Low-Complexity Distributed Scheduling Algorithms for Wireless Networks

Abhinav Gupta, *Member, IEEE*, Xiaojun Lin, *Member, IEEE*, and R. Srikant, *Fellow, IEEE*

***Abstract*— We consider the problem of designing distributed scheduling algorithms for wireless networks. We present two algorithms both of which achieve throughput arbitrarily close to that of maximal schedules, but whose complexity is low due to the fact that they do not necessarily attempt to find maximal schedules. The first algorithm requires each link to collect local queue-length information in its neighborhood, and its complexity is otherwise independent of the size and topology of the network. The second algorithm, presented for the node-exclusive interference model, does not require nodes to collect queue-length information even in their local neighborhoods, and its complexity depends only on the maximum node degree in the network.**

***Index Terms*— Wireless Scheduling Algorithms, Low-Complexity and Distributed Algorithms, Provable Efficiency Ratios, Maximal Scheduling.**

I. Introduction

In this paper, we present distributed algorithms for link scheduling in wireless networks. Since interfering links in a wireless network cannot transmit at the same time, a scheduling policy is required to resolve the contention between various links attempting transmission. The well-known max-weight and back-pressure scheduling algorithms introduced in [2], [3] are throughput-optimal, i.e., they can stabilize the system under the largest set of offered load vectors. However, they are centralized algorithms and have high computational complexity. Using the max-weight or back-pressure algorithms for scheduling, a number of recent papers have studied the problem of joint congestion control, routing, and scheduling in multihop wireless networks [4]–[11]; see [12] for a survey. The focus of this paper is on designing distributed scheduling algorithms with low complexity and low implementation overhead. We consider two simple collision models in this paper: one where each link is associated with an interference set such that the link cannot be scheduled if any other link in its interference set is scheduled. The other model, called the node-exclusive interference model, is a special case of the first model where the interference set of a link consists of all links that share a common node with the first link. The first model covers a wide range of collision models that arise in practical wireless networks while the second model is applicable to Bluetooth or FH-CDMA type networks [13], [14].

Manuscript received July 10, 2008; accepted January 29, 2009. Recommended by Associate Editor Saswati Sarkar. This work has been partially supported by NSF grants CNS-0626703, CNS-0721286, CNS-0721484, CNS-0519691 and a grant from Purdue Research Foundation. An earlier version of this paper has appeared in the Proceedings of IEEE INFOCOM, 2007 [1].

Abhinav Gupta was with Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL 61801, when the research in this work was carried out. He is now with Qualcomm (email: abhinavg@qualcomm.com).

Xiaojun Lin is with Center for Wireless Systems and Applications (CWSA) and School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906 (email: linx@ecn.purdue.edu).

R. Srikant is with Coordinated Science Laboratory and Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL 61801 (email: rsrikant@uiuc.edu).

The study of low-complexity scheduling algorithms has its roots in the high-speed switching literature where maximal matching has been studied as an alternative to the max-weight algorithm. Upper bounds on the throughput loss due to the use of maximal matching have been derived in [15], [16]. Recently, these ideas have been successfully applied to wireless networks in [6], [17]–[19]. These papers show that low-complexity maximal-matching-type algorithms achieve a provably lower-bounded fraction of the maximum possible throughput, where the lower bound is a function of the local topology of the network. In particular, it was shown in [6], [17] that the lower bound is $1/2$ for the node-exclusive interference model while [18], [19] show that the lower bound is the inverse of the maximum number of links that can be simultaneously scheduled in an interference set.

The main drawback of the algorithms in [6], [17]–[19] is that they focus primarily on computational complexity but do not consider distributed implementation. For example, in the node-exclusive interference model, each valid schedule is a matching. (A matching in a graph is a set of edges such that no two edges share a common node). A maximal matching can be found as follows: each node requests a connection to one of its neighbors. A connection is accepted if the node receiving the request is not already part of the matching; otherwise, the node requests again. However, such a process, if not implemented in a structured fashion, would require many rounds of requests and incur a huge overhead, negating the benefits of the simplicity of maximal matching. We note that this problem is unique to wireless networks. In contrast, in many high-speed switches, a matching can be implemented by a central controller. Even if a central controller is not available, input and output ports are just one hop from each other and thus message passing is relatively easy in high-speed switches.

In view of the discussion above, the goal of this paper is to devise low-complexity, low-overhead distributed algorithms for multi-hop wireless networks. We will present two distributed algorithms which we summarize below:

(a) The first algorithm, which we call Q-SCHED, uses queue-length information in a local neighborhood of each link to perform scheduling. Q-SCHED is a randomized algorithm which works in two phases: in the first phase, each link tosses a coin to determine if it will participate in the schedule. In the second phase, the links that decide to participate use a one-step collision resolution protocol to determine if they will be part of the schedule or not. Such a two-phase algorithm was originally proposed in [20] and later generalized in [21], but the key contribution in this paper is to modify the algorithm to achieve dramatically larger throughput and to extend the algorithm to the case with multi-path routing. We will show that the computational complexity of Q-SCHED is independent of the network size and throughput, although in order to obtain the queue-length information in a local neighborhood, Q-SCHED requires communication overhead that is a function of the maximum node-degree of the network.

(b) The second algorithm, which we call BP-SIM (short for bipartite simulation), is also a randomized algorithm but does not require queue-length information. BP-SIM is presented for the node-exclusive interference model, and is an adaptation of the algorithm in [22], [23] for the case of wireless networks. The algorithm proceeds by emulating a bipartite graph: each node randomly decides to be a *left* or a *right* node. Then connection requests are made from *left* to *right* nodes. The key distinction between wireline networks considered in [22], [23] and multi-hop wireless networks is that the connection requests collide in the wireless networks and a contention resolution protocol is required. We design such a protocol and show that the overall complexity of BP-SIM is a function only of the maximum node degree and not of the size of the networks.

One of the main reasons that Q-SCHED and BP-SIM require lower complexity than maximal scheduling is that they do not attempt to compute a maximal schedule. Under maximal scheduling, at each time slot, every backlogged link has the following property: either the link is scheduled or some other link in its local neighborhood is scheduled. Let A_l denote the event that such a property holds for a given link l. Consider a randomized algorithm that requires a constant number of iterations to achieve this property with probability $\mathbf{P}[A_l] \geq 1-\epsilon$ for a given link l. A straightforward way of approximating a maximal schedule is to make sure that, with probability $1-\epsilon$, this property holds for *all* links, i.e.,

$$\mathbf{P}[\cap_l A_l] \geq 1-\epsilon. \tag{1}$$

It turns out that, in order to satisfy such a requirement, we would need $\Omega(\log L)$ number of iterations where L is the total number of links in the network [22], [23]. As a result, the complexity of computing a maximal schedule increases with the size of the network. In contrast, in this paper we show that, in order to achieve a throughput guarantee similar to that of maximal scheduling, we do not need to achieve (1). In particular, for BP-SIM, we only ensure that

$$\mathbf{P}[A_l] \geq 1-\epsilon, \text{ for all links } l. \tag{2}$$

In Q-SCHED, we ensure that

$$\mathbf{P}[A_l] \geq 1-\epsilon \tag{3}$$

for any link l such that the sum of the queue-length in its neighborhood is (roughly speaking) the largest among all neighborhoods. Clearly, the requirements in (2) and (3) are much easier to satisfy than (1). Both Q-SCHED and BP-SIM exploit this insight to achieve throughput guarantees comparable to maximal scheduling, but with complexity that does not increase with the size of the network.

The assumption that we make in designing the above algorithms is that time is slotted and synchronized in the network. Synchronizing time slots in a large network used to be a difficult problem, but recent advances in clock synchronization algorithms have made it possible to synchronize clocks in large networks with very low complexity, see [24], [25]. In addition to scheduling, another important issue is power control which we do not address in this paper. We refer the readers to [11] for distributed implementation of power control in multi-hop wireless networks. In addition, the work in [26] also considers a low-complexity randomized algorithm (the Random Transmitter Selection algorithm) under a more general setting with multi-receiver diversity. However, the efficiency ratio of the Random Transmitter Selection algorithm tends to be lower than the algorithms developed in this paper (e.g., its efficiency ratio will decrease as the node-degree increases even under the node-exclusive interference model).

The rest of the paper is organized as follows. In Section II we present the network model that is used in the rest of the paper. In Section III we present the Q-SCHED algorithm, study its performance and discuss a simpler variant of the algorithm specifically for the node-exclusive interference model. In Section IV we present BP-SIM scheduling algortihm, study its performance and discuss simulation results. Then, in Section V, we extend the results to multihop networks and to the case when each source-destination pair can have multiple paths through the network. We then conclude in Section VI.

II. Model

We consider a wireless network of $\mathcal{N}$ nodes. Let $G(V,E)$ be the directed connectivity graph of the network where V is the set of nodes and E is the set of links. For each $v \in V$, another node $v' \in V$ is a neighbor of v if they are end points of a link. Let $N(v)$ be the set of neighbors of v. The degree of node $v, d(v)$, is defined as the number of neighbors of v, i.e., $d(v) = |N(v)|$, where $|\mathcal{K}|$ refers to the cardinality of the set $\mathcal{K}$.

For each link $l \in E$, let $b(l)$ and $e(l)$ denote the transmitter node and receiver node, respectively. Two links are neighbors if they share a common node. Every link $l \in E$ interferes with a set of other links. Let $\mathcal{E}_l$ be the interference set of l. We adopt the convention that $l \in \mathcal{E}_l$, i.e.,

$$\mathcal{E}_l = \{l\} \cup \{l' : l' \in E \text{ and } l' \text{ interferes with } l\}.$$

We assume that the interference relationship is symmetric, i.e., if $k \in \mathcal{E}_l$ then $l \in \mathcal{E}_k$. This interference set varies with different communication models [6], [18], [19], [27]. In the node-exclusive interference model, also known as the one-hop interference model, $\mathcal{E}_l$ is the set of one-hop neighbors of l, including l. A valid schedule in this model is a matching. This model has been studied in [6].

