and applicability to a variety of network scenarios. It remains an interesting open problem whether one can develop scheduling policies that combine the benefits of all of them.

## REFERENCES

- [1] X. Lin and S. Rasool, "Constant-Time Distributed Scheduling Policies for Ad Hoc Wireless Networks," in *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, December 2006, pp. 1258–1263.
- [2] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.
- [3] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, December 1992.
- [4] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic Power Allocation and Routing for Time Varying Wireless Networks," in *Proceedings of IEEE INFOCOM*, San Francisco, April 2003, pp. 745–755.
- [5] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Stable Scheduling Policies for Fading Wireless Channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, April 2005.
- [6] R. L. Cruz and A. V. Santhanam, "Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks," in *Proceedings of IEEE INFOCOM*, San Francisco, April 2003, pp. 702–711.
- [7] X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Wireless Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005, pp. 1804–1814.

- [8] —, “The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.
- [9] P. Chaporkar, K. Kar, and S. Sarkar, “Throughput Guarantees Through Maximal Scheduling in Wireless Networks,” in *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2005.
- [10] —, “Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks,” in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.
- [11] X. Wu, R. Srikant, and J. R. Perkins, “Queue-Length Stability of Maximal Greedy Schedules in Wireless Network,” in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.
- [12] X. Wu and R. Srikant, “Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks,” in *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006, pp. 1–12.
- [13] T. Weller and B. Hajek, “Scheduling Nonuniform Traffic in a Packet-switching System with Small Propagation Delay,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 813–823, December 1997.
- [14] J. G. Dai and B. Prabhakar, “The Throughput of Data Switches with and without Speedup,” in *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000, pp. 556–564.
- [15] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, “On the Stability of Input-Queued Switches with Speed-Up,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 104–118, February 2001.
- [16] L. Tassiulas, “Linear Complexity Algorithms for Maximum Throughput in Radio Networks and Input Queued Switches,” in *Proceedings of IEEE INFOCOM*, vol. 2, New York, March-April 1998, pp. 533–539.
- [17] E. Modiano, D. Shah, and G. Zussman, “Maximizing Throughput in Wireless Networks via Gossiping,” in *Proc. ACM SIGMETRICS*, June 2006, pp. 27–38.
- [18] A. Israeli and A. Itai, “A Fast and Simple Randomized Parallel Algorithm for Maximal Matching,” *Information Processing Letters*, vol. 22, no. 2, pp. 77–80, January 1986.
- [19] J. G. Dai, “On Positive Harris Recurrence of Multiclass Queueing Networks: A Unified Approach via Fluid Limit Models,” *Annals of Applied Probability*, vol. 5, no. 1, pp. 49–77, February 1995.
- [20] A. N. Rybko and A. L. Stolyar, “Ergodicity of Stochastic Processes Describing the Operation of Open Queueing Networks,” *Problems of Information Transmission*, vol. 28, pp. 199–220, 1992, translated from *Problemy Peredachi Informatsii*, vol. 28, no. 3, pp. 3–26, 1992.
- [21] B. Hajek and G. Sasaki, “Link Scheduling in Polynomial Time,” *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, September 1988.
- [22] S. Sarkar and L. Tassiulas, “End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks,” in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 564–569.
- [23] Y. Yi and S. Shakkottai, “Hop-by-hop Congestion Control over a Wireless Multi-hop Network,” in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004, pp. 2548–2558.
- [24] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, New Jersey: Prentice-Hall, 1982.
- [25] A. Dimakis and J. Walrand, “Sufficient Conditions for Stability of Longest Queue First Scheduling: Second Order Properties Using Fluid Limits,” *Advances in Applied Probability*, vol. 38, no. 2, pp. 505–521, June 2006.
