4-1-2006

Fairness and Throughput Guarantees with Maximal Scheduling in Multi-hop Wireless Networks

Saswati Sarkar
University of Pennsylvania, swati@seas.upenn.edu

Prasanna Chaporkar
INRIA

Koushik Kar
Rensselaer Polytechnic Institute

Suggested Citation:
Sarkar, S., P. Chaporkar, and K. Kar, "Fairness and Throughput Guarantees with Maximal Scheduling in Multi-hop Wireless Networks," *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium*. pp. 1- 13, 03-06 April 2006

Fairness and Throughput Guarantees with Maximal Scheduling in Multi-hop Wireless Networks

Saswati Sarkar
University of Pennsylvania
Philadelphia, PA, USA
swati@seas.upenn.edu

Prasanna Chaporkar
INRIA
Paris, France
Prasanna.Chaporkar@ens.fr

Koushik Kar
Rensselaer Polytechnic Institute
Troy, NY, USA
koushik@ecse.rpi.edu

***Abstract*— We investigate the fairness and throughput properties of a simple distributed scheduling policy, maximal scheduling, in the context of a general ad-hoc wireless network. We design a fully distributed algorithm that combines a token generation scheme with maximal scheduling policy so as to attain maxmin fair rates within the feasible region of maximal scheduling. We next present throughput guarantees of maximal scheduling that quantify the performance loss of each session due to the use of local information based scheduling. We show that the performance loss for each session depends on the maximum "interference degree" in its neighborhood. We also demonstrate that the performance penalties can not be localized any further.**

I. Introduction

Fairness and throughput are two important metrics that characterize the performance of any wireless network. The question of providing throughput and fairness guarantees through intelligent link (packet) scheduling has received significant attention in recent literature. However, most of the scheduling algorithms proposed in this context have either been centralized [8], [10], [11], [12], or did not have any analytical guarantee [6], [7]. Characterization of the fairness and throughput properties of distributed scheduling strategies have remained largely unexplored; in this paper, we take a step towards addressing these important issues.

We first investigate fair allocation of bandwidth using distributed scheduling in multi-hop wireless networks. Attaining fairness guarantees using distributed scheduling requires obtaining throughput guarantees through distributed resolution of medium access contention, and the latter remained an open problem for some time. Recently, some progress has been made towards solving the above open problem using a simple distributed scheduling strategy, *maximal scheduling*. The maximal scheduling policy 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 can not simultaneously transmit with (u, v) is scheduled for transmission; the scheduling is otherwise arbitrary. The maximal scheduling policy can be implemented in a distributed manner with only local state information at each node [9]. It has been shown that maximal scheduling is guaranteed to attain a certain fraction of the throughput region in arbitrary wireless networks [4], [14]; this fraction turns out to be a constant in certain special cases [4], [5], [13]. Due to its simplicity and the analytical guarantees obtained above, maximal scheduling is likely to find extensive applications in large scale multihop wireless networks in near future.

Nevertheless, maximal scheduling is really a class of policies, and some policies in this class could allocate bandwidth very unfairly. Recently, Lin *et al.* [5] and Bui *et. al.* [2] have shown that in a specific interference model, the node exclusive spectrum sharing model, maximal scheduling can be used for maximizing the network utility and congestion control. One of our important contributions is to show that maxmin fairness can be attained in wireless networks with arbitrary interference models within the framework of maximal scheduling, without sacrificing the simplicity and the distributed nature of these policies. Using the characterizations for 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. 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.

We next compare the throughput region of maximal scheduling with the maximum possible throughput region of the network. This comparison characterizes the penalty due to the use of only local information in the scheduling. A common feature of all the existing results in this context has been that same performance bounds are obtained for all sessions [2], [4], [5], [13], [14]. This uniform characterization therefore bounds the performance of the network in terms of that of the worst session. However, depending on the interference in individual neighborhoods, different sessions may be able to

This work was supported by the National Science Foundation under grants NCR-0238340, CNS-0435306 and CNS-0435141.

accommodate different arrival rates. The natural next question now is whether it is possible to obtain better non-uniform bounds by considering the constraints of individual sessions. Let the interference degree of a link l in a session's path be the number of links that interfere with l but do not interfere with each other. We prove that under maximal scheduling the performance of each session can be characterized by the interference degree of only the links in its path, and the interference degrees of the neighbors of these links. Thus the performance penalty for each session, due to the use of local information based scheduling, depends only on the neighborhoods of the links in its path. 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 prove that the performance penalties under maximal scheduling can not be localized any further. Specifically, the interference degrees of the links of a session alone can not determine its throughput guarantee.

The paper is organized as follows. We describe our system model in Section II. We describe the fairness and throughput guarantees in Sections III and IV respectively. We prove the analytical results in the appendix.

II. System Model

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 network consists of N end-to-end sessions, indexed as $1, \ldots, N$. Each end-to-end 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 (u, v), where u and v represent the transmitter and the receiver, respectively, of the corresponding session-links. 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., $q(j) = \{i : j \in P_i\}$. We assume that there are a total of M session-links in the network (over all sessions), and these are indexed by $1, \ldots, M$.

We now introduce the notion of interference. A session-link j *interferes* with session-link k if k can not 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. 1(a)).

We now describe the arrival process. We assume that time is slotted. 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. We assume that at most $\alpha_{\max}$ packets arrive for any session in any slot. Further, there exists a constant $\hat{\alpha} > 1$ and an arrival rate vector $\vec{\lambda} = (\lambda_1, \ldots, \lambda_N)$ such that the empirical average of the 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 $i \in \{1, \ldots, m\}$, $T \geq \widehat{t}_\delta$, and $\delta > 0$,

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

Note that a large class of arrival processes, e.g., periodic, i.i.d., and Markovian arrival processes with finite state space, satisfy the above assumption. For simplicity, we will sometimes consider a special case of the above general model. Specifically, we will consider the "bounded-burstiness" arrival model where there exists a burstiness vector $\vec{\sigma} = (\sigma_1, \ldots, \sigma_N)$ such that

$$|A_i(t) - \lambda_i t| \leq \sigma_i \ \forall \ t. \tag{2}$$

Whenever we use the above special case, we will explicitly state so.