We assume that time is divided into slots of equal length. Associated with each link l is a stochastic arrival process

$\{A_l(n)\}$, where $A_l(n)$ is the number of packet arrivals to link l in the slot n. We assume that each packet is of unit length. Let $\lambda_l = E[A_l(n)]$. For simplicity we assume that the arrival process is *i.i.d.* across time, i.e., $A(n) := \{A_1(n), A_2(n), ...A_{|E|}(n)\}$ is *i.i.d.* across n, although the results of this paper can also be extended to more general arrival processes. It is further assumed that the arrival process has bounded second moments, i.e., $\text{Cov}\big(A_l(n), A_k(n)\big) < \infty$ for any two links l and k. Let $D_l(n)$ denote the number of packets that link l can serve in the time slot n. The capacity of each link is the number of packets that the link can serve in one time slot and is denoted by c_l. Let $d_l(n)$ be the indicator function that indicates whether link l is scheduled or not. Then, $D_l(n) = c_l d_l(n)$. Also we define $a_l(n) := A_l(n)/c_l$. The system state is defined as

$$Q(n) := \big(q_1(n), q_2(n), ..., q_{|E|}(n)\big)$$

where $q_l(n)$ is the number of packets queued at link l at time n, and the dynamics are given by

$$q_l(n+1) = \big[q_l(n) + A_l(n) - D_l(n)\big]^+$$

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

Let $\vec{\lambda} = [\lambda_1, ..., \lambda_{|E|}]$. We define the capacity region under a given scheduling policy as the set of offered load vector $\vec{\lambda}$ under which the system can remain stable. Let Λ denote the largest capacity region under all scheduling policies [3]. A scheduling policy is said to guarantee an efficiency ratio of γ if it can stablize the system at any offered load in $\gamma\Lambda$.

III. Algorithm 1: Q-SCHED

For Q-SCHED, we assume that at the beginning of each time-slot, every link l knows the queue-lengths of all links k in its interference set $\mathcal{E}_l$ and also the queue-lengths of all links in the interference sets of $k \in \mathcal{E}_l$. A slight variation of this algorithm for the node-exclusive model will also be discussed where the queue-length information of only the immediate interferers is required. We now present the algorithm.

A. Scheduling Policy

Each time slot is divided into two parts: a scheduling slot and a data transmission slot. The links that are to be scheduled are chosen in the scheduling slot and the chosen links transmit their packets in the data transmission slot. The scheduling slot is further divided into M mini-slots. For ease of exposition, in what follows, we will drop the index n from the notation $q_j(n)$ when there is no confusion. The algorithm proceeds as follows: at the beginning of time-slot n, each link l first computes

$$P_l = \alpha \frac{\frac{q_l}{c_l}}{\max_{i \in \mathcal{E}_l} [\sum_{k \in \mathcal{E}_i} \frac{q_k}{c_k}]} , \tag{4}$$

where $\alpha = \log(M)$. Each link then picks a backoff time from $\{1, 2, ..., M+1\}$ where picking $M+1$ implies that the link will not attempt to transmit in this time slot. The backoff time (Y) is chosen as follows:

$$\begin{aligned} &\Pr\{Y = M+1\} = e^{-P_l}, \\ &\Pr\{Y = m\} = e^{-P_l \frac{m-1}{M}} - e^{-P_l \frac{m}{M}}, m = 1, 2, ..., M. \end{aligned} \tag{5}$$

When the backoff timer for a link expires, it begins transmission unless it has already heard a transmission from one of its interfering links. If two or more links that interfere begin transmissions simultaneously, there is a collision and none of the transmissions is successful. Further, any link that hears the collision will not attempt transmission in the rest of their time-slot. (Note that here we have assumed that each link can always overhear its interfering links. In practice, we can precede each link's transmission by an RTS/CTS exchange immediately before the transmission (for more details, see Policy Q in Section IV of [20]). Then, it is sufficient if either end-points of the link can overhear the RTS or CTS transmissions from interfering links. Note that such an RTS/CTS procedure also addresses the potential hidden-terminal issues.)

The Q-SCHED algorithm can be thought of as a two-phase algorithm. In the first phase, each link l first decides whether or not it would participate in the schedule for that time slot. In our algorithm, this phase corresponds to choosing $\{1, 2, ..., M\}$ or $(M+1)$ respectively. In the next phase, each participating link chooses a number between 1 and M and attempts to transmit starting from that mini-slot. This backoff procedure serves to reduce collision, and thus should lead to a higher capacity compared with a policy without backoff, e.g., [28]. While data transmission may start at any mini-slot, the length of each packet (plus the corresponding acknowledgment packet if required) is assumed to be smaller than the data transmission slot so that a transmission ends within the time-slot. The above idea of using two phases was first introduced in [20], [21], and is essential to achieve high efficiency ratios with a constant number of backoff mini-slots. Here the probabilities in (5) have been modified to achieve a higher guaranteed throughput. Further, we complete the proof in [20], [21] by establishing stochastic stability.

B. Analysis

We now proceed to analyze this scheduling policy. Define the Lyapunov function

$$V(n) = \max_{i \in E} \sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}. \tag{6}$$

***Lemma** 1:* The Q-SCHED scheduling policy guarantees that for any $\epsilon > 0$ and constants $C_1, C_2 > 0$, there exists a constant R such that if $V(n) \geq R$, then for any $\eta \in [0, 1]$ and for any link k such that

$$\sum_{l \in \mathcal{E}_k} \frac{q_l}{c_l} \geq \eta(V(n) - C_1 - C_2\epsilon), \tag{7}$$

the following holds,

$$\sum_{l \in \mathcal{E}_k} \Pr\{\text{Link } l \text{ is scheduled}\} \geq \eta \left(1 - \frac{\log(M) + 1}{M} - \epsilon\right).$$

Proof: See Appendix I. ■

We present the following proof for the special case of bounded arrivals, i.e., we assume that there exists a constant θ such that $A_l(n) \leq \theta$ for all l and n. The proof can be extended to cover more general arrival processes by upper-bounding the number of arrivals in a time-slot with high probability.

***Lemma** 2:* Q-SCHED scheduling policy guarantees that for any $\delta > 0$, there exists a positive integer constant H and a positive constant B, such that if $V(n) \geq B$ and

$$\sum_{l \in \mathcal{E}_k} \frac{\lambda_l}{c_l} < 1 - \frac{\log(M)+1}{M} - 4\epsilon, \text{ for all links } k \in E,$$

then in the time-slot $n + H$, the following holds,

$$\Pr\{\sum_{l \in \mathcal{E}_k} \frac{q_l(n+H)}{c_l} \leq V(n) - H\epsilon\} \geq 1 - \delta$$

for all links $k \in E$.

Proof: See Appendix II. ■

We now prove the stability of Q-SCHED.

***Theorem** 1:* Consider the Markov chain $\{Q(n)\}$. Under Q-SCHED scheduling algorithm this Markov chain is positive recurrent if for some $\epsilon > 0$

$$\sum_{l \in \mathcal{E}_k} \frac{\lambda_l}{c_l} < 1 - \frac{\log(M)+1}{M} - 4\epsilon \text{ for all links } k \in E.$$

Proof: Note that from Lemma 2 we can infer that for any $\delta > 0$ there exists a constant B and a positive integer constant H such that if $V(n) \geq B$, then

$$\Pr\{V(n+H) - V(n) \leq -H\epsilon | Q(n)\} \geq 1 - L\delta,$$

where L is the total number of links in the network.

Since the arrivals and departures are both upper-bounded, there exists a constant C such that

$$\left| \sum_{l \in \mathcal{E}_k} \frac{q_l(n+1)}{c_l} - \sum_{l \in \mathcal{E}_k} \frac{q_l(n)}{c_l} \right| \leq C \tag{8}$$

for all time-slot n and link k. This implies that $V(n+H) - V(n) \leq HC$. Denoting $E_X[\cdot] = E[\cdot|X]$, we have

$$\begin{aligned} E_{Q(n)}\big[V(n+H) - V(n)\big] &\leq -H\epsilon(1 - L\delta) + HCL\delta \\ &= H\big((C+\epsilon)L\delta - \epsilon\big). \end{aligned}$$

Thus, if $\delta \leq \frac{\epsilon}{2(C+\epsilon)L}$ we get

$$E_{Q(n)}\big[V(n+H) - V(n)\big] \leq -\frac{H\epsilon}{2} < 0$$

whenever $V(n) > B$. Since the set $\mathcal{B} = \{Q(n) : V(n) \leq B\}$ is bounded, by Foster's theorem [29] we have proved that the Markov chain $\{Q(n)\}$ is positive recurrent. ■

C. The Efficiency Ratio and Overhead of Q-SCHED

Here we adopt the definition of the *interference degree* in [18], [19], [27]. The interference degree of a link l is the maximum number of links within its interference range that can be activated simultaneously without interfering with each other. The interference degree $\mathcal{K}$ of a network is the maximum interference degree over all links. A nice property of this notion is that for a number of common interference models, the value of interference degree can be bounded independently from the network topology. For example, for the node-exclusive interference model, the interference degree is bounded by 2. For the so-called bi-directional equal-power model, which resembles the way IEEE 802.11 DCF (Distributed Coordination Function) operates, the interference degree is bounded by 8 [27].

Consider a network whose interference degree is $\mathcal{K}$. It implies that in the interference range of any given link k, at most $\mathcal{K}$ links can be scheduled at the same time. Therefore, the offered load that the network can possibly support must satisfy

$$\sum_{l \in \mathcal{E}_k} \frac{\lambda_l}{c_l} < \mathcal{K} \text{ for all links } k.$$

According to Theorem 1, Q-SCHED can stabilize the network for any offered load such that, for some $\epsilon > 0$,

$$\sum_{l \in \mathcal{E}_k} \frac{\lambda_l}{c_l} < 1 - \frac{\log(M)+1}{M} - 4\epsilon \text{ for all links } k.$$

Since the parameter ϵ can be arbitrarily small, this implies that the guaranteed efficiency ratio of Q-SCHED can be arbitrarily close to $\frac{1}{\mathcal{K}}\left(1 - \frac{\log(M)+1}{M}\right)$, where M again is the number of backoff mini-slots. As $M \to \infty$ and $\epsilon \to 0$, this guaranteed efficiency ratio approaches $1/\mathcal{K}$. Recall that maximal scheduling can guarantee an efficiency ratio of $1/\mathcal{K}$ [18], [19]. Hence, the guaranteed efficiency ratio of Q-SCHED can be arbitrarily close to that of maximal scheduling.

