Throughput and Fairness Guarantees Through Maximal Scheduling in Wireless Networks

Prasanna Chaporkar, Koushik Kar, *Member, IEEE*, Xiang Luo, and Saswati Sarkar, *Member, IEEE*

***Abstract*—The question of providing throughput guarantees through distributed scheduling, which has remained an open problem for some time, is addressed in this paper. It is shown that a simple distributed scheduling strategy, *maximal scheduling*, attains a guaranteed fraction of the maximum throughput region in arbitrary wireless networks. The guaranteed fraction depends on the "interference degree" of the network, which is the maximum number of transmitter–receiver pairs that interfere with any given transmitter–receiver pair in the network and do not interfere with each other. Depending on the nature of communication, the transmission powers and the propagation models, the guaranteed fraction can be lower-bounded by the maximum link degrees in the underlying topology, or even by constants that are independent of the topology. The guarantees are tight in that they cannot be improved any further with maximal scheduling. The results can be generalized to end-to-end multihop sessions. Finally, enhancements to maximal scheduling that can guarantee fairness of rate allocation among different sessions, are discussed.**

***Index Terms*—Fairness guarantees, maximal scheduling, throughput guarantees, wireless networks.**

I. Introduction

MAXIMIZING the network throughput by appropriately scheduling sessions is a key design goal in wireless networks. Tassiulas *et al.* characterized the maximum attainable throughput region and also provided a scheduling strategy that attains this throughput region in any given wireless network [21]. The policy, however, is centralized and can have exponential complexity depending on the network topology considered. Later, Tassiulas [20] and Shah *et al.* [18] provided linear complexity randomized scheduling schemes that attain the maximum achievable throughput region; both scheduling strategies, however, require centralized control.

Manuscript received February 19, 2006; revised April 28, 2007. This work was supported by the National Science Foundation under Grants NCR-0238340, CNS-0435306, CNS-0448316 and CNS-0435141. The material in this paper was presented in part at the 43rd Annu. Allerton Conf. Communication, Control and Computing, Monticello, IL, September 2005; the Infomation Theory and Applications, Inaugural Workshop, San Diego, CA, February 2006; and the 4th Workshop on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Boston, MA, April 2006.

P. Chaporkar is with the Department of Electrical Engineering, Indian Institute of Technology, Powai, Mumbai, India 400 076 (e-mail: chaporkar@ee.iitb.ac.in).

K. Kar and X. Luo are with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy NY 12180 USA (e-mail: koushik@ecse.rpi.edu; luox3@rpi.edu).

S. Sarkar is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19107 USA (e-mail: swati@seas.upenn.edu).

Communicated by E. Modiano, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2007.913537

Designing a distributed scheduling policy that attains the throughput region in wireless networks has remained elusive. Recently, Lin *et al.* [10] proved that a distributed maximal matching scheduling strategy is guaranteed to attain at least half of this region for the node-exclusive spectrum sharing model. In the node-exclusive spectrum sharing model, only scheduling constraint is that a node cannot communicate with multiple nodes simultaneously. This specific interference model holds only when every node has a unique frequency in its two-hop neighborhood.

Different wireless networks have significantly different interference constraints. Bluetooth networks satisfy the node-exclusive spectrum sharing model. On the other hand, IEEE 802.11 networks have limited number of frequencies that may not permit the allocation of unique frequencies in a two-hop neighborhood. Furthermore, the interference regions of nodes involved in transmissions may vary widely depending on the signal propagation conditions, and may be different for different transmitter–receiver pairs. A basic question that remains open is whether a distributed scheduling strategy can attain a guaranteed fraction of the maximum achievable throughput region for arbitrary interference models. Our investigation takes a step forward in solving this open problem.

Our contribution is to characterize the maximum-throughput region attained by a distributed scheduling strategy under arbitrary topologies and interference models. The simple scheduling policy we consider, referred to as *maximal scheduling*, only ensures that if a transmitter u has a packet to transmit to a receiver v, either (u, v) or a transmitter–receiver pair that cannot simultaneously transmit with (u, v) is scheduled for transmission; the scheduling is otherwise arbitrary. Our investigation of this maximal scheduling policy has been motivated by the following observations. In the specific node-exclusive spectrum sharing model, the maximal scheduling policy becomes the maximal matching policy considered by Lin *et al.*, and is therefore guaranteed to attain at least half of the maximum throughput region [10]. Dai *et al.* [7] has also obtained a similar guarantee for the maximal matching policy in input-queued switches where the scheduling constraints are similar to that in the node-exclusive spectrum sharing model. Last but not the least, the simplicity and localized nature of maximal scheduling imply that it can be readily implemented in a distributed manner with low overhead and computation cost. Using the randomized distributed algorithm described in [11], a maximal schedule can be computed in $O(\log^2 n)$ communication rounds, where n represents the number of nodes in the network and a communication round involves message exchanges by each node with its two-hop neighbors. It is therefore interesting and important to examine

whether maximal scheduling can provide any throughput guarantee under arbitrary interference models and topologies.

Towards this goal, we characterize the fraction of the maximum throughput region attained by maximal scheduling in any given topology and interference model. Let $K(\mathcal{N})$ be the maximum interference degree in an arbitrary wireless network $\mathcal{N}$, where the "interference degree" of any transmitter–receiver pair (u, v) is the maximum number of transmitter–receiver pairs that interfere with (u, v) but do not interfere with each other. We prove that maximal scheduling is guaranteed to attain at least $1/K(\mathcal{N})$ of the maximum throughput region in the given network $\mathcal{N}$. Also, there exists an arrival process in the given network $\mathcal{N}$ for which maximal scheduling will attain at most $1/K(\mathcal{N})$ of the maximum-throughput region. Given a network, the maximum interference degree may be computed using geometric or graph-theoretic techniques. These results therefore allow us to obtain performance guarantees for maximal scheduling for arbitrary node locations, propagation conditions, interference models, and channel allocations.

We argue that the maximum throughput region attained by maximal scheduling is significantly different for different interference models. We first consider a "bidirectional equal power" interference model in which the network has a single frequency, and all communications use the same power and involve bidirectional message exchanges (e.g., RTS, CTS, data, ACK exchanges in IEEE 802.11). Using a combination of Lyapunov theory and geometric packing, we prove that in this interference model, maximal scheduling is guaranteed to attain at least $1/8$th of the maximum throughput region. This result therefore guarantees that as in the node-exclusive spectrum sharing model, a distributed scheduling can attain a constant fraction of the maximum throughput region in this case as well. Furthermore, we show that the guarantee cannot be improved any further in this case as there exists topologies for which maximal scheduling will attain at most $1/8$th of the maximum throughput region. We then consider a "unidirectional equal power" interference model in which all communications involve unidirectional message exchanges. The network still has a single frequency and all communications use the same power. In this case, however, the performance of maximal scheduling can become arbitrarily bad. More precisely, given any constant Z, there exist topologies in which maximal scheduling will attain less than $1/Z$ of the maximum throughput region. On the other extreme, as discussed before, in the node-exclusive spectrum sharing model, maximal scheduling is guaranteed to attain at least half of the maximum throughput region [10]. We also demonstrate that in this case there exist topologies in which maximal scheduling, and hence maximal matching, will attain at most $1/2$ of the maximum throughput region.

The comparisons between the throughput region of maximal scheduling and the maximum possible throughput region of the network characterize the penalty due to the use of only local information in the scheduling. The characterizations of the throughput region of maximal scheduling discussed above bound the performance of the network in terms of that of the worst transmitter–receiver pair. The natural next question to ask is whether it is possible to obtain better nonuniform bounds by considering the constraints of individual sessions. We prove that under maximal scheduling the performance of each transmitter–receiver pair can be characterized by the interference degrees of itself and its neighbors. Our results can be nicely generalized to multihop sessions, where the performance penalty for each session, due to the use of local information based scheduling, depends only on the interference degree of the links in its path and their neighbors. The result is somewhat counterintuitive, as the overall performances of sessions may depend on each other even when they are separated by several hops. Furthermore, we show that the performance penalties under maximal scheduling cannot be localized any further. Specifically, the interference degrees of the links of a session alone cannot determine its throughput guarantee.

Maximal scheduling is really a class of policies, and some policies in this class could allocate bandwidth very unfairly. Recently, Lin *et al.* [10] and Bui *et al.* [3] have shown that in the node-exclusive spectrum sharing model, maximal scheduling can be used for maximizing the network utility and congestion control. We obtain global fairness guarantees in wireless networks with arbitrary interference models using maximal scheduling. First, using the characterizations of the throughput region for maximal scheduling, we characterize the feasible set of service rate allocations for maximal scheduling, and prove that a combination of a token generation scheme together with maximal scheduling attains maxmin fairness in this feasible set. We next show that the rate vector attained by the above combination is fairer than the overall maxmin fair rate vector times the reciprocal of the maximum interference degree in the network. The token generation scheme allows each session to estimate its maxmin fair rate in a distributed manner. Sessions contend for channel access in accordance with this estimate, and the contention is resolved using maximal scheduling. The token generation and the contention resolution can be executed in parallel. The maxmin fair rates need not be computed explicitly, and no knowledge of the statistics of the packet arrival process is necessary for executing the algorithm. The computation need not restart when the topology or the arrival rates change. The scheme is therefore robust.

The paper is organized as follows. We describe the system model and the maximal scheduling policy in Section II. We then describe some specific communication and interference models in Section III. We characterize the throughput regions of maximal scheduling for arbitrary wireless networks in Section IV, and for some representative interference models in Section IV-B. In Section V, we generalize the analytical results and the framework so as to provide different throughput guarantees for different sessions, stronger notions of stability, and end-to-end performance guarantees. We describe how maximal scheduling can be enhanced so as to guarantee fairness in Section VI. We conclude in Section VII.

II. System Model

We consider scheduling at the multiple-access channel (MAC) layer in a wireless network. We assume that time is slotted. The topology in a wireless network can be modeled as a directed graph $G = (V, E)$, where V and E, respectively, denote the sets of nodes and links. A link exists from a node u to another node v if and only if v can receive u's signals. The

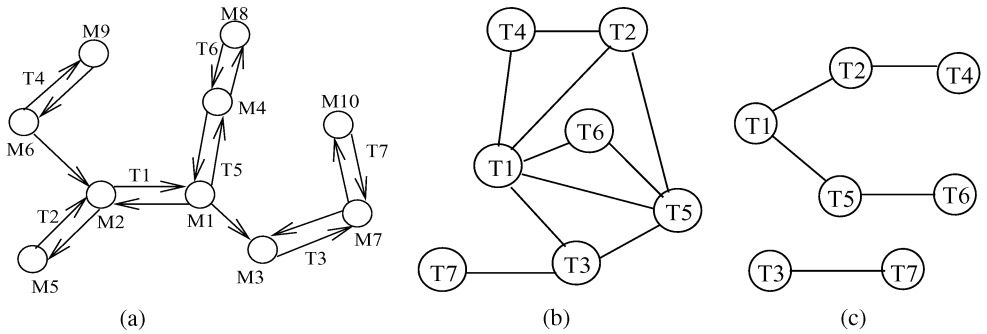

Fig. 1. Panel (a) shows a directed graph with $V = \{M1, \ldots, M10\}$. The arrows between the nodes indicate the directed links. There are seven sessions: $T1, \ldots, T7$. Nodes $M2, M5, M3, M6, M1, M8$, and $M10$ are the transmitters of sessions $T1, T2, T3, T4, T5, T6$, and $T7$, respectively. Node $M2$ has three neighbors: $M1, M5, M6$. Nodes $M1$ and $M2$ have degree 5; hence, the degrees of edges $(M1, M2)$ and $(M2, M1)$ are 10. Here, $\delta_G = 10$. Both the out-degree of $M1$ and in-degree of $M2$ are 3. Thus, the directed degree of $(M1, M2)$ is 6. Here, $\Delta_G = 6$. Sessions $T5$ and $T6$ interfere with each other, as $M4$ has a single transceiver. Panels (b) and (c) show the interference graphs for the network shown in (a) under bidirectional and unidirectional communication models, respectively. As panels (b) and (c) show, the interference sets of $T6$ are $\{T1, T5\}$ and $\{T5\}$ under the bidirectional and unidirectional communication models, respectively. (a) Network $\mathcal{N}'$, (b), (c)

link set E depends on the transmission power levels of nodes and the propagation conditions in different directions.

We now introduce terminologies that we use throughout the paper. Some of these are well known in graph theory; we mention these for completeness.

Definition 1: A node u is a *neighbor* of a node v, if there exists a link from u to v, i.e., $(u, v) \in E$.

The *degree of a node* u is the number of links in E originating from or ending at u. The *degree of a link* $e = (u, v)$ is defined as the sum of the degrees of u and v. The *maximum link degree* in G, δ_G, is the maximum degree of any link in E.

The *out-degree* of a node u is the number of links in E originating from u. The *in-degree* of a node u is the number of links in E ending at u. The *directed degree of a link* $e = (u, v)$ is defined as the sum of the out-degree of u and in-degree of v. The *maximum directed link degree* in G, Δ_G, is the maximum directed degree of any link in E.

At the MAC layer, each session traverses only one link. In the following discussion, therefore, we only consider single-hop sessions (generalization of our results to multi-hop sessions is discussed in Section V-C). We allow multiple sessions to traverse the same link. If a session i traverses link (u, v) then u and v are i's transmitter and receiver, respectively. Without loss of generality, we assume that every node in V is either the transmitter or the receiver of at least one session. If this assumption does not hold, we can consider G to be a subgraph obtained from the original topology by removing the nodes that are not the end points of sessions.

Definition 2: A session i *interferes* with session j if j cannot successfully transmit a packet when i is transmitting.

In Section III, we will describe broad classes of communication and interference models and how to obtain the pairwise interference relations in each case.

A wireless network $\mathcal{N}$ can be described by the topology $G = (V, E)$, the 3-tuple specifications of the sessions and the pairwise interference relations between them. We consider a network with N sessions.

Definition 3: The *interference set* of a session i, S_i, is the set of sessions j such that either i interferes with j or j interferes with i.

Note that if $j \in S_i$, then $i \in S_j$.

Definition 4: The *interference graph* $I^{\mathcal{N}} = (V_I^{\mathcal{N}}, E_I^{\mathcal{N}})$ of a network $\mathcal{N}$ is an undirected graph in which the vertex set $V_I^{\mathcal{N}}$ corresponds to the set of sessions in $\mathcal{N}$ and there is an edge between two vertices i and j if $j \in S_i$.

We elucidate these definitions through examples in Fig. 1.

We now describe the arrival process. We assume that at most $\alpha_{\max} > 1$ packets arrive for any session in any slot. Let $A_i(n)$ be the number of packets that session i generates in interval $(0, n], i = 1, \ldots, N$. We assume that any packet arriving in a slot arrives at the beginning of the slot, and may be transmitted in the slot. The arrival process $\{A_i(\cdot), i = 1, \ldots, N\}$ satisfies a strong law of large numbers (SLLN). Thus, there exist nonnegative real numbers $\lambda_i, i = 1, \ldots, N$ such that with probability (w.p.) 1

$$\lim_{n \to \infty} A_i(n)/n = \lambda_i, \qquad i = 1, \ldots, N. \tag{1}$$

The condition (1) on the arrival processes is mild. Several arrival processes including all jointly stationary and ergodic arrival processes satisfy (1). For simplicity, we will sometimes consider special cases of the above general model (Sections V-B, V-C, VI), and explicitly state whenever we do so.

Definition 5: The *arrival rate* of session i is λ_i, $i = 1, \ldots, N$. The *arrival rate vector* $\vec{\lambda}$ is an N-dimensional vector whose components are the arrival rates.

Definition 6: A *scheduling policy* is an algorithm that decides in each slot the subset of sessions that would transmit packets.

Clearly, a subset S of sessions can transmit packets simultaneously in a slot if no two sessions in S interfere with each other and every session in S has a packet to transmit. We assume that all the packets have the same length and one packet can be transmitted in a single slot. Thus, if a session is scheduled in a slot, it transmits a packet in the slot.

Let $D_i(n)$ be the number of packets that session i transmits in interval $(0, n], i = 1, \ldots, N$. Clearly, the transmissions depend on the scheduling policy.

Definition 7: The network is said to be *stable* if w.p. 1

$$\lim_{n\to\infty} D_i(n)/n = \lambda_i, \qquad i = 1, \ldots, N. \tag{2}$$

Thus, a network is stable if the arrival and departures rates are equal for each session.

Definition 8: The *throughput region* of a scheduling policy is the set of arrival rate vectors $\vec{\lambda}$ such that the network is stable under the policy for any arrival process that satisfies (1) and has arrival rate vector $\vec{\lambda}$.

Definition 9: An arrival rate vector $\vec{\lambda}$ is said to be *feasible* if it is in the throughput region of some scheduling policy.

Definition 10: The *maximum throughput region* Λ is the set of feasible arrival rate vectors. Note, that Λ depends on the network $\mathcal{N}$.

Example 1: Consider the network shown in Fig. 3(a). Consider a scheduling policy π_1, that serves session $(n \bmod 9) + 1$ in slot n, where "$\bmod$" is the modulo operator. Under π_1, each session $i \in \{1, \ldots, 9\}$ can transmit at the rate of at most $1/9$. Thus, the throughput region of π_1, Λ^{π_1}, is characterized as follows:

$$\Lambda^{\pi_1} = \{(\lambda_1, \ldots, \lambda_9) \ : \ \lambda_i \leq 1/9 \, \forall i\}.$$

In this case, since the only scheduling constraint is that session 1 cannot be scheduled simultaneously with any of the sessions $2, 3, \ldots, 9$, the maximum throughput region Λ is given by

$$\Lambda = \left\{(\lambda_1, \ldots, \lambda_9) \ : \ \lambda_1 + \max_{2\leq i\leq 9}\{\lambda_i\} \leq 1\right\}.$$

Therefore, in this example, scheduling policy π_1 achieves only a small fraction of the maximum throughput region.

We now describe the "maximal scheduling" policy we consider. This policy schedules a subset S of sessions such that i) every session in S has a packet to transmit, ii) no session in S interferes with any other session in S, iii) if a session i has a packet to transmit, then either i or a session in S_i, is included in S. Clearly, many subsets of sessions satisfy the above criteria in each slot, e.g., in Fig. 1(b), $\{T1, T7\}, \{T2, T3, T6\}$ satisfy the above criteria in any slot in which all sessions have packets to transmit. Maximal scheduling can select any such subset. If each session knows its interference set, maximal scheduling can be implemented in a distributed manner using standard algorithms [13]. In most cases of practical interest, sessions can determine their interference sets using local message exchange.

III. Interference Models

The pairwise interference relations between the sessions depend on topology $G = (V, E)$ and the nature of communication. The topology G is determined by the transmission powers, propagation conditions, and node locations. Communication can either be bidirectional or unidirectional. In the former, when a session is scheduled, both the transmitter and the receiver transmit sequentially. For example, the transmitter may transmit data and control messages while the receiver may transmit control messages. Such bidirectional communications occur in IEEE 802.11. Thus, there must be links in both directions between a session's transmitter and receiver. In unidirectional communication, when a session is scheduled, it transmits packets from only the transmitter to the receiver. For example, unidirectional communication occurs in IEEE 802.11 when control messages are disabled (e.g., in broadcast mode).