A *scheduling policy* is an algorithm that decides in each slot the subset of session-links that would transmit packets in the slot. Clearly, a subset of session-links can transmit packets in any slot if no two session-links in the subset interfere with each other, and every session-link in the subset has a packet to transmit.

We assume that every packet has length 1 slot. Let $D_j(n)$ be the number of packets that session-link j transmits in interval $(0, n]$, $j = 1, \ldots, M$. Let L_i be the session-link corresponding to the last hop of session i. Clearly the transmissions depend on the scheduling policy. If for some constant d_i, the limit $\lim_{n\to\infty} D_{L_i}(n)/n = d_i$ with probability 1, then d_i is denoted as the departure rate of session i.

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

$$\lim_{n\to\infty} D_{L_i}(n)/n = d_i = \lambda_i, \ i = 1, \ldots, N. \tag{3}$$

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

Definition 2: 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 $\vec{\lambda}$. The *maximum throughput region* is the union of the throughput region of all policies.

We now describe the "maximal scheduling" policy we consider. This policy schedules a subset S of session-links such that (i) every session-link in S has a packet to transmit, (ii) no session-link in S interferes with any other session-link in S, (iii) if a session-link i has a packet to transmit, then either i or a session-link in S_i, is included in S. Clearly, many subsets of session-links satisfy the above criteria in each slot. Maximal scheduling can select any such subset, and can be implemented in distributed manner using standard algorithms [9].

Let Λ and Λ^{MS} respectively denote the maximum throughput region, and the throughput region attained by maximal scheduling.

If a rate vector $\vec{\lambda} = (\lambda_1, \ldots, \lambda_N)$ satisfies

$$\sum_{k \in S_j \cup \{j\}} \lambda_{q(k)} \quad \leq \quad 1, \quad \forall \ j = 1, \ldots, M, \tag{4}$$

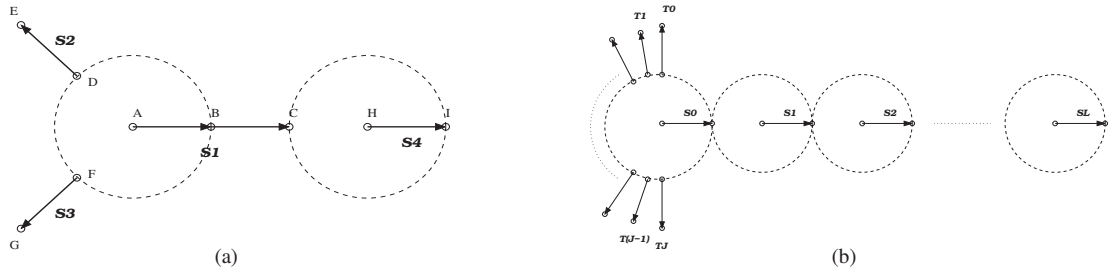

Fig. 1. In both figures, all sessions and session-links are unidirectional, and the arrows show the direction of data transfer. The circles indicate the interference regions of session-links AB and HI (Fig. (a)) and S0, S1, ..., SL (Fig. (b)).
In Fig. (a), session S1 consists of two session-links, AB and BC, whereas sessions S2, S3, S4 are single-hop sessions. Session-link AB interferes with session-links DE (session S2) and FG (session S3) and session-link HI (session S4) interferes with session-link BC. Now, $S_{\text{AB}} = \{\text{BC, DE, FG}\}, S_{\text{BC}} = \{\text{AB, HI}\}, S_{\text{DE}} = S_{\text{FG}} = \{\text{AB}\}, S_{\text{HI}} = \{\text{BC}\}$. 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 ($1, 2$ 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.
In Fig. (b), network consists of single-hop sessions only. Session S0 interferes with sessions T0, ..., TJ, whereas session Si interferes with session S(i-1), for i=1,2, ..., L. Thus, $K_i(\mathcal{N}) = 1$ for $i \in \{\text{T0}, \ldots \text{TJ, SL}\}$, $K_i(\mathcal{N}) = 2$ for $i \in \{\text{S1}, \ldots \text{S(L-1)}\}$, $K_{\text{S0}}(\mathcal{N}) = J{+}2$, $\beta_i(\mathcal{N}) = J{+}2$ for $i \in \{\text{T0}, \ldots \text{TJ, S0, S1}\}$, and $\beta_i(\mathcal{N}) = 2$ for $i \in$ S2, ..., SL, $K(\mathcal{N}) = (J + 2)$. If J and L are large, but L $\gg$ J, then K_i, β_i for most sessions are substantially smaller than $K(\mathcal{N})$.

then $\vec{\lambda} \in \Lambda^{\text{MS}}$ [3], [4]. Thus, the above constraints specify a sufficiency condition for the stability of a network under maximal scheduling. The constraints follow since any maximal scheduling always schedules at least one session-link in $S_j \cup \{j\}$ if session-link j has a packet to transmit. The above constraints are also necessary for the stability of any network under maximal scheduling in the following sense. Given any network $\mathcal{N}$, if an arrival rate vector $\vec{\lambda}$ does not satisfy (4), then for some maximal scheduling policy, $\vec{\lambda} \notin \Lambda^{\text{MS}}$ in $\mathcal{N}$ [3].

Fairness issues are particularly relevant when the arrival rate vector is not in Λ^{MS}, because then maximal scheduling can not serve all sessions at their arrival rates, and therefore it is necessary to fairly allocate the service rates or departure rates of sessions. We introduce the notion of the *feasible set* Δ^{MS} *of departure rate vectors* $\vec{d} = (d_1, \ldots, d_N)$ which can be described as follows:

$$\sum_{k \in S_j \cup \{j\}} d_{q(k)} \quad \leq \quad 1, \quad \forall\ j = 1, \ldots, M, \tag{5}$$

(interference constraints)

$$d_i \quad \leq \quad \lambda_i \quad \forall\ i = 1, \ldots, N. \tag{6}$$

The "interference constraints" (5) capture the interference relations and are analogous to constraints (4) for the stability region. The constraints (6) follow since the departure rates can not exceed the arrival rates.

