Low-Complexity Distributed Scheduling Algorithms for Wireless Networks

Abhinav Gupta
Coordinated Science Lab
and
Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Email: agupta12@uiuc.edu

Xiaojun Lin
School of Electrical and
Computer Engineering
Purdue University
Email: linx@ecn.purdue.edu

R. Srikant
Coordinated Science Lab
and
Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Email: rsrikant@uiuc.edu

***Abstract*— We consider the problem of distributed scheduling in wireless networks. We present two different algorithms whose performance is arbitrarily close to that of maximal schedules, but which require low complexity 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 independent of the size and topology of the network. The second algorithm is 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.**

I. Introduction

In this paper we consider the problem of 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 backpressure scheduling algorithms introduced in [1], [2] are throughput-optimal, i.e., they achieve the maximum possible throughput. However, they are centralized algorithms and have high computational complexity. Using the max-weight or backpressure algorithms for scheduling, a number of recent papers have studied the problem of joint congestion control, routing, and scheduling in multihop wireless networks [3]–[10]; see [11] 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 [12], [13].

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 use of maximal matching have been derived in [14], [15]. Recently, these ideas have been successfully applied to wireless networks in [5], [16]–[18]. 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 [5], [16] that the lower bound is 1/2 for the node-exclusive interference model while [17], [18] 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 [5], [16]–[18] 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. This problem is unique to wireless networks since a matching can be implemented by a central controller in many high-speed switches. 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 [19], but the key contribution in this paper is to modify the algorithm to achieve dramatically larger throughput. We will show that the complexity and overhead of Q-SCHED is independent of the network size and throughput. (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. For ease of exposition and notational simplicity, we present BP-SIM only for the node-exclusive interference model. BP-SIM is an adaption of the algorithm in [20], [21] 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 [20], [21] 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. Numerical examples based on analytical formulas indicate that the complexity is quite low for moderately sized networks (up to 225 nodes that we have simulated).

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 [22], [23]. In addition to scheduling, another important issue is power control which we do not address in this paper. We refer the readers to [10] for distributed implementation of power control in multi-hop wireless networks.

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 Q-SCHED scheduling algorithm, provide its analysis and discuss a slight variation of the algorithm that can be specifically used for the node-exclusive interference model. In Section IV we present BP-SIM scheduling algortihm, provide its analysis and discuss simulation results. We then conclude in Section V.

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 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. 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 [5], [13] and is a commonly used model for bluetooth and FH-CDMA networks [12], [13].

We assume that time is divided into slots of equal length. With each link l is associated a stochastic arrival process $\{A_l(n)\}$, where $A_l(n)$ is the number of packet arrivals to link l in the slot n. The arrival processes are stationary and let $\lambda_l = E[A_l(n)]$. It is further assumed 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. It is also assumed that the arrival process has bounded second moments, i.e., $Cov\big(A_l(n), A_k(n)\big) < \infty$. Let $D_l(n)$ denote the number of departures from link l 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

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

and the dynamics of $q_l(n)$ are given by

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

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

III. Algorithm 1: Q-SCHED

For this algorithm, it is assumed that at the beginning of each time-slot every link l has the knowledge about 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 neighbors 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 $q_j(n)$ where there is no confusion. The algorithm proceeds as follows: at the beginning of time-slot n, each link l first computes

$$P_l = \alpha \frac{\frac{q_l}{c_l}}{\max_{i \in \mathcal{E}_l} [\sum_{k \in \mathcal{E}_i} \frac{q_k}{c_k}]} \ , \tag{1}$$

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:

$$\Pr\{Y = M+1\} = e^{-P_l}, \quad (2)$$
$$\Pr\{Y = m\} = e^{-P_l \frac{m-1}{M}} - e^{-P_l \frac{m}{M}}, \quad \forall m \in \{1, 2, ...M\}.$$

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 links that hears the collision will not attempt transmission in the rest of their time-slot.

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 the algorithm this 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. [24]. While data transmission may start at any mini-slot, the length of each packet is assumed to be smaller than the data transmission slot so that a transmission ends within the time-slot. The idea of using two phases is essential to getting $O(1)$ complexity and was introduced in [19]. Here the probabilities in (2) have been modified to achieve a higher guaranteed throughput. Further, we complete the proof in [19] by establishing stochastic stability.