We assume that each node has a single transceiver. Thus, a node can be involved in at most one transmission. In other words, sessions that have a node in common interfere with each other. We initially assume that all transmissions use the same frequency. Thus, node v cannot receive any packet successfully if more than one of its neighbors are transmitting simultaneously (we do not assume capture). Thus, a transmission on link $(u, v) \in E$ is successful in a slot if and only if no neighbor of v other than u transmits in the slot. For example, in Fig. 1(a), transmission along $(M5, M2)$ is successful if $M1$ and $M6$ do not transmit. For bidirectional communication, when a session (i, u, v) is scheduled, transmissions proceed along both (u, v) and (v, u). For unidirectional communication, when a session (i, u, v) is scheduled, transmissions proceed only along (u, v). The above constraints provide the interference relations for both the bidirectional and unidirectional communication models.

In the *bidirectional communication model*, a session i interferes with session j if i and j have a common endpoint, or one endpoint (transmitter or receiver) of j is a neighbor of an end point of i. For example, in Fig. 1(a), $T1, T5, T7$ interfere with $T3$. This is also clearly evident from Fig. 1(b). In the *unidirectional communication model*, session i interferes with session j if i and j have a common endpoint, or j's receiver is a neighbor of i's transmitter. For example, in Fig. 1(a), only $T7$ interferes with $T3$. Observe that the interference relations may be asymmetric, i.e., i may interfere with j but j may not interfere with i. For example, under the bidirectional communication model, in Fig. 1(a), $T1$ interferes with $T3$ but $T3$ does not interfere with $T1$.

We now describe several important special cases. First, assume that the propagation conditions are identical in all directions. Each node transmits at a fixed power level which can be different for different nodes. The power level of a node u determines its transmission range, and all nodes within u's transmission range receive u's signal. Thus, the link set E has the following structure: a link exists from u to v if and only if the distance between u and v is less than or equal to u's transmission range. In the bidirectional communication model, session i interferes with session j if one endpoint of j is within the transmission range of an endpoint of i. In the unidirectional communication model, session i interferes with session j if j's receiver is within the transmission range of i's transmitter.

Let us further assume that all nodes transmit at the same power. Thus, all nodes have the same transmission range d which is determined by the transmission power. Now, the link set E has the following structure: a link exists from u to v if and only if the distance between u and v is less than d. Now, in the bidirectional communication model, a session i interferes with session j if one endpoint of j is within distance d from

an end point of i (*bidirectional equal power model*). In the unidirectional interference model, a session i interferes with session j if j's receiver is within distance d from i's transmitter (*unidirectional equal power model*). Refer to Fig. 3(a) and (b) for examples of both cases. Note that now the interference relation is symmetric in the bidirectional communication model, i.e., if node i interferes with node j, then node j also interferes with node i. However, interference relationships could still be asymmetric in the unidirectional communication model.

We also consider a scenario where the network has a large number of frequencies such that every node has a unique frequency in its two-hop neighborhood. Now, for both bidirectional and unidirectional communications, only the sessions that have a common endpoint interfere. This model arises in Bluetooth communications, and is commonly referred to as the *node-exclusive spectrum sharing model* (Fig. 4).

We observe that the pairwise interference relations are significantly different in each of the cases discussed above. There is, however, one important similarity. If session i interferes with another session j, the distance between the transmitters of i and j is at most three hops. Thus, a session can use local message exchange to determine its interference set. Hence, maximal scheduling can be implemented in distributed manner in each of these cases. But, given the significant difference between the interference relations, it is not clear how similar the performance of maximal scheduling will be in these different cases. In the next section, we first characterize the performance of maximal scheduling in arbitrary networks, and subsequently characterize the throughput regions in each of the above cases using the general results.

IV. Performance Guarantees of Maximal Scheduling

We first design a framework for characterizing the throughput region of maximal scheduling Λ^{MS} for an arbitrary wireless network (Section IV-A), and subsequently characterize the throughput regions in several special cases of interest (Section IV-B). Finally, using simulations, we evaluate the throughput regions under specific arrival patterns and some representative networks (Section IV-C).

A. Arbitrary Networks and Interference Models

We first introduce a new definition.

Definition 11: The *interference degree of a session* i is i) the maximum number of sessions in its interference set S_i that can simultaneously transmit, if S_i is nonempty and ii) 1 if S_i is empty.

The interference degrees depend on the links traversed by the sessions and the topology $G = (V, E)$ as well as the node locations, propagation conditions, and interference models. For example, in Fig. 1(b), $S_{T1} = \{T2, T3, T4, T5, T6\}$, and the largest set of sessions in S_{S1} that can simultaneously transmit is $\{T3, T4, T6\}$. Thus, the interference degree of $T1$ is 3.

Definition 12: The *interference degree of a network* $\mathcal{N}$, $K(\mathcal{N})$, is the maximum interference degree of sessions in the network.

In Fig. 1(b) and (c), the interference degrees of the network are 3 and 2, respectively. Session $T1$ has these interference degrees in both cases.

We next show that for an arbitrary wireless network and interference model the throughput region of maximal scheduling Λ^{MS} can be tightly characterized in terms of $K(\mathcal{N})$.

Theorem 1: In any wireless network $\mathcal{N}$, if $\vec{\lambda} \in \Lambda$ in $\mathcal{N}$, $\vec{\lambda}/K(\mathcal{N}) \in \Lambda^{\mathrm{MS}}$ in $\mathcal{N}$.

Before providing a formal proof of Theorem 1, we describe the intuition behind it. From (2), under some scheduling policy, the packet arrival rate λ_j for each session j equals j's departure rate. Thus, for each session i, the sum of its arrival rate and the arrival rates of the sessions in its interference set S_i must equal the sum of the corresponding departure rates. Clearly, for each i at most $K(\mathcal{N})$ sessions in $\{i\} \cup S_i$ can simultaneously transmit packets in any slot. Thus, the sum of the departure rates of sessions in $\{i\} \cup S_i$, and hence the sum of the corresponding arrival rates, is at most $K(\mathcal{N})$. Thus, when the arrival rate vector is $\vec{\lambda}/K(\mathcal{N})$ instead of $\vec{\lambda}$, the sum of the arrival rates of sessions in $\{i\} \cup S_i$ is at most 1. Let the arrival rate vector be $\vec{\lambda}/K(\mathcal{N})$, and let maximal scheduling be used. For any session i, maximal scheduling always serves one packet from $\{i\} \cup S_i$ in any slot in which i has a packet to transmit. Thus, whenever i has a packet to transmit, the sum of the departure rates for these sessions is 1, which is greater than or equal to the sum of the arrival rates of these sessions. Now, since the departure rate of any session cannot exceed its arrival rate, for all i, the sum of the departure rates from the sessions in $\{i\} \cup S_i$ equals the sum of the corresponding arrival rates. It follows that the departure rate of each session i equals i's arrival rate. Thus, the system is stable. Hence $\vec{\lambda}/K(\mathcal{N}) \in \Lambda^{\mathrm{MS}}$. The proof of Theorem 1 is provided next.

Proof: We prove Theorem 1 using the Lemmas 1 and 2, stated below.

Lemma 1: Let $\vec{\lambda} \in \Lambda$. Then, $\sum_{j \in S_i \cup \{i\}} \lambda_j \leq K(\mathcal{N})$ for all sessions $i = 1, \ldots, N$.

Proof: We assume that there exists a session i such that

$$\sum_{j \in S_i \cup \{i\}} \lambda_j > K(\mathcal{N}) \tag{3}$$

and show that $\vec{\lambda} \notin \Lambda$.

Consider an arbitrary scheduling policy π. Under $\pi, \sum_{j \in S_i \cup \{i\}} D_j(n) \leq nK(\mathcal{N})$ for every $n \geq 0$ as at most $K(\mathcal{N})$ nodes among $S_i \cup \{i\}$ can be scheduled concurrently . Thus

$$\begin{aligned}
& \liminf_{n \to \infty} \sum_{j \in S_i \cup \{i\}} \frac{D_j(n)}{n} \leq K(\mathcal{N}) \\
\Rightarrow \quad & \sum_{j \in S_i \cup \{i\}} \liminf_{n \to \infty} \frac{D_j(n)}{n} \leq K(\mathcal{N}) \\
& < \sum_{j \in S_i \cup \{i\}} \lambda_j \quad \text{(from (3))} \\
\Rightarrow \quad & \liminf_{n \to \infty} \frac{D_j(n)}{n} < \lambda_j, \text{ for some } j \in S_i \cup \{i\}.
\end{aligned}$$

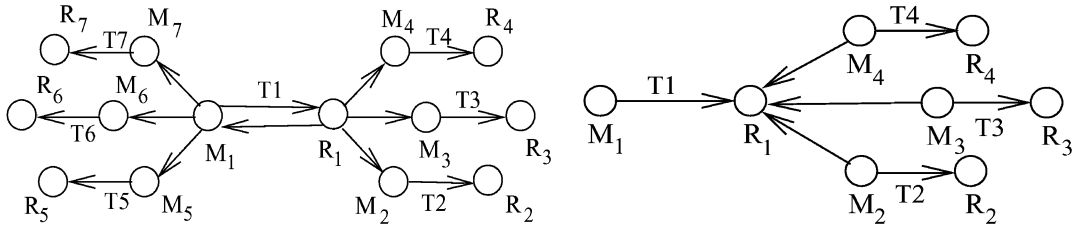

Fig. 2. Part (a) shows a network $\mathcal{N}_1'$ with bidirectional communication model and seven sessions: $(T1, M_1, R_1), \ldots, (T7, M_7, R_7)$. Session $T1$ interferes with all the remaining sessions, and none of the remaining sessions interferes with each other. Thus, $K(\mathcal{N}_1') = 6$. The degree of (M_1, R_1) is 10, which is also equal to δ_G. Thus, $K(\mathcal{N}_1') = \delta_G - 4 = \max(\delta_G - 4, 1)$. Part (b) shows a network $\mathcal{N}_2'$ with unidirectional communication model and four sessions: $(T1, M_1, R_1), \ldots, (T4, M_4, R_4)$. Sessions $T2, T3$, and $T4$ interfere with $T1$, but not with each other. Thus, $K(\mathcal{N}_2') = 3$. The directed degree of (M_1, R_1) is 5, which is also equal to Δ_G. Thus, $K(\mathcal{N}_2') = \Delta_G - 2 = \max(\Delta_G - 2, 1)$. In both graphs, arrows indicate directed links between the nodes. (a) Network $\mathcal{N'}_1$, (b) Network $\mathcal{N'}_2$.

Thus, if $\lim_{n\to\infty} \frac{D_j(n)}{n}$ exists, then its value is less than λ_j. Therefore, the network is not stable under π. Alternatively, if the limit does not exist, then also the network is not stable under π. Thus, $\vec{\lambda} \notin \Lambda$. The result follows. □

Lemma 2: Let

$$\vec{\lambda} \in \{\vec{\lambda} : \text{if } \lambda_i > 0, \sum_{j \in S_i \cup \{i\}} \lambda_j \leq 1,\ i = 1, \ldots, N\}.$$

Then $\vec{\lambda} \in \Lambda^{\mathrm{MS}}$.

The Proof of Lemma 2 is rather long, and is therefore provided in Appendix I.A.

Theorem 1 follows from Lemmas 1 and 2. □

Next we prove a result which shows that the characterization of Λ^{MS} provided by Theorem 1 is tight.

Theorem 2: Consider an arbitrary wireless network $\mathcal{N}$ and a constant Z such that $Z < K(\mathcal{N})$. There exists an arrival rate vector $\vec{\lambda}$ such that $\vec{\lambda} \in \Lambda$ in $\mathcal{N}$, but $\vec{\lambda}/Z \notin \Lambda^{\mathrm{MS}}$ in $\mathcal{N}$.

Proof: Consider an arbitrary network $\mathcal{N}$ with interference degree $K(\mathcal{N})$. By Definition 11, there exists an i such that the interference degree of session i is $K(\mathcal{N})$. Consider sessions $j_1, \ldots, j_{K(\mathcal{N})} \in S_i$ such that they are pairwise noninterfering. Now, consider the following arrival rate vector $\vec{\lambda}$: $\lambda_j = Z/K(\mathcal{N})$ if $j \in \{j_1, \ldots, j_{K(\mathcal{N})}\}$, and $\lambda_j = (K(\mathcal{N}) - Z)/K(\mathcal{N})$ if $j = i$, and $\lambda_j = 0$ otherwise. Thus, effectively the network consists only of sessions i and $j_1, \ldots, j_{K(\mathcal{N})}$. Note that since $1 \leq Z < K(\mathcal{N}), \lambda_j > 0$ for every $j \in \{i, j_1, \ldots, j_{K(\mathcal{N})}\}$. Now, consider a scheduling policy π that schedules i w.p. $(K(\mathcal{N}) - Z)/K(\mathcal{N})$ and sessions $j_1, \ldots, j_{K(\mathcal{N})}$ concurrently in the remaining slots. Clearly, the network is stable under π. Thus, $\vec{\lambda} \in \Lambda$.

Now, consider the arrival rate vector $\vec{\lambda}/Z$ and the following arrival pattern. A packet corresponding to session j_u arrives in slots n if $u = n \bmod K(\mathcal{N}) + 1$, where "mod" is the modulo operator. In every slot, a packet corresponding to session i arrives w.p. $\frac{K(\mathcal{N}) - Z}{ZK(\mathcal{N})}$. Clearly, the arrivals are in accordance with $\vec{\lambda}/Z$. Let maximal scheduling schedule i only when none of the sessions in S_i have a packet to transmit. Note that under maximal scheduling and the described arrival pattern, j_u is scheduled in slot n such that $u = n \bmod K(\mathcal{N}) + 1$, and thus i is never scheduled. Since $\lambda_i/Z7gt; 0, i$ is not stable. Thus, $\vec{\lambda}/Z \notin \Lambda^{MS}$. □

We now obtain tight bounds for $K(\mathcal{N})$ for arbitrary bidirectional and unidirectional communications models, in terms of the maximum link degrees δ_G and Δ_G in the underlying topology G. These bounds and the resulting characterizations of Λ^{MS} hold even when transmission powers differ across nodes, and propagation conditions in different directions are different.

Lemma 3: In a wireless network $\mathcal{N}$ with bidirectional communication and underlying topology $G = (V, E)$, $K(\mathcal{N}) \leq \max(\delta_G - 4, 1)$. Moreover, there exists a wireless network $\mathcal{N}_1$ with bidirectional communication and underlying topology $G = (V, E)$, such that $K(\mathcal{N}_1) = \max(\delta_G - 4, 1)$.

Proof: Consider a network $\mathcal{N}$ that has bidirectional communication and underlying topology $G = (V, E)$. Select a session i from u to v. Since we are considering bidirectional communication, $(u, v) \in E$ and $(v, u) \in E$. Note that at most one session along every link from u and v, and every link to u and v can be scheduled concurrently in the interference region of i without interfering with each other. Let $d_{(u,v)}$ denote the degree of link (u, v). Now, i's interference degree $k_i(\mathcal{N})$ satisfies the following inequality:

$$\begin{aligned} k_i(\mathcal{N}) &\leq \sum_{\substack{j \in V \\ j \neq v}} \left[\mathbf{1}_{\{(j,u) \in E\}} + \mathbf{1}_{\{(u,j) \in E\}}\right] \\ &\quad + \sum_{\substack{j \in V \\ j \neq u}} \left[\mathbf{1}_{\{(j,v) \in E\}} + \mathbf{1}_{\{(v,j) \in E\}}\right] \\ &= \sum_{j \in V} \left[\mathbf{1}_{\{(j,u) \in E\}} + \mathbf{1}_{\{(u,j) \in E\}}\right] \\ &\quad + \sum_{j \in V} \left[\mathbf{1}_{\{(j,v) \in E\}} + \mathbf{1}_{\{(v,j) \in E\}}\right] - 4 \\ &= d_{(u,v)} - 4 \\ \Rightarrow \quad \max_i \{k_i(\mathcal{N})\} &\leq \max_{(u,v) \in E} \{d_{(u,v)}\} - 4 \\ \Rightarrow \quad K(\mathcal{N}) &\leq \delta_G - 4. \end{aligned} \tag{4}$$

Now, Fig. 2(a) shows an example of a network that achieves the equality in (4). □

Lemma 4: In a wireless network $\mathcal{N}$ with unidirectional communication and underlying topology $G = (V, E)$, $K(\mathcal{N}) \leq \max(\Delta_G - 2, 1)$. Moreover, there exists a wireless network $\mathcal{N}_1$ with unidirectional communication and underlying topology $G = (V, E)$, such that $K(\mathcal{N}_1) = \max(\Delta_G - 2, 1)$.

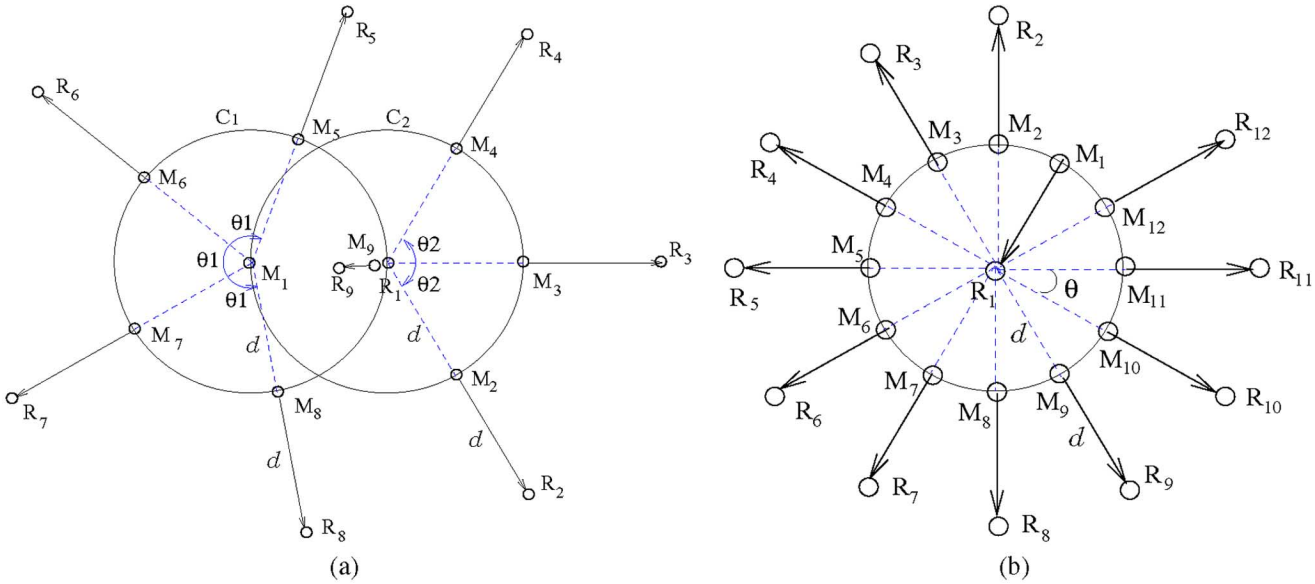

Fig. 3. Part (a) shows a network with interference constraints given by the bidirectional equal power model and transmission range d. There are nine sessions: $1, \ldots, 9$. Session i has transmitter M_i and receiver R_i. The interference area of session 1 is the union of circles C_1 and C_2. Here, $\theta_1 = 70$ deg, and $\theta_2 = 61$ deg. Distance between i) M_i and R_i is d for every $i = 1, \ldots, 8$, ii) M_9 and R_9 is $\epsilon > 0$, where ϵ is a small positive number, iii) M_1 and M_i is d for every $i = 2, \ldots, 9$, ii) M_j and M_k is greater than d for every $j, k \in \{2, \ldots, 9\}, j \neq k$ and iv) M_9 and R_1 is ϵ. *Thus, session 1 interferes with all the other eight sessions, but none of the other sessions interfere with each other*. Part (b) shows a network with interference constraints given by the unidirectional equal power model and transmission range d. There are 12 sessions: $1, \ldots, 12$. Session i has transmitter M_i and receiver R_i. The distance between M_i and R_i, and R_1 and M_i is d for every i. Thus, session 1 interferes will all the other 11 sessions, but none of the other sessions interfere with each other. We refer to sessions $2, \ldots, 12$ as noninterfering sessions. Here, θ is $\pi/6$. Note that $2\pi/\theta - 1$ noninterfering sessions can be accommodated. Thus, for any given Z, $Z+1$ noninterfering sessions can be accommodated by choosing $\theta = 2\pi/(Z+2)$.