Note that $\Delta^{\text{MS}} \subseteq \Lambda^{\text{MS}}$. When $\vec{\lambda} \in \Lambda^{\text{MS}}$, the departure rate vector satisfies $d_i = \lambda_i$ for each i and hence both (5) and (6) hold. When $\vec{\lambda} \notin \Lambda^{\text{MS}}$, 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, when the network consists of only single-hop sessions, 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.

We now formally define the notion of maxmin fairness that we seek to attain. For any N-dimensional vector a, let $\mathcal{I}(a)$ denote a non-decreasing 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 $\hat{a}_1 \leq \hat{a}_2 \leq \ldots \leq \hat{a}_N$. A departure rate vector $\vec{d^*}$ is said to be maxmin fair if $\vec{d^*} \in \Delta^{\text{MS}}$, and for any other departure rate vector $\vec{d} \in \Delta^{\text{MS}}$, the first non-zero 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 [1]. Note that $\vec{d^*} \in \Lambda^{\text{MS}}$ as $\Delta^{\text{MS}} \subseteq \Lambda^{\text{MS}}$. Finally, if $\vec{\lambda} \in \Lambda^{\text{MS}}$, then $\vec{d^*} = \vec{\lambda}$.

Finally, 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 3: 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) $d_i \geq d_k$ for all other sessions k whose session-links are associated with the constraint and (c) the inequality in the constraint is an equality.

Lemma 1: A departure rate vector $\vec{d} \in \Delta^{MS}$ 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 [1].

III. Max-Min Fairness under Maximal Scheduling

We propose a modular approach for attaining maxmin fairness using maximal scheduling. The first module estimates the maxmin fair bandwidth share of each session in each node in the session's path, and releases packets for transmission in accordance with these estimates. The second module schedules

the transmission of the released packets so as to attain the estimates. Note that the modules operate in parallel.

We first describe the algorithm for the special case that $\lambda_i > 1$ for each i and thus every session always has a packet to transmit (saturated sessions) and every session spans one link. We summarize the algorithm in Figure 2. We next motivate the changes required for the general case when all sessions may not always have packets to transmit, and sessions traverse multiple links. At the end of the section, we present the performance guarantees.

Let every session consists of only one session-link ($N = M$) and $\lambda_i > 1$ for each i. Fair bandwidth is estimated by a token generation process. The source node for each session i maintains a token bucket for i (Fig. 1(a)). The token bucket consists of a token-queue for each session in $S_i \cup \{i\}$. Every token bucket generates tokens for all token-queues in it. The token generation process is so designed that each token-queue receives tokens at a rate that equals the maxmin fair departure rate of the corresponding session (we shortly describe how this can be done). 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 describe the token generation process for each token-bucket. A session i is associated with $b_i = |S_i| + 1$ token-buckets, one for each of the sessions it interferes with, and itself. Let us denote these token-buckets as $1, \ldots, b_i$. 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 session i be sampled in slot t. Then, k' generates a token for session i in slot t if and only if $C_{i,k'}(t) < W + \min\left(C_{i,k'-1}(t), C_{i,k'+1}(t)\right)$. Thus, session i receives a token at bucket k' unless the number of tokens for session i at k' substantially exceeds that at the adjacent buckets; this prohibitive difference is the window parameter, W. In slot t, k' samples the next session in the bucket in a round robin order if and only if k' does not generate a token for session i. Note that token-bucket 1 and b_i have only one adjacent token-bucket for session i, and thus decide whether to generate a token based on the number of tokens at only one adjacent token-bucket. Tokens are never removed from a bucket.

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. Since $\lambda_i > 1$ for each i, constraints (5) subsume constraints (6), and hence the latter can be ignored. Note that each token-bucket corresponds to constraint (5) 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

---

Procedure Token Generation (node m)
begin
For all t and i, let $C_{i,0}(t) = C_{i,b_i+1}(t) = \infty$.
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 $C_{i,k}(t) < C_{i,k+1}(t) + W$ 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 (source 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. 2. Pseudo code of the fair departure rate allocation algorithm for saturated sessions

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 differ 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.

Note that 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. Now, 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. 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.

Let d_i^* be the max-min fair departure rate of session i. Then the following result holds.

Lemma 2: Consider token-bucket k of session i. For the bounded-burstiness arrival model, 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 [10], [12]. However, the constraints characterizing the feasibility set for maximal scheduling are

Procedure Token generation for a session at the bucket at its source node (bucket k)
begin
For all t and i, let $C_{i,0}(t) = C_{i,b_i+1}(t) = \infty$.
Let the kth bucket of session i be at i's source node.
Let $A_i^{\text{NR}}(t)$ be the number of packets of session i at slot t that have been generated at its source but not been released.
Sampling procedure is the same as that in Figure 2 for all sessions associated with the kth bucket.
Token generation procedure for all sessions other than i is similar to that in Figure 2.
When session i is sampled at its kth bucket in slot t:
if $C_{i,k}(t) < C_{i,k+1}(t)+W$ and $C_{i,k}(t) < C_{i,k-1}(t)+W$ and $A_i^{\text{NR}}(t) > 0$, **then**
generate 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 in the round robin order.
end

Fig. 3. Pseudo code of the token generation process at the buckets associated with the source nodes of sessions when sessions may not be saturated.

significantly different from those characterizing the feasibility set in [10], [12]; therefore, the scheme differs significantly in the two cases.

We now describe the packet scheduling policy. Whenever the source node of a session i generates a new token for i at i's token-bucket at the source (the one associated with sessions in $S_i \cup \{i\}$), i releases a new packet. 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.