B. Analysis

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

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

***Lemma** 1:* 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 $V(n) \geq R$ and $\sum_{l \in \mathcal{E}_k} \frac{q_l}{c_l} \geq V(n) - C_1 - C_2\epsilon$ then

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

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 $A_l(n) \leq \theta \ \forall l, n$. 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 $\sum_{l \in \mathcal{E}_k} \frac{\lambda_l}{c_l} < 1 - \frac{\log(M)+1}{M} - 4\epsilon \quad \forall k \in E$, then in the time-slot $n+H$

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

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 $\epsilon > 0$

$$\sum_{k \in \mathcal{E}_l} \frac{\lambda_k}{c_k} < 1 - \frac{\log(M)+1}{M} - 4\epsilon \ \forall l.$$

Proof: Note that from Lemma 2 we also 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 for any $\delta > 0$,

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

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

Since the arrival and departure processes are both upper-bounded, there exists a constant C such that

$$\left| \sum_{l \in \mathcal{E}_k} \frac{q_l(n+1)}{c_l} - \sum_{l \in \mathcal{E}_k} \frac{q_l(n)}{c_l} \right| \leq C \quad \forall n, \forall k. \quad (3)$$

This implies that $V(n+1) - V(n) \leq HC$. 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 [25] we have proved that the Markov chain $\{Q(n)\}$ is positive recurrent. ■

C. A Special Case

The scheduling algorithm discussed above is valid for any interference model including the node-exclusive interference model, as long as the symmetry conditions are satisfied. However in the special case of node-exclusive interference model, an even simpler variant of this algorithm can be used. The only difference in the scheduling policy is in the calculation of the term P_l. For the new scheduling policy 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 $\mathcal{F}_j$ is the set of links incident on j. The nodes $b(l)$ and $e(l)$ are the transmitter and the receiver nodes, respectively, of link l. In this case we choose $\alpha = \log(2M)/2$.

Thus in this scheduling policy each link requires a knowledge of the queue-lengths of only those links that interfere with it. We can prove that in this case if

$$\sum_{l \in \mathcal{F}_i} \frac{\lambda_l}{c_l} < 1/2 - \frac{\log(2M)}{2M} \qquad \forall i \in V,$$

then $Q(n)$ is positive recurrent and the system is queue-length stable.

The factor of 1/2 appears because the set $\mathcal{F}_i$ where $i \in V$ is on an average about half the size of the set $\mathcal{E}_l$ where $l \in E$. However the performances of the two algorithms are comparable. We do not provide the proof since it is very similar to the proof in Section III-B.

Similar results have been obtained in [26] using a different approach. However, we complete the proof by establishing stochastic stability and we also provide an alternative algorithm, in the next section, which does not require nodes to collect any queue-length information from its neighbors. Such an algorithm is especially useful when collecting queue-length information is not feasible, as discussed in the following section.

IV. Algorithm 2: BP-SIM

While Q-SCHED is an O(1) complexity algorithm, 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 normally needs to be performed asynchronously and thus may reduce the performance of the algorithm. In this section, we present an alternative algorithm that requires no queue-length information. As in the case of Q-SCHED, BP-SIM does not attempt to find maximal schedule but rather ensures that, for 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 [17] for the stability of maximal schedules can be adapted easily even when the schedule is not precisely maximal. While BP-SIM is not on O(1) complexity algorithm, our analysis followed by numerical examples indicate that the complexity of the algorithm is low for networks with even a few hundred nodes.

BP-SIM is presented for the node-exclusive interference model. Further it is assumed that the maximum degree of any node in the network is upper-bounded by d^*, i.e., $\forall v \in V, d(v) \leq d^*$. The algorithm proceeds in rounds. There are K rounds in the algorithm, where K is a constant to be chosen later.

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 $K \times M$ mini-slots, where each round requires M of these mini-slots. Initially none of the links are in the schedule. In each round a matching $\mathcal{M}_i$ is formed by adding links from the graph G to the matching $\mathcal{M}_{i-1}$.The matching $\mathcal{M}_K$ is the final schedule.