The proof for the first part of Lemma 4 is similar to that for the first part of Lemma 4. Now, Fig. 2(b) shows a network where $K(\mathcal{N}_1) = \max(\Delta_G - 2, 1)$.

B. Specific Interference Models

We characterize $K(\mathcal{N})$ for some representative interference models. These characterizations together with Theorems 1 and 2 characterize the throughput regions of maximal scheduling for these models.

Lemma 5:

1) For the bidirectional equal power model, $K(\mathcal{N}) \leq 8$ for any network $\mathcal{N}$, and there exists a network $\mathcal{N}$ such that $K(\mathcal{N}) = 8$.
2) For the unidirectional equal power model, given any constant Z, there exists a network $\mathcal{N}$ such that $K(\mathcal{N}) > Z$.
3) For the node-exclusive spectrum sharing model, $K(\mathcal{N}) \leq 2$ for any network $\mathcal{N}$, and there exists a network $\mathcal{N}$ such that $K(\mathcal{N}) = 2$.

We prove that $K(\mathcal{N}) \leq 8$ for the bidirectional equal power model in Appendix I.B. We present the intuition behind the result here. From the interference constraints, for any i, at least one endpoint of each session in S_i must be within a distance d (transmission radius) from either i's transmitter or i's receiver. Also, the distance between i's transmitter and receiver is at most d. Thus, at least one endpoint of each session in S_i must be in the union of two circles of radius d and centered around i's transmitter and receiver respectively (Fig. 3(a)). We refer to the area in this union as i's *interference area*. We prove using geometric arguments that at most eight points can be present in this interference area such that the distance between any two points exceeds d. Clearly, if sessions j and k need to simultaneously transmit packets, the distance between an endpoint of j and an endpoint of k must exceed d. The result follows. It is worth noting that several results on packing of unit disk graphs in the existing literature show that $K(\mathcal{N})$ must be upper-bounded by a constant in the bidirectional equal power model. In particular, extending results in [9] with arguments used in the proof of Lemma 1 of [1] show that $K(\mathcal{N})$ cannot exceed 12. Furthermore, results in [12] imply that $K(\mathcal{N})$ must be upper-bounded by 9. Note that the existing results are closely related to the packing of unit disks within a *circular* region. The interference region of a session is formed by the union of two disks and may not therefore be circular; upper bounding it by a circular region and using existing results (as done in [1]) leads to a loose bound in this case. We therefore use structural properties of the area formed by the union of two closely located disks to obtain a better upper bound on $K(\mathcal{N})$, namely 8, which turns out to be tight.

We next prove the rest of Lemma 5.

Proof: Consider the bidirectional equal power model. We prove that $K(\mathcal{N}) \leq 8$ in Appendix I.B. Fig. 3(a) shows a network $\mathcal{N}$ with bidirectional equal power model such that $K(\mathcal{N}) = 8$.

Consider the unidirectional equal power model and any constant Z. In the network $\mathcal{N}$ of Fig. 3(b), for $\theta < 2\pi/(Z+2)$, $K(\mathcal{N}) > Z$ under unidirectional equal power model.

Consider the node-exclusive spectrum sharing model, and a session (i, u, v). Any session j in S_i must traverse either u or

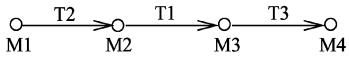

Fig. 4. This figure shows a network with four nodes $M1, \ldots, M4$ and three sessions $T1, T2$, and $T3$. Under node-exclusive spectrum sharing model, $T1$ interferes with both $T2, T3$, but $T2$ and $T3$ do not interfere with each other.

v. Thus, if $|S_i| \geq 3$, then at most two of any three sessions in S_i must traverse the same node, and hence must interfere. Thus, $K(\mathcal{N}) \leq 2$. Fig. 4 shows an example of a network $\mathcal{N}$ under node-exclusive spectrum sharing model with $K(\mathcal{N}) = 2$. The lemma follows. □

We now describe the significance of the above results. For the bidirectional equal power model, it follows from part 1) of Lemma 5 and Theorems 1, 2 that a) if $\vec{\lambda} \in \Lambda, \vec{\lambda}/8 \in \Lambda^{\mathrm{MS}}$, and b) for any constant $Z < 8$, there exists a network $\mathcal{N}$ and an arrival rate vector $\vec{\lambda}$, such that $\vec{\lambda} \in \Lambda$ in $\mathcal{N}$, but $\vec{\lambda}/Z \notin \Lambda^{\mathrm{MS}}$ in $\mathcal{N}$. Thus, Λ^{MS} is $1/8$th of the maximum throughput region Λ in this case.

For the unidirectional equal power model, it follows from part 2) of Lemma 5 and Theorem 2, that for any positive constant Z, there exists a network $\mathcal{N}$, an arrival rate vector $\vec{\lambda}$, such that $\vec{\lambda} \in \Lambda$ in $\mathcal{N}$, but $\vec{\lambda}/Z \notin \Lambda^{\mathrm{MS}}$ in $\mathcal{N}$. Thus, maximal scheduling cannot attain any constant fraction (however small) of the maximum throughput region.

Next note that for the node-exclusive spectrum sharing model, maximal scheduling is the same as maximal matching. Lin *et al.* [10] have proved that maximal matching attains at least $1/2$ the maximum throughput region in this model. This result also follows from part 3) of Lemma 5 and Theorem 1. In addition, part 3) of Lemma 5 and Theorem 2 show that this characterization is tight. Specifically, for the node-exclusive spectrum sharing model, for any positive constant Z such that $Z < 2$, there exists a network and an arrival rate vector $\vec{\lambda}$, such that $\vec{\lambda} \in \Lambda$ in $\mathcal{N}$, but $\vec{\lambda}/Z \notin \Lambda^{\mathrm{MS}}$ in $\mathcal{N}$. It is worth noting here that in the context of input-queued switches, Chuang *et al.* [6] have proved a result that is related to (although significantly different from) part 3) of Lemma 5. More precisely, the authors in [6] show that for an $N \times N$ input-queued switch, a speedup of $2 - \frac{1}{N}$ is necessary to emulate an output-queued switch with first-in first-out (FIFO) scheduling discipline.

Thus, the performance guarantees for maximal scheduling will critically depend on the interference relations, and slight changes in interference conditions can significantly alter the guarantees.

C. Numerical Results

We showed that the lower bound for the throughput region of maximal scheduling presented in Theorem 1 is tight (Theorem 2) by considering specific topologies, specific traffic patterns, and specific scheduling policies within the class of maximal scheduling policies. Using representative simulation results, we now demonstrate that the performance attained by maximal scheduling with respect to the throughput-optimal policy is usually significantly better than this bound, particularly in the presence of randomness in the packet arrival process and the scheduling policy.

We consider two network topologies. The first network $\mathcal{N}_1$, shown in Fig. 3(a), has nine single-hop sessions $T1, T2, \ldots, T9$ and bidirectional equal power model; the interference graph of this network is shown in Fig. 5(a). The second one ($\mathcal{N}_2$) is a network with eight single-hop sessions, $T1, T2, \ldots, T8$, whose interference graph is shown in Fig. 5(b).

The packet arrival process is Bernoulli; the packet arrival rate at all sessions is the same, and equal to λ. Therefore, the throughput region is characterized by a single parameter λ^*, which corresponds to the maximum value of λ that can be supported in the network by any scheduling policy. The maximum attainable throughput per session, λ^*, in networks $\mathcal{N}_1$ and $\mathcal{N}_2$ can be computed as $(1/2)$ and $(1/3)$, respectively. The maximum throughput attained by a scheduling algorithm $\mathcal{A}$ is measured as $\lambda_{\mathcal{A}}$, the maximum value of per-session arrival rate λ that leads to bounded delays (and finite queue lengths) under $\mathcal{A}$; $\lambda_{\mathcal{A}}$ is calculated through simulations. Table I shows $\lambda_{\mathcal{A}}/\lambda^*$ for three different maximal scheduling algorithms, which are described next.

Greedy MS: In this algorithm, the sessions are picked for scheduling greedily according to a predetermined order, skipping over sessions that are not backlogged or interfere with a session that has already been chosen. For network $\mathcal{N}_1$, the sessions are chosen according to the sequence $T2, T3, \ldots$, followed by $T1$, i.e., the scheduling policy gives preference to links that correspond to the peripheral nodes in the interference graph, over the link that corresponds to the central node. Note that this is the same scheduling policy that achieved the lower bound of $(1/8)$ on the throughput guarantee attained in network topology $\mathcal{N}_1$. For network $\mathcal{N}_2$, the sessions are chosen according to the sequence $T1, T2, T3, \ldots$.

Randomized MS: Here, sessions are chosen at random, ignoring sessions that are not backlogged or interfere with a session that has already been chosen.

Distributed MS: In this case, we use the randomized distributed maximal schedule construction algorithm described in [11]. This algorithm constructs a maximal schedule in $O(\log |V|)$ communication rounds.

The results demonstrate that the throughput ratio attained by the maximal scheduling algorithms with respect to the optimum is significantly better than $1/K(\mathcal{N})$ under randomized traffic patterns and scheduling policies. For example, the distributed MS algorithm attains a throughput ratio of 0.54 and 0.9, respectively, for the two networks $\mathcal{N}_1$ and $\mathcal{N}_2$, whereas the corresponding bounds are $(1/8) = 0.125$ and $(1/3) = 0.33$, respectively. Tight performance characterization of maximal scheduling under randomness in traffic patterns and schedules is a difficult question, and remains open for future research.

V. Generalizations of Throughput Guarantees

In this section, we generalize our analytical results in several ways. First, note that the characterizations of Λ^{MS} obtained so far demonstrate that maximal scheduling does not attain the maximum throughput region of a network. This is clearly expected as maximal scheduling uses only local information and the maximum throughput region has so far only been obtained by centralized scheduling policies [21], [20]. The contribution of these results is to characterize the penalty

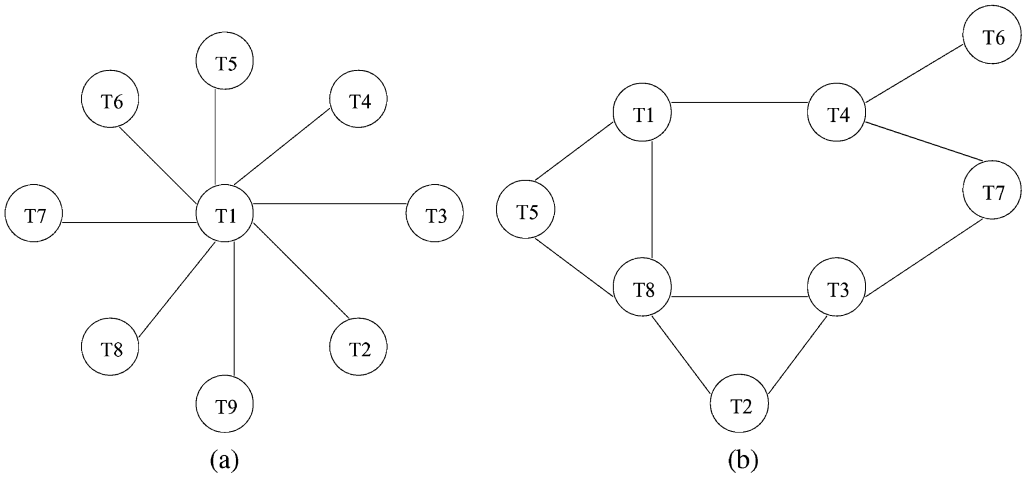

Fig. 5. Interference graphs of networks used in the simulations. (a) Interference graph of $\mathcal{N}_1$. (b) Interference graph of $\mathcal{N}_2$.

TABLE I
PERFORMANCE RATIOS OF VARIOUS MAXIMAL SCHEDULING ALGORITHMS WITH RESPECT TO THE OPTIMUM (OBTAINED THROUGH SIMULATIONS)

Algorithm $\mathcal{A}$	$\lambda_{\mathcal{A}}/\lambda^*$ for network $\mathcal{N}_1$	$\lambda_{\mathcal{A}}/\lambda^*$ for network $\mathcal{N}_2$
Greedy MS	0.38	0.75
Randomized MS	0.59	0.92
Distributed MS	0.54	0.9

due to the use of such limited information, and provide tight "uniform" bounds on the penalty in the arbitrary networks. The bounds are "uniform" because they uniformly apply to all sessions. In Section V-A, we generalize Theorems 1 and 2 to obtain better throughput guarantees for specific sessions by allowing different bounds for different sessions (Lemma 6).

We have so far considered the notion of stability which guarantees that the arrival rates of sessions equal their respective departure rates. This does not however provide guarantees on the expected queue lengths of the sessions. In Section V-B, we characterize the performance of maximal scheduling under a stronger notion of stability which guarantees that the expected queue lengths of all sessions are finite (Lemma 8).

Finally, in Section V-C, we relax the assumption that each sessions traverses only one hop, and provide throughput guarantees for maximal scheduling when sessions traverse arbitrary number of hops (Lemmas 9 and 10).

Proofs of all the results in this section are presented in the Appendix.

A. Nonuniform Bounds

In Theorems 1 and 2, the uniform bound of $1/K(\mathcal{N})$ is obtained by considering the worst session, and it is possible that for most sessions the penalty is less. We now prove that it is possible to obtain better nonuniform bounds by considering the constraints of individual sessions. Let $K_i(\mathcal{N})$ denote the interference degree of any session i in network $\mathcal{N}$ (Definition 11). We show that the performance of each session i can be characterized by its *two-hop interference degree*, $\beta_i(\mathcal{N})$, which is the maximum of the interference degrees in its neighborhood (i.e., $\beta_i(\mathcal{N}) = \max_{j \in S_i \cup \{i\}} K_j(\mathcal{N})$), but not by its interference degree $(K_i(\mathcal{N}))$ alone.

Lemma 6: If $(\lambda_1, \ldots, \lambda_N) \in \Lambda$, then

$$(\lambda_1/\beta_1(\mathcal{N}), \ldots, \lambda_N/\beta_N(\mathcal{N})) \in \Lambda^{\mathrm{MS}}.$$

Thus, due to the use of local information based scheduling, the performance of each session i decreases by a factor of $\beta_i(\mathcal{N})$; the penalty for each session therefore depends only on its two-hop neighborhood. Note that in many networks $\beta_i(\mathcal{N})$ may be significantly less than $K(\mathcal{N})$ for most sessions i (Fig. 6(a)). The following result shows that a similar characterization in terms of the single-hop neighborhood does not hold in general.

Lemma 7: There exists a wireless network $\mathcal{N}$ and an arrival rate vector $(\lambda_1, \ldots, \lambda_N)$ such that $(\lambda_1, \ldots, \lambda_N) \in \Lambda$ in $\mathcal{N}$, but

$$(\lambda_1/K_1(\mathcal{N}), \ldots, \lambda_N/K_N(\mathcal{N})) \notin \Lambda^{\mathrm{MS}}.$$

B. Stronger Notion of Stability

In this subsection, we consider a stronger notion of stability, *queue length stability*, which guarantees that the expected queue lengths of sessions are finite in stable systems. We provide guarantees on the stability region of maximal scheduling under this notion and under some stronger assumptions on the arrival process. We first mention the additional assumptions on the arrival process and formally define the notion of queue-length-stability.

Let $\alpha_j(t)$ denote the number of arrivals of session j in slot t. We assume that the arrival process $(\alpha_1(\cdot), \ldots, \alpha_N(\cdot))$ constitute an irreducible, aperiodic Markov chain with a finite number of states. We refer to this assumption as the *jointly Markovian* assumption. Note that such an arrival process satisfies (1). Let

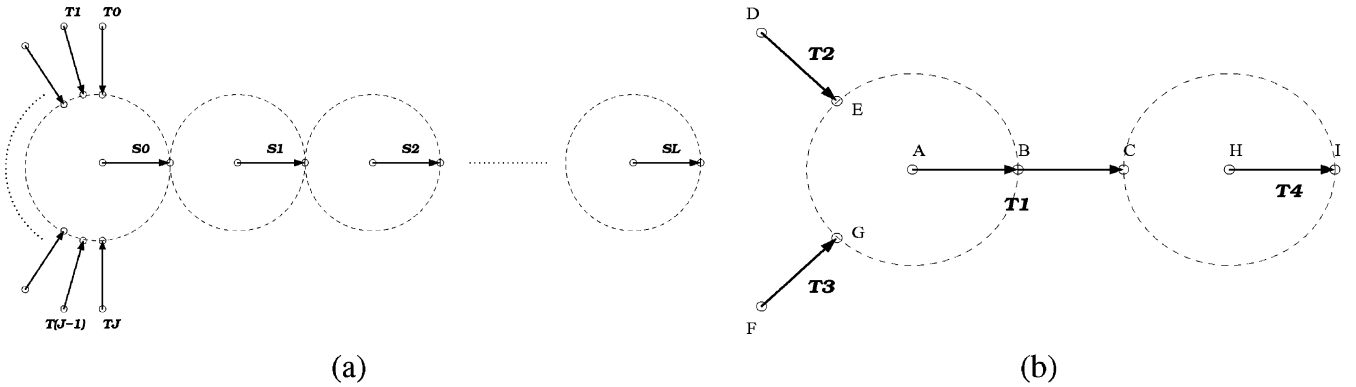

Fig. 6. In both graphs, all sessions and session links are unidirectional, and the arrows show the direction of data transfer. The circles indicate the interference regions of session-links $U0, U1, \ldots, UL$ (a) and AB and HI (b). In (a), the network consists of single-hop sessions only. Session $U0$ interferes with sessions $T0, \ldots, TJ$, whereas session Ui interferes with session $U(i-1)$, for $i = 1, 2, \ldots, L$. Thus, $K_i(\mathcal{N}) = 1$ for $i \in \{T0, \ldots, TJ, UL\}, K_i(\mathcal{N}) = 2$ for $i \in \{U1, \ldots, U(L-1)\}, K_{U0}(\mathcal{N}) = J+2, \beta_i(\mathcal{N}) = J+2$ for $i \in \{T0, \ldots, TJ, U0, U1\}$, and $\beta_i(\mathcal{N}) = 2$ for $i \in U2, \ldots, UL, K(\mathcal{N}) = (J+2)$. If J and L are large, but $L \gg J$, then K_i, β_i's for most sessions are substantially smaller than $K(\mathcal{N})$. In (b), session $T1$ consists of two session links, AB and BC, whereas sessions $T2, T3, T4$ are single-hop sessions. Session link AB interferes with session links DE (session $T2$) and FG (session $T3$) and session link HI (session $T4$) interferes with session link BC. Now, $S_{AB} = \{BC, DE, FG\}$, $S_{BC} = \{AB, HI\}$, $S_{DE} = S_{FG} = \{AB\}, S_{HI} = \{BC\}$. Thus, token-buckets at nodes A, B, D, F, H consist of token-queues corresponding to session links $\{AB, BC, DE, FG\}, \{AB, BC, HI\}, \{AB, DE\}, \{AB, FG\}$, and $\{BC, HI\}$. Thus, token-buckets associated with session link $AB(BC)$ are at nodes A, B, D, $F(A, B, H)$; these are denoted buckets $1, \ldots, 4$ of $AB(1, 2, 3$ of $BC)$. The token generation for AB at bucket 4 depends on that for AB at bucket 3 and BC at bucket 1 of BC.