Packets that contend for scheduling and are transmitted by maximal scheduling arrive as per the release process. The release rate vector is maxmin fair (Lemma 2) 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 1: For the bounded-burstiness arrival model, there exists a constant W_0, such that when $W \geq W_0$, $\lim_{n\to\infty} D_{L_i}(n)/n = d_i^*,\ i = 1, \ldots N$.

We now consider two important generalizations. First, assume that $\lambda_i \leq 1$ for some or all sessions i. Thus, sessions may not always have packets to transmit. The only modification in the algorithm is that the bucket at the source node of a session now does not generate a new token for the session if all of its packets have already been released (Figure 3). Note that the modification applies to all sessions; therefore, the algorithm need not know which sessions are saturated. If a session is saturated, then the modification will not be executed as its source will always have packets that have not been released.

We next allow sessions to traverse multiple hops. Thus, $N \geq M$ and a session consists of multiple session-links. We first describe the modifications in the token-generation procedure. We must consider session-links instead of sessions in this case. Therefore, session-links, rather than sessions,

Procedure Token Generation (node m)
begin
For session-link i, let l and m respectively be the previous and next session-links of the same session.
For each slot t and session-link i,
if i is the first-session-link of its session, **then**
$C_{i,0}(t) = \infty, C_{i,b_i+1}(t) = C_{m,0}(t)$
else
if i is the last session-link of its session, **then**
$C_{i,0}(t) = C_{l,b_l+1}(t), C_{i,b_i+1}(t) = \infty$
else
$C_{i,0}(t) = C_{l,b_l}(t)$ and $C_{i,b_i+1}(t) = C_{m,0}(t)$.
Let $A_i^{\text{NR}}(t)$ be the number of packets of session-link i at slot t that are in its waiting-queue.
Let $\Theta_{i,k}(t) = A_i^{\text{NR}}(t)$ if the kth bucket of session-link i is at i's source-node, and $\Theta_{i,k}(t) = \infty$ otherwise.
Each bucket samples the session-links associated with it in round robin order.
When session-link 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-link 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-link at the kth bucket in the round robin order.
end

Procedure Queue Management (session-link i)
begin
When a new packet is generated for session-link i or a new packet arrives at the source of session-link i from a previous session-link, add the new-packet in the waiting-queue for session-link i.
Transfer a session-link i 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. 4. Pseudo code of the fair departure rate allocation algorithm when sessions traverse multiple hops

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 the session-links are now similar to that for single-hop sessions. The only 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. 1(a)). 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 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 non-empty released queues contend for scheduling. The rest of the scheduling remains the same as that for the case of single-hop sessions. Refer to Figure 4 for a pseudo-code.

Both Lemma 2 and Theorem 1 hold for both these generalizations (we prove Lemma 2 and Theorem 1 for the

first generalization); for the second generalization the term 'session' must now be replaced with 'session-link' in the statement of Lemma 2.

Before we conclude this section, we make a few 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.

IV. Non-uniform Throughput Guarantees with Maximal Scheduling

In this section, we relate the throughput region attained by maximal scheduling to the maximum throughput region by providing neighborhood-specific throughput guarantee for each session under maximal scheduling.

We consider the notion of "interference degree" of a session-link, as introduced in [4]. The *interference degree* of a session-link i in network $\mathcal{N}$, $K_i(\mathcal{N})$ is (i) the maximum number of sessions in its interference set S_i that can simultaneously transmit, if S_i is non-empty, and (ii) 1, if S_i is empty. The *two-hop interference degree* of session-link i, is defined as $\beta_i(\mathcal{N}) = \max_{j \in S_i \cup \{i\}} K_i(\mathcal{N})$. The *interference degree of a network* $\mathcal{N}$, $K(\mathcal{N})$, is the maximum interference degree of session-links in the network.

First consider maximal scheduling without any enhancements. We have earlier shown in networks with single-hop sessions that if $\vec{\lambda} \in \Lambda$, then $\vec{\lambda}/K(\mathcal{N}) \in \Lambda^{\text{MS}}$ [4]. Note that $K_i(\mathcal{N})$ determines the congestion in the neighborhood of a session-link i. Thus, the existing bound characterizes the performance of the entire network in terms of the worst session-link, the session-link that has interference degree $K(\mathcal{N})$. In many networks $K_i(\mathcal{N})$ and $\beta_i(\mathcal{N})$ may be significantly less than $K(\mathcal{N})$ for most session-links i (Figure 1(b)). Our contribution in this paper has been to conclusively determine whether the performance of individual session-links can be characterized in terms of the interference-degrees in their neighborhoods. We prove that the performance of each session can be characterized by the $\beta_i(\mathcal{N})$s, but not by the $K_i(\mathcal{N})$s, of the corresponding session-links.

We initially assume that each session traverses only one link, and therefore consists of a single session-link.

Theorem 2: If $(\lambda_1, \ldots, \lambda_N) \in \Lambda$, *then* $(\lambda_1/\beta_1(\mathcal{N}), \ldots, \lambda_N/\beta_N(\mathcal{N})) \in \Lambda^{\text{MS}}$.

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. The following result shows that a similar characterization in terms of the single-hop neighborhood does not hold in general.

Theorem 3: 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* $(\lambda_1/K_1(\mathcal{N}), \ldots, \lambda_N/K_N(\mathcal{N})) \notin \Lambda^{\text{MS}}$.

Next we consider the case where a session can traverse several hops. Now, let $\tilde{\beta}_i(\mathcal{N})$ denote the maximum two-hop interference degree of all session-links of session i, i.e., $\tilde{\beta}_i(\mathcal{N}) = \max_{j \in P_i} \beta_j(\mathcal{N})$. In this case, we first show that maximal scheduling attains a weaker notion of stability, as described below. For any session-link $j = 1, \ldots, M$, let $\hat{A}_j(n)$ denote the number of arrivals for the session-link in the time interval $(0, n]$. Furthermore, 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 $B_{j,t} = 0$.

Theorem 4: 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, $\mathbb{E}[B_{j,t}] < \infty$.