We next comment on the complexity and overhead of Q-SCHED. Once each link obtains the queue-length information from its neighboring links, Q-SCHED only takes M mini-slots to compute a schedule. Note that in practice the length of a mini-slot cannot be arbitrarily small. As a result, the value M corresponds to the amount of time required for computing a schedule. Thus, the computation time of Q-SCHED is independent of the size and the topology of the network. However, it does incur additional communication overhead for Q-SCHED to exchange the queue-length information. This communication overhead increases quadratically with the the number of links in the two-hop neighborhood, which is bounded when the maximum node-degree of the network is bounded. Hence, given a small positive number ϵ and a maximum node-degree, Q-SCHED can guarantee an efficiency ratio of $1/\mathcal{K} - \epsilon$ with both complexity and overhead independent of the size of the network. In contrast, while maximal-scheduling guarantees an efficiency ratio of $1/\mathcal{K}$, its complexity increases logarithmically with the size of the network [23].

D. A Special Case

The scheduling algorithm discussed above is valid for any interference model including the node-exclusive interference model, as long as the interference relationship is symmetric. However, in the special case of node-exclusive interference model, an even simpler variant of this algorithm can be used. The only difference is in the calculation of the term P_l : each link l computes

$$P_l = \alpha \frac{\frac{q_l}{c_l}}{\max[\sum_{k \in \mathcal{F}_{b(l)}} \frac{q_k}{c_k}, \sum_{k \in \mathcal{F}_{e(l)}} \frac{q_k}{c_k}]}$$

where $\mathcal{F}_j$ denotes the set of links incident on j, and $b(l)$ and $e(l)$ denote the transmitter node and the receiver node, respectively, of link l. In this case we choose $\alpha = \log(2M)/2$. Thus, each link requires the knowledge of the queue-lengths of only those links that interfere with it. We can prove that in this case if

$$\sum_{l \in \mathcal{F}_i} \frac{\lambda_l}{c_l} < \frac{1}{2} - \frac{\log(2M)}{2M} \text{ for all nodes } i \in V, \quad (9)$$

then $Q(n)$ is positive recurrent and the system is queue-length stable. We do not provide the proof since it is very similar to the proof in Section III-B.

Remark: Q-SCHED policy is considerably similar to the Policy V (for the node-exclusive interference model) and Policy W (for the two-hop interference model) obtained in [30]. Here we comment briefly on the main differences in the results. First, under the two-hop interference model discussed in [30], the performance guarantee of Q-SCHED proved in our paper is often higher than that of Policy W in [30]. Specifically, in this paper we prove that Q-SCHED can guarantee an efficiency ratio close to $1/\mathcal{K}$, where the interference degree $\mathcal{K}$ is defined as the maximum number of links in an interference range that can be activated simultaneously. In [30, Proposition 3], it is proved that Policy W can guarantee an efficiency ratio close to $1/\hat{n}$, where $\hat{n}$ is the maximum number of links in a one-hop neighborhood. Although $\mathcal{K}$ can be equal to $\hat{n}$ in the worst case, it is often much smaller than $\hat{n}$ in practical topologies. For example, under the bi-directional equal-power model in [27], $\mathcal{K}$ is bounded by 8 while $\hat{n}$ can be unbounded. Hence, Q-SCHED is able to provide a higher performance guarantee than Policy W. Second, in this paper, our proof of stochastic stability is different from the fluid-limit approach in [30]. Additionally, in the next section, we also provide an alternative algorithm which does not require nodes to collect any queue-length information from its neighbors. Such an algorithm allows further trade-off between the scheduling efficiency and the signaling overhead required to collect queue length information.

IV. Algorithm 2: BP-SIM

While Q-SCHED can compute a schedule in a constant number of mini-slots, it requires knowledge of queue-length information. This may or may not be difficult to obtain. At moderate loads, queue-lengths will not be very large; thus, queue-lengths can be transmitted using a small number of bits along with the data packets. Even if the queue-lengths are large, changes in queue-lengths can be transmitted using a small number of bits if the arrival process is bounded. However, in practice such queue-length information exchange has to be performed asynchronously and thus may reduce the performance of the algorithm. In this section, we present an alternative algorithm, named BP-SIM (short for bipartite simulation), that requires no queue-length information. As in the case of Q-SCHED, BP-SIM does not attempt to find a maximal schedule. Instead, BP-SIM ensures that, for any backlogged link l, the probability that a link in $\mathcal{E}_l$ is scheduled is high. This stems from the key observation that the proof in [18] for the stability of maximal schedules can be adapted easily even when the schedule is not maximal. We will show with both analytical and numerical results that the complexity of BP-SIM is low for networks with even a few hundred nodes.

BP-SIM is presented for the node-exclusive interference model. To be more specific, we refer to the following FH-CDMA system that adheres to the node-exclusive interference model. Each node is assigned a unique orthogonal fast-hopping sequence. At each time, a node can be either in a *sending* mode, or in a *receiving* mode. If it is in a receiving mode, its frequency hops according to its own hopping sequence. If it is in a sending mode, it uses the hopping sequence of the intended receiver in order to send information to the receiving node. Clearly, if two or more sending nodes want to send information to the same receiver at the same time, their messages will collide. Hence, such a system can be modeled well using the node-exclusive interference model. Finally, it is assumed that the maximum degree of any node in the network is upper-bounded by d^*, i.e., $d(v) \leq d^*$ for all nodes v.

A. Scheduling Policy

As in Q-SCHED, each time slot is divided into two parts: a scheduling slot and a data transmission slot. The links that are to be scheduled are chosen in the scheduling slot and the chosen links transmit their packets in the data transmission slot. The scheduling slot is further divided into K rounds, where K is a constant to be chosen later. Each round contains $2M$ mini-slots. Initially none of the links are in the schedule. In each round i a matching $\mathcal{M}_i$ is formed by adding links from the graph G to the matching $\mathcal{M}_{i-1}$ (obtained from the previous round). The matching $\mathcal{M}_K$ is the final schedule.

We define a few terms used in describing the algorithm. Each node v maintains and updates a list of *potentially available* neighboring nodes as follows. At the beginning, this list is set to contain all neighboring nodes of node v. Then, in each round if node v discovers that a neighboring node u has been matched (using a procedure to be described below), then node u will be removed from the list. (Note that even if a neighboring node u is matched, it may still be in node v's list of *potentially available* neighbors if v does not know whether u is matched or not.)

We said that a link is backlogged if its queue backlog is greater than or equal to the capacity of the link. A neighboring node of node v is said to be a *backlogged neighbor* of v if the link from v to this neighboring node is backlogged.

We now describe the BP-SIM algorithm. The $2M$ mini-slots in each round are further divided into two subgroups: mini-slots 1 to M form the requesting group, and mini-slots $M+1$ to $2M$ form the responding group. In each round, BP-SIM emulates a bipartite graph by first dividing the nodes randomly into *left* nodes and *right* nodes. Specifically, the round proceeds as follows. For each node v, if it is not matched yet, and it has at least one backlogged and potentially available neighbor, such a node v becomes either a *left* node or a *right* node with probability $1/2$ each, independently of other nodes. Otherwise, (i.e., if the node v is already matched, or if it has no backlogged neighbors, or if none of the backlogged neighbors are potentially available), such a node v becomes *right*.

For any node, say v_l, that becomes *left*, it chooses a partner, say v_{lr}, uniformly and randomly from the set of backlogged and potentially available neighbors of v_l. Node v_l then randomly and independently chooses a mini-slot between 1 and M uniformly, and sends a scheduling request to v_{lr} in that mini-slot. This scheduling request contains the ID of node v_l.

For any node, say v_r, that becomes *right*, it switches to a receiving mode in mini-slots 1 to M and listens to the scheduling requests from its neighbors. Note that some of the scheduling requests that node v_r receives may be transmitted at the same mini-slot, in which case they will collide and node v_r will not be able to decode the IDs contained in the requests. Otherwise, node v_r will be able to decode the IDs from non-colliding requests.

After M mini-slots of the requesting group end, the *left* nodes switch to a receiving mode. The *right* nodes then send responding messages to the *left* nodes as follows. There are three cases.

Case 1: if node v_r has already been matched, then for any mini-slot when it receives a non-colliding request, it will choose a mini-slot in the responding group to reply to the requester with a message that node v_r is a matched node. The requester will then remove node v_r from its list of *potentially available* neighbors. Note that since each left node requested at most one right node, such a reply message will always succeed without collision. Further, since a right node needs to reply to at most M senders, the right node can reply to one request at a time, and complete within M mini-slots. (Such a free-of-collision property also holds for the acknowledgment packets in the next two cases.)

Case 2: If node v_r has not been matched yet, and if the first request that it received does not collide, it will then use the first mini-slot of the responding group to acknowledge the request from the node that first requested it. Node v_r and the node that first requested it are then matched with each other. In subsequent mini-slots, node v_r will behave as a matched node, using the procedure in case 1 to reply to other non-colliding requests with a message that node v_r has been matched.

Case 3: if node v_r has not been matched yet, and if at the first mini-slot when v_r received a request, there was a collision due to simultaneous requests from more than one neighbors, then v_r does not acknowledge or reply to any requests in that round.

We note that only *right* nodes acknowledge requests. It is possible that a *left* node sends a request to a *left* node, in which case the request will never be acknowledged in any manner.

If v_r acknowledges the request of v_l then v_r and v_l are both matched. At the end of the round, the matching contains all ordered-pairs of nodes that have been matched so far.

The above process is repeated in every round. After the K rounds, the links that correspond to matched pairs of nodes begin transmission in the data transmission slot.