$Q_i(n)$ be the number of packets waiting for transmission at the source of session i at the beginning of slot n.

Definition 13: The network is said to be *queue-length-stable* if there exists nonnegative real numbers $q_i, i = 1, \ldots, N$, such that w.p. 1

$$\lim_{n\to\infty} \sum_{m=1}^{n} Q_i(m)/n = q_i, \qquad i = 1, \ldots, N. \tag{5}$$

The *queue-length-stability region* of a scheduling policy is the set of arrival rate vectors $\vec{\lambda}$ such that the network is stable under the policy for any arrival process that satisfies the jointly Markovian assumption and has arrival rate vector $\vec{\lambda}$. The *maximum queue-length-stability region* Λ_{Q} is the union of the queue-length-stability region of all scheduling policies.

Note that if a network is queue-length-stable it is also stable, but the converse is not true. Thus, queue-length-stability is a stronger notion of stability.

We now obtain a lower-bound[1] on the queue-length-stability region of maximal scheduling $\Lambda_{\mathrm{Q}}^{\mathrm{MS}}$.

Lemma 8: Consider a jointly Markovian arrival process with the arrival rate vector $(\lambda_1^{'}, \ldots, \lambda_N^{'})$ such that $\lambda_1^{'} < \lambda_1/\beta_1(\mathcal{N}), \ldots, \lambda_N^{'} < \lambda_N/\beta_N(\mathcal{N})$, where $(\lambda_1, \ldots, \lambda_N) \in \Lambda_{\mathrm{Q}}$. Then, $(\lambda_1^{'}, \ldots, \lambda_N^{'}) \in \Lambda_{\mathrm{Q}}^{\mathrm{MS}}$.

C. Multihop Sessions

We now obtain performance guarantees for maximal scheduling when sessions traverse an arbitrary number of links. We first mention the differences from the model in Section II. The network has N end-to-end sessions, and the route of each session is assumed fixed. We allow multiple sessions to traverse the same link(s). Each session can be viewed as a collection of several hop-by-hop connections, one for each link it traverses; each of these hop-by-hop connections is called a *session-link* of the session considered. Each session-link is of the form (i, u, v), where i is an identifier for the session, u and v represent the transmitter and the receiver, respectively, of the corresponding session-link. Session-links of different sessions can be associated with the same physical link, and are distinguished by their session-identifiers (for simplicity of notation, in examples where only one session-link traverses each link we denote the session-links only by the sources and destinations of the associated links). We assume that there are a total of M session-links in the network (over all sessions), and these are indexed by $1, \ldots, M$. For any session i, let P_i denote the set of its session-links. Let $q(j)$ denote the session of session-link j, i.e., $q(j) = \{i : j \in P_i\}$.

The notions of interference, interference-set, and interference-degrees are now defined for session-links instead of sessions. Specifically, a session-link j *interferes* with session-link k if k cannot successfully transmit a packet when j is transmitting. The *interference set* of session-link j, S_j denotes the set of session-links k such that either k interferes with j or j interferes with k (Fig. 6(b)). The *interference degree* of a session-link j in network $\mathcal{N}, K_j(\mathcal{N})$ is i) the maximum number of session-links in its interference set S_j that can simultaneously transmit, if S_j is nonempty, and ii) 1, if S_j is empty. The *two-hop interference degree of session-link* j, is defined as $\beta_j(\mathcal{N}) = \max_{m\in S_j\cup\{j\}} K_m(\mathcal{N})$. The *two-hop interference degree of session* $i, \tilde{\beta}_i(\mathcal{N})$ denotes the maximum two-hop interference degree of all session-links of session i, i.e., $\tilde{\beta}_i(\mathcal{N}) = \max_{j\in P_i} \beta_j(\mathcal{N})$. The *interference degree of a network* $\mathcal{N}, K(\mathcal{N})$ is the maximum interference degree of session-links in the network.

The packet arrival and departure processes now need to be defined for session-links. Let $A_j(n)$ denote the number of arrivals for session-link j in the time interval $(0, n], j = 1, \ldots, M$. The arrival process at the first session-link of any session consists only of exogenous packets, and satisfies the SLLN as described

[1]We presented this result at the ITA workshop [5]. Wu *et al.* [25] also obtained this result independently, and presented it at the same workshop.

in (1). Thus, if F_i denotes session-link corresponding to the first link for session i, then there exists nonnegative real numbers $\lambda_i, i = 1, \ldots, N$ such that w.p. 1

$$\lim_{n\to\infty} A_{F_i}(n)/n = \lambda_i, \qquad i = 1, \ldots, N. \tag{6}$$

Now, $D_j(n)$ denotes the number of packets that session-link j transmits in interval $(0, n], j = 1, \ldots, M$. Note that if j and $j+1$ are consecutive session-links of a session, then $A_{j+1}(n) = D_j(n)$. Now, let L_i be the session-link corresponding to the last hop of session i. If for some constant d_i the limit $\lim_{n\to\infty} D_{L_i}(n)/n = d_i$ w.p. 1, then d_i is denoted as the departure rate of session i.

Definition 14: The network is said to be *stable* if there exists a departure rate vector $\vec{d} = (d_1, \ldots, d_N)$ such that w.p. 1, for each session i

$$\lim_{n\to\infty} D_{L_i}(n)/n = d_i = \lambda_i, \qquad i = 1, \ldots, N. \tag{7}$$

Thus, again a network is stable if the arrival and departures rates are equal for each session. Now, using the above definition for stability, the maximum throughput region Λ, and the throughput region for maximal scheduling, Λ^{MS} can be defined as in Section II. Note that maximal scheduling can be described similarly to that in Section II; the only difference is that session-links must now be used instead of sessions in the description.

We now provide lower bounds on Λ^{MS}, under an enhancement of maximal scheduling that has been proposed by Wu *et al.* [23], [24]. Under this enhancement, every session-link that does not originate from the source of the session has a regulator that in each slot generates a token with a probability that equals the arrival rate of the session. Every such session-link also maintains two types of queues, a *waiting queue* and a *release queue*. Packets arriving at such a session-link are initially stored in its waiting queue. Whenever the regulator generates a new token, if the waiting queue is nonempty, a packet is transferred from the waiting queue to the release queue. A session-link that originates from the source of the session maintains only the release queue, and all exogenous packets waiting for transmission are stored there. Maximal scheduling only considers the release queues of session-links for service and contention resolution. We refer to this enhancement as *regulator enhancement*.

Lemma 9: If $\vec{\lambda} \in \Lambda$, then

$$(\lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N/\tilde{\beta}_N(\mathcal{N})) \in \Lambda^{\mathrm{MS}}$$

in $\mathcal{N}$ under regulator enhancement.

Since $K(\mathcal{N}) \geq \tilde{\beta}_i(\mathcal{N}), i = 1, \ldots, N$, Lemma 9 also implies that if $\vec{\lambda} \in \Lambda$, then $\vec{\lambda}/K(\mathcal{N}) \in \Lambda^{MS}$ in $\mathcal{N}$ under regulator enhancement.

The use of regulators requires that the arrival rate for each session must be known at each session-link. We now investigate whether performance guarantees can be provided for maximal scheduling without using regulators. We consider a special case of the general arrival process described in (6). We refer to this special case as *exponentially convergent arrival processes*. We assume that there exists a constant $\hat{\alpha} > 1$ such that the empirical average of the exogenous arrivals in the system in T slots converges to $\vec{\lambda}$ at a rate faster than $\frac{1}{T^{\hat{\alpha}}}$. Mathematically, there exists $\widehat{t}_\delta$ such that for every $i \in \{1, \ldots, N\}, T \geq \widehat{t}_\delta$ and $\delta > 0$

$$\boldsymbol{P}\left\{\left|\frac{\sum_{t=1}^{T} A_{F_i}(t)}{T} - \lambda_i\right| > \delta\right\} < \frac{1}{T^{\hat{\alpha}}}. \tag{8}$$

Again, a large class of arrival processes, e.g., periodic, independent and identically distributed (i.i.d.), and positive recurrent Markovian arrival processes with finite state space, satisfy the preceding assumption. We show that, without any enhancements,[2] for exponentially convergent arrival processes, maximal scheduling attains the following weaker notion of stability. We define a random variable $B_{j,t}$ as follows. If session-link j has a packet to transmit at time t, then $B_{j,t}$ is the length of its remaining busy period, otherwise $B_{j,t} = 0$.

Lemma 10: Consider exponentially convergent arrival processes. Let the arrival rate vector $(\lambda_1^{'}, \ldots, \lambda_N^{'})$ be such that

$$\lambda_1^{'} < \lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N^{'} < \lambda_N/\tilde{\beta}_N(\mathcal{N})$$

where $(\lambda_1, \ldots, \lambda_N) \in \Lambda$. Then under maximal scheduling, the packet queue of every session-link will almost surely become empty infinitely often. Furthermore, for every session-link j and time $t, \boldsymbol{E}[B_{j,t}] < \infty$.

The above result implies that almost surely

$$\limsup_{n\to\infty} \frac{D_j(n) - A_j(n)}{n} = 0, \qquad \forall\, j = 1, \ldots, M.$$

Thus, if the arrival rate vector satisfies the condition in Lemma 10, and for each session-link, the limits of the departure and the arrival rates exist almost surely, then almost surely $\lim_{n\to\infty} D_{L_i}(n)/n = \lambda_i \;\forall\, i = 1, \ldots N$, and the system is stable under maximal scheduling. But, there is no guarantee that these limits exist. Thus, this is a weaker notion of stability than that in Definition 14. Whether the stronger notion of stability holds in this case or not remains an open question.

VI. Maxmin Fairness Under Maximal Scheduling

We have so far characterized the throughput region for maximal scheduling Λ^{MS} under different system assumptions. We now describe the issues involved when the arrival rate vector is not in Λ^{MS}. Then maximal scheduling cannot serve all sessions at their arrival rates, and therefore it is necessary to fairly allocate the service rates or departure rates of sessions. We describe how to enhance maximal scheduling so as to ensure maxmin fair allocation of rates in the feasible set for maximal scheduling. We also prove that the rate vector attained by this enhancement differs from the maxmin fair rate vector in the overall network feasible set, by a factor that is at most the reciprocal of the network interference degree. We first consider networks with single-hop sessions (Section VI-A) and subsequently networks with multihop sessions (Section VI-B). In both cases, we will consider a

[2]Each session-link therefore has only one queue for storing the packets waiting for transmission.

special case of the general arrival model presented in (1). Specifically, we will consider the *bounded-burstiness* arrival model where a) $\lambda_i > 0 i = 1, \ldots, N$[3] and there exists a burstiness vector $\vec{\sigma} = (\sigma_1, \ldots, \sigma_N)$ such that

$$|A_i(t) - \lambda_i t| \leq \sigma_i, \qquad \forall\, t. \tag{9}$$

A. Single-Hop Sessions

We assume that every session spans one link. Thus, the framework presented in Section II applies. We introduce our fairness notions and additional assumptions in Section VI-A.1, and subsequently describe the enhancement used for attaining maxmin fairness and the performance guarantees in Section VI-A.2.

1) Fairness Notion and Terminologies: We first present a characterization of the feasible set under maximal scheduling. The result follows easily from the use of proof techniques of Lemma 2 and Theorem 2; the proof is therefore omitted for brevity.

Lemma 11:

$$\Lambda^{\mathrm{MS}} = \Bigg\{ \vec{\lambda} = (\lambda_1, \ldots, \lambda_N) : \text{ if } \lambda_i > 0, \sum_{j \in S_i \cup \{i\}} \lambda_j \leq 1, \ \forall\, i = 1, \ldots, N \Bigg\}. \tag{10}$$

The preceding lemma motivates the following definition.

Definition 15: The *feasible set* Δ^{MS} *of departure rate vectors under maximal scheduling* is the set of vectors $\vec{d} = (d_1, \ldots, d_N)$ that satisfy the following conditions:

$$\sum_{j \in S_i \cup \{i\}} d_j \leq 1, \quad \forall\, i = 1, \ldots, N \quad \text{(interference constraints)} \tag{11}$$

$$0 \leq d_i \leq \lambda_i, \quad \forall\, i = 1, \ldots, N. \tag{12}$$

The "interference constraints" (11) capture the interference relations and are analogous to constraints (10) for the stability region. Note that the constraints $\lambda_i > 0$ are omitted since they hold by our assumption. The constraints (12) follow since the departure rates cannot exceed the arrival rates.

Note that $\Delta^{\mathrm{MS}} \subseteq \Lambda^{\mathrm{MS}}$. When $\vec{\lambda} \in \Lambda^{\mathrm{MS}}$, the departure rate vector satisfies $d_i = \lambda_i$ for each i and hence both (11) and (12) hold. When $\vec{\lambda} \notin \Lambda^{\mathrm{MS}}$, depending on the maximal scheduling policy used, the departure rate vector can be any element of Δ^{MS}, and hence can be unfair for some sessions. For example, if maximal scheduling provides absolute priority to a session i, and $\lambda_i > 1$, then $d_i = 1$ and the departure rates of sessions in S_i are 0. This motivates our goal of ensuring fairness using maximal scheduling.

[3]This assumption requires that the arrival rate for each active session is positive. Note that if a session i is not active we do not need to consider it at all. Thus, we assume that there are N active sessions denoted $1, \ldots, N$. In this section, a session will always refer to an active session, though for brevity we omit the adjective "active."

We now define the notion of maxmin fairness that we seek to attain.

Definition 16: For any N-dimensional vector a, let $\mathcal{I}(a)$ denote a nondecreasing ordering of the components of a. Therefore, if $a = (a_1, a_2, \ldots, a_N)$ and $\mathcal{I}(a) = (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_N)$, then $(\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_N)$ is a permutation of $(a_1, a_2, \ldots, a_N)$, satisfying $\hat{a}_1 \leq \hat{a}_2 \leq \cdots \leq \hat{a}_N$. A departure rate vector $\vec{d}^*$ is said to be maxmin fair if $\vec{d}^* \in \Delta^{\mathrm{MS}}$, and for any other departure rate vector $\vec{d} \in \Delta^{\mathrm{MS}}$, the first nonzero component in $\mathcal{I}(\vec{d}^*) - \mathcal{I}(\vec{d})$ is positive.

Intuitively, a departure rate vector is maxmin fair if it is not possible to increase any of its components without decreasing any other component of equal or lesser value [2]. Note that $\vec{d}^* \in \Lambda^{\mathrm{MS}}$ as $\Delta^{\mathrm{MS}} \subseteq \Lambda^{\mathrm{MS}}$. Finally, if $\vec{\lambda} \in \Lambda^{\mathrm{MS}}$, then $\vec{d}^* = \vec{\lambda}$.

Next, we present a condition that is both necessary and sufficient for any departure rate vector to be maxmin fair. We first introduce the notion of a bottleneck constraint.

Definition 17: For any departure rate vector $\vec{d}$, an interference constraint is a *bottleneck constraint* for a session i if a) i is involved in the constraint, b) $d_i \geq d_k$ for all other sessions k whose sessions are associated with the constraint, and c) the inequality in the constraint is an equality.

Lemma 12: A departure rate vector $\vec{d} \in \Delta^{\mathrm{MS}}$ is maxmin fair if and only if the following holds: for every session i, either $d_i = \lambda_i$, or the session has a bottleneck constraint.

We omit the proof for the above lemma as the proof is similar to that for the well-known bottleneck condition for maxmin fairness in wireline networks [2].

Finally, although for notational simplicity we refer to $\vec{d}^*$ as the maxmin fair departure rate vector, it is maxmin fair only in the feasible set of maximal scheduling Δ^{MS}. The feasible set for the network Δ is the union of the feasible sets of all policies, and may be a strict superset of Δ^{MS}. Thus, the maxmin fair departure rate vector in the network $(\vec{m}^*)$, which we refer to as the *globally maxmin fair departure rate vector*, is the rate vector which is maxmin fair in Δ. We now describe the relation between $\vec{d}^*$ and $\vec{m}^*$. We first describe the notion of "relative fairness" introduced in [16]. A departure rate vector $\vec{a}$ is fairer than another departure rate vector $\vec{b}$ if the first nonzero component in $\mathcal{I}(\vec{a}) - \mathcal{I}(\vec{b})$ is positive. Note that by this definition a departure rate vector is maxmin fair in any feasible set if it is fairer than any other departure rate vector in the same feasible set. Now, since $\vec{m}^* \in \Delta$, $\vec{m}^*/K(\mathcal{N}) \in \Delta^{\mathrm{MS}}$. Thus, from the definition of $\vec{d}^*$, $\vec{d}^*$ is either fairer than $\vec{m}^*/K(\mathcal{N})$ or $\vec{d}^* = \vec{m}^*/K(\mathcal{N})$.

2) Maxmin Fair Rate Allocation Algorithm: We propose a modular approach for attaining maxmin fairness using maximal scheduling. A *Token Generation* module estimates the maxmin fair bandwidth share of each session in each node in the session's path, and generates tokens in accordance with the estimates. A *Packet Release* module releases packets for transmission in accordance with the number of tokens generated. A *Packet Scheduling* module schedules the transmission of the released packets so as to attain the estimates. Note that all the modules operate in parallel. We first describe each module, next explain the intuition behind their design, and finally, present the performance guarantees.