In the first round, to find $\mathcal{M}_1$, BP-SIM emulates a bipartite graph, each node randomly deciding to be *left* or *right*. The round proceeds as follows: each node having at least one neighbor in G decides to be *left* or *right* with probability 1/2 independently. If a node, say v_l, becomes *left* then it chooses a partner, say v_{lr}, from one of its backlogged neighbors in G uniformly at random. The node v_{lr} is backlogged with respect to v_l if the link from v_l to v_{lr} has a backlog greater than or equal to the capacity of the link. A node that does not have a backlogged neighbor does not become a *left* node. Node v_l randomly and independently chooses a mini-slot between 1 and M uniformly, and requests v_{lr} in that mini-slot to be scheduled for that time-slot.

If a node, say v_r, becomes *right* then it waits for a request to be scheduled from one of its neighbors. Node v_r acknowledges the request from the node that first requested it. If at the first time when v_r received a request, there was a collision due to simultaneous requests from more than one neighbor, then v_r does not acknowledge any request in that round.

If v_r acknowledges the request of v_l then v_r and v_l are matched. The matching $\mathcal{M}_1$ then consists of v_l and v_r and of the link between them. Scheduling requests from a *left* node to a *left* node are not acknowledged.

In the subsequent rounds, same process is repeated except that the nodes which are already a part of $\mathcal{M}_{i-1}$ neither become *left* nor they acknowledge requests from any other nodes. This ensures that the links added to $\mathcal{M}_{i-1}$ will also appear in matching $\mathcal{M}_i$. In all the rounds *left* nodes choose a partner from one of their neighbors in G and hence do not need to know which of their neighbors are already scheduled.

After the K rounds, the links that are scheduled begin transmission in the data transmission slot.

This algorithm is an extension of the maximal matching algorithms discussed in [20] and [21]. The novel feature of BP-SIM is the contention resolution protocol that is necessary in a wireless network to reduce the chance of collisions between the connection requests. We also observe that even when no collisions occur due to simultaneous requests to a node, the time to compute perfect maximal matching increases as $\log(n)$, where n is the number of nodes. In contrast, in our algorithm the scheduling time depends only on d^* and not on n as we will see later.

B. Analysis

We now proceed to the analysis of 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 prove the stability of BP-SIM.

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

$$\sum_{k \in \mathcal{E}_l} \frac{\lambda_k}{c_k} < \kappa,$$

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

Proof: Define the Lyapunov function

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

This is the same Lyapunov function as the one used in [17]. Using the results in [17] 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

$$\begin{aligned} &E_{Q(n)}(V(n+1) - V(n)) \\ &\le 2\sum_l \frac{q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - E\left[\sum_{k\in\mathcal{E}_l} d_k(n) \right] \right) + B \\ &= 2\sum_{l:q_l>0} \frac{q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - E\left[\sum_{k\in\mathcal{E}_l} d_k(n) \right] \right) + B \\ &\le 2\sum_{l:q_l>0} \frac{q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - \kappa \right) + B_1 \\ &\le -2\epsilon \sum_{l:q_l>0} \frac{q_l(n)}{c_l} + B_1, \end{aligned}$$

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 [17, Th. 1], the system is queue-length stable if

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

∎

C. Numerical examples and Simulations

Theorem 2 shows that by choosing K large enough the throughput of BP-SIM can be arbitrarily close to that of maximal schedule. We next present some numerical results on how large K needs to be for practical choice of κ.

According to the numerical evaluations based on analytical functions derived in Appendix III when $d^* = M = 5$, we require $K \ge 7$ for $\kappa = 0.9$. Therefore in 35 $(K \times M)$ mini-slots BP-SIM guarantees that for any 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 $K \ge 8$ to achieve $\kappa = 0.9$.

We have simulated the scheduling policy BP-SIM to analyze its scheduling efficiency. 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. there x and y were chosen randomly and independently. The radius of transmission of the nodes was chosen so as to make the graph connected.

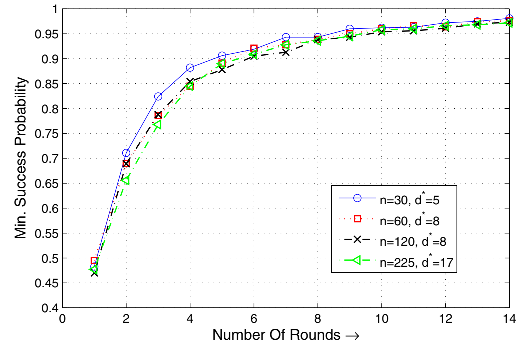

Fig. 1. Scheduling Efficiency of BP-SIM: Curves for all fours cases are close in the figure, thus showing the nice scaling property of the algorithm.