The above result implies that almost surely $\limsup_{n \to \infty} \frac{D_j(n) - \hat{A}_j(n)}{n} = 0 \quad \forall\ j = 1, \ldots, M$. Thus, if the arrival rate vector satisfies the condition in Theorem 4, and for each session link the limits of the departure and the arrival rates exist almost surely, then almost surely $\lim_{n \to \infty} D_{L_i}(n)/n = \lambda_i \quad \forall\ i = 1, \ldots N$, 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 the one defined in Secton II. Whether the stronger notion of stability, holds in this case or not, remains an open question.

We now consider some enhancements of maximal scheduling that obtain strong stability results for multi-hop sessions for any arrival rate vector $(\lambda_1', \ldots, \lambda_N')$ for which $\lambda_1^{'} \leq \lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N^{'} \leq \lambda_N/\tilde{\beta}_N(\mathcal{N})$, where $(\lambda_1, \ldots, \lambda_N) \in \Lambda$. Both enhancements combine maximal scheduling with a token-generation scheme. The first enhancement is the algorithm proposed in Section III. The second enhancement has been proposed by Wu *et. al.* [13], and uses a different token generation strategy, but has the same scheduling strategy as the algorithm in Section III. We now describe the token generation scheme for the second enhancement. Every session-link has a regulator that generates tokens at the arrival-rate of the session. As stated before, Wu *et. al.* [13], [14] show that this enhancement bounds the performance of all sessions in terms of that of the worst session in the network. We however prove that for both these enhancements the performance of each session can be bounded in terms of the worst-case two-hop interference degree of all session-links of a session.

Theorem 5: If $(\lambda_1, \ldots, \lambda_N) \in \Lambda$, *then* $(\lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N/\tilde{\beta}_N(\mathcal{N})) \in \Lambda^{\text{MS}}$ *in* $\mathcal{N}$.

We prove Theorem 5 in technical report [3].

APPENDIX

A. Proof of Lemma 2

We prove Lemma 2 for arbitrary $\vec{\lambda}$ and when each session spans one link. First, we show that if a session generates packets at rate r or higher, and if it is sampled at rate r or higher at every bucket associated with it, then it receives tokens at rate r or higher from each of its buckets (Lemma 3). We next show that a session's sampling rate at any of its buckets equals its maxmin fair rate (Lemma 4). Now, the result follows, as by definition, a session's maxmin fair rate is less than or equal to its packet generation rate. We prove Lemmas 3 and 4 in sections B and C. Thus, like in the current section, throughout sections B and C, we will assume that every session spans one link.

We introduce some terminologies and subsequently state Lemmas 3 and 4. Let $S_{i,n}(t)$ be the number of times session i is sampled at token-bucket n in the interval $(0,t]$, $L = \max_i b_i$, $\sigma = \max_i \sigma_i$, and β, γ are constants that are specified later.

Lemma 3: Consider an arbitrary K and a sequence of K disjoint intervals, $(t_l, w_l]$, $l = 1, \ldots, K$, that satisfies the following property for session i, for every positive integer M' and every sequence of sub-intervals $(x_m, y_m]$, $m = 1, \ldots, M'$, $(x_m, y_m] \subset (t_l, w_l]$, for some l: At every bucket n associated with i,

$$\sum_{m=1}^{M'} (S_{i,n}(y_m) - S_{i,n}(x_m)) \geq r \sum_{m=1}^{M'} (y_m - x_m) - e - M'f, \tag{7}$$

where e and f are constants that do not depend on M' and the sub intervals $(x_m, y_m]$, $m = 1, \ldots, M'$. Let $\lambda_i \geq r$ and $W \geq 3^{b_i - 1}(f + \sigma_i)/2$. Then, at every bucket n associated with i,

$$\begin{aligned} \sum_{l=1}^{K} (C_{i,n}(w_l) - C_{i,n}(t_l)) \quad \geq \quad & r \sum_{l=1}^{K} (w_l - t_l) - 2^{b_i - 1} e \\ & - K 3^{b_i - 1}(f + \sigma_i). \end{aligned} \tag{8}$$

Lemma 4: Consider any positive integer K, and an arbitrary non-decreasing sequence of times $x_1, y_1, \ldots, x_K, y_K$. Let $W \geq 3^{L-1}(\varepsilon_1(F) + \sigma)/2$, where $\varepsilon_1(F)$ is defined in (12) to (17). For every bucket n associated with session i,

$$\begin{aligned} \sum_{k=1}^{K} (S_{i,n}(y_k) - S_{i,n}(x_k)) \quad \geq \quad & d_i^* \sum_{k=1}^{K} (y_k - x_k) - \beta \\ & - K\gamma, \end{aligned} \tag{9}$$

$$\begin{aligned} \sum_{k=1}^{K} (C_{i,n}(y_k) - C_{i,n}(x_k)) \quad \geq \quad & d_i^* \sum_{k=1}^{K} (y_k - x_k) - \beta \\ & - K\gamma, \end{aligned} \tag{10}$$

$$\begin{aligned} \sum_{k=1}^{K} (C_{i,n}(y_k) - C_{i,n}(x_k)) \quad \leq \quad & d_i^* \sum_{k=1}^{K} (y_k - x_k) + \beta \\ & + K\gamma. \end{aligned} \tag{11}$$

Here, β and γ are constants that do not depend on $x_1, y_1, \ldots, x_K, y_K$.

We introduce the notion of "rank" of a session for defining β and γ. A session has rank p if its maxmin fair rate is $\hat{d}_p$, the pth lowest among the maxmin fair rates of different sessions. Let F be the number of distinct ranks, $F \leq N$.

$$\begin{aligned} \varsigma_1(1) &= 0. && (12) \\ \varepsilon_1(1) &= 1. && (13) \\ \varsigma_2(p) &= 2^{L-1}\varsigma_1(p). && (14) \\ \varepsilon_2(p) &= 3^{L-1}(\varepsilon_1(p) + \sigma). && (15) \\ \varsigma_3(p) &= 2\sigma + \max(L, 2)\,(\varsigma_2(p) + \varepsilon_2(p)) \\ & \quad + 2LW. && (16) \\ \varepsilon_3(p) &= \varepsilon_2(p). && (17) \\ \varsigma_1(p+1) &= (L-1)\varsigma_3(p). && (18) \\ \varepsilon_1(p+1) &= (L-1)\varepsilon_3(p) + 1. && (19) \end{aligned}$$

Now, $\beta = \varsigma_3(F)$ and $\gamma = \varepsilon_3(F)$.