Procedure Token Generation (session i)
begin
For all t and sessions i, let $C_{i,0}(t) = C_{i,b_i+1}(t) = \infty$.
Let $A_i^{\mathrm{NR}}(t)$ be the number of packets of session i at slot t that have been generated at its source but not been released.
Let $\Theta_{i,k}(t) = A_i^{\mathrm{NR}}(t)$ if $k = 1$, and $\Theta_{i,k}(t) = \infty$ otherwise.
Each bucket samples the sessions associated with it in round robin order.
When session i is sampled at its kth bucket in slot t:
if $\Theta_{i,k}(t) > 0$ and $C_{i,k}(t) < C_{i,k+1}(t) + W$ and $C_{i,k}(t) < C_{i,k-1}(t) + W$, **then**
generate a token for session i at its kth bucket ($C_{i,k}(t+1) = C_{i,k}(t) + 1$);
else
do not generate token for session i at its kth bucket ($C_{i,k}(t+1) = C_{i,k}(t)$), and
sample the next session at the kth bucket in the round robin order.
end

Procedure Packet Release (session i)
begin
Release a new session i packet for transmission at session i source node when a token is generated for the session at the bucket at its source.
end

Procedure Packet Scheduling For Transmission
begin
Transmit the released packets using maximal scheduling.
end

Fig. 7. Pseudocode of the fair departure rate allocation algorithm when each session traverses one hop.

a) Description of the modules: We first describe the token generation process. The source node for each active session i maintains a token-bucket for i (Fig. 6(b)). The token-bucket consists of a token-queue for each session in $S_i \cup \{i\}$. Each token-bucket generates tokens for all token-queues in it. For example, in Fig. 6(b), the token-bucket at node A generates tokens for token-queues AB, BC, DE, and FG. A session i is thus associated with (i.e., has token-queues at) $b_i = |S_i| + 1$ token-buckets, one for each of the sessions it interferes with, and itself (Fig. 6(b)). Let us denote these token-buckets as $1, \ldots, b_i$, and let token-bucket 1 be that at the session's source. For example, in Fig. 6(b), $b_i = 4$ for $i = AB$. Each token-bucket samples all sessions in the bucket in a round-robin order. Let $C_{i,k}(t)$ be the number of tokens generated for session i at bucket k in the interval $(0, t]$. Let token-bucket k' $(1 < k' < b_i)$ associated with i be sampled in slot t. Then, k' generates a token for session i in slot t if and only if

$$C_{i,k'}(t) < W + \min(C_{i,k'-1}(t), C_{i,k'+1}(t)).$$

Thus, session i receives a token at bucket k' unless the number of tokens in the token-queue for i at bucket k' substantially exceeds those at the adjacent buckets; this prohibitive difference is the window parameter W. For example, in Fig. 6(b), let $W = 5$ and assume when bucket 2 of AB samples the token-queue of AB, AB has 4, 6, 5 tokens in buckets 1, 2, 3, respectively; then bucket 2 generates a token for AB in this slot. However, if AB had 4, 9, 5 tokens in buckets 1, 2, 3, respectively, at that time, it would not have received a token. If no token is generated for session i in bucket k' in slot t, the next session in the bucket is sampled in that slot. Note that token-bucket 1 and b_i have only one adjacent token-bucket for session i; thus, tokens are generated for session i in these buckets in a manner similar to that described above, but by comparing with the number of tokens of session i in only one adjacent token-bucket. Furthermore, if $k' = 1$ (i.e., k' is at i's source), k' generates a token to i in slot t if and only if the number of packets generated for i at i's source in $(0, t]$ exceeds $C_{i,k'}(t)$. Tokens are never removed from the token-queues in the buckets.

We now describe the packet release process. Whenever the source node of a session i generates a new token for i at i's token-bucket at the source, a new packet for i is released.

Finally we describe the packet scheduling policy. Only the sessions that have released packets waiting for transmission contend for scheduling, and are scheduled as per maximal scheduling. When these sessions are scheduled, they transmit only released packets.

Fig. 7 summarizes the modules.

b) Intuition behind the design: The design of the token generation process ensures that each token-queue receives tokens at a rate that equals the maxmin fair departure rate of the corresponding session (in the next paragraph we describe why this is the case). Whenever a new token is generated for a session i at the token-bucket for i at i's source, i's source releases a new packet for transmission. Thus, the packet release rates are maxmin fair and hence belong to Λ^{MS}. Only the released packets are eligible for transmission. Thus, maximal scheduling transmits the released packets at the rates at which they are released. Hence, the rate allocations are maxmin fair.

We now explain why the token generation rate for each session at each token-bucket associated with the session equals the session's maxmin fair rate. For this explanation, we assume that $\lambda_i > 1$ for each i; all performance guarantees however hold for arbitrary $\vec{\lambda}$. Since $\lambda_i > 1$ for each i, constraints (11) subsume constraints (12). Note that each token-bucket corresponds to constraint (11) for some $j \in \{1, \ldots, M\}$. Since the goal is to allocate maxmin fair rates, each constraint should try to allocate equal rates to all sessions in the constraint. This motivates the round-robin sampling of the sessions at each token-bucket. Again, all constraints involving a session must offer the same

rate to the session. This is attained by relating the token generation process for a given session at a given token-bucket to that at the adjacent token-buckets for the same session. The number of tokens for a session at two adjacent buckets associated with the session diffesr by at most W at any time t, and the difference is at most $b_i W$ for that at any two buckets associated with the session. Thus, the rates of token generation for a session are nearly the same at any two buckets associated with the session.

Since $\lambda_i > 1$ for each i, every session has a bottleneck constraint under the maxmin fair rate allocation. Now, the maxmin fair rate of a session is determined by the bandwidth offered by the bottleneck constraint which offers the least bandwidth to the session. The bucket corresponding to the bottleneck constraint of a session is denoted as the *bottleneck bucket* for the session. By the discussion in the previous paragraph, a session's token generation rate at any token-bucket equals that at its bottleneck bucket, which turns out to be the session's maxmin fair rate. The fairness guarantees follow. Note that if a session has a low maxmin fair rate, then its bottleneck constraint offers it a low rate, and it does not receive tokens several times it is sampled at other buckets; other sessions with less severe constraints receive these tokens.

c) Performance guarantees: The following lemma is instrumental in obtaining the fairness guarantees, and can be motivated by the intuition behind the design of the token generation process.

Lemma 13: Consider token-bucket k of session i. For the bounded-burstiness arrival model and arbitrary $\vec{\lambda}$, there exists constants ϱ, W_0, such that if $W \geq W_0$, then for any interval $(n_1, n_2]$

$$\left| \frac{C_{i,k}(n_2) - C_{i,k}(n_1)}{n_2 - n_1} - d_i^* \right| \leq \frac{\varrho}{n_2 - n_1}.$$

The token generation scheme here is based on the same design principle as that for an existing centralized fair bandwidth allocation algorithm [17], [22]. However, the constraints characterizing the feasibility set for maximal scheduling are different from those characterizing the feasibility set in [17], [22]. We relate the given network $\mathcal{N}$ and the token generation scheme here to a new network $\mathcal{N}'$ where the feasibility constraints and the token generation scheme are the same as those in [17], and prove the above lemma using a result obtained in [17].

Proof: We first obtain a fictitious network $\mathcal{N}'$ from $\mathcal{N}$. Each token-bucket in $\mathcal{N}$ constitutes a node in $\mathcal{N}'$, and there exists a link between any two nodes in $\mathcal{N}'$. Each session i in $\mathcal{N}$ corresponds to a (potentially) multihop session i' in $\mathcal{N}'$. Now, i' in $\mathcal{N}'$ traverses nodes that correspond to its token-buckets $1, \ldots, b_i$ in $\mathcal{N}$, and the source node for i' in $\mathcal{N}'$ is the node that corresponds to its bucket 1. Let a packet arrive at the source node of session i' in $\mathcal{N}'$ whenever a packet arrives for i in $\mathcal{N}$. Let a rate allocation for sessions be feasible in $\mathcal{N}'$ if and only if a) the sum of the rates allocated to sessions traversing a node is upper-bounded by 1 and b) the rate for each session is upper-bounded by its arrival rate. Note that the feasible set of rate allocations in $\mathcal{N}'$ is the same as Δ^{MS}. Thus, $\vec{d}^*$ is the maxmin fair allocation in $\mathcal{N}'$ (note that the definition of maxmin fair allocation applies for any set of vectors with nonnegative real components).

We now describe a token generation process for $\mathcal{N}'$. Each node samples all sessions traversing it in a round-robin order. Let node u sample session i' in slot t. If u is not a source node for a session, it generates a token for session i' in slot t if and only if the number of tokens for i' at u exceeds that at the nodes adjacent to u in the path of i' by at most $W-1$. If u is the source node of i', u generates a token to i' in slot t if and only if the above condition holds and the number of packets that arrived for i' in $(0, t]$ exceeds the number of tokens of i' at u. Lemma 2 in [17, p. 9] proves the following property for the above token generation scheme in any network in which a rate allocation for sessions is feasible if and only if a) the sum of the rates allocated to sessions traversing a node is upper-bounded by 1 and b) the rate for each session is upper-bounded by its arrival rate. For the bounded-burstiness arrival model and arbitrary $\vec{\lambda}$, there exists constants ϱ, W_0, such that if $W \geq W_0$, then for any interval $(n_1, n_2]$, the number of tokens generated for any session at any node in the network in the interval differs from the session's maxmin fair rate by at most $\frac{\varrho}{n_2 - n_1}$.

The result follows from the above lemma and the observation that a token is generated for i' at u in $\mathcal{N}'$ at time t if and only if session i receives a token at its corresponding token-bucket at time t. □

Packets that contend for scheduling and are transmitted by maximal scheduling arrive as per the release process. Since a new packet is released every time a new packet is generated, the above lemma implies that the release rate vector is maxmin fair and is therefore in Λ^{MS}. Maximal scheduling therefore provides departure rates equal to the packet release rates. Thus, as the following result states, a combination of token generation and maximal scheduling attains the maxmin fair departure rates for every session.

Theorem 3: For the bounded-burstiness arrival model and arbitrary $\vec{\lambda}$, there exists a constant W_0, such that when $W \geq W_0$, $\lim_{n\to\infty} D_{L_i}(n)/n = d_i^*,\ i = 1, \ldots N$.

Proof: Let $A_i^{\mathrm{R}}(t)$ be the number of packets of session i that have been released at its source node in $(0, t]$. Note that a packet is released for session i at its source if and only if a new token is generated for session i at the bucket at its source. Thus, $\forall\, t, A_i^{\mathrm{R}}(t) = C_{i,n}(t)$ where n is the bucket at i's source. Now, from Lemma 13, there exists constants ϱ, W_0, such that when $W \geq W_0,\ \forall\, t, \left|\frac{A_i^{\mathrm{R}}(t)}{t} - d_i^*\right| \leq \frac{\varrho}{t}$. Thus, the packet release rate vector is $\vec{d}^* \in \Lambda^{\mathrm{MS}}$. Since only the released packets are available for scheduling and the release rate vector is in Λ^{MS}, the departure rate vector exists and equals the release rate vector. The result follows. □

B. Multihop Sessions

We next allow sessions to traverse multiple hops. Thus, the framework in Section V-C applies. The *feasible set* Δ^{MS} *of departure rate vectors* $\vec{d} = (d_1, \ldots, d_N)$ can be described by (12) and

$$\sum_{k \in S_j \cup \{j\}} d_{q(k)} \leq 1, \qquad \forall\, j = 1, \ldots, M. \tag{13}$$

Procedure Token Generation (session-link j)
begin
For session-link j, let l and m respectively be the previous and next session-links of the same session.
For each slot t and session-link j,
if i is the first-session-link of its session, **then**
$C_{j,0}(t) = \infty, C_{j,b_j+1}(t) = C_{m,0}(t)$
else
if j is the last session-link of its session, **then**
$C_{j,0}(t) = C_{l,b_l+1}(t), C_{j,b_j+1}(t) = \infty$
else
$C_{j,0}(t) = C_{l,b_l}(t)$ and $C_{j,b_j+1}(t) = C_{m,0}(t)$.
Let $A_j^{\mathrm{NR}}(t)$ be the number of packets of session-link j at slot t that are in its waiting-queue.
Let $\Theta_{j,k}(t) = A_j^{\mathrm{NR}}(t)$ if the kth bucket of session-link j is at the source-node of session of j, and $\Theta_{j,k}(t) = \infty$ otherwise.
Each bucket samples the session-links associated with it in round robin order.
When session-link j is sampled at its kth bucket in slot t:
if $\Theta_{j,k}(t) > 0$ and $C_{j,k}(t) < C_{j,k+1}(t) + W$ and $C_{j,k}(t) < C_{j,k-1}(t) + W$, **then**
generate a token for session-link j at its kth bucket ($C_{j,k}(t+1) = C_{j,k}(t) + 1$);
else
do not generate token for session j at its kth bucket ($C_{j,k}(t+1) = C_{j,k}(t)$), and
sample the next session-link at the kth bucket in the round robin order.
end

Procedure Queue Management (session-link j)
begin
When a new packet is generated for session-link j or a new packet arrives at the source of session-link j from a previous session-link, add the new-packet in the waiting-queue for session-link j.
Transfer a session-link j packet from its waiting-queue to its release-queue at its source node when a token is generated for it at the bucket at its source.
end

Procedure Packet Scheduling For Transmission
begin
Transmit the packets in the release-queues of the session-links using maximal scheduling.
end

Fig. 8. Pseudocode of the fair departure rate allocation algorithm when sessions traverse multiple hops.

Using the above description for Δ^{MS}, the maxmin fair departure rate vector can now be defined as in Section VI-A.

Definition 18: For any departure rate vector $\vec{d}$, an interference constraint is a *bottleneck constraint* for a session i if a) a session-link j of i is involved in the constraint, b) $d_{q(j)} \geq d_{q(k)}$ for all other session-links k whose sessions are associated with the constraint, and c) the inequality in the constraint is an equality.

Again, with the above definition for a bottleneck constraint, Lemma 12 provides a necessary and sufficient condition for a departure rate vector to be maxmin fair.

We now describe the modifications required in the algorithm presented in Fig. 7 for attaining maxmin fairness in this general case. We first describe the modifications in the token generation procedure. Now, session-links, rather than sessions, are associated with token-buckets, and the source of each session-link j maintains the bucket consisting of session-links in $S_j \cup \{j\}$. Again, token-buckets sample session-links rather than sessions. The token generation process for a session-link is similar to that for a single-hop session. The main difference is that the token generation process for a session-link j at the first (last) token-bucket of j must also depend on the number of tokens generated at the last (first) token-bucket for the previous (next) session-link k of the same session (Fig. 6(b)). We now describe the packet scheduling policy. The source of each session-link maintains two packets queues: a *waiting* packet queue, and a *released* packet queue. On arrival, a packet is queued at the waiting packet queue. A packet is forwarded from the waiting queue to the released queue when a new token is generated at the token-bucket for the session-link at the session-link's source. Only session-links with nonempty released queues contend for scheduling. The rest of the scheduling remains the same as that for the case of single-hop sessions. Refer to Fig. 8 for a pseudocode.

Both Lemma 13 and Theorem 3 hold; the term "session" must now be replaced with "session-link" in the statement of Lemma 13.

We now make a few concluding remarks on our maxmin fair packet scheduling algorithm. Note that the token-buckets associated with a session-link i need to know the number of tokens generated for i at other token-buckets associated with i. Also note that a token-bucket associated with i is either at i's source or at j's source, where $j \in S_i$. Thus, a token-bucket at the source of a session-link k need only know the number of tokens generated at a token-bucket at the source of a session-link l if and only if both k and l interfere with each other or with a common session-link. Since only session-links in close proximity interfere with each other in a wireless network, the token generation process requires communication among nodes in proximity as well. Finally, the analytical guarantees hold even when nodes know the number of tokens generated at other nodes after some delay, as long as the delay is upper-bounded by a constant.

VII. Discussion and Conclusion

In this paper, we have addressed the long-standing open question of attaining throughput guarantees with distributed scheduling in wireless networks. We have studied the performance of a simple distributed scheduling policy, maximal scheduling, which had earlier been investigated in context of node-exclusive spectrum sharing model and input-queued switches. We have obtained tight performance guarantees for maximal scheduling under arbitrary interference models and topologies, and have characterized the throughput region attained by maximal scheduling in terms of the interference degree of the network. The characterizations demonstrate that the performance bounds depend heavily on the nature of communication and interference models. We prove that maximal scheduling is guaranteed to attain a constant fraction of the maximum throughput region for certain communication and interference models, while it is also guaranteed to not attain a constant fraction in the worst case for some other models. Our results can be generalized to networks with multicast communication, arbitrary number of frequencies, and end-to-end sessions. Finally, we enhance maximal scheduling to guarantee fairness of rate allocation.

The class of maximal scheduling policies is quite broad, and our performance bounds apply to all policies in this class. However, it remains to be seen whether certain policies in this class can attain better performance bounds, while still being amenable to low-complexity distributed implementation. Similar questions remain open for distributed scheduling policies outside this class as well. Recently, Sharma *et al.* [19] have lower-bounded the complexity of policies that attain the maximum stability region, or approximate the maximum stability region within constant factor, in arbitrary topologies. These results may help answer some of the above open questions.

Appendix I
Proofs of Analytical Results in Section IV (Lemmas 2 and 5)

A. Proof of Lemma 2

Recall that $Q_i(n)$ denotes the queue length of session i in the beginning of the nth slot. Then, for any scheduling policy

$$Q_i(n+1) = Q_i(0) + A_i(n) - D_i(n), \quad \forall\, n \geq 1 \text{ and } i = 1, \ldots, N. \tag{14}$$

We first define fluid limits. The definitions are similar to those used by Dai *et al.* [7].

1) Definition of Fluid Limits: We denote by $\mathbb{N}$ and $\mathbb{R}$ the set of nonnegative integers and reals, respectively. For a random process $\{f(t)\}_{t\geq 0}$, we denote its value at time t along a sample path ω by $f(t,\omega)$.

Note that the domain of the functions $A(\cdot), D(\cdot)$ and $Q(\cdot)$ is $\mathbb{N}$. Now, we define these functions for arbitrary $t \in \mathbb{R}$ by using a piecewise linear interpolation. The piecewise linear interpolation of a function $f : \mathbb{N} \to \mathbb{R}$ is defined as follows. For $t \in (n, n+1]$

$$f(t) = f(n) + (t-n)(f(n+1) - f(n)).$$

Note that $f(t)$ defined as above is a continuous function.

Consider any scheduling policy. From any sender i, at most one packet can be served in a slot. Also, the maximum number of packets arriving in a slot at i is bounded by $\alpha_{\max}$. Thus, for every $i, \omega, t \geq 0$ and $\delta > 0$

$$A_i(t+\delta,\omega) - A_i(t,\omega) \leq \delta\alpha_{\max} \tag{15}$$
$$D_i(t+\delta,\omega) - D_i(t,\omega) \leq \delta \tag{16}$$
$$Q_i(t+\delta,\omega) - Q_i(t,\omega) \leq \delta\alpha_{\max}. \tag{17}$$

Now, let us define a family of functions for any given function $f(\cdot)$ as follows:

$$f^r(t,\omega) \stackrel{\text{def}}{=} \frac{f(rt,\omega)}{r}, \qquad \text{for every } r > 0.$$

It follows from (15), (16), and (17), that for every $r > 0$

$$A_i^r(t+\delta,\omega) - A_i^r(t,\omega) \leq \delta\alpha_{\max} \tag{18}$$
$$D_i^r(t+\delta,\omega) - D_i^r(t,\omega) \leq \delta \tag{19}$$
$$Q_i^r(t+\delta,\omega) - Q_i^r(t,\omega) \leq \delta\alpha_{\max}. \tag{20}$$

Thus, all the above functions are Lipschitz continuous, and hence uniformly continuous on any compact interval. Clearly, the above functions are also bounded on any compact interval. Fix a compact interval $[0,t]$. Now, consider any sequence r_n such that $r_n \to \infty$ as $n \to \infty$. Then, by Arzela–Ascoli theorem [14], there exists a subsequence r_{n_k} and continuous functions $\bar{A}_i(\cdot), \bar{D}_i(\cdot)$ and $\bar{Q}_i(\cdot)$ such that for every i, ω

$$\lim_{k\to\infty} \sup_{\hat{t}\in[0,t]} \left| A_i^{r_{n_k}}(\hat{t},\omega) - \bar{A}_i(\hat{t},\omega) \right| = 0 \tag{21}$$
$$\lim_{k\to\infty} \sup_{\hat{t}\in[0,t]} \left| D_i^{r_{n_k}}(\hat{t},\omega) - \bar{D}_i(\hat{t},\omega) \right| = 0 \tag{22}$$
$$\lim_{k\to\infty} \sup_{\hat{t}\in[0,t]} \left| Q_i^{r_{n_k}}(\hat{t},\omega) - \bar{Q}_i(\hat{t},\omega) \right| = 0. \tag{23}$$

We now define fluid limits.