This algorithm is an extension of the maximal matching algorithms discussed in [22] and [23]. There are two novel features of the proposed BP-SIM algorithm. Firstly, BP-SIM uses a contention resolution protocol that is necessary in a wireless network to reduce the chance of collisions between the connection requests. Secondly, BP-SIM uses a discovery procedure to keep track of neighboring nodes that are already matched. We note that it is important to keep track of such information, because otherwise a left node may repeatedly request a matched node. The discovery procedure described above carefully resolves this issue. Finally, we also note that the algorithms in [22] and [23] are designed to compute maximal matchings. Thus, even if we ignore the collisions due to simultaneous requests to a node, they take $O(\log n)$ time to compute a schedule. In contrast, in BP-SIM algorithm the scheduling time depends only on d^* and not on n as we will see later.

B. Analysis

We now analyze this scheduling policy.

***Lemma** 3:* For any $\kappa \in (0,1)$ there exists a constant K that depends on d^*, κ, and M but is independent of network size, such that for each backlogged link l the probability that at least one backlogged link in $\mathcal{E}_l$ is scheduled after K rounds is greater than or equal to κ.

Proof: See Appendix III. ■

We next prove the stability of BP-SIM.

***Theorem** 2:* Consider the Markov chain $\{Q(n)\}$. For the node-exclusive interference model and under the scheduling policy BP-SIM, this Markov Chain is positive recurrent if

$$\sum_{k \in \mathcal{E}_l} \frac{\lambda_k}{c_k} < \kappa, \text{ for all links } l,$$

where $\kappa \in (0,1)$ and K are appropriately chosen according to Lemma 3.

Proof: Define the Lyapunov function

$$V(n) = \sum_l \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{q_k(n)}{c_k} \right). \tag{10}$$

This is the same Lyapunov function as the one used in [18]. Using the results in [18] we get

$$\begin{aligned} &V(n+1) - V(n) \\ &\quad = 2 \sum_l \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} (a_k(n) - d_k(n)) \right) \\ &\qquad + \sum_l (a_l(n) - d_l(n)) \left(\sum_{k \in \mathcal{E}_l} (a_k(n) - d_k(n)) \right), \end{aligned}$$

from which we get

$$\begin{aligned} &E[V(n+1) - V(n) | Q(n)] \\ &\leq 2 \sum_l \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - E\left[\sum_{k \in \mathcal{E}_l} d_k(n) \right] \right) + B \\ &\leq 2 \sum_{l: q_l \geq c_l} \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - E\left[\sum_{k \in \mathcal{E}_l} d_k(n) \right] \right) + B_1 \qquad (11) \\ &\leq 2 \sum_{l: q_l \geq c_l} \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - \kappa \right) + B_1 \\ &\leq -2\epsilon \sum_{l: q_l \geq c_l} \frac{q_l(n)}{c_l} + B_1, \end{aligned}$$

where $B, B_1 > 0$ are some constants and $\epsilon = \kappa - \max_l \sum_{k\in\mathcal{E}_l} \frac{\lambda_k}{c_k}$. Thus using [18, Th. 1], the system is queue-length stable if

$$\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} < \kappa, \qquad \text{for all links } l \in E.$$

■

C. Numerical Examples and Simulations

Recall that under the node-exclusive interference model [6], [17]–[19], maximal matching can be shown to achieve queue-length stability for any offered load $\vec{\lambda}$ such that

$$\sum_{k\in\mathcal{E}_l} \frac{\lambda_k}{c_k} < 1, \text{ for all links } l.$$

According to Theorem 2, by choosing κ close enough to 1, the throughput of BP-SIM can be arbitrarily close to that of maximal schedule. Further, note that any feasible offered load under the node-exclusive interference model must satisfy

$$\sum_{k\in\mathcal{E}_l} \frac{\lambda_k}{c_k} < 2, \text{ for all links } l.$$

Hence, by choosing κ close enough to 1, the efficiency ratio of BP-SIM can be made arbitrarily close to $1/2$. For any fixed $\kappa < 1$, the complexity of BP-SIM depends on the value of M and K, and is independent of the size of the network.

Although in the above results we are unable to derive an analytic expression for the minimum number of rounds K as a function of κ, we next present some numerical results on how large K needs to be for practical choice of κ. We first run numerical evaluations based on the analytical bounds derived in Appendix III. When $d^* = M = 5$, in order to guarantee $\kappa = 0.9$, we require $K \geq 29$. Therefore in $K \times 2M = 290$ mini-slots, BP-SIM guarantees that for any backlogged link l at least one link in its interference set is scheduled with a probability greater than or equal to 0.9. If $d^* = M = 10$ we require $K \geq 53$ to guarantee $\kappa = 0.9$.

The above analytical bounds are found to be quite conservative. We have also simulated the BP-SIM policy to analyze its actual performance. Simulations were performed on networks of four different sizes. The number of nodes (n) in the four cases were 30, 60, 120 and 225. The maximum node degrees (d^*) in the four cases were 5, 8, 8 and 17, respectively. The nodes were placed randomly on a rectangular area, i.e. their coordinates were chosen *i.i.d.* and uniformly. The radius of transmission of the nodes was chosen so as to make the graph connected.

In Figure 1 we plot the minimum success probability of a link versus number of rounds. Success probability of a link here refers to the probability that either the link or one of its interfering links is scheduled. Minimum success probability is the lowest success probability for all links in the network and among all random graphs simulated. The maximum backoff time in each case was $M = 4$.

In each case shown in Figure 1 we only needed $K = 11$ rounds (i.e., $K \times 2M = 88$ mini-slots) for the minimum success probability to be greater than 0.9. Hence, the simulation results show that in practice BP-SIM works even better than the minimum performance guarantees we have proved.

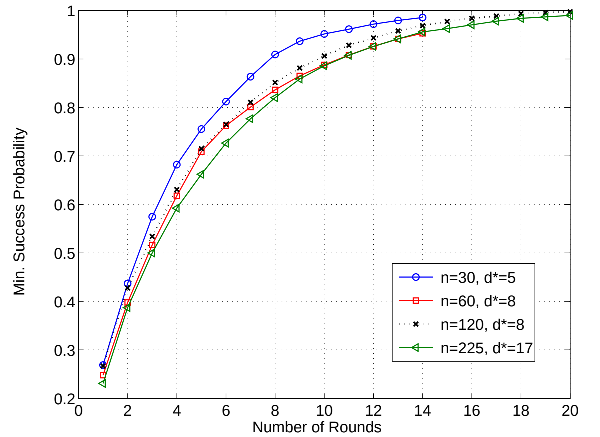

Fig. 1. Scheduling Efficiency of BP-SIM: Each curve shows, as the number of rounds increases, the minimum value (over all links) of the probability that either a given link or one of its interfering links is scheduled. Note that the curves for all four cases are close in the figure, thus showing the nice scalability property of the algorithm.

V. Joint Multipath Routing and Scheduling

The results in Section III and Section IV have focused on the case with single-hop flows. In other words, the arriving packets are directly offered to each link, and once they are served, they immediately leave the system. In this section, we extend the results to the case of multi-hop flows and with multipath routing. We will show how to design joint routing algorithms to work with the scheduling algorithms developed in the previous sections.

A. The Extended System Model with Multipath Routing

In addition to the system model in Section II, we now assume that there are $\mathcal{S}$ users in the system. Each user is associated with $J(s)$ alternate paths. Let $H_{sj}^l = 1$ if the path j of user s uses link l, and $H_{sj}^l = 0$, otherwise. Let $X_s(n)$ be the number of (unit-length) packets generated by user s at the beginning of time-slot n. For simplicity, we assume again that $[X_1(n), ..., X_{\mathcal{S}}(n)]$ is *i.i.d.* across time-slots. Let $x_s = E[X_s(n)]$, and $\vec{x} = [x_1, ..., x_{\mathcal{S}}]$. We can then define the capacity region Λ' in terms of $\vec{x}$ as the set of offered-load vectors $\vec{x}$ such that there exists $p_{sj}^* \geq 0$, for all $s = 1, ..., \mathcal{S}$, $j = 1, ..., J(s)$ such that $\sum_{j=1}^{J(s)} p_{sj}^* = 1$ for all s and

$$\left[\sum_{s=1}^{\mathcal{S}} \sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}^*\right] \in \Lambda,$$

where Λ is the capacity region defined in Section II in terms of the per-link rate $\vec{\lambda}$, and p_{sj}^* can be viewed as the long-term average fraction of traffic from user s that is routed to path j. In this section, we are interested in joint routing algorithms that can work with the scheduling algorithms in the previous sections to stabilize the system for a large fraction of the capacity region Λ'.

B. Joint Routing Algorithms Working with Q-SCHED

We first present a joint routing algorithm that can work with Q-SCHED to achieve a large fraction of the capacity region Λ'. Let β be a positive constant.

Joint Routing and Q-SCHED Algorithm:

- At each time slot n, each user s computes the routing probability $\vec{p}_s(n) = [p_{sj}(n)]$ as the solution to the following problem:

$$\min_{\vec{p}_s \geq 0} \quad \frac{B_s}{2}\sum_{j=1}^{J(s)}(p_{sj})^2 + \sum_{j=1}^{J(s)} p_{sj} \sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i \in \mathcal{E}_l} \left[\sum_{k \in \mathcal{E}_i} \frac{q_k(t)}{c_k}\right]^\beta$$

$$\text{subject to} \quad \sum_{j=1}^{J(s)} p_{sj} = 1, \tag{12}$$

  where B_s is a positive constant chosen for each user s. Each user s then routes every arriving packet at time-slot n to path j with probability $p_{sj}(n)$. Let $P_{sj}(n)$ denote the actual fraction of packets routed to path j at time-slot n. Note that $E[X_s(n)P_{sj}(n)] = x_s p_{sj}(n)$.
- At each time slot n, use Q-SCHED to compute the schedule to be used at this time slot. Let $D_l(n) = c_l$ if link l is scheduled by Q-SCHED at time slot n.