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. The maximum backoff time in each case was $M = 4$.

In each case shown in Figure 1 we only needed $K = 6$ rounds for the minimum success probability to be greater than 0.9. Thus in a total of 24 $(K \times M)$ mini-slots each link in the network reaches a success probability greater than 0.9.

The above simulation results show that in practice BP-SIM works much better than the minimum performance guarantees we have proved. This is indeed true as our bounds are very conservative and we do not take into the account that when links are scheduled in one round, BP-SIM performs better in the next round since the degrees of nodes decrease.

V. 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 perfect maximal matching. With high probability, Q-SCHED schedules links in those interference sets where the total queue-length is large. BP-SIM ensures that the probability that a link is scheduled in the interference set of each link is high. Q-SCHED is an $O(1)$ complexity algorithm, i.e. , its complexity is independent of the network size. On the other hand, the complexity of BP-SIM depends only on the maximum node degree. One may argue that in random radio networks, the node degree should be $O(\log(n))$ for connectivity. If this is the case, for $n = 200$, $\log(n)$ is smaller than 10 and our simulations indicate that the number of mini-slots required to guarantee a throughput at least equal to 0.9 times that of maximal schedules is 24. On the other hand our analysis upper-bounds the number of mini-slots to be 80. A mini-slot can be thought as the contention slot in 802.11. In today's standard the number of mini-slots varies from 32 to 1024. Thus, both algorithms have small overheads and can provide guaranteed throughput in ad hoc networks as shown in Theorems 1 and 2.

Appendix I
Proof of Lemma 1

Proof: 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 chooses 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

$$\Pr\{S_j\} \geq \sum_{k=1}^{M} \Pr\{Y_j = k\} \prod_{\substack{h\in\mathcal{E}_j\\ h\neq j}} \Pr\left(Y_h > k\right)$$

$$= \sum_{k=1}^{M} \left(e^{-P_j\frac{k-1}{M}} - e^{-P_j\frac{k}{M}}\right) \prod_{\substack{h\in\mathcal{E}_j\\ h\neq j}} e^{-P_h\frac{k}{M}} \qquad (4)$$

$$= \left(e^{\frac{P_j}{M}} - 1\right) \sum_{k=1}^{M} e^{-P_j\frac{k}{M}} \prod_{\substack{h\in\mathcal{E}_j\\ h\neq j}} e^{-P_h\frac{k}{M}}$$

$$= \left(e^{\frac{P_j}{M}} - 1\right) \sum_{k=1}^{M} e^{(-\frac{k}{M}\sum_{h\in\mathcal{E}_j} P_h)} \qquad (5)$$

where (4) is obtained by using the probability distribution described in Section III-A. We now find the upper bound on the term $\sum_{h\in\mathcal{E}_j} P_h$ that appears in (5).

$$\sum_{h\in\mathcal{E}_j} P_h = \alpha \sum_{h\in\mathcal{E}_j} \frac{\frac{q_h}{c_h}}{\max_{l\in\mathcal{E}_h} \sum_{k\in\mathcal{E}_l} \frac{q_k}{c_k}} \qquad (6)$$

$$\leq \alpha \sum_{h\in\mathcal{E}_j} \frac{\frac{q_h}{c_h}}{\sum_{k\in\mathcal{E}_j} \frac{q_k}{c_k}} = \alpha. \qquad (7)$$

To obtain (7) we recall from the definition of an interference set that if $h \in \mathcal{E}_j$, then $j \in \mathcal{E}_h$. This implies that the denominator in (6) is never less than $\sum_{k\in\mathcal{E}_j} \frac{q_k}{c_k}$. Using (7) and (5) we get

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

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

$$\sum_{j\in\mathcal{E}_k} \Pr\{S_j\} \geq \sum_{k=1}^{M} e^{(-\alpha\frac{K}{M})} \sum_{j\in\mathcal{E}_k} \frac{P_j}{M}. \qquad (8)$$