Now, for any given $\vec{\lambda}$, Lemma 2 follows from (10) and (11) of Lemma 4 with $\varrho = \beta + \gamma$ and $W_0 = 3^{L-1}(\varepsilon_1(F) + \sigma)/2$. ■

B. Proof of Lemma 3

We first present the intuition behind the proof. The proof is by induction on the number of buckets associated with a session. The sessions with one bucket form the base case. Note that any such session receives a token at its bucket every time it is sampled at its bucket and has a packet that has not been released, since no adjacent bucket applies back-pressure. Now, the lemma follows for the base case from the lower bounds on the sampling and packet generation rates. We next assume that the lemma holds for all sessions with p buckets, and then prove the lemma for sessions with $p+1$ buckets. Consider a session with $p+1$ buckets and adjacent buckets n and $n+1$ associated with it. Bucket $n+1$ does not prevent the generation of any token at n unless the number of tokens at n is W more than that at $n+1$. If the number of tokens at n is W more than that at $n+1$, n does not prevent any token generation at $n+1$, and the buckets $n+1, n+2, \ldots$ generate tokens oblivious to the presence of the buckets $1, \ldots, n$, as though they constitute a session with fewer buckets. By induction hypothesis, and from the sampling and packet generation rates, the session receives tokens at rate r or higher at $n+1$ in these intervals. In all these slots, the number of tokens at n exceeds that at $n+1$ by W. Thus, n's token generation rate is lower bounded by $n+1$'s token generation rate which is at least r. In other slots, $n+1$ does not prevent the generation of any token at n. Thus, the token generation at the buckets $1, \ldots, n$ resembles that for a session with fewer buckets. Thus, by induction hypothesis and the assumption on the sampling rate, in all slots, n generates tokens at rate r or higher for the session.

Proof: We prove by induction on the number of buckets p associated with a session.

First consider a session i with one bucket n. Let n not be at the source node of i. The lemma holds from the assumption on the sampling rate (condition (7)). Now, let n be at the source node of i. Let $A_i^{\text{NR}}(t)$ be the number of packets of session i

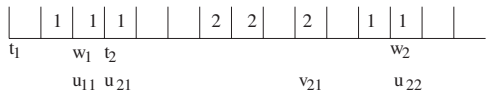

Fig. 5. We show two intervals $(t_1, w_1]$ and $(t_2, w_2]$, and some type 1 and 2 slots. We also show the corresponding u and v slots. Here $(t_1, u_{11}]$, $(t_2, u_{21}]$, $(v_{21}, u_{22}]$ are example sub-intervals that end in $u-$slots and start from the nearest $v-$slot or t_i-slot.

at its source at time t that have not been released. We now define a slot z_l. If $A_i^{\text{NR}}(t) > 0$ for all $t \in (t_l, w_l]$, $z_l = t_l$, else $z_l = \max_{t \in (t_l, w_l], A_i^{\text{NR}}(t)=0} t$. If $z_l > t_l$,

$$\begin{aligned} C_{i,n}(z_l) - C_{i,n}(t_l) &= A_i(z_l) - A_i(t_l) + A_i^{\text{NR}}(t_l) \\ &\geq A_i(z_l) - A_i(t_l) \\ &\geq r(z_l - t_l) - \sigma_i. \end{aligned} \tag{20}$$

The last inequality follows from (2) and since $r \leq \lambda_i$. Clearly, (20) also holds if $z_l = t_l$. Bucket n generates a token for session i every time it samples i in $(z_l, w_l]$, $\forall\ l$.

$$\begin{aligned} & \sum_{l=1}^{K} (C_{i,n}(w_l) - C_{i,n}(z_l)) \\ = & \sum_{l=1}^{K} (S_{i,n}(w_l) - S_{i,n}(z_l)) \\ \geq & \ r \sum_{l=1}^{K} (w_l - z_l) - e - Kf \text{ (from (7))}. \end{aligned} \tag{21}$$

$$\begin{aligned} & \sum_{l=1}^{K} (C_{i,n}(w_l) - C_{i,n}(t_l)) \\ = & \sum_{l=1}^{K} (C_{i,n}(w_l) - C_{i,n}(z_l)) + \sum_{l=1}^{K} (C_{i,n}(z_l) - C_{i,n}(t_l)) \\ \geq & \ r \sum_{l=1}^{K} (w_l - t_l) - e - K(f + \sigma_i) \text{ (from (20) and (21))}. \end{aligned}$$

Thus, (8) holds in the base case.

We now assume that (8) holds for all sessions with p or fewer buckets, and prove (8) for an arbitrary session i with $p+1$ buckets. Consider an arbitrary bucket n associated with i. If the number of tokens of i at n does not exceed that at buckets adjacent to n by W or more in the intervals (t_l, w_l), $l = 1, \ldots, K$, then the token generation process for i at n is not affected by back-pressure, and the proof is similar to the base case. Thus, we assume that there exists a bucket B that is adjacent to n, and $C_{i,n}(t) = C_{i,B}(t) + W$ at some time t in these intervals. Clearly $B \in \{n-1, n+1\}$. We consider the case that $B = n+1$. The proof when $B = n-1$ is similar.