Remark: The main idea of (12) is to assign higher routing probabilities to those paths j that are less congested, i.e., to those paths j with smaller cost

$$\sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i \in \mathcal{E}_l} \left[\sum_{k \in \mathcal{E}_i} \frac{q_k(t)}{c_k}\right]^\beta . \tag{13}$$

The quadratic term $\frac{B_s}{2}\sum_{j=1}^{J(s)}(p_{sj})^2$ is to alleviate an otherwise "oscillation" problem for this type of multipath routing algorithms [31]. Note that if $B_s = 0$, then it is easy to see that the solution of (12) will have $p_{sj} > 0$ *only* for those paths j whose costs (given by (13)) are the smallest among all alternate paths of user s. Thus, even if the queue-length varies slightly, the routing probabilities could switch completely from one path to the other. On the other hand, when $B_s > 0$, we can show that the solution to (12) is a continuous function of the queue-length. Hence, the oscillation problem is eliminated. For convenience, we use a quadratic function here, while in fact any strictly-convex function of $\vec{p}_s$ should serve the same purpose.

According to this algorithm, the evolution of the queue-length is given by[1]

$$q_l(n+1) = \left[q_l(n) + \sum_{s=1}^{\mathcal{S}}\sum_{j=1}^{J(s)} H_{sj}^l X_s(n) P_{sj}(n) - D_l(n)\right]^+ . \tag{14}$$

[1]Note that here we have adopted the simplifying assumption that the packets routed to path j are offered to all links on the path instantaneously. This assumption simplifies the analysis. There are standard techniques in the literature [18], [32]–[35] that can extend our result to the case when the hop-by-hop packet-forwarding dynamics are taken into account. We refer the readers to these references for details.

We now state the main result of the section.

***Theorem** 3:* For any $\epsilon > 0$, there exists a positive number β_0 such that for all $\beta \geq \beta_0$, the above joint routing and Q-SCHED algorithm can stabilize all queues under all offered load vector $\vec{x}$ that satisfies the following condition: there exist $\vec{\lambda}^* = [\lambda_l^*]$ and $\vec{p}^* = [p_{sj}^*]$ such that

$$\lambda_l^* = \sum_{s=1}^{\mathcal{S}}\sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}^*, \text{ for all links } l,$$

$$\sum_{l \in \mathcal{E}_i} \frac{\lambda_l^*}{c_l} \leq \frac{1-\epsilon}{1+\epsilon}\left(1 - \frac{\log M + 1}{M} - \epsilon\right), \text{ for all links } i.$$

Remark: When the interference degree is $\mathcal{K}$, it is each to verify that any feasible offered load $\vec{x} \in \Lambda'$ must satisfy the following: there must exist $\vec{\lambda}^*$ and $\vec{p}^*$ with

$$\lambda_l^* = \sum_{s=1}^{\mathcal{S}}\sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}^*, \text{ for all links } l,$$

$$\sum_{l \in \mathcal{E}_i} \frac{\lambda_l^*}{c_l} \leq \mathcal{K}, \text{ for all links } i.$$

Therefore, Theorem 3 implies that the above algorithm can guarantee close to $1/\mathcal{K}$ of the capacity region Λ'.

In order to prove Theorem 3, we will need the following lemma.

***Lemma** 4:* For any $\epsilon > 0$, there exists $\beta_0 \geq 0$ such that for all $\beta \geq \beta_0$ and $Q(n) \neq 0$, the following holds,

$$\frac{\sum_{i=1}^{L}\left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}\right)^{(1+\beta)}}{\max_{i \in E}\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}} \geq (1-\epsilon)\sum_{i=1}^{L}\left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}\right)^\beta .$$

where $L = |E|$ is the total number of links in the system.

Proof: Let

$$a_i = \frac{\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}}{\max_{k \in E}\sum_{l \in \mathcal{E}_k} \frac{q_l(n)}{c_l}}.$$

Then $0 \leq a_i \leq 1$ for all links $i \in E$. Further, there exists a link i such that $a_i = 1$. Without loss of generality, we can assume that $a_1 = 1$. To prove the lemma, it suffices to show that for large β,

$$1 + \sum_{i \neq 1} a_i^{(1+\beta)} \geq (1-\epsilon)\left(1 + \sum_{i \neq 1} a_i^\beta\right),$$

for all $0 \leq a_i \leq 1$, $i = 2, 3, ..., L$.

Fix $\epsilon \in (0,1)$. Note that for any $\beta > 0$, we have,

$$a_i^{(1+\beta)} \geq a_i^\beta(1-\epsilon), \text{ if } a_i \geq 1-\epsilon,$$

and

$$a_i^{(1+\beta)} \geq 0, \text{ if } a_i \leq 1-\epsilon.$$

In both cases, we have

$$a_i^{(1+\beta)} \geq a_i^\beta(1-\epsilon) - (1-\epsilon)^{(1+\beta)}.$$

Hence,

$$1 + \sum_{i \neq 1} a_i^{(1+\beta)} \geq 1 + \sum_{i \neq 1} a_i^\beta(1-\epsilon) - L(1-\epsilon)^{(1+\beta)}.$$

Now, given $\epsilon > 0$, we can pick β_0 such that

$$L(1-\epsilon)^{(1+\beta_0)} \leq \epsilon,$$

Then, for any $\beta > \beta_0$, we have,

$$1 + \sum_{i \neq 1} a_i^{(1+\beta)} \geq \left(1 + \sum_{i \neq 1} a_i^{\beta}\right)(1-\epsilon).$$

The result of the lemma then follows. ∎

We can now prove Theorem 3.

Proof: (of Theorem 3) We will use the following Lyapunov function:

$$U(n) = \frac{1}{\beta+1} \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{(1+\beta)}.$$

The motivation for considering $U(n)$ is that $U(n)^{1/(1+\beta)}$ approximates $V(n) = \max_{i \in E} \sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}$ (see (6)) when β is large. Since $V(n)$ is a Lyapunov function for the Q-SCHED algorithm, we expect that $U(n)$ will also become a Lyapunov function when β is large. Let $\lambda_l(n) = \sum_{s=1}^{\mathcal{S}} \sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}(n)$, where $p_{sj}(n)$ is the routing probability chosen for time-slot n. To compute the drift of $U(n)$, recall that $E[X_s(n)P_{sj}(n)] = x_s p_{sj}(n)$. Hence,

$$\begin{aligned} & E[U(n+1) - U(n)|Q(n)] \\ \leq \; & \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta} \left[\sum_{l \in \mathcal{E}_i} \frac{\lambda_l(n)}{c_l} - \sum_{l \in \mathcal{E}_i} \Pr\{S_l\} \right] \\ & + g(Q(n)), \end{aligned}$$

where S_l is the event that link l is scheduled at time slot n (see Lemma 1 in Section III), and

$$g(Q(n)) = \sum_{i=1}^{L} o\left[\left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta} \right].$$

(We have use the notation $h(x) = o(f(x))$ to mean $\lim_{x \to +\infty} h(x)/f(x) = 0$.) According to Lemma 1, there exists a constant R such that if $V(n) \geq R$, then

$$\sum_{l \in \mathcal{E}_i} \Pr\{S_l\} \geq \frac{\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}}{V(n)} (1 - \frac{\log M + 1}{M} - \epsilon),$$

for all links i, where $V(n) = \max_{i \in E} \sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l}$. Hence, if we choose β_0 as in Lemma 4, then for all $\beta \geq \beta_0$, we have,

$$\begin{aligned} & \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta} \left[\sum_{l \in \mathcal{E}_i} \Pr\{S_l\} \right] \\ \geq \; & \frac{(1 - \frac{\log M+1}{M} - \epsilon)}{V(n)} \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{(\beta+1)} \\ \geq \; & (1-\epsilon)(1 - \frac{\log M + 1}{M} - \epsilon) \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta}. \end{aligned}$$

By the assumption,

$$\sum_{l \in \mathcal{E}_i} \frac{\lambda_l^*}{c_l} \leq \frac{1-\epsilon}{1+\epsilon} \left(1 - \frac{\log M + 1}{M} - \epsilon \right), \text{ for all links } i.$$

Hence,

$$\begin{aligned} & E[U(n+1) - U(n)|Q(n)] \\ \leq \; & \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta} \left[\sum_{l \in \mathcal{E}_i} \frac{\lambda_l(n)}{c_l} - (1+\epsilon) \sum_{l \in \mathcal{E}_i} \frac{\lambda_l^*}{c_l} \right] \\ & + g(Q(n)) \qquad (15) \\ = \; & \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta} \left[\sum_{l \in \mathcal{E}_i} \frac{\sum_{s=1}^{\mathcal{S}} \sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}(n)}{c_l} \right. \\ & \left. -(1+\epsilon) \sum_{l \in \mathcal{E}_i} \frac{\sum_{s=1}^{\mathcal{S}} \sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}^*}{c_l} \right] + g(Q(n)) \\ = \; & \sum_{s=1}^{\mathcal{S}} x_s \left[\sum_{j=1}^{J(s)} p_{sj}(n) \sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i \in \mathcal{E}_l} \left(\sum_{k \in \mathcal{E}_i} \frac{q_k(n)}{c_k} \right)^{\beta} \right. \\ & \left. -(1+\epsilon) \sum_{j=1}^{J(s)} p_{sj}^* \sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i \in \mathcal{E}_l} \left(\sum_{k \in \mathcal{E}_i} \frac{q_k(n)}{c_k} \right)^{\beta} \right] \\ & + g(Q(n)). \end{aligned}$$

According to the multi-path routing algorithm (12), we have for each user s,

$$\begin{aligned} & \sum_{j=1}^{J(s)} p_{sj}(n) \sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i \in \mathcal{E}_l} \left(\sum_{k \in \mathcal{E}_i} \frac{q_k(n)}{c_k} \right)^{\beta} \\ & - \sum_{j=1}^{J(s)} p_{sj}^* \sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i \in \mathcal{E}_l} \left(\sum_{k \in \mathcal{E}_i} \frac{q_k(n)}{c_k} \right)^{\beta} \\ \leq \; & -\frac{B_s}{2} \sum_{j=1}^{J(s)} (p_{sj}(n))^2 + \frac{B_s}{2} \sum_{j=1}^{J(s)} (p_{sj}^*)^2 \\ \leq \; & B_s J(s). \end{aligned}$$

Hence,

$$\begin{aligned} & E[U(n+1) - U(n)|Q(n)] \\ \leq \; & -\epsilon \sum_{i=1}^{L} \left(\sum_{l \in \mathcal{E}_i} \frac{q_l(n)}{c_l} \right)^{\beta} \left[\sum_{l \in \mathcal{E}_i} \frac{\lambda_l^*}{c_l} \right] \\ & + \sum_{s=1}^{\mathcal{S}} x_s B_s J(s) + g(Q(n)). \end{aligned}$$

This then provides the negative drift for stability. ∎

C. Joint Routing Algorithms Working with BP-SIM

A similar multipath routing algorithm can be design for working with BP-SIM. In fact, since the Lyapunov function for BP-SIM (see Equation (10) in the proof of Theorem 2) is of a quadratic form, we do not need large β as in Theorem 3.