Now from the probability distribution given in Section III-A

$$P_j = \alpha \frac{\frac{q_j}{c_j}}{\max_{m\in\mathcal{E}_j} [\sum_{k\in\mathcal{E}_m} \frac{q_k}{c_k}]} \geq \alpha \frac{\frac{q_j}{c_j}}{V(n)} \qquad (9)$$

where (9) follows from the definition of $V(n)$. Now summing over all $j \in \mathcal{E}_k$ we get

$$\sum_{j\in\mathcal{E}_k} P_j \geq \alpha \frac{\sum_{j\in\mathcal{E}_k} \frac{q_j}{c_j}}{V(n)} \geq \alpha \frac{V(n) - C_1 - C_2\epsilon}{V(n)} \qquad (10)$$

$$= \alpha \left(1 - \frac{C_1 + C_2\epsilon}{V(n)}\right) \qquad (11)$$

where (10) follows from the assumption $\sum_{l\in\mathcal{E}_k} \frac{q_l}{c_l} \geq V(n) - C_1 - C_2\epsilon$. Using (11) in (8) we get

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

$$= \frac{\alpha}{M} \frac{1 - e^{-\alpha}}{1 - e^{-\frac{\alpha}{M}}} e^{-\frac{\alpha}{M}} \left(1 - \frac{C_1 + C_2\epsilon}{V(n)}\right).$$

Since $\alpha = \log(M)$ and $M > 1$, we can see that

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

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 a sufficiently large R we have $\frac{C_1+C_2\epsilon}{V(n)} \leq \epsilon$ and this gives

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

$$\geq 1 - \frac{\log(M) + 1}{M} - \epsilon.$$

This ends the proof of Lemma 1. ■

Appendix II
Proof of Lemma 2

Proof: For any given H and with C as defined in (3), 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.

In the other case, i.e. , 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

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

$$\geq V(t) - 2C(t - n) - H(C + \epsilon)$$

$$\geq V(t) - 2HC - H(C + \epsilon)$$

$$= V(t) - H(3C + \epsilon)$$

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

$$\sum_{l\in\mathcal{E}_k} \Pr\{S_l\} \geq 1 - \frac{\log(M) + 1}{M} - \epsilon \qquad (12)$$

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 $V(t) \leq V(n) + C(t-n)$ 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 in the time-slot t and $X_t = 0$ otherwise. Then using (12) we get

$$\Pr\{X_t = 1\} \geq 1 - \frac{\log(M)+1}{M} - \epsilon.$$

Let $Y = \sum_{t=n+1}^{n+H} X_t$. Using Chernoff bounds we can now show that there exists a constant $\tau_1 > 0$ such that

$$\Pr\left\{Y \leq H\left(1 - \frac{\log(M)+1}{M} - 2\epsilon\right)\right\} \leq e^{-H\tau_1}. \quad (13)$$

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

$$\Pr\left\{Z \geq H\left(1 - \frac{\log(M)+1}{M} - 3\epsilon\right)\right\} \leq e^{-H\tau_2} \quad (14)$$

for some $\tau_2 > 0$. Thus from (13) and (14) 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 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

Proof: The proof is divided into two parts. We first find lower bounds on the probability that a link or one of the links in its interference set is scheduled and then analyze the bound. In the discussion that follows we only consider backlogged links since links with zero backlog do not participate in the schedule for that round.

A. Bounds

In a graph $\tilde{G}(\tilde{V}, \tilde{E})$, for a vertex $A \in \tilde{V}$, another vertex $B \in \tilde{V}$ is a neighbor of A if there is an edge between A and B. An edge $e \in \tilde{E}$ is equivalently represented as AB if it connects the two vertices A and B.

For the analysis we need to find the lower bound on $P_{\mathcal{E}_{AB}}$, the probability that in a given stage at least one edge from $\mathcal{E}_{AB}$ is matched.

Let M_A be the event that vertex A is matched and M_{AB} the event that the edge AB is matched in the given stage. Let P_A be the probability of the event M_A. Then

$$P_{\mathcal{E}_{AB}} = \Pr(M_A \ \ or \ M_B) \geq \max(P_A, P_B).$$

To find the lower bound we need to consider different degrees of the vertices A and B. Let $x_1 = d(A)$ and $x_2 = d(B)$. We know $1 \leq x_1, x_2 \leq d^*$. The case in which either of the degrees is one is a special case since the vertex with degree 1 always acknowledges a request, i.e., there is no contention at its end.