- [26] A. Brzezinski, G. Zussman, and E. Modiano, “Enabling Distributed Throughput Maximization in Wireless Mesh Networks - A Partitioning Approach,” in *ACM Mobicom*, Los Angeles, CA, September 2006, pp. 26–37.
- [27] D. E. Drake and S. Hougardy, “A Linear-Time Approximation Algorithm for Weighted Matchings in Graph,” *ACM Transactions on Algorithms*, vol. 1, no. 1, pp. 107–122, July 2005.
- [28] G. Sharma, R. R. Mazumdar, and N. B. Shroff, “On the Complexity of Scheduling in Wireless Networks,” in *ACM Mobicom*, Los Angeles, CA, September 2006, pp. 227–238.
- [29] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M. V. Marathe, and S. Thite, “The Distance-2 Matching Problem and its Relationship to the MAC-layer Capacity of Ad hoc Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1069–1079, August 2004.
- [30] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2000.
- [31] A. Eryilmaz, R. Srikant, and J. R. Perkins, “Stable Scheduling Policies for Broadcast Channels,” in *ISIT 2002*, Lausanne, Switzerland, June-July 2002, p. 382.
- [32] R. Durrett, *Probability: Theory and Examples*, 3rd ed. Brooks/Cole, 2005.
- [33] Y. S. Chow, “On a Strong Law of Large Numbers for Martingales,” *The Annals of Mathematical Statistics*, vol. 38, no. 2, p. 610, April 1967.
- [34] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: Springer, 2000.
- [35] A. Gupta, X. Lin, and R. Srikant, “Low-Complexity Distributed Scheduling Algorithms for Wireless Networks,” in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 1631–1639.
- [36] C. Joo and N. B. Shroff, “Performance of Random Access Scheduling Schemes in Multi-hop Wireless Networks,” in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 19–27.
- [37] P. R. Kumar and T. I. Seidman, “Dynamic Instabilities and Stabilization Methods in Distributed Real-Time Scheduling of Manufacturing Systems,” *IEEE Transactions on Automatic Control*, vol. 35, no. 3, pp. 289–298, March 1990.
- [38] S. H. Lu and P. R. Kumar, “Distributed Scheduling Based on Due Dates and Buffer Priorities,” *IEEE Transactions on Automatic Control*, vol. 36, no. 12, pp. 1406–1416, December 1991.
- [39] C. Humes, “A regulator stabilization technique: Kumar-Seidman revisited,” *IEEE Transactions on Automatic Control*, vol. 39, no. 1, pp. 191–196, January 1994.
- [40] X. Wu and R. Srikant, “Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks with Node-Exclusive Spectrum Sharing,” in *Proceedings of IEEE CDC*, Seville, Spain, December 2005, pp. 5342–5347.
- [41] X. Lin and N. B. Shroff, “The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Multihop Wireless Networks,” *Technical Report*, Purdue University, <http://min.ecn.purdue.edu/~linx/papers.html>, 2004.
- [42] S. Ray and S. Sarkar, “Arbitrary Throughput Versus Complexity Trade-offs in Wireless Networks using Graph Partitioning,” in *Proceedings of Information Theory and Applications Second Workshop*, University of California, San Diego, January 2007.
- [43] K. Jung and D. Shah, “Low Delay Scheduling in Wireless Network,” in *Proceedings of IEEE ISIT*, Nice, France, June 2007, pp. 1396 – 1400.
- [44] S. Sanghavi, L. Bui, and R. Srikant, “Distributed Link Scheduling with Constant Overhead,” in *ACM SIGMETRICS*, San Diego, CA, June 2007, pp. 313–324.



**Xiaojun Lin** (S'02 / M'05) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, Indiana, in 2000 and 2005, respectively. He is currently an Assistant Professor of Electrical and Computer Engineering at Purdue University.

Dr. Lin's research interests are resource allocation, optimization, network pricing, routing, congestion control, network as a large system, cross-layer design in wireless networks, mobile ad hoc and sensor networks. He received the IEEE INFOCOM 2008 best paper award and 2005 best paper of the year award from *Journal of Communications and Networks*. His paper was also one of two runner-up papers for the best-paper award at IEEE INFOCOM 2005. He received the NSF CAREER award in 2007.



**Shahzada B. Rasool** (S'04) received the B.S. (with honors) degree from University of Engineering and Technology, Lahore, Pakistan and M.S. degree from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2000 and 2005, respectively. He is currently working towards his Ph.D. at School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. His research interests include analysis of medium access control and scheduling algorithms for wireless networks, and signal design and processing for sensors.