Definition 19: Any $(\bar{A}_i, \bar{D}_i, \bar{Q}_i)$ is called a fluid limit for $\mathcal{N}$ if there exists r_{n_k} such that all the relations (21) to (23) are satisfied.

Now, we state some important properties of the fluid limits which we use to prove Lemma 2.

Lemma 14: Every fluid limit satisfies, $\bar{A}_i(t) = \lambda_i t$ with probability (w.p.) 1 for every session i and $t \geq 0$.

Lemma 15: Any fluid limit $(\overline{A}_i, \overline{D}_i, \overline{Q}_i)$ for $\mathcal{N}$ satisfies the following equality for every i and $t \geq 0$ w.p. 1:

$$\overline{Q}_i(t) = \overline{Q}_i(0) + \lambda_i t - \overline{D}_i(t). \tag{24}$$

Lemma 16: Let $\overline{Q}_i(0) = 0$ for every i. Also, let $\sum_{j\in S_i\cup\{i\}} \lambda_j \leq 1$ if $\lambda_i > 0$, $i = 1, \ldots, N$. Then, under maximal scheduling, every fluid limit satisfies $\overline{Q}_i(t) = 0$ for every $t \geq 0$ w.p. 1 for every i.

The proofs of Lemmas 14–16 are provided later, after the proof Lemma 2. We now prove Lemma 2.

Proof: First, we show that $\lim_{r\to\infty} D_i^r(t) = \lambda_i t$ w.p. 1 for every t. Then, the result follows by choosing $t = 1$.

Under maximal scheduling, if $\overline{Q}_i(0) = 0$ and $\sum_{j\in S_i\cup\{i\}}\lambda_i \leq 1$ for every i for which $\lambda_i > 0$, then $\overline{Q}_i(t) = 0$ w.p. 1 for every i and $t \geq 0$ (Lemma 16). Thus, by Lemma 15, $\overline{D}_i(t) = \lambda_i t$ w.p. 1 for every $t \geq 0$. Since $\overline{D}_i(\cdot)$ is a fluid limit, there exists a subsequence r_{n_k} such that $\lim_{k\to\infty} r_{n_k} = \infty$ and $\lim_{k\to\infty} D_i^{r_{n_k}}(t) = \overline{D}_i(t) = \lambda_i t$ w.p. 1 (Section I-A.1). Thus, $\liminf_{r\to\infty} D_i^r(t) \leq \lambda_i t$ w.p. 1. Now, we argue that $\liminf_{r\to\infty} D_i^r(t) = \lambda_i t$ w.p. 1.

Suppose, $\liminf_{r\to\infty} D_i^r(t) < \lambda_i t$ w.p. 1. Then, there exists a subsequence $\widehat{r}_{n_k}$ such that $\lim_{k\to\infty}\widehat{r}_{n_k} = \infty$ and $\lim_{k\to\infty} D_i^{\widehat{r}_{n_k}}(t) = \lambda_i t - \epsilon$ w.p. 1 for some $\epsilon > 0$. Now, note that

$$Q^{\widehat{r}_{n_k}}(t) = Q^{\widehat{r}_{n_k}}(0) + A^{\widehat{r}_{n_k}}(t) - D^{\widehat{r}_{n_k}}(t) \quad \text{(from (14))}.$$

Now, by taking limit as $k \to \infty$ on both sides of the above equation we obtain

$$\begin{aligned}\overline{Q}_i^1(t) &= \overline{Q}_i^1(0) + \lambda_i t - \overline{D}_i^1(t) \quad \text{w.p. } 1 \quad \text{(from Lemma 14)}\\ &= \epsilon \ (\text{since } \overline{D}_i^1(t) = \lim_{k\to\infty} D_i^{\widehat{r}_{n_k}}(t) = \lambda_i t - \epsilon).\end{aligned}$$

Since, $\overline{Q}_i^1(t)$ is also a fluid limit under maximal scheduling, the above equation contradicts Lemma 16. Thus

$$\liminf_{r\to\infty} D_i^r(t) = \lambda_i t \quad \text{w.p. } 1.$$

Now, for every $r > 0$, $D_i^r(t) \leq A_i^r(t)$ as the number of departures from i can at most be equal to the arrivals for i till time rt. Thus, clearly

$$\limsup_{r\to\infty} D_i^r(t) \leq \lambda_i t \quad \text{w.p. } 1.$$

This shows that

$$\lim_{r\to\infty} D_i^r(t) = \lambda_i t \quad \text{w.p. } 1.$$

Now, select $t = 1$, and consider subsequence r_n such that $r_n = n$. Here, for every i

$$\begin{aligned}\lim_{n\to\infty} D_i^{r_n}(1) &= \lambda_i \quad \text{w.p. } 1\\ \lim_{n\to\infty} \frac{D_i(n)}{n} &= \lambda_i \quad \text{w.p. } 1. \qquad \square\end{aligned}$$

We now prove the supporting lemmas used to prove Lemma 2.

2) Proof of Lemma 14:

Proof: Since $\overline{A}_i(t)$ is a fluid limit, by Definition 19, there exists a sequence r_{n_k} such that $\lim_{k\to\infty} r_{n_k} = \infty$ and

$$\begin{aligned}\overline{A}_i(t) &= \lim_{k\to\infty} A_i^{r_{n_k}}(t) (\text{from (21)})\\ &= \lim_{k\to\infty} \frac{A_i(r_{n_k}t)}{r_{n_k}}\\ &= \lim_{k\to\infty} \frac{A_i(r_{n_k}t)}{r_{n_k}t} t\\ &= \lambda_i t \quad \text{w.p. } 1 \text{ (since } A_i(\cdot) \text{ satisfy SLLN)}.\end{aligned}$$

The result follows. $\square$

3) Proof of Lemma 15:

Proof: Since $\overline{Q}_i(\cdot), \overline{A}_i(\cdot)$ and $\overline{D}_i(\cdot)$ are fluid limits, there exists a sequence r_{n_k} such that $\lim_{k\to\infty} r_{n_k} = \infty$ and they are obtained as a uniform limits of functions $Q_i^{r_{n_k}}(\cdot), A_i^{r_{n_k}}(\cdot)$, and $D_i^{r_{n_k}}(\cdot)$, respectively. Now, from (14) it follows that for every r_{n_k} and $t \geq 0$

$$Q_i^{r_{n_k}}(t) = Q_i^{r_{n_k}}(0) + A_i^{r_{n_k}}(t) - D_i^{r_{n_k}}(t).$$

The result follows from Lemma 14 after taking the limit $k \to \infty$ on both sides of the above equality. $\square$

4) Proof of Lemma 16:

Proof: We prove the required by contradiction. Let $\overline{Q}_i(t) \neq 0$ for every t and i. Then, there exists a session $i, \hat{t}, y_1 > 0$, and $x_1 > 0$ such that

$$\sum_{j\in S_i\cup\{i\}} \overline{Q}_j(\hat{t}) = y_1 \tag{25}$$

$$\sum_{j\in S_i\cup\{i\}} \overline{Q}_j(t) < y_1 \quad \text{for every } t \in [0,\hat{t}) \tag{26}$$

$$\overline{Q}_i(\hat{t}) = x_1. \tag{27}$$

We justify (25) to (27) by constructing $x_1, y_1, \hat{t}$ that satisfy (25) to (27). Let $t' = \inf\{t : t \geq 0, \max_k \bar{Q}_k(t) > 0\}$. Since $\bar{Q}_k(t) \neq 0$ for some t and some k, t' is well defined. From the definition of t' there exists an i such that $t' = \inf\{t : t \geq 0, \bar{Q}_i(t) > 0\}$. From the continuity of $\bar{Q}_k(t)$ for all t, k, the definition of t', and since $\bar{Q}_k(0) = 0$ for all $k, \bar{Q}_k(t_1) = 0$ for all $t_1 \leq t'$ and k. From the continuity of $\bar{Q}_i(t)$ for all t, there exists an $\epsilon > 0$ such that (s.t.) $\sum_{j\in S_i\cup\{i\}} \bar{Q}_j(t) \geq \bar{Q}_i(t) > 0$ for all $t \in (t', t'+\epsilon]$. Let $y_1 = \max_{t:t\in[0,t'+\epsilon]} \sum_{j\in S_i\cup\{i\}} \bar{Q}_j(t)$. Let $\hat{t}$ be the first time at which $\sum_{j\in S_i\cup\{i\}} \bar{Q}_j(t) = y_1$. Now, $\hat{t} \in (t', t'+\epsilon]$, since $\bar{Q}_k(t_1) = 0$ for all k and all $t_1 \leq t'$, and $\sum_{j\in S_i\cup\{i\}} \bar{Q}_j(t) \geq \bar{Q}_i(t) > 0$ for all $t \in (t', t'+\epsilon]$. Let $x_1 = \bar{Q}_i(\hat{t})$. Clearly, $x_1 > 0$. Let $\lambda_i \leq 0$. From Lemma 15, since $\bar{Q}_i(0) = 0, \bar{Q}_i(\hat{t}) \leq -\bar{D}_i(\hat{t})$. Since $\bar{D}_i(\cdot)$ is the fluid limit of $D_i(\cdot)$, and $D_i(t) \geq 0$ at all t, $\bar{D}_i(\hat{t}) \geq 0$. Thus, $x_1 \leq 0$, which is a contradiction. Thus, $\lambda_i > 0$, and hence, $\sum_{j\in S_i\cup\{i\}} \lambda_j \leq 1$. Clearly, $x_1 \leq y_1$ as $\overline{Q}_j(\cdot) \geq 0$ for every j. Since $\overline{Q}_i(\cdot)$ is a continuous function, there exists $t' \in [0,\hat{t})$ such that

$$\overline{Q}_i(t) \geq \frac{x_1}{2} \quad \text{for every } t \in [t',\hat{t}]. \tag{28}$$

Now, since $\overline{Q}_j(\cdot)$ is a fluid limit, by Definition 19, there exists a sequence r_{n_k} such that $\lim_{k\to\infty} r_{n_k} = \infty$ and $\lim_{k\to\infty} Q_j^{r_{n_k}}(t) = \overline{Q}_j(t)$ for every j and t in an interval $[0,\hat{t}]$. Thus, we can draw two conclusions. First, for sufficiently large r_{n_k}, $Q_i^{r_{n_k}}(t) > x_1/4$ for every $t \in [t',\hat{t}]$. Thus, $Q_i(r_{n_k}t) > r_{n_k}x_1/4$. This implies that for every $r_{n_k} > 4/x_1$

$$Q_i(r_{n_k}t) > 1 quad \text{for every } t \in [t',\hat{t}]. \tag{29}$$

The second conclusion is that for every sufficiently large r_{n_k}, there exists $\epsilon > 0$ such that

$$\begin{aligned}&\sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(\hat{t}) - \sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(t') > \epsilon\\ &\Rightarrow \lim_{k\to\infty}\left[\sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(\hat{t}) - \sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(t')\right]\\ &\quad \geq \epsilon.\end{aligned} \tag{30}$$

Relation (30) follows from (25), (26), $t' < \widehat{t}$, and the definition of fluid limits. Select r_{n_k} large enough such that (29) holds. For all such r_{n_k}

$$\begin{aligned}
&\sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(\widehat{t}) - \sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(t') \\
&\quad= \sum_{j\in S_i\cup\{i\}} \left[A_j^{r_{n_k}}(\widehat{t}) - A_j^{r_{n_k}}(t')\right] \\
&\qquad- \left[\sum_{j\in S_i\cup\{i\}} D_j^{r_{n_k}}(\widehat{t}) - \sum_{j\in S_i\cup\{i\}} D_j^{r_{n_k}}(t')\right] \text{ (from (14))}.
\end{aligned} \tag{31}$$

Since maximal scheduling is used and (29) holds, at least one packet from some session in $S_i\cup\{i\}$ departs in every slot. Thus

$$\sum_{j\in S_i\cup\{i\}} D_j^{r_{n_k}}(\widehat{t}) - \sum_{j\in S_i\cup\{i\}} D_j^{r_{n_k}}(t') \geq (\widehat{t}-t').$$

Now, from (31)

$$\begin{aligned}
&\sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(\widehat{t}) - \sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(t') \\
&\quad\leq \sum_{j\in S_i\cup\{i\}} \left[A_j^{r_{n_k}}(\widehat{t}) - A_j^{r_{n_k}}(t')\right] - (\widehat{t}-t') \\
\Rightarrow &\lim_{k\to\infty} \left[\sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(\widehat{t}) - \sum_{j\in S_i\cup\{i\}} Q_j^{r_{n_k}}(t')\right] \\
&\quad\leq \lim_{k\to\infty} \sum_{j\in S_i\cup\{i\}} \left[A_j^{r_{n_k}}(\widehat{t}) - A_j^{r_{n_k}}(t')\right] - (\widehat{t}-t') \\
&\quad= \left(\sum_{j\in S_i\cup\{i\}} \lambda_j - 1\right)(\widehat{t}-t') \text{ w.p. 1 (from Lemma 14)} \\
&\quad\leq 0.
\end{aligned} \tag{32}$$

Note that (32) contradicts (30). Thus, the result follows. □

B. Proof of Lemma 5

We prove that $K(\mathcal{N}) \leq 8$ in any network $\mathcal{N}$ under the bidirectional equal power model. We consider an arbitrary session (S_0, T_0, R_0) and show that K_0, the maximum number of sessions that interfere with S_0 but do not interfere with each other, must satisfy $K_0 \leq 8$. The result follows.

We assume that the nodes are deployed on a two-dimensional Euclidean plane. Let the distance between the transmitting node T_0 and receiving node R_0 be $\rho \leq r$, where r is the transmission range of any node.

Without loss of generality let us assume that the line joining T_0 and R_0 is aligned along the x-axis. Let $\mathcal{D}_{T_0}$ and $\mathcal{D}_{R_0}$ represent disks of radius r around T_0 and R_0, respectively. Then the interference area of session S_0 is $\mathcal{D}_{T_0}\cup\mathcal{D}_{R_0}$.

In the following, a node is said to be the *transceiver node* of a session if it is either the transmitting node or the receiving node of that session; thus, each session has two transceiver nodes. Note that if a session interferes with S_0, at least one of its transceiver nodes must lie in $\mathcal{D}_{T_0}\cup\mathcal{D}_{R_0}$. Now for each of the sessions that interfere with S_0 but do not interfere with each other, choose any one transceiver node of that session that lies in $\mathcal{D}_{T_0}\cup\mathcal{D}_{R_0}$; let $\mathcal{U}_0$ denote the set of the transceiver nodes thus chosen. We will show $K_0 \leq 8$ by showing $U_0 = |\mathcal{U}_0| \leq 8$.

We first argue that $U_0 \leq 9$, which follows almost immediately from Lemma 3.1 of [12]. The following observation and lemma follows from Lemma 3.1 (and its proof) in [12].

Observation 1: Let $W_1, W_2 \in \mathcal{U}_0$. If $W_1, W_2 \in \mathcal{D}_{T_0}$ $(W_1, W_2 \in \mathcal{D}_{R_0})$, and none of them coincide with T_0 (R_0), then the line segment joining W_1 and W_2 subtends an angle greater than $\frac{\pi}{3}$ at T_0 (R_0).

Lemma 17: The number of nodes in $\mathcal{U}_0$ that lie in $\mathcal{D}_{T_0}$ $(\mathcal{D}_{R_0})$ can be no greater than 5.

Note that $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$ is contained in four $\frac{\pi}{3}$ sectors. Therefore, at most four nodes in $\mathcal{U}_0$ can lie in $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$ (follows from Observation 1). Since at most five nodes in $\mathcal{U}_0$ can lie in $\mathcal{D}_{T_0}$ (Lemma 17), it follows that $U_0 \leq 9$.

Now we proceed to tighten this upper bound by showing $U_0 \leq 8$; the proof of this fact is rather tedious, and is described next. Towards this end, let us assume, for the sake of contradiction, that $U_0 = 9$.

Corollary 1: If $U_0 = 9$, then the number of nodes in $\mathcal{U}_0$ that lie in $\mathcal{D}_{T_0}\setminus\mathcal{D}_{R_0}$, $\mathcal{D}_{T_0}\cap\mathcal{D}_{R_0}$, and $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$ are 4, 1, and 4, respectively.

Proof: Let U_1, U_2, and U_3, respectively, denote the nodes in $\mathcal{U}_0$ that lie in $\mathcal{D}_{T_0}\setminus\mathcal{D}_{R_0}$, $\mathcal{D}_{T_0}\cap\mathcal{D}_{R_0}$,, and $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$. Then, $U_1+U_2+U_3 = 9$. Without loss of generality, assume $U_1 \geq U_3$.

We first argue that $U_2 \neq 0$. Note that if $U_2 = 0$, then $U_1 + U_3 = 9$, implying $U_1 \geq 5$, which is impossible since $\mathcal{D}_{T_0}\setminus\mathcal{D}_{R_0}$ is contained in four $\frac{\pi}{3}$ sectors. This implies that $U_2 > 0$.

Now we argue that $U_2 \leq 1$. Let us assume, for the sake of contradiction, that $U_2 \geq 2$. Then, $U_1+U_3 = 9-U_2 \leq 7$. Thus, $U_3 \leq 3$. Therefore, $U_1+U_2 = 9-U_3 \geq 6$, which is impossible (from Lemma 17). Therefore, $U_2 \leq 1$. Since $U_2 > 0$ (as shown previously), we have $U_2 = 1$.

Therefore, $U_1 + U_3 = 8$. Since $U_1 \leq 4, U_3 \leq 4$ (each of $\mathcal{D}_{T_0}\setminus\mathcal{D}_{R_0}$ and $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$ are contained in four $\frac{\pi}{3}$ sectors), we must have $U_1 = U_3 = 4$. □

From Corollary 1, we see that if $K_0 = 9$, then $\mathcal{D}_{T_0}\setminus\mathcal{D}_{R_0}$ and $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$ must each contain four nodes in $\mathcal{U}_0$. For the sake of contradiction, let us assume that this is true. Note that none of these eight nodes can lie at the centers of the two disks, i.e., at T_0 or R_0. Also, exactly one of these eight points must lie in each of the $\frac{\pi}{3}$ sectors of $\mathcal{D}_{T_0}\setminus\mathcal{D}_{R_0}$ and $\mathcal{D}_{R_0}\setminus\mathcal{D}_{T_0}$. Let X_1, X_2, X_3, and X_4, respectively, denote the nodes in $\mathcal{U}_0$ that lie in sectors A_1T_0C, CT_0D, DT_0E, and ET_0B_1. Let Y_1, Y_2, Y_3, and Y_4, respectively, denote the nodes in $\mathcal{U}_0$ that lie in sectors A_2R_0F, FR_0G, GR_0H, and HR_0B_2. Join X_1, X_2, X_3, X_4 with T_0, and Y_1, Y_2, Y_3, Y_4 with R_0 (refer to Fig. 9). Now, construct the octagon by joining $X_1X_2, X_2X_3, X_3X_4, Y_1Y_2, Y_2Y_3, Y_3Y_4$, and X_1Y_1, X_4Y_4. Note that the length of each side of this octagon must be greater than r. Let line segment X_1Y_1 intersect line segments T_0A and R_0A (possibly extended) at points I_1 and I_2, respectively. Let line segment X_4Y_4 intersect line segments T_0B and R_0B (possibly extended) at points J_1 and J_2, respectively.