Before we proceed it is helpful to define two terms $F_1(x)$ and $F_2(x)$ for $x \geq 2$, as follows:

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

and

$$F_2(x) = \sum_{j=2}^{x}\binom{x}{j}\left(\frac{1}{2}\right)^{x}\left(1 - \frac{j}{M}\sum_{l=1}^{M}\left(1-\frac{l}{M}\right)^{j-1}\right).$$

The term $F_1(x)$ is the upper bound on the probability that when a given vertex, say v_l, chooses to request another vertex, say v, it is not acknowledged, where $d(v) = x$. The event is broken down into cases when j of the remaining $x-1$ neighbors of v become *left*.

The term $F_2(x)$ is the upper bound on the probability that a vertex, say v, with degree $d(v) = x$ receives two or more requests and acknowledges none, given that v is *right*. The event is broken down into cases when j of the neighbors of v become *left*.

To obtain these probabilities we observe that the probability that among m contending nodes, a particular node, say r, wins (chooses the unique lowest backoff time) is $\frac{1}{M}\sum_{l=1}^{M}\left(1-\frac{l}{M}\right)^{m-1}$. Here l is the backoff time selected by the link r (which happens with probability $\frac{1}{M}$) and $\left(1-\frac{l}{M}\right)^{m-1}$ is the probability that the remaining contenders choose a larger backoff time. The probability than any one of them wins is simply m times this. In deriving the upper bounds we have assumed that the *left* neighbors of v always request v.

Now we will proceed to find a lower bound on $P_{\mathcal{E}_{AB}}$ considering different values of x_1 and x_2.

Case 1 : $x_1 = x_2 = 1$

In this case the edge AB has no neighbor. It is straightforward that $P_{\mathcal{E}_{AB}} = 0.5$.

Case 2 : $x_1 = 1, x_2 \geq 2$

In this case we first find a lower bound on P_A. Let R be the set of all nodes that become *right* and L the set of nodes that become *left*.

$$\begin{aligned}P_A &= \Pr(M_A \mid A \in R)\Pr(A \in R) \\ &\quad + \Pr(M_A \mid A \in L)\Pr(A \in L) \\ &= \frac{1}{2}(\Pr(M_A \mid A \in R) + \Pr(M_A \mid A \in L)). \quad (15)\end{aligned}$$

Let M_A^c be complement of the event M_A, then

$$\begin{aligned}\Pr(M_A^c \mid A \in R) &= \Pr(B \in R \mid A \in R) \\ &\quad + \Pr(B \ \text{ does not request } \ A \mid B \in L, A \in R) \\ &\qquad \times \Pr(B \in L \mid A \in R).\end{aligned}$$

Since B has x_2 neighbors, it will request A with $\frac{1}{x_2}$ probability. Thus we get

$$\Pr\left(M_A^c \mid A \in R\right) = \frac{1}{2} + \frac{1}{2}(1 - \frac{1}{x_2}) = 1 - \frac{1}{2x_2}. \quad (16)$$

Similarly,

$$\begin{aligned}\Pr\left(M_A^c \mid A \in L\right) &= \Pr\left(B \in L \mid A \in L\right) \\ &\quad + \Pr\left(M_A^c \mid A \in L, B \in R\right) \\ &\quad \times \Pr\left(B \in R \mid A \in L\right) \\ &= \frac{1}{2} + \frac{1}{2}\Pr\left(M_{AB}^c \mid A \in L, B \in R\right).\end{aligned}$$

When A requests B, which is *right*, B does not acknowledge the request if some other neighbor of B requests it before or in the same mini-slot as A. If N is a set of vertices then L_N is defined as $L_N = L \cap N$. Similarly $R_N = R \cap N$. We break the event into cases where j of the neighbors of B are *left*. Thus,

$$\begin{aligned}&\Pr\left(M_{AB}^c \mid A \in L, B \in R\right) \\ &= \sum_{j=1}^{x_2-1} \Big[\Pr\left(M_A^c \mid A \in L, B \in R, |L_{E(B)}| = j+1\right) \\ &\qquad \times \Pr\left(|L_{E(B)}| = j+1 \mid A \in L, B \in R\right)\Big].\end{aligned}$$