Let a slot t where $C_{i,n}(t)$ exceeds $C_{i,n+1}(t)$ by W be a type 1 slot, and a slot t where $C_{i,n+1}(t)$ exceeds $C_{i,n}(t)$ by W be a type 2 slot; a slot may neither be type 1 nor type 2. Consider each $(t_l, w_l]$ interval separately. Consider the sequences of type 1 and 2 slots that are obtained after removing the slots without numbers. The last slot in such a sequence of type-1 (2) slots is denoted a "u" ("v") slot. The mth "u-slot" ("v-slot") of the lth interval is u_{lm} (v_{lm}) (Figure 5). Note that

$$C_{i,n}(u_{lm}) = C_{i,n+1}(u_{lm}) + W \ \forall\ l, m. \tag{22}$$

$$C_{i,n+1}(v_{lm}) = C_{i,n}(v_{lm}) + W \ \forall\ l, m. \tag{23}$$

$$C_{i,n}(t) \leq C_{i,n+1}(t) + W, \ \forall\ t. \tag{24}$$

Consider a sub-interval that ends at a u slot and starts from a t_j (not inclusive) or a $v-$slot (not inclusive), whichever is the nearest to the $u-$slot (Figure 5). Let there be J_l such sub-intervals in $(t_l, w_l]$, and $\sum_{l=1}^{K} J_l = I_1$. These sub-intervals do not consist of any type 2 slot. Thus, n does not prevent any session i token generation at $n+1$ in these sub-intervals. Hence, in these sub-intervals, the token generation for i in buckets $n+1, \ldots, p+1$ resembles that in the buckets of a session with $p+1-n$ buckets, where $n > 0$. Condition (7) holds for i in each of these buckets for every set of sub-intervals of these I_1 sub-intervals, since any such sub-interval is in $(t_l, w_l]$ for some l. Thus, the number of tokens generated for i in these I_1 sub-intervals in each of these buckets can be lower bounded using the induction hypothesis. The sub-intervals in $(t_l, w_l]$ are $(t_l, u_{l1}]$ and $(v_{lm-1}, u_{lm}]$, $m > 1$, if $v_{l1} > u_{l1}$ as in Figure 5; the sub-intervals are $(v_{lm}, u_{lm}]$, $m \geq 1$, otherwise. We assume that $v_{l1} > u_{l1}$ for all l; the argument is similar if $v_{l1} < u_{l1}$ for some or all l. From induction hypothesis,

$$\begin{aligned} & \sum_{l=1}^{K} ((C_{i,n+1}(u_{l1}) - C_{i,n+1}(t_l)) \\ & + \sum_{m=2}^{J_l} (C_{i,n+1}(u_{lm}) - C_{i,n+1}(v_{lm-1}))) \\ \geq & \ r \sum_{l=1}^{K} \left((u_{l1} - t_l) + \sum_{m=2}^{J_l} (u_{lm} - v_{lm-1}) \right) \\ & -2^{p-1} e - I_1 3^{p-1} (f + \sigma_i). \end{aligned} \tag{25}$$

$$\begin{aligned} & C_{i,n}(u_{l1}) - C_{i,n}(t_l) \\ \geq & \ C_{i,n+1}(u_{l1}) + W - C_{i,n+1}(t_l) - W \text{ (from (22) and (24))} \\ = & \ C_{i,n+1}(u_{l1}) - C_{i,n+1}(t_l). \end{aligned} \tag{26}$$

From (22) and (23),

$$\begin{aligned} C_{i,n}(u_{lm}) - C_{i,n}(v_{lm-1}) &= C_{i,n+1}(u_{lm}) - C_{i,n+1}(v_{lm-1}) \\ &\quad +2W. \end{aligned} \tag{27}$$

$$\begin{aligned} & \sum_{l=1}^{K} ((C_{i,n}(u_{l1}) - C_{i,n}(t_l)) \\ & + \sum_{m=2}^{J_l} (C_{i,n}(u_{lm}) - C_{i,n}(v_{lm-1}))) \\ \geq & \ \sum_{l=1}^{K} ((C_{i,n+1}(u_{l1}) - C_{i,n+1}(t_l))) \end{aligned}$$

$$
\begin{aligned}
&+\sum_{m=2}^{J_l}\left(C_{i,n+1}(u_{lm})-C_{i,n+1}(v_{lm-1})\right)\\
&+2W(I_1-K) \text{ (from (26) and (27))}\\
\geq\ & r\sum_{l=1}^{K}\left((u_{l1}-t_l)+\sum_{m=2}^{J_l}(u_{lm}-v_{lm-1})\right)\\
&-2^{p-1}e-K3^{p-1}(f+\sigma_i)\\
&+(I_1-K)(2W-3^{p-1}f-3^{p-1}\sigma_i) \text{ (from (25))}
\end{aligned}
\tag{28}
$$

Now, consider the sub-intervals obtained after removing these I_1 sub-intervals from $\cup_{l=1}^{K}(t_l,w_l]$. These new sub-intervals do not contain any type 1 slot. Thus, $n+1$ does not prevent any session i token generation at n. Hence, the session i token generation in buckets $1,\ldots,n$ resembles that of a session with n buckets, where $n\leq p$. The number of session i tokens generated at n in these sub-intervals can be lower bounded from the induction hypothesis. There are at most I_1+K such sub-intervals, which are of the form $(u_{lm},v_{lm}]$ and $(u_{J_l},w_l]$, since we assume that $v_{l1}>u_{l1}\ \forall\ l$.

$$
\begin{aligned}
&\text{Thus, } \sum_{l=1}^{K}\Bigg((C_{i,n}(w_l)-C_{i,n}(u_{J_l}))\\
&+\sum_{m=1}^{J_l-1}(C_{i,n}(v_{lm})-C_{i,n}(u_{lm}))\Bigg)\\
\geq\ & r\sum_{l=1}^{K}\left((w_l-u_{J_l})+\sum_{m=1}^{J_l-1}(v_{lm}-u_{lm})\right)\\
&-2^{p-1}e-(I_1-K)3^{p-1}(f+\sigma_i)\\
&-2K3^{p-1}(f+\sigma_i).
\end{aligned}
\tag{29}
$$

Adding (28) and (29),

$$
\begin{aligned}
&\sum_{l=1}^{K}(C_{i,n}(w_l)-C_{i,n}(t_l))\\
\geq\ & r\sum_{l=1}^{K}(w_l-t_l)-2^pe-K3^p(f+\sigma_i)\\
&+(I_1-K)(2W-3^p(f+\sigma_i)).
\end{aligned}
\tag{30}
$$