Note that the angle subtended at T_0 by $I_1X_1X_2X_3X_4J_1$ (which is a collection of the line segments $I_1X_1, X_1X_2, \ldots, X_4J_1$), is equal to $\frac{4\pi}{3}$. Similarly, the angle subtended at R_0

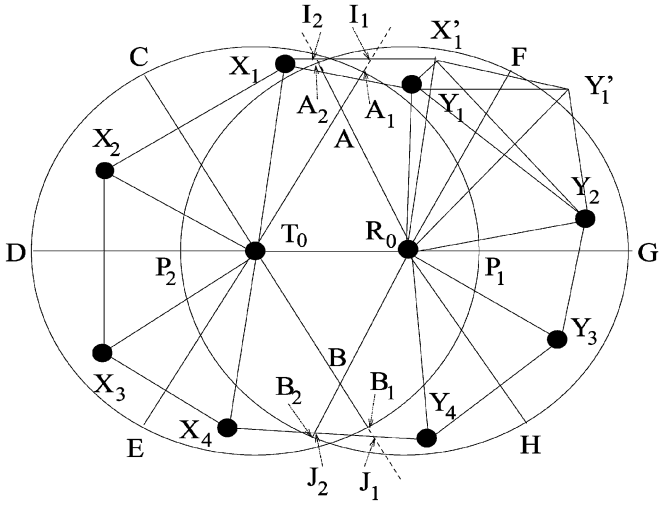

Fig. 9. Diagram used in proof of Lemma 5.

by $I_2Y_1Y_2Y_3Y_4J_2$ (which is a collection of the line segments $I_2Y_1, Y_1Y_2, \ldots, Y_4J_2$), is equal to $\frac{4\pi}{3}$. In the following, we show however that the angle subtended at T_0 by $I_1X_1X_2X_3X_4J_1$ plus the angle subtended at R_0 by $I_2Y_1Y_2Y_3Y_4J_2$ must be greater than $\frac{8\pi}{3}$, thus arriving at a contradiction.

We will show that the angle subtended by $X_2X_1I_1$ at T_0 plus the angle subtended by $I_2Y_1Y_2$ at R_0 is greater than π. Without loss of generality, assume that X_1 has a higher y-coordinate than Y_1 (recall that T_0R_0 is aligned along the x-axis). As shown in Fig. 9, choose $X_1^{'}$ such that $X_1T_0R_0X_1^{'}$ is a parallelogram. Join $X_1^{'}$ with Y_1 and Y_2. Note, $\angle X_1T_0I_1 + \angle I_2R_0X_1^{'} = \pi - \angle I_1T_0R_0 - \angle I_2R_0T_0 = \pi - \frac{\pi}{3} - \frac{\pi}{3} = \frac{\pi}{3}$.

We consider the following two cases separately: i) Y_1 lies within parallelogram $X_1T_0R_0X_1^{'}$, and ii) Y_1 lies outside parallelogram $X_1T_0R_0X_1^{'}$. Let us consider case i) first (Fig. 9 shows this case). In this case, we claim that $\angle X_1^{'}R_0Y_2 > \frac{\pi}{3}$. To see this, choose $Y_1^{'}$ such that $X_1Y_1Y_1^{'}X_1^{'}$ is a parallelogram. Join $Y_1^{'}$ with R_0 and Y_2. Note that $|X_1^{'}Y_1^{'}| = |X_1Y_1| > r$. Note that Y_2 must lie "below" $Y_1Y_1^{'}$, since it is easy to see that there is no point in sector FR_0G that is "above" $Y_1Y_1^{'}$ and whose distance from Y_1 is greater than r.

Note that $|Y_1Y_1^{'}| = |X_1X_1^{'}| = |T_0R_0| = \rho$ (by construction). Therefore, it is easy to see that $Y_1^{'}$ must lie outside $\mathcal{D}_{R_0}$. Thus, line segment $X_1^{'}Y_2$ must intersect line segment $Y_1Y_1^{'}$. In the triangle $Y_1Y_2Y_1^{'}$, $|Y_1Y_2| > r$ and $|Y_1Y_1^{'}| = \rho \leq r$. Therefore, $\angle Y_1Y_1^{'}Y_2 > \angle Y_1Y_2Y_1^{'}$. Thus, $\angle X_1^{'}Y_1^{'}Y_2 \geq Y_1Y_1^{'}Y_2 > \angle Y_1Y_2Y_1^{'} \geq \angle X_1^{'}Y_2Y_1^{'}$. Thus, comparing angles in the triangle $X_1^{'}Y_2Y_1^{'}$, we get $|X_1^{'}Y_2| > |X_1^{'}Y_1^{'}| > r$.

Note that since X_1 lies in sector CT_0A_1, it follows that $X_1^{'}$ must lie in sector A_2R_0F. Therefore, $X_1^{'}$ lies in $\mathcal{D}_{R_0}$. In the triangle $X_1^{'}R_0Y_2$, therefore, we have $|X_1^{'}R_0| \leq r, |Y_2R_0| \leq r$, and $|X_1^{'}Y_2| > r$. Therefore, $\angle X_1^{'}R_0Y_2 > \frac{\pi}{3}$.

Thus, if Y_1 lies in the parallelogram $X_1T_0R_0X_1^{'}$, we have $\angle X_1T_0I_1 + \angle I_2R_0X_1^{'} + \angle X_1^{'}R_0Y_2 > \frac{\pi}{3} + \frac{\pi}{3} = \frac{2\pi}{3}$. Moreover, since $\angle I_2R_0X_1^{'} + \angle X_1^{'}R_0Y_2 = \angle I_2R_0Y_1 + \angle Y_1R_0Y_2$, we have $\angle X_1T_0I_1 + \angle I_2R_0Y_1 + \angle Y_1R_0Y_2 > \frac{2\pi}{3}$. From Observation 1, $\angle X_2T_0X_1 > \frac{\pi}{3}$. Therefore, $\angle X_2T_0X_1 + \angle X_1T_0I_1 + \angle I_2R_0Y_1 + \angle Y_1R_0Y_2 > \pi$. In other words, the angle subtended by $X_2X_1I_1$ at T_0 plus the angle subtended by $I_2Y_1Y_2$ at R_0 is greater than π.

Now let us consider the case where Y_1 does not lie inside the parallelogram $X_1T_0R_0X_1^{'}$. Since Y_1 has a lower y-coordinate than X_1, it follows that Y_1 must lie below the line $X_1X_1^{'}$. Thus, Y_1 must lie to the "right" of line $R_0X_1^{'}$. Thus, $\angle X_1T_0I_1 + \angle I_2R_0Y_1 > \angle X_1T_0I_1 + I_2R_0X_1^{'} = \frac{\pi}{3}$. From Observation 1, we get $\angle X_2T_0X_1 > \frac{\pi}{3}, \angle Y_1R_0Y_2 > \frac{\pi}{3}$. Therefore, we obtain $\angle X_2T_0X_1 + \angle X_1T_0I_1 + \angle I_2R_0Y_1 + \angle Y_1R_0Y_2 > \pi$, implying that the angle subtended by $X_2X_1I_1$ at T_0 plus the angle subtended by $I_2Y_1Y_2$ at R_0 is greater than π.

Using similar arguments as above, it follows that the angle subtended by $X_3X_4J_1$ at T_0 plus the angle subtended by $J_2Y_4Y_3$ is greater than π. From Observation 1, we obtain $\angle X_2T_0X_3 \geq \frac{\pi}{3}, \angle Y_2R_0Y_3 \geq \frac{\pi}{3}$. Combining all of the above results, we see that the angle subtended at T_0 by $I_1X_1X_2X_3X_4J_1$ plus the angle subtended at R_0 by $I_2Y_1Y_2Y_3Y_4J_2$ must be greater than $\pi + \pi + \frac{\pi}{3} + \frac{\pi}{3} = \frac{8\pi}{3}$. Thus, we arrive at a contradiction showing that our assumption that $U_0 = 9$ was incorrect. Therefore, $U_0 \leq 8$. □

APPENDIX II
PROOFS OF ANALYTICAL RESULTS IN SECTION V-A (LEMMA 6 AND 7)

A. Proof of Lemma 6

The proof of Lemma 6 uses a generalized version of Lemma 1 which is stated next.

Lemma 18: If $\vec{\lambda} \in \Lambda$, then $\sum_{j \in S_i \cup \{i\}} \lambda_j/\beta_j(\mathcal{N}) \leq 1$ for all sessions $i = 1, \ldots, N$.

Lemma 6 follows from Lemma 18 (proved below) and Lemma 2. □

1) Proof of Lemma 18: The broad outline of the proof is similar to that of Lemma 1. We assume that there exists a session i such that

$$\sum_{j \in S_i \cup \{i\}} \frac{\lambda_j}{\beta_j(\mathcal{N})} > 1$$

and show that $\vec{\lambda} \notin \Lambda$.

Now, note that $K_i(\mathcal{N}) \leq \beta_j(\mathcal{N})$ for every session $j \in S_i \cup \{i\}$. This is because if $j \in S_i$, then $i \in S_j$. Thus

$$\sum_{j \in S_i \cup \{i\}} \frac{\lambda_j}{K_i(\mathcal{N})} > 1$$
$$\Rightarrow \sum_{j \in S_i \cup \{i\}} \lambda_j > K_i(\mathcal{N}). \quad (33)$$

Now consider an arbitrary scheduling policy π. Under π, $\sum_{j \in S_i \cup \{i\}} D_j(n) \leq nK_i(\mathcal{N})$ for every $n \geq 0$ as at most $K_i(\mathcal{N})$ nodes among $S_i \cup \{i\}$ can be scheduled concurrently. Thus

$$\liminf_{n \to \infty} \sum_{j \in S_i \cup \{i\}} \frac{D_j(n)}{n} \leq K_i(\mathcal{N})$$
$$\Rightarrow \sum_{j \in S_i \cup \{i\}} \liminf_{n \to \infty} \frac{D_j(n)}{n} \leq K_i(\mathcal{N}) < \sum_{j \in S_i \cup \{i\}} \lambda_j \quad \text{(from (33))}$$

$$\Rightarrow \quad \liminf_{n\to\infty} \frac{D_j(n)}{n} < \lambda_j \text{ for some } j \in S_i \cup \{i\}.$$

Thus, if $\lim_{n\to\infty} \frac{D_j(n)}{n}$ exists, then its value is less than λ_j. Thus, the network is not stable under π. Alternatively, if the limit does not exist, then also the network is not stable under π. Thus, $\vec{\lambda} \notin \Lambda$. The result follows. □

B. Proof of Lemma 7

Consider a network $\mathcal{N}$ with three single-hop sessions i_1, i_2, and i_3 such that $S_{i_1} = \{i_2, i_3\}$ and $S_{i_2} = S_{i_3} = \{i_1\}$. Thus, $K_{i_1}(\mathcal{N}) = 2$ and $K_{i_2}(\mathcal{N}) = K_{i_3}(\mathcal{N}) = 1$. Let $\lambda_{i_1} = \lambda_{i_2} = \lambda_{i_3} = 1/2$. Note that a policy that schedules session i_1 in odd slots and i_2 and i_3 in the even slots stabilizes the system. Hence, $\vec{\lambda} \in \Lambda$.

Now, consider the arrival rate vector

$$(\lambda_{i_1}/K_{i_1}(\mathcal{N}), \lambda_{i_2}/K_{i_2}(\mathcal{N}), \lambda_{i_3}/K_{i_3}(\mathcal{N})) = (1/4, 1/2, 1/2)$$

which corresponds to the following arrival process: i_2 (i_3, resp.) generates a packet every even (odd, resp.) slot, and i_1 generates a packet in slots $1, 5, 9, \ldots$. Note that a maximal scheduling policy that schedules i_1 only when i_2 and i_3 do not have a packet to transmit, never schedules i_1, and is therefore unstable. Thus, $(\lambda_{i_1}/K_{i_1}(\mathcal{N}), \lambda_{i_2}/K_{i_2}(\mathcal{N}), \lambda_{i_3}/K_{i_3}(\mathcal{N})) \notin \Lambda^{\mathrm{MS}}$. □

Appendix III
Proof of Analytical Results in Section V-B (Lemma 8)

A. Proof of Lemma 8

Proof: Let $\vec{\lambda} \in \Lambda_{\mathrm{Q}}$. Then, under $\vec{\lambda}$, for some scheduling policy π, there exists a nonnegative real vector $(q_1, \ldots, q_N)$ such that for all $i, \lim_{n\to\infty} \sum_n Q_i(n)/n = q_i$ w.p. 1. Now, since $Q_i(n) = Q_i(0) + A_i(n-1) - D_i(n-1)$

$$\sum_n Q_i(n)/n = Q_i(0) + \sum_n \frac{A_i(n-1) - D_i(n-1)}{n}.$$

Thus, for all i

$$\lim_{n\to\infty} \frac{A_i(n-1) - D_i(n-1)}{n} = 0 \quad \text{w.p. 1.}$$

Since for all $i, \lim_{n\to\infty} A_i(n-1)/n = \lim_{n\to\infty} A_i(n)/n = \lambda_i$ w.p. 1, for all $i, \lim_{n\to\infty} D_i(n)/n = \lim_{n\to\infty} D_i(n-1)/n = \lambda_i$ w.p. 1. Thus, $\vec{\lambda} \in \Lambda$. Thus, from Lemma 18, for all i, $\sum_{j\in S_i\cup\{i\}} \lambda_j/\beta_j(\mathcal{N}) \le 1$. Thus

$$\sum_{j\in S_i\cup\{i\}} \lambda'_j < 1 \quad \forall\, i. \tag{34}$$

Let the arrival rate vector be $(\lambda'_1, \ldots, \lambda'_N)$. Consider a maximal scheduling policy. Let the state of the arrival process in the end of slot n be $\vec{B}(n)$. Clearly, $(\vec{Q}(n), \vec{B}(n))$ constitutes an irreducible aperiodic Markov chain. Let $\theta_j(t)$ denote the number of departures for session j in slot t.

Consider the Lyapunov function $f(t)$, where

$$f(t) = \sum_i \sum_{j\in S_i\cup\{i\}} Q_i(t)Q_j(t).$$

Clearly, $f(t) > 0$ if $Q_i(t) > 0$ for some i

$$\begin{aligned}
&\mathbb{E}[f(n+1) - f(n)|\vec{Q}(n), \vec{B}(n)] \\
&= \sum_i \sum_{j\in S_i\cup\{i\}} \mathbb{E}[Q_i(n+1)Q_j(n+1) \\
&\qquad - Q_i(n)Q_j(n)|\vec{Q}(n), \vec{B}(n)] \\
&= \sum_i \sum_{j\in S_i\cup\{i\}} \mathbb{E}[(Q_i(n) + \alpha_i(n) - \theta_i(n)) \\
&\qquad \times (Q_j(n) + \alpha_j(n) - \theta_j(n)) \\
&\qquad - Q_i(n)Q_j(n)|\vec{Q}(n), \vec{B}(n)] \\
&= \sum_i \sum_{j\in S_i\cup\{i\}} \mathbb{E}[(Q_i(n) + \alpha_i(n) - \theta_i(n)) \\
&\qquad \times (Q_j(n) + \alpha_j(n) - \theta_j(n)) \\
&\qquad - Q_i(n)Q_j(n)|\vec{Q}(n), \vec{B}(n)] \\
&\le \sum_i \sum_{j\in S_i\cup\{i\}} \mathbb{E}[Q_i(n)\alpha_j(n) - Q_i(n)\theta_j(n) \\
&\qquad + Q_j(n)\alpha_i(n) - Q_j(n)\theta_i(n)|\vec{Q}(n), \vec{B}(n)] \\
&\quad + (N+1)N(\alpha_{\max}^2 + 1).
\end{aligned}$$

Now

$$\sum_i \sum_{j\in S_i\cup\{i\}} Q_i(n)\alpha_j(n) = \sum_i \sum_{j\in S_i\cup\{i\}} Q_j(n)\alpha_i(n),$$

and

$$\sum_i \sum_{j\in S_i\cup\{i\}} Q_i(n)\theta_j(n) = \sum_i \sum_{j\in S_i\cup\{i\}} Q_j(n)\theta_i(n).$$

Thus

$$\begin{aligned}
&\mathbb{E}[f(n+1) - f(n)|\vec{Q}(n), \vec{B}(n)] \\
&\quad \le 2\sum_i Q_i(n) \sum_{j\in S_i\cup\{i\}} \mathbb{E}[\alpha_j(n) - \theta_j(n)|\vec{Q}(n), \vec{B}(n)] \\
&\qquad + (N+1)N(\alpha_{\max}^2 + 1) \\
&\mathbb{E}[f(n+\tau) - f(n)|\vec{Q}(n), \vec{B}(n)] \\
&\quad \le 2\sum_i Q_i(n) \Bigg[\sum_{j\in S_i\cup\{i\}} \sum_{k=0}^{\tau-1} \alpha_j(n+k) \\
&\qquad - \mathbb{E}\Big[\sum_{j\in S_i\cup\{i\}} \sum_{k=0}^{\tau-1} \theta_j(n+k)|\vec{Q}(n), \vec{B}(n) \Big] \Bigg] \\
&\qquad + (N+1)N(\alpha_{\max}^2 + 1)\tau.
\end{aligned}$$

Under maximal scheduling, if $Q_i(n) > \tau + 1$, $\sum_{j\in S_i\cup\{i\}} \theta_j(l) = 1$ for each $l \in [n, n+\tau-1]$. Thus, if $Q_i(n) > \tau + 1, \sum_{j\in S_i\cup\{i\}} \sum_{k=0}^{\tau-1} \theta_j(n+k) = \tau$. Next, let $\delta = 1 - \max_i \sum_{j\in S_i\cup\{i\}} \lambda'_j$. From (34), $\delta > 0$. Since the arrival process is a positive recurrent Markov chain, thus for any $\vec{Q}(n), \vec{B}(n)$ there exists τ_0 such that for all $\tau \ge \tau_0$

$$\sum_{k=0}^{\tau-1} \alpha_j(n+k) \le \tau(\lambda'_j + \delta/2N).$$

Thus, for all $\vec{Q}(n)$, and for $\tau \geq \tau_0$

$$\begin{aligned}\mathbb{E}[f(n+\tau)-f(n)|\vec{Q}(n)=\vec{Q},\vec{B}(n)=\vec{B}] \\ \leq -\delta\tau \sum_{i:Q_i(n)>\tau+1} Q_i(n) \\ +(N+1)N(\alpha_{\max}^2+\alpha_{\max}+1)(\tau+1).\end{aligned}$$

Thus, for $\tau \geq \tau_0$

$$\mathbb{E}[f(n+\tau)-f(n)|\vec{Q}(n)=\vec{Q},\vec{B}(n)=\vec{B}]<\infty$$

for all $\vec{Q},\vec{B}$, and

$$\mathbb{E}[f(n+\tau)-f(n)|\vec{Q}(n)=\vec{Q},\vec{B}(n)=\vec{B}]<-1$$

for all $\vec{Q},\vec{B}$ such that

$$\max_i Q_i > \max\left(\tau+1, \frac{(N+1)N(\alpha_{\max}^2+\alpha_{\max}+1)(\tau+1)}{\delta\tau}\right).$$

Hence, by Foster's theorem [8, Theorem 2.2.3], for each $\tau \geq \tau_0, t \in (0,\tau-1), (\vec{Q}(t),\vec{B}(t)), (\vec{Q}(t+\tau),\vec{B}(t+\tau)), (\vec{Q}(t+2\tau),\vec{B}(t+2\tau)),\ldots$, is a positive recurrent Markov chain. Also, all these Markov chains have the same set of states, and same transition probabilities. Thus, under maximal scheduling, there exists a nonnegative real vector $(q_1,\ldots,q_N)$ such that for all i, $\lim_{n\to\infty}\sum_{m=1}^{n} Q_i(m)/n = q_i$ w.p. 1. Thus, $(\lambda_1',\ldots,\lambda_N') \in \Lambda_{\mathrm{Q}}^{\mathrm{MS}}$. □

Appendix IV
Proofs of Analytical Results in Section V-C (Lemmas 9 and 10)

A. Proof of Lemma 9

We prove Lemma 9 using the following supporting lemmas.