Therefore, $\Pr\left(M_{AB}^c \mid A \in L, B \in R\right) \leq F_1(x_2)$ which along with (15) and (16) gives

$$P_A \geq 0.25 + \frac{1}{4\,x_2} - \frac{1}{4}F_1(x_2).$$

Now we find the lower bound on P_B. If $B \in R$ then B is not matched if it does not receive a request from one of its neighbors or when it receives, it receives more than one request. As in previous case

$$\begin{aligned}\Pr\left(M_B^c \mid B \in R\right) &= \Pr\left(\text{All neighbors of B are either } \textit{right} \text{ or do not request B}\right) \\ &\quad + \sum_{j=2}^{x_2} \binom{x_2}{j}\left(\frac{1}{2}\right)^{x_2} \Pr\left(M_B^c \mid B \in R, |L_{E(B)}| = j\right).\end{aligned}$$

This implies that

$$\begin{aligned}\Pr\left(M_B^c \mid B \in R\right) &\leq \frac{1}{2}\left(1 - \frac{1}{2\,d^*}\right)^{x_2-1} \\ &\quad + \sum_{j=2}^{x_2} \binom{x_2}{j}\left(\frac{1}{2}\right)^{x_2} \Pr\left(M_B^c \mid B \in R, |L_{E(B)}| = j\right).\end{aligned}$$

When j of the neighbors of B are *left* some of them request B with probabilities depending on their degrees. The worst case assumption is that each of them always requests B so that there is maximum contention. With this assumption,

$$\begin{aligned}\Pr\left(M_B^c \mid B \in R\right) &\leq \frac{1}{2}\left(1 - \frac{1}{2\,d^*}\right)^{x_2-1} \\ &\quad + \sum_{j=2}^{x_2} \binom{x_2}{j}\left(\frac{1}{2}\right)^{x_2} \left(1 - \frac{j}{M}\sum_{l=1}^{M}\left(1 - \frac{l}{M}\right)^{j-1}\right)\end{aligned}$$

which implies that

$$\Pr\left(M_B^c \mid B \in R\right) \leq \frac{1}{2}\left(1 - \frac{1}{2\,d^*}\right)^{x_2-1} + F_2(x_2). \quad (17)$$

When B becomes *left*, let C be the vertex that B decides to request. Then

$$\begin{aligned}\Pr\left(M_B^c \mid B \in L\right) &= \Pr\left(C \in L\right) \mid B \in L) \\ &\quad + \Pr\left(M_B^c \mid B \in L, C \in R\right) \\ &\qquad \times \Pr\left(C \in L \mid B \in R\right) \\ &= \frac{1}{2} + \frac{1}{2}\Pr\left(M_B^c \mid B \in L, C \in R\right).\end{aligned}$$

The probability that $C = A$, i.e. , the probability that B chooses A, is $\frac{1}{x_2}$. Since $x_1 = 1$ whenever A is requested by B, A acknowledges. Thus

$$\begin{aligned}\Pr\left(M_B^C \mid B \in L\right) &= \frac{1}{2} + \\ &\quad \frac{x_2 - 1}{2x_2}\Pr\left(M_B^c \mid B \in L, C \in R, C \neq A\right).\end{aligned}$$

Since $d(C) \leq d^*$, the worst case assumption is that $d(C) = d^*$ because the higher the value of $d(C)$, lower is the probability that the time-slot chosen by B is the first one and unique. Thus

$$\Pr\left(M_B^c \mid B \in L\right) \leq \frac{1}{2} + \frac{x_2 - 1}{2x_2}F_1(d^*). \quad (18)$$

Thus using (17) and (18) we get

$$P_B \geq \frac{3}{4} - \frac{1}{4}\left(1 - \frac{1}{2\,d^*}\right)^{x_2-1} - \frac{1}{2}F_2(x_2) - \frac{x_2 - 1}{4x_2}F_1(d^*).$$

Case 3 : $x_1, x_2 \geq 2$

Similar to the expression for P_B in the previous section we have

$$\Pr\left(M_B^c \mid B \in R\right) \leq \left(1 - \frac{1}{2\,x_1}\right)\left(1 - \frac{1}{2\,d^*}\right)^{x_2-1} + F_2(x_2)$$

The extra factor appears because $x_1 \neq 1$ and the probability that A is either *right* or does not request B is $\left(1 - \frac{1}{2\,x_1}\right)$ rather than $\frac{1}{2}$. Similarly we have

$$\Pr\left(M_B^c \mid B \in L\right) \leq \frac{1}{2} + \frac{x_2 - 1}{2x_2}F_1(d^*) + \frac{1}{2x_2}F_1(x_1).$$