Joint Routing and BP-SIM Algorithm:

- At each time slot n, each user s computes the routing fractions $\vec{p}_s(n) = [p_{sj}(n)]$ as the solution to the following problem:

$$\min_{\vec{p}_s \geq 0} \quad \frac{B_s}{2}\sum_{j=1}^{J(s)}(p_{sj})^2 + \sum_{j=1}^{J(s)} p_{sj} \sum_{l=1}^{L} \frac{H_{sj}^l}{c_l} \sum_{i\in\mathcal{E}_l}\left[\sum_{k\in\mathcal{E}_i}\frac{q_k(t)}{c_k}\right]$$

$$\text{subject to} \quad \sum_{j=1}^{J(s)} p_{sj} = 1,$$

where B_s is a positive constant chosen for each user s. Each user s then routes every arriving packet at time-slot n to path j with probability $p_{sj}(n)$. Let $P_{sj}(n)$ denote the actual fraction of packets routed to path j at time-slot n. Note again that $E[X_s(n)P_{sj}(n)] = x_s p_{sj}(n)$.

- At each time slot n, use BP-SIM to compute the schedule to be used at this time slot. Let $D_l(n) = c_l$ if link l is scheduled by BP-SIM at time slot n.

The evolution of the queue-length is again given by (14). We can show the following result, which is similar to Theorem 3.

***Theorem** 4:* The above joint routing and BP-SIM algorithm can stabilize all queues under all offered load vector $\vec{x}$ that satisfies the following condition: there exist $\vec{\lambda}^* = [\lambda_l^*]$ and $\vec{p}^* = [p_{sj}^*]$ such that

$$\lambda_l^* = \sum_{s=1}^{\mathcal{S}}\sum_{j=1}^{J(s)} H_{sj}^l x_s p_{sj}^*, \text{ for all links } l,$$

$$\sum_{l\in\mathcal{E}_i}\frac{\lambda_l^*}{c_l} \leq \kappa, \text{ for all links } i.$$

where $\kappa \in (0,1)$ and the number of rounds K (that BP-SIM executes within each time-slot) are appropriately chosen according to Lemma 3.

To prove Theorem 4, we can use the same Lyapunov function (10) as in the proof of Theorem 2, and proceed to Equation (11), which is comparable to Equation (15) in the proof of Theorem 3. We can then establish the negative drift by comparing with λ_l^* (similar to the steps after Equation (15) in the proof of Theorem 3), and by using Lemma 3.

Since the interference degree under the node-exclusive interference model is 2, Theorem 4 implies that, by choosing κ close enough to 1, the above algorithm can guarantee close to $1/2$ of the capacity region Λ'.

VI. Conclusion

We have presented two low-complexity, distributed algorithms for scheduling in multi-hop wireless networks. The algorithms approximate the performance of maximal matching-type scheduling arbitrarily closely. However, a key feature that allows the two algorithms to have low complexity is that neither algorithm attempts to find a maximal matching. With high probability, Q-SCHED schedules links in those interference sets where the total queue-length is large. On the other hand, BP-SIM ensures that the probability that at least one link is scheduled in the interference set of each backlogged link is high. The complexity of both algorithms can be made to only depend on the maximum node-degree and an approximation factor (as to how close one wants to approach the performance guarantee of maximal schedules), but are otherwise independent of the size of the network.

Recently, there have been a number of papers addressing complexity and decentralization questions in multi-hop wireless networks. We now briefly comment on the contribution of this paper in the overall context of this line of research. Complexity issues in max-weight scheduling have been addressed through longest-queue-first and maximal scheduling in [6], [17]–[19], [30], [36], [37], through the use of randomized algorithms in [38]–[40], or through partitioning approaches in [41], [42]. The issues of decentralization and signaling overhead have been addressed in [40], [43]. The Q-SCHED algorithm in this paper addresses both complexity and overhead for fairly general models of wireless networks, while BP-SIM addresses these issues in the context of node-exclusive interference model only. However, as with prior papers, it is clear that there is a tradeoff between complexity, decentralization overhead, and performance. Further, our philosophy here is that an algorithm that has low complexity and overhead under a bounded node-degree assumption would be quite reasonable in practice since most ad hoc networks are expected to be of relatively small node-degrees in the near future. However, a detailed comparison of the algorithms in the growing literature of low-complexity, decentralized algorithms is currently lacking. Thus, a comprehensive simulation or theoretical study of these algorithms would be a good avenue for future investigation.

Appendix I
Proof of Lemma 1

Proof: Fix any link k such that Inequality (7) holds. Consider a link $j \in \mathcal{E}_k$. We will first find a lower bound on the probability that j is scheduled.

Link j gets scheduled when it attempts transmission and each of the other attempting links in its interference set choose a bigger backoff time. Let S_j be the event that j is scheduled and let Y_l be the backoff time chosen by a link l. Then we get

$$\begin{aligned}\Pr\{S_j\} &\geq \sum_{m=1}^{M}\Pr\{Y_j = m\}\prod_{\substack{h\in\mathcal{E}_j\\h\neq j}}\Pr(Y_h > m)\\ &= \sum_{m=1}^{M}\left(e^{-P_j\frac{m-1}{M}} - e^{-P_j\frac{m}{M}}\right)\prod_{\substack{h\in\mathcal{E}_j\\h\neq j}} e^{-P_h\frac{m}{M}} \qquad (16)\\ &= \left(e^{\frac{P_j}{M}}-1\right)\sum_{m=1}^{M} e^{-P_j\frac{m}{M}}\prod_{\substack{h\in\mathcal{E}_j\\h\neq j}} e^{-P_h\frac{m}{M}}\\ &= \left(e^{\frac{P_j}{M}}-1\right)\sum_{m=1}^{M} e^{(-\frac{m}{M}\sum_{h\in\mathcal{E}_j}P_h)} \qquad (17)\end{aligned}$$

where (16) is obtained by using the probability distribution described in Section III-A. We now find an upper bound on the term $\sum_{h\in\mathcal{E}_j} P_h$ that appears in (17).

$$\sum_{h\in\mathcal{E}_j} P_h = \alpha \sum_{h\in\mathcal{E}_j} \frac{\frac{q_h}{c_h}}{\max_{l\in\mathcal{E}_h}\sum_{i\in\mathcal{E}_l}\frac{q_i}{c_i}} \quad (18)$$

$$\leq \alpha \sum_{h\in\mathcal{E}_j} \frac{\frac{q_h}{c_h}}{\sum_{i\in\mathcal{E}_j}\frac{q_i}{c_i}} = \alpha, \quad (19)$$

where in Equation (19) we have used the assumption that if $h \in \mathcal{E}_j$, then $j \in \mathcal{E}_h$. This implies that the denominator in (18) is never less than $\sum_{i\in\mathcal{E}_j}\frac{q_i}{c_i}$. Using (17) and (19), we get

$$\Pr\{S_j\} \geq \left(e^{\frac{P_j}{M}} - 1\right) \sum_{m=1}^{M} e^{(-\alpha\frac{m}{M})} \geq \frac{P_j}{M}\sum_{m=1}^{M} e^{(-\alpha\frac{m}{M})}.$$

Hence, summing over all $j \in \mathcal{E}_k$, we have

$$\sum_{j\in\mathcal{E}_k} \Pr\{S_j\} \geq \sum_{m=1}^{M} e^{(-\alpha\frac{m}{M})} \sum_{j\in\mathcal{E}_k}\frac{P_j}{M}. \quad (20)$$

Now from the probability distribution given in Section III-A, we get

$$\sum_{j\in\mathcal{E}_k} P_j \geq \alpha\frac{\sum_{j\in\mathcal{E}_k}\frac{q_j}{c_j}}{V(n)} \geq \alpha\eta\frac{V(n) - C_1 - C_2\epsilon}{V(n)} \quad (21)$$

$$= \alpha\eta\left(1 - \frac{C_1 + C_2\epsilon}{V(n)}\right), \quad (22)$$

where (21) follows from the assumption that $\sum_{l\in\mathcal{E}_k}\frac{q_l}{c_l} \geq \eta(V(n) - C_1 - C_2\epsilon)$. Using (22) in (20) we get

$$\sum_{j\in\mathcal{E}_k} \Pr\{S_j\} \geq \frac{\alpha\eta}{M}\sum_{m=1}^{M} e^{(-\alpha\frac{m}{M})}\left(1 - \frac{C_1 + C_2\epsilon}{V(n)}\right)$$
$$= \frac{\alpha\eta}{M}\frac{1 - e^{-\alpha}}{1 - e^{-\frac{\alpha}{M}}} e^{-\frac{\alpha}{M}}\left(1 - \frac{C_1 + C_2\epsilon}{V(n)}\right).$$

Since $\alpha = \log(M)$ and $M > 1$, we can see that $\frac{\alpha}{M}/(1 - e^{-\alpha/M}) \geq 1$, $e^{-\alpha/M} \geq 1 - \log(M)/M$ and $1 - e^{-\alpha} = 1 - 1/M$. Hence,

$$\sum_{j\in\mathcal{E}_k} \Pr\{S_j\} \geq \eta\left(1 - \frac{\log(M) + 1}{M}\right)\left(1 - \frac{C_1 + C_2\epsilon}{V(n)}\right).$$

Now if $V(n) \geq R$ then $\frac{C_1+C_2\epsilon}{V(n)} \leq \frac{C_1+C_2\epsilon}{R}$. Thus, for sufficiently large R, we have $\frac{C_1+C_2\epsilon}{V(n)} \leq \epsilon$ and this gives

$$\sum_{j\in\mathcal{E}_k} \Pr\{S_j\} \geq \eta\left(1 - \frac{\log(M) + 1}{M}\right)(1 - \epsilon)$$
$$\geq \eta\left(1 - \frac{\log(M) + 1}{M} - \epsilon\right).$$

This ends the proof of Lemma 1. ■

Appendix II
Proof of Lemma 2

Proof: For any given H, since the arrivals and departures are both upper-bounded, there exists a constant C such that

$$\left|\sum_{l\in\mathcal{E}_k}\frac{q_l(n+1)}{c_l} - \sum_{l\in\mathcal{E}_k}\frac{q_l(n)}{c_l}\right| \leq C \text{ for all } n \text{ and } k.$$

For any given link k, we consider two cases.