Lemma 19: If $\vec{\lambda} \in \Lambda$, then $\sum_{k\in S_j\cup\{j\}} \lambda_{q(k)}/\tilde{\beta}_{q(k)}(\mathcal{N}) \leq 1$, for all session-links $j=1,\ldots,M$.

Lemma 20: Let

$$\vec{\lambda} \in \left\{\vec{\lambda}: \text{if } \lambda_{q(k)}>0, \sum_{k\in S_j\cup\{j\}} \lambda_{q(k)} \leq 1,\ j=1,\ldots,M\right\}.$$

Then $\vec{\lambda} \in \Lambda^{\mathrm{MS}}$.

Lemma 9 follows from Lemmas 19 and 20, proved below. □

1) Proof of Lemma 19: Let there exist a session-link j such that

$$\sum_{k\in S_j\cup\{j\}} \frac{\lambda_{q(k)}}{\tilde{\beta}_{q(k)}(\mathcal{N})} > 1.$$

We will show that $\vec{\lambda} \notin \Lambda$. Now, since $\beta_k(\mathcal{N}) \leq \tilde{\beta}_{q(k)}(\mathcal{N})$

$$\sum_{k\in S_j\cup\{j\}} \frac{\lambda_{q(k)}}{\beta_k(\mathcal{N})} > 1.$$

Also, note that $K_j(\mathcal{N}) \leq \beta_k(\mathcal{N})$ for every session-link $k \in S_j\cup\{j\}$. Thus

$$\begin{aligned}&\sum_{k\in S_j\cup\{j\}} \frac{\lambda_{q(k)}}{K_j(\mathcal{N})} > 1 \\ &\Rightarrow \sum_{k\in S_j\cup\{j\}} \lambda_{q(k)} > K_j(\mathcal{N}). \qquad (35)\end{aligned}$$

Now consider an arbitrary scheduling policy π. Under $\pi, \sum_{k\in S_j\cup\{j\}} D_k(n) \leq nK_j(\mathcal{N})$ for every $n \geq 0$ as at most $K_j(\mathcal{N})$ nodes among $S_j\cup\{j\}$ can be scheduled concurrently. Thus

$$\begin{aligned}&\liminf_{n\to\infty} \sum_{k\in S_j\cup\{j\}} \frac{D_k(n)}{n} \leq K_j(\mathcal{N}) \\ \Rightarrow\ &\sum_{k\in S_j\cup\{j\}} \liminf_{n\to\infty} \frac{D_k(n)}{n} \leq K_j(\mathcal{N}) \\ &< \textstyle\sum_{k\in S_j\cup\{j\}} \lambda_{q(k)} \quad \text{(from (35))} \\ \Rightarrow\ &\liminf_{n\to\infty} \frac{D_k(n)}{n} < \lambda_{q(k)} \quad \text{for some } k \in S_j\cup\{j\} \\ \Rightarrow\ &\liminf_{n\to\infty} \frac{D_{L_{q(k)}}(n)}{n} < \lambda_{q(k)}.\end{aligned}$$

The last inequality follows since $D_{L_{q(k)}}(n) \leq D_k(n)$ for all k,n. Thus, if $\lim_{n\to\infty} \frac{D_{L_{q(k)}}(n)}{n}$ exists, then its value is less than $\lambda_{q(k)}$. Hence, the network is not stable under π. Alternatively, if the limit does not exist, then also the network is not stable under π. Thus, $\vec{\lambda} \notin \Lambda$. The result follows. □

2) Proof of Lemma 20: We outline this proof as it is similar to that for Lemma 2. Recall that $A_j(n)$ and $D_j(n)$, respectively, denote the arrivals in and departures from session-link j in the time duration $(0,n]$. With regulators, the source of each session-link has two queues: waiting-queue and release-queue. If we only focus on the release-queue of session-link j, note that departure process from the queue is the same as $D_j(\cdot)$. Let $\tilde{A}_j(n)$ denote the arrivals at the release-queue of session-link j in the time duration $(0,n]$. Then, $\tilde{A}_j(n) = A_j(n)$ if j is the first session-link in its session, and $\tilde{A}_j(n) \leq A_j(n)$ otherwise. Let $\tilde{Q}_j(n)$ denote the queue length at the release-queue of session-link j at the beginning of the nth slot.

For $j=1,\ldots,M$, the fluid limits of $\tilde{A}_j(\cdot), D_j(\cdot), \tilde{Q}_j(\cdot)$ are defined as in Section I-A.1; let $\overline{A}_j(\cdot), \overline{D}_j(\cdot), \overline{Q}_j(\cdot)$ denote the respective fluid limits.

Now, we state and prove some important properties of the fluid limits which we use to prove Lemma 20.

Lemma 21: Every fluid limit satisfies $\overline{A}_j(t) \leq \lambda_{q(j)}t$ w.p. 1 for every session-link $j=1,\ldots,M$, and $t \geq 0$.

Proof: The proof is similar to that for Lemma 14 when j is the first session-link of its session. When j is not the first session-link of its session, the proof follows because due to the regulator, the release-queue of j receives packet w.p. at most $\lambda_{q(j)}$ in any slot n. □

Lemma 22: Any fluid limit $(\overline{A}_j, \overline{D}_j, \overline{Q}_j), j = 1, \ldots, M$, satisfies the following equality for every j and $t \geq 0$ with probability (w.p.) 1:

$$\overline{Q}_j(t) = \overline{Q}_j(0) + \overline{A}_j(t) - \overline{D}_j(t). \tag{36}$$

The proof is similar to that for Lemma 15.

Lemma 23: Let $\overline{Q}_j(0) = 0$ for every j. Also, let $\sum_{k \in S_j \cup \{j\}} \lambda_{q(k)} \leq 1$ if $\lambda_{q(j)} > 0$, $j = 1, \ldots, M$. Then, under maximal scheduling, every fluid limit satisfies $\overline{Q}_j(t) = 0$ for every $t \geq 0$ w.p. 1 for every j.

The lemma follows from Lemma 21. The arguments are similar to that in the Proof of Lemma 16.

We now prove Lemma 20.

Proof: We prove the following for each session-link $j = 1, \ldots, M$:

i) $\overline{A}_j(t) = \lambda_{q(j)} t$ w.p. 1 for every $t \geq 0$;
ii) $\overline{D}_j(t) = \lambda_{q(j)} t$ w.p. 1 for every $t \geq 0$;
iii) $\lim_{t \to \infty} D_j(t)/t = \lambda_{q(j)}$ w.p. 1.

We prove using induction on the position of the session-links in the paths of their sessions.

First, let j be the first session-link of some session (i.e., the session-link originating at the source of the session). The arrivals in the release-queue of the first session-link are the exogenous arrivals. Now, i) follows from (6). From Lemmas 22 and 23, $\overline{D}_j(t) = \overline{A}_j(t)$ w.p. 1 for every $t \geq 0$. Now, ii) follows from i). Finally, using arguments similar to those in the proof Lemma 2, and using i) and ii), we obtain iii).

Now, let i) and ii) hold for the 1st, 2nd, $\ldots$, pth session-links in the path of the session in consideration. We now prove i) and ii) for a session-link j that is the $(p+1)$th in the path of the session. Let session-link k be the session-link of session $q(j)$ that terminate at the source of session-link j. Let $\hat{Q}_j(n)$ be the queue length at the waiting-queue of session-link j at the beginning of the nth slot. Now

$$\hat{Q}_j(n+1) = \hat{Q}_j(0) + D_k(n) - \tilde{A}_j(n).$$

From iii) of induction hypothesis $\lim_{t \to \infty} D_k(t)/t = \lambda_{q(j)}$ w.p. 1. Note that $\tilde{A}_j(n) = 1$ w.p. $\lambda_{q(j)}$ if $\hat{Q}_j(n) > 0$. Thus, the waiting-queue of session-link j is a queue which receives packets as per an arrival process that satisfies SLLN with rate $\lambda_{q(j)}$ and is served w.p. $\lambda_{q(j)}$ whenever it is nonempty. It follows that the departure-process of this queue satisfies SLLN with rate $\lambda_{q(j)}$. Thus, i) follows. Now, ii) and iii) follow as in the base case.

The lemma follows from iii). □

B. Proof of Lemma 10

We prove Lemma 10 using Lemma 19 and another supporting lemma, Lemma 24, which we state and prove next.

Lemma 24: Consider an arrival rate vector $\vec{\lambda}'$ such that $\sum_{k \in S_j \cup \{j\}} \lambda'_{q(k)} < 1$ for all session-links $j = 1, \ldots, M$. Then the packet queue of every session-link will almost surely become empty infinitely often. Furthermore, for every session-link j and time t, $\mathbb{E}[B_{j,t}] < \infty$.

Proof: Let $\alpha_j(t)$ and $\theta_j(t)$ denote the number of arrivals and departures, respectively, for session-link j in slot t. Let $Q_j(t)$ be the number of packets for the session of session-link j waiting for transmission at the source of session-link j at the end of slot t. Let $S_j \cup \{j\} = \mathcal{X}_j$, and $\widehat{n} = |\mathcal{X}_j|$. If session-link j satisfies $Q_j(\nu) > 0$ for every $\nu \in [t, t+\tau]$, then for every $\nu \in [t, t+\tau]$

$$\sum_{k \in \mathcal{X}_j} \theta_k(\nu) \geq 1. \tag{37}$$

$$\begin{aligned} Q_j(t) + \sum_{\nu=t+1}^{t+\tau} \alpha_j(\nu) &\leq \sum_{\nu=1}^{t+\tau} A_{q(j)}(\nu) \\ &\leq t\alpha_{\max} + \sum_{\nu=t+1}^{t+\tau} A_{q(j)}(\nu). \end{aligned} \tag{38}$$

Now we have

$$\begin{aligned} &\mathbb{P}\{B_{j,t} > \tau\} \\ &\leq \mathbb{P}\left\{ \bigcap_{v=t}^{t+\tau} \left\{ \sum_{k \in \mathcal{X}_j} Q_k(t) + \sum_{\nu=t+1}^{v} \sum_{k \in \mathcal{X}_j} \alpha_k(\nu) \right.\right. \\ &\qquad \left.\left. - \sum_{\nu=t+1}^{v} \sum_{k \in \mathcal{X}_j} \theta_k(\nu) > 0 \right\} \right\} \\ &\leq \mathbb{P}\left\{ \bigcap_{v=t+1}^{t+\tau} \left\{ \sum_{k \in \mathcal{X}_j} Q_k(t) \right.\right. \\ &\qquad \left.\left. + \sum_{\nu=t+1}^{v} \left(\sum_{k \in \mathcal{X}_j} \alpha_k(\nu) - 1 \right) > 0 \right\} \right\} \text{ (from (37))} \\ &\leq \mathbb{P}\left\{ \sum_{k \in \mathcal{X}_j} Q_k(t) \right. \\ &\qquad \left. + \sum_{\nu=t+1}^{t+\tau} \sum_{k \in \mathcal{X}_j} \alpha_k(\nu) - \tau > 0 \right\} \\ &\leq \mathbb{P}\left\{ \frac{t\widehat{n}\alpha_{\max}}{\tau} + \frac{1}{\tau} \sum_{\nu=t+1}^{t+\tau} \sum_{k \in \mathcal{X}_j} A_{F_{q(k)}}(\nu) - 1 > 0 \right\} \\ &\qquad \text{(from (38))} \\ &= \mathbb{P}\left\{ \frac{t\widehat{n}\alpha_{\max}}{\tau} + \sum_{k \in \mathcal{X}_j} \left(\frac{1}{\tau} \sum_{\nu=t+1}^{t+\tau} A_{F_{q(k)}}(\nu) \right.\right. \\ &\qquad \left.\left. - \lambda'_{q(k)} \right) > 1 - \sum_{k \in \mathcal{X}_j} \lambda'_{q(k)} \right\}. \end{aligned}$$

Let $\delta = 1 - \sum_{k \in \mathcal{X}_j} \lambda'_{q(k)}$. Clearly, $\delta > 0$. Thus

$$\begin{aligned} &\mathbb{P}\{B_{j,t} > \tau\} \\ &\leq \mathbb{P}\left\{ \left\{ \frac{t\widehat{n}\alpha_{\max}}{\tau} > \frac{\delta}{\widehat{n}+1} \right\} \right. \end{aligned}$$

$$\times \bigcup_{k\in\mathcal{X}_j}\left\{\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{F_{q(k)}}(\nu)-\lambda'_{q(k)}>\frac{\delta}{\widehat{n}+1}\right\}\Bigg\}$$

$$\le \mathbb{P}\left\{\frac{t\widehat{n}\alpha_{\max}}{\tau}>\frac{\delta}{\widehat{n}+1}\right\} + \sum_{k\in\mathcal{X}_j}\mathbb{P}\left\{\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{F_{q(k)}}(\nu)-\lambda'_{q(k)}>\frac{\delta}{\widehat{n}+1}\right\}$$

$$= \sum_{k\in\mathcal{X}_j}\mathbb{P}\left\{\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{F_{q(k)}}(\nu)-\lambda'_{q(k)}>\frac{\delta}{\widehat{n}+1}\right\}, \quad \text{if } \tau>\frac{\widehat{n}(\widehat{n}+1)t\alpha_{\max}}{\delta}.$$

Therefore, from (8), the packet queue of every session-link will almost surely become empty infinitely often. Also

$$\mathbb{E}[B_{j,t}]=\sum_{\tau=1}^{\infty}\mathbb{P}\{B_{j,t}>\tau\}<\infty.$$

Lemma 10 follows from Lemma 19 and Lemma 24. □

REFERENCES

[1] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," *IEEE J. Sel. Areas Commun.: Special Issue on Multihop Wireless Mesh Networks*, vol. 24, no. 11, pp. 1960–1971, Nov. 2006.

[2] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1992.

[3] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint asynchronous congestion control and distributed scheduling for multihop wireless networks," in *Proc. INFOCOM*, Barcelona, Spain, Apr. 2006.

[4] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees through maximal scheduling in multihop wireless networks," in *Proc. 43rd Annu. Allerton Conf. Communication, Control and Computing*, Monticello, IL, Sep. 2005.

[5] P. Chaporkar, K. Kar, and S. Sarkar, "Achieving queue length stability through maximal scheduling in wireless networks," in *Proc. Information Theory and Applications Inaugural Workshop*, San Diego, La Jolla, CA, Feb. 2006.

[6] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input output queued switch," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1030–1039, Dec. 1999.

[7] J. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. INFOCOM*, Tel-Aviv, Israel, Mar. 2000, pp. 556–564.

[8] G. Fayolle, V. A. Malyshev, and M. V. Menshikov, *Topics in the Constructive Theory of Countable Markov Chains*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[9] S. Kravitz, "Packing cylinders into cylindrical containers," *Math. Mag.*, vol. 40, pp. 65–70, 1967.

[10] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proc. INFOCOM*, Miami, FL, Mar. 2005, vol. 3, pp. 1804–1814.

[11] M. Luby, "An efficient randomized algorithm for input-queued switch scheduling," *SIAM J. Comput.*, vol. 15, no. 4, pp. 1036–1055, Nov. 1986.

[12] M. V. Marathe, H. Breu, H. B. Hunt, III, S. S. Ravi, and D. J. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol. 25, pp. 59–68, 1995.

[13] D. Peleg, *Distributed Computing: A Locality-sensitive Approach*. Philadelphia, PA: Soc. Ind. Appl. Math., 2000.

[14] W. Rudin, *Real and Complex Analysis*. New York: McGraw Hill, 1987.

[15] S. Sarkar, P. Chaporkar, and K. Kar, "Fairness and throughput guarantees with maximal scheduling in multihop wireless networks," in *Proc. 4th Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Boston, MA, Apr. 2006.

[16] S. Sarkar and K. N. Sivarajan, "Fairness in wireless mobile networks," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2412–2426, Aug. 2002.

[17] S. Sarkar and L. Tassiulas, "End-to-end bandwidth guarantees through fair local spectrum share in wireless ad-hoc networks," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1246–1259, Sep. 2005.

[18] D. Shah, P. Giaccone, and B. Prabhakar, "An efficient randomized algorithm for input-queued switch scheduling," *IEEE Micro.*, vol. 22, no. 1, pp. 19–25, Jan./Feb. 2002.

[19] G. Sharma, N. Shroff, and R. Mazumdar, "On the complexity of scheduling in multi-hop wireless systems," in *Proc. IEEE Int. Workshop on Foundations and Algorithms for Wireless Networking (FAWN)*, Pisa, Italy, Mar. 2006.

[20] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proce. INFOCOM*, San Francisco, CA, Mar. 1998, pp. 533–539.

[21] L. Tassiulas and A. Ephremidis, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[22] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless ad hoc networks," *IEEE J. Sel. Areas Commun., Special Issue on Ad Hoc Networks, Part I*, vol. 23, no. 1, pp. 163–173, Jan. 2005.

[23] X. Wu and R. Srikant, "Regulated maximal matching: A distributed scheduling algorithm for multihop wireless networks with node-exclusive spectrum sharing," in *Proc. IEEE CDC-ECC'05*, Seville, Spain, Dec. 2005, pp. 5342–5347.

[24] X. Wu and R. Srikant, "Bounds on the capacity region of multihop wireless networks under distributed greedy scheduling," in *Proc. INFOCOM*, Barcelona, Spain, Apr. 2006.

[25] X. Wu, R. Srikant, and J. R. Perkins, "Queue length stability of maximal greedy schedules in wireless networks," in *Proc. Information Theory and Applications Inaugural Workshop*, San Diego, La Jolla, CA, Feb. 2006.