The extra term appears because even if B requests A, it might not be matched because A has other neighbors that could be requesting it too. Thus we have

$$\begin{aligned}P_B &\geq \frac{3}{4} - \frac{1}{4}\left(1 - \frac{1}{2x_1}\right)\left(1 - \frac{1}{2d^*}\right)^{x_2-1} - \frac{1}{2}F_2(x_2) \\ &\quad - \left(\frac{x_2 - 1}{4x_2}\right)F_1(d^*) - \frac{1}{4x_2}F_1(x_1).\end{aligned}$$

Similarly

$$\begin{aligned}P_A &\geq \frac{3}{4} - \frac{1}{4}\left(1 - \frac{1}{2x_2}\right)\left(1 - \frac{1}{2d^*}\right)^{x_1-1} - \frac{1}{2}F_2(x_1) \\ &\quad - \left(\frac{x_1 - 1}{4x_1}\right)F_1(d^*) - \frac{1}{4x_1}F_1(x_2).\end{aligned}$$

B. Analysis

The way we have defined $F_1(x)$ it is not defined for $x = 1$. Similarly $F_2(x)$ is not defined for $x = 1$ too. Let us define $\widetilde{F_1}(x)$ and $\widetilde{F_2}(x)$ as

$$\widetilde{F_1}(x) = \sum_{j=0}^{x-1} \binom{x-1}{j} \left(\frac{1}{2}\right)^{x-1} \left(1 - \frac{1}{M}\sum_{l=1}^{M}\left(1 - \frac{l}{M}\right)^j\right)$$

and

$$\widetilde{F_2}(x) = \sum_{j=1}^{x} \binom{x}{j} \left(\frac{1}{2}\right)^{x} \left(1 - \frac{j}{M}\sum_{l=1}^{M}\left(1 - \frac{l}{M}\right)^{j-1}\right).$$

With this definition $\widetilde{F_1}(x) = F_1(x)$ when $x \geq 2$ and $\widetilde{F_1}(1) = 0$ and likewise for $\widetilde{F_2}(x)$. Let

$$\begin{aligned}\widetilde{P_A} = \frac{3}{4} - \frac{1}{4}\left(1 - \frac{1}{2\,x_2}\right)\left(1 - \frac{1}{2\,d^*}\right)^{x_1-1} \\ - \frac{1}{2}\widetilde{F_2}(x_1) - \frac{x_1-1}{4x_1}\widetilde{F_1}(d^*) - \frac{1}{4x_1}\widetilde{F_1}(x_2)\end{aligned}$$

and

$$\begin{aligned}\widetilde{P_B} = \frac{3}{4} - \frac{1}{4}\left(1 - \frac{1}{2\,x_1}\right)\left(1 - \frac{1}{2\,d^*}\right)^{x_2-1} \\ - \frac{1}{2}\widetilde{F_2}(x_2) - \frac{x_2-1}{4x_2}\widetilde{F_1}(d^*) - \frac{1}{4x_2}\widetilde{F_1}(x_1).\end{aligned}$$

Then $P_A \geq \widetilde{P_A}$ and $P_B \geq \widetilde{P_B}$. Since $P_{\mathcal{E}_{AB}} \geq \max(P_A, P_B)$ we get

$$P_{\mathcal{E}_{AB}} \geq \max(\widetilde{P_A}, \widetilde{P_B}).$$

Thus for any edge AB we have

$$P_{\mathcal{E}_{AB}} \geq \min_{x_1, x_2 \in [1, d^*]} \max(\widetilde{P_A}, \widetilde{P_B}).$$

Let

$$p^* = \min_{x_1, x_2 \in [1, d^*]} \max(\widetilde{P_A}, \widetilde{P_B}),$$

then $P_{\mathcal{E}_{AB}} \geq p^*$. The bound is useful as long as $p^* > 0$, which we can ensure by choosing M sufficiently large. For a given value of d^* and M, p^* is a constant and thus K stages where

$$K = \min_{K \geq 1}\{k : (1 - p^*)^k \leq 1 - \kappa\}$$

will guarantee that with a probability greater than κ each link or one of its neighbors is matched. Note that K is also a constant, depending only on d^* and M. It is independent of the network size and of the network topology as long as the maximum degree requirement is satisfied. ■

REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] X. Lin and N. B. Shroff, "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.

[6] 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.

[7] A. Eryilmaz and R. Srikant, "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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," to appear in The Proceedings of IEEE INFOCOM, May 2007.