Case 1: if

$$\sum_{l\in\mathcal{E}_k}\frac{q_l(n)}{c_l} \leq V(n) - H(C + \epsilon),$$

then

$$\Pr\{\sum_{l\in\mathcal{E}_k}\frac{q_l(n+H)}{c_l} \leq V(n) - H\epsilon\} = 1.$$

This can be seen from the fact that $\sum_{l\in\mathcal{E}_k}\frac{q_l}{c_l}$ cannot increase by more than C in a single time-slot. Thus, in this case the Lemma holds trivially.

Case 2: if $\sum_{l\in\mathcal{E}_k}\frac{q_l(n)}{c_l} > V(n) - H(C + \epsilon)$, then for all $t \in [n+1, n+H]$ we have

$$\sum_{l\in\mathcal{E}_k}\frac{q_l(t)}{c_l} \geq V(n) - H(C + \epsilon) - C(t - n)$$
$$\geq V(t) - 2C(t - n) - H(C + \epsilon)$$
$$\geq V(t) - H(3C + \epsilon).$$

Thus using Lemma 1 with $\eta = 1$, there exists a positive constant B (as a function of H) such that if $V(n) \geq B$ then

$$\sum_{l\in\mathcal{E}_k}\Pr\{S_l\} \geq 1 - \frac{\log(M) + 1}{M} - \epsilon \quad (23)$$

for all $t \in [n+1, n+H]$, where S_l is the event that link l is scheduled. Note that we did not need to impose a condition on each $V(t)$ separately because $V(t) \geq V(n) - C(t-n)$ for all t. Thus a sufficiently large B would guarantee the condition of Lemma 1 for all $V(t)$.

We define the event X_t such that $X_t = 1$ if at least one service occurs among the links in $\mathcal{E}_k$ in time-slot t, and $X_t = 0$ otherwise. Note that from Lemma 1 and (23), we have in fact obtained that, for all $t = n+1, ..., n+H$,

$$\Pr\{X_t = 1 | X_{t-1}, X_{t-2}, ..., X_{n+1}\} \geq 1 - \frac{\log(M) + 1}{M} - \epsilon.$$

Let $Y = \sum_{t=n+1}^{n+H} X_t$. For any $\theta \geq 0$, using the Chernoff bound we must have

$$\Pr\left\{Y \leq H\left(1 - \frac{\log(M) + 1}{M} - 2\epsilon\right)\right\}$$
$$= \Pr\left\{H - Y \geq H\left(\frac{\log(M) + 1}{M} + 2\epsilon\right)\right\}$$
$$\leq \frac{\mathbf{E}[\exp(\theta(H - Y)]}{\exp\left[\theta H\left(\frac{\log(M)+1}{M} + 2\epsilon\right)\right]}$$
$$\leq \exp\left(H\left\{\log\left[1 + (e^\theta - 1)(\frac{\log(M) + 1}{M} + \epsilon)\right] - \theta\left[\frac{\log(M) + 1}{M} + 2\epsilon\right]\right\}\right).$$

By appropriately choosing some $\theta > 0$, it is then easy to show that there exists a constant $\tau_1 > 0$ such that

$$\Pr\left\{Y \leq H\left(1 - \frac{\log(M) + 1}{M} - 2\epsilon\right)\right\} \leq e^{-H\tau_1}. \quad (24)$$

Similarly we can show that the aggregate arrivals $Z = \sum_{l\in\mathcal{E}_k}\sum_{t=n+1}^{n+H}\frac{A_l(t)}{c_l}$ must satisfy[2]

$$\Pr\left\{Z \geq H\left(1-\frac{\log(M)+1}{M}-3\epsilon\right)\right\} \leq e^{-H\tau_2} \qquad (25)$$

for some $\tau_2 > 0$. Thus from (24) and (25) we have

$$\Pr\left\{\sum_{l\in\mathcal{E}_k}\frac{q_l(n+H)}{c_l} \leq V(n)-H\epsilon\right\} \geq \Pr\{Z \leq Y - H\epsilon\}$$
$$\geq 1-e^{-H\tau_1}-e^{-H\tau_2}.$$

Thus, by choosing a large enough H and a correspondingly large enough B, we get

$$\Pr\left\{\sum_{l\in\mathcal{E}_k}\frac{q_l(n+H)}{c_l} \leq V(n)-H\epsilon\right\} \geq 1-\delta.$$

This ends the proof of Lemma 2. ■

APPENDIX III
PROOF OF LEMMA 3

Before we proceed it is helpful to define three terms $F_1(x)$, $F_4(x)$ and $F_3(x)$. First, $F_1(1) = 0$ and for any $x \geq 2$,

$$F_1(x) = \sum_{j=1}^{x-1}\binom{x-1}{j}\left(\frac{1}{2}\right)^{x-1}\left(1-\frac{1}{M}\sum_{l=1}^{M}\left(1-\frac{l}{M}\right)^j\right).$$

The term $F_1(x)$ is an upper bound on the probability that, when a given *left* node (say v_l) chooses to request another *right* node (say v) *that is not matched* and whose degree is $d(v) = x$, the node v_l is not acknowledged by v. Note that if $x = 1$, node v will always acknowledge the request from v_l. For $x \geq 2$, to derive this upper bound, we have assumed that each neighbor of v (other than v_l) becomes left with probability $1/2$ and the *left* neighbors of v always request v. This assumption maximizes the amount of contention and hence maximizes the probability that the request from v_l is not acknowledged. The event is broken down into cases when exactly j of the remaining $x-1$ neighbors of v become *left*, which occurs with probability $\binom{x-1}{j}\left(\frac{1}{2}\right)^{x-1}$. The term $\frac{1}{M}\sum_{l=1}^{M}\left(1-\frac{l}{M}\right)^j$ is the probability that, among $j+1$ contending nodes, a particular node (say v_l) wins (i.e., chooses the unique lowest backoff time). Here l represents the backoff time selected by the node v_l (which happens with probability $\frac{1}{M}$) and $\left(1-\frac{l}{M}\right)^j$ is the probability that the remaining contenders choose a larger backoff time. It is easy to check that $F_1(x) \leq F_1(d^*)$ for any $x \leq d^*$.

Define $F_4(x)$ such that $F_4(1) = 0$ and for $x \geq 2$,

$$F_4(x) = \sum_{j=1}^{x-1}\binom{x-1}{j}\left(\frac{1}{2}\right)^{x-1}\left(1-\frac{j+1}{M}\sum_{l=1}^{M}\left(1-\frac{l}{M}\right)^j\right).$$

The term $F_4(x)$ is the upper bound on the probability that, when a given *left* node (say v_l) chooses to request another *right* node (say v) *that is not matched* and whose degree is $d(v) = x$, the node v receives two or more requests and acknowledges none of them (due to collision). Note that if $x = 1$, node v will always acknowledge the request from v_l. For $x \geq 2$, to derive this upper bound, we have assumed that each neighbor of v (other than v_l) becomes left with probability $1/2$ and the *left* neighbors of v always request v. This assumption maximizes the amount of contention and hence maximizes the probability that the first two requests collide in a mini-slot. Similar to $F_1(x)$, the event is broken down into cases when exactly j of the remaining $x-1$ neighbors of v become *left*. The factor $j+1$ is because any one of the $j+1$ contending nodes (including v_l itself) can potentially win. We can check that $F_4(x) \leq F_4(d^*)$ for any $x \leq d^*$.

Further, define $F_3(x)$ such that $F_3(1) = 0$ and for $x \geq 2$,

$$F_3(x) = 1-\left(1-\frac{1}{2M}\right)^{x-1}.$$

Using similar techniques, we can show that the term $F_3(x)$ is the upper bound on the probability that, when a given *left* node (say v_l) chooses to request another *right* node (say v) *that is matched* and whose degree is $d(v) = x$, the node v_l does not receive a reply from v that v is matched.

Consider any directed link AB that is backlogged. We need to find a lower bound on the probability of the event M_{AB} that either A or B is matched at the end of K rounds. Towards this end, define $p(a, k)$ as a lower bound on the probability that either A or B will be matched after the next k rounds, conditioned on the event that neither A nor B has been matched yet, and A currently has a number of potentially available neighbors. We further require that, for each k, $p(a, k)$ is a non-increasing function of a. Note that $p(a, k)$ is well defined for $a \geq 1$. For ease of exposition, we define $p(0, k) = 1$ for all k. The intuition behind the last definition is that if A has no potentially available neighbors, then B must be matched.

For any k, assume that $p(a, k)$ is given for all $a \geq 0$ and it is non-increasing in a. We next derive an expression that relates $p(a, k+1)$ to $p(a, k)$ for all $a \geq 1$. Consider $p(a, k+1)$ for a given $a \geq 1$, and consider the round that immediately follows. Since by definition neither A nor B has been matched yet, according to the BP-SIM algorithm node A will become left or right with probability $1/2$. If A becomes right, the worst case is that neither A nor B will be matched in this round. Further, since A is a right node, A will not learn of any additional matched neighbors. Hence, conditioned on A becomes a right node in the first round, a lower bound on the probability that neither A nor B is matched in $k+1$ rounds is $p(a, k)$.

On the other hand, if node A becomes left, with probability $1/a$ it will request the node B. Node B becomes a right node with probability at least $1/2$. Conditioned on the event that A requests B and node B becomes right, B will acknowledge one request with probability at least $1 - F_4(d^*)$, in which case B will be matched. If the node B does not acknowledge any requests (which occurs with probability at most $F_4(d^*)$), then node A's number of potentially available neighbors does not change. Combining the above arguments, we can conclude that, conditioned on the event that A requests B in the first

[2]This is the place where we use the assumption that the arrivals are *i.i.d.* across time. However, the property (25) also holds for more general arrival processes, e.g., Markov-modulated Poisson arrivals.

round, a lower bound on the probability that neither A nor B is matched in $k+1$ rounds is

$$\begin{aligned}&\frac{1}{2}(1-F_4(d^*))+(1-\frac{1}{2}(1-F_4(d^*))p(a,k)\\=\ &\frac{1-F_4(d^*)}{2}+\frac{1+F_4(d^*)}{2}p(a,k).\end{aligned}$$

Next, consider that case when A becomes left, and with probability $1-1/a$ it requests a neighbor other than B. There are two sub-cases.

Sub-case 1: if this neighbor is a matched node, then with probability at least $1-F_3(d^*)$ node A will be able to receive a reply, and hence its number of potentially available neighbors can decrease by 1. If A does not receive a reply, then its number of potentially available neighbors does not change. Recall that $p(a-1,k)\geq p(a,k)$ by our definition. Hence, conditioned on the event that A requests a neighbor that is matched, a lower bound on the probability that neither A nor B is matched in $k+1$ rounds is

$$(1-F_3(d^*))p(a-1,k)+F_3(d^*)p(a,k).$$

Sub-case 2: if this neighbor is an unmatched node, then it becomes a right node with probability at least $1/2$. Hence, the request from A will be acknowledged with probability at least $\frac{1}{2}(1-F_1(d^*))$, in which case A will be matched. If the request from A is not acknowledged (which occurs with probability at most $1-\frac{1}{2}[1-F_1(d^*)]$), the worst case will be that A does not receive any reply from the neighbor being requested, and hence node A's number of potentially available neighbors does not change. Combining the above arguments, we can conclude that, conditioned on the event that A requests a neighbor (other than B) that is not matched, a lower bound on the probability that neither A nor B is matched in $k+1$ rounds is

$$\begin{aligned}&\frac{1}{2}(1-F_1(d^*))+(1-\frac{1}{2}(1-F_1(d^*))p(a,k)\\=\ &\frac{1-F_1(d^*)}{2}+\frac{1+F_1(d^*)}{2}p(a,k).\end{aligned}$$

Finally, taking the smaller value between the above two sub-cases, we can then conclude the following: conditioned on the event that A becomes left in the first round and it requests a neighbor other than B, a lower bound on the probability that neither A nor B is matched in $k+1$ rounds is

$$\begin{aligned}\min\Big[&(1-F_3(d^*))p(a-1,k)+F_3(d^*)p(a,k),\\&\frac{1-F_1(d^*)}{2}+\frac{1+F_1(d^*)}{2}p(a,k)\Big].\end{aligned}$$

Combining all of the discussions above, we can derive a value for $p(a,k+1)$ as

$$\begin{aligned}&p(a,k+1)\\=\ &\min\Big\{p(a-1,k+1),\\&\frac{1}{2}p(a,k)+\frac{1}{2a}\left[\frac{1-F_4(d^*)}{2}+\frac{1+F_4(d^*)}{2}p(a,k)\right]\\&+\frac{1}{2}(1-\frac{1}{a})\min\Big[\\&\quad(1-F_3(d^*))p(a-1,k)+F_3(d^*)p(a,k),\\&\quad\frac{1-F_1(d^*)}{2}+\frac{1+F_1(d^*)}{2}p(a,k)\Big]\Big\}\end{aligned}\qquad(26)$$

where in the first term we have incorporated the requirement that $p(a,k+1)$ must be non-decreasing in a.

We can then use iteration (26) to compute all values of $p(a,k), a\geq 1, k=1,2,....$ The initial condition for $k=0$ is given by $p(a,0)=0$ if $a\geq 1$ and $p(0,k)=1$ for all k. The probability of interest in Lemma 3 is then $\Pr(M_{AB})\geq p(d^*,K)$. It remains to show the following claim.

***Lemma** 5:* Given d^* and M, for any $a\geq 0$, $p(a,k)\to 1$ as $k\to\infty$.

Proof: We prove by induction on a. By our definition that $p(0,k)=1$ for all k, the claim holds trivially for $a=0$. Assume that the claim holds for a, we next show that it must also hold for $a+1$. To see this, note that by the induction hypothesis, for any $\epsilon>0$, we can find K_0 such that $p(a,k)\geq(1-\epsilon)$ for $k\geq K_0$. Let

$$F=\max\{\frac{1+F_1(d^*)}{2},F_3(d^*),\frac{1+F_4(d^*)}{2}\}.$$

Note that $F<1$. Then, using (26), we have, for all $k\geq K_0$,

$$\begin{aligned}&p(a+1,k+1)\geq\min\{1-\epsilon,\\&\quad\frac{1}{2}p(a+1,k)+\frac{1}{2}\left[(1-F)(1-\epsilon)+Fp(a+1,k)\right]\}.\end{aligned}$$

Since $1-F>0$, this implies that $\liminf_{k\to\infty}p(a+1,k)\geq 1-\epsilon$. Since ϵ can be arbitrarily small, the results of Lemma 5 then follows. ■

Hence, the result of Lemma 3 must hold.

References

[1] A. Gupta, X. Lin, and R. Srikant, "Low-Complexity Distributed Scheduling Algorithms for Wireless Networks," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007.

[2] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1067–1073, May 1997.

[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. 1936–1949, December 1992.

[4] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.

[5] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proceedings of IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004, pp. 1484–1489.

[6] ——, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005, pp. 1804–1814.

[7] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005, pp. 1794–1803.

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

[9] I. Paschalidis, W. Lai, and D. Starobinski, "Asymptotically optimal transmission policies for large-scale low-power wireless sensor networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005, pp. 2458–2469.

[10] M. Neely and E. Modiano, "Fairness and optimal stochastic control for heterogeneous networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005, pp. 1723–1734.

[11] M. Chiang, "To layer or not to layer: Balancing transport and physical layers in wireless multihop networks," in *Proceedings of IEEE INFOCOM*, March 2004, pp. 2525–2536.

[12] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, August 2006.

[13] B. A. Miller and C. Bisdikian, *BLUETOOTH REVEALED: The Insider's Guide to an Open Specification for Global Wireless Communications*. Upper Saddle River, NJ: Prentice Hall, 2001.

[14] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, pp. 910–917, September 1988.

[15] 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, 1997.

[16] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proceedings of IEEE INFOCOM*, 2000, pp. 556–564.

[17] X. Lin and N. Shroff, "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.

[18] X. Wu, R. Srikant, and J. R. Perkins, "Queue-length stability of maximal greedy schedules in wireless networks," in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[19] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in wireless networks," in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[20] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, December 2006.

[21] X. Lin and S. B. Rasool, "Constant-Time Distributed Scheduling Policies for Ad Hoc Wireless Networks," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 231–242, February 2009.

[22] K. Jung and D. Shah, "A distributed matching algorithm," Massachusetts Institute of Technology, 2006, Technical Report.

[23] 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, 1986.

[24] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: Algorithms and analysis," in *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, December 2006.

[25] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proceedings of IEEE Conference on Decision and Control,* San Diego, CA, December 2006.

[26] M. J. Neely and R. Urgaonkar, "Optimal Backpressure Routing for Wireless Networks with Multi-Receiver Diversity," *Ad Hoc Networks (Elsevier)*, vol. 7, no. 5, pp. 862–881, July 2009.

[27] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput Guarantees in Maximal Scheduling in Wireless Networks," in *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2005.

[28] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *Proceeding of MobiHoc*, Urbana-Champaign, IL, May 2005, pp. 157–168.

[29] S. Asmussen, *Applied Probability and Queues*. New York: Springer-Verlag, 2003.

[30] 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.

[31] X. Lin and N. B. Shroff, "Utility Maximization for Communication Networks with Multi-path Routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, May 2006.

[32] 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.

[33] 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.

[34] C. Humes, "A regulator stabilization technique: Kumar-Seidman revisited," *IEEE Transactions on Automatic Control*, vol. 39, pp. 191–196, January 1994.

[35] 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.

[36] 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.

[37] 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.

[38] E. Modiano, D. Shah, and G. Zussman, "Maximizing Throughput in Wireless Networks via Gossiping," in *Proc. ACM SIGMETRICS*, June 2006.

[39] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial Complexity Algorithms for Full Utilization of Multi-hop Wireless Networks," in *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[40] S. Sanghavi, L. Bui, and R. Srikant, "Distributed Link Scheduling with Constant Overhead," in *ACM SIGMETRICS*, San Diego, June 2007.

[41] 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.

[42] K. Jung and D. Shah, "Low Delay Scheduilng Algorithms," in *Proceedings of IEEE ISIT*, Nice, France, June 2007.

[43] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc network," in *Proceedings of IEEE INFOCOM*, Phoenix, Arizona, April 2008.

Abhinav Gupta (S'03 / M'08) received his B.S. degree in Electrical Engineering from the University of Illinois at Urbana-Champaign, in 2005, and the M.S. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2006. He is currently working as Systems Engineer at Qualcomm.

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 in Electrical and Computer Engineering 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.

R. Srikant (S'90 / M'91 / SM'01 / F'06) received his B.Tech. from the Indian Institute of Technology, Madras in 1985, his M.S. and Ph.D. from the University of Illinois at Urbana-Champaign in 1988 and 1991, respectively, all in Electrical Engineering. He was a Member of Technical Staff at AT&T Bell Laboratories from 1991 to 1995. He is currently with the University of Illinois at Urbana-Champaign, where he is a Professor in the Department of Electrical and Computer Engineering, and a Research Professor in the Coordinated Science Lab.

He was an associate editor of Automatica, the IEEE Transactions on Automatic Control, and the IEEE/ACM Transactions on Networking. He has also served on the editorial boards of special issues of the IEEE Journal on Selected Areas in Communications and IEEE Transactions on Information Theory. He was the chair of the 2002 IEEE Computer Communications Workshop in Santa Fe, NM and a program co-chair of IEEE INFOCOM, 2007. His research interests include communication networks, stochastic processes, queueing theory, information theory, and game theory.