Note that $p+1\leq b_i$ and thus, $W\geq 3^p(f+\sigma_i)/2$. We have implicitly assumed that at least one type-1 slot exists in each interval $(t_l,w_l]$; this justifies the summation from $l=1$ to K in (25). Under this assumption, $I_1\geq K$. Hence, (8) holds for session i at bucket n. If there is no type-1 slot in $(t_l,w_l]$ for some l, then the summation in (25) must be over the intervals $(t_l,w_l]$ that have at least one type-1 slot. Let K_1 be the number of such intervals. Now, (I_1-K) must be replaced with (I_1-K_1). Since $I_1\geq K_1$, (8) holds at all buckets associated with i. ■

C. Proof of Lemma 4

We outline the proof for the special case that all sessions always have packets to transmit. We use induction on the rank p of a session. For the base case ($p=1$), using a property of the round robin sampling, we show that all sessions are sampled at a rate $\hat{d}_1$ or higher at every bucket. Now, (10), the lower bound on the token generation rate follows from Lemma 3. Next, we show (11), i.e., the token generation rates are upper bounded by $\hat{d}_1$ for all sessions with rank 1. This follows because the sampling and hence the token generation rate is upper bounded by $\hat{d}_1$ at the bottleneck bucket, and due to back-pressure the token generation rates for a session are equal at different buckets in the session's path. Now, consider the induction case, i.e., arbitrary p. The token generation rates of sessions with rank lower than p are upper bounded by their respective maxmin fair rates which are upper bounded by $\hat{d}_p$. Sessions of rank p or higher are sampled in a certain minimum fraction of the slots in which the sessions with rank lower than p do not receive tokens. Therefore, the lower bound on the sampling rate of sessions with rank p or higher follows. Again, the lower bound on the token generation rate follows from Lemma 3. We prove, as in the base case, the upper bound on the token generation rate for sessions with rank p.

In the formal proof, we relax the assumption that all sessions always have packets to transmit, i.e., we consider arbitrary $\vec{\lambda}$. We would like to clarify the usage of a particular notation before proceeding further. We have so far numbered token-buckets based on the sessions traversing them. In this terminology, bucket n of session i is i's nth bucket, and $C_{i,n}(t),S_{i,n}(t)$ are the number of tokens generated for session i at and the number of times session i is sampled at its nth bucket respectively. In the following proof, we number token-buckets separately. Thus, for example, we consider token-bucket n and all sessions associated with n. Now, $n(i)$ will denote the number for the bucket n among i's buckets. Thus, we need to use $C_{i,n(i)}(t),S_{i,n(i)}(t)$ instead of $C_{i,n}(t),S_{i,n}(t)$. For simplicity, we still use $C_{i,n}(t),S_{i,n}(t)$. Thus, in the following proof, $C_{i,n}(t),S_{i,n}(t)$ really stand for $C_{i,n(i)}(t),S_{i,n(i)}(t)$ respectively. Note that this inconsistency is limited to the following proof only, and does not lead to any error, because none of the analytical guarantees in other lemmas (including those that are used in the following proof and those whose proof use Lemma 4) depend on the token-bucket number.

Proof: We prove the following for ranks $p=1,\ldots,F$, by induction on p.

For each bucket n, for each session i that is associated with n and has rank greater than or equal to p, for any positive integer K, and for any nondecreasing sequence of times $x_1,y_1,\ldots,x_K,y_K$,

$$
\sum_{k=1}^{K}(S_{i,n}(y_k)-S_{i,n}(x_k))\geq\hat{d}_p\sum_{k=1}^{K}(y_k-x_k)-\varsigma_1(p)-K\varepsilon_1(p).
\tag{31}
$$

For each bucket n, for each session i that is associated with n and has rank greater than or equal to p, for any positive integer K, and for any nondecreasing sequence of

times $x_1, y_1, \ldots, x_K, y_K$,

$$\sum_{k=1}^{K} (C_{i,n}(y_k) - C_{i,n}(x_k)) \geq \hat{d}_p \sum_{k=1}^{K} (y_k - x_k) - \varsigma_2(p) - K\varepsilon_2(p). \tag{32}$$

If a session i has rank p, and $d_i^* = \lambda_i$,

$$A_i^{\text{NR}}(t) \leq \sigma_i + \varsigma_2(p) + \varepsilon_2(p) \ \forall \ t. \tag{33}$$

For each bucket n, for each session i that is associated with n and has rank p, for any positive integer K, and for any nondecreasing sequence of times $x_1, y_1, \ldots, x_K, y_K$,

$$\sum_{k=1}^{K} (C_{i,n}(y_k) - C_{i,n}(x_k)) \leq \hat{d}_p \sum_{k=1}^{K} (y_k - x_k) + \varsigma_3(p) + K\varepsilon_3(p). \tag{34}$$

We first prove (31) to (34) for $p = 1$. Note that $\hat{d}_1 = \min(1/L, \min_i \lambda_i)$. Consider a bucket n. Let $\mathcal{X}$ be the set of sessions associated with n. Since at least one session is sampled at n in a slot, in any interval $(x_k, y_k]$,

$$\sum_{j \in \mathcal{X}} (S_{j,n}(y_k) - S_{j,n}(x_k)) \quad \geq \quad y_k - x_k.$$

Since sessions are sampled in round robin order, $S_{i,n}(y_k) - S_{i,n}(x_k) \geq S_{j,n}(y_k) - S_{j,n}(x_k) - 1$ for any two sessions i, j associated with n. Thus, for any session i associated with n,

$$\begin{aligned} |\mathcal{X}| \left(S_{i,n}(y_k) - S_{i,n}(x_k) + 1\right) &\geq y_k - x_k, \\ S_{i,n}(y_k) - S_{i,n}(x_k) &\geq \frac{y_k - x_k}{|\mathcal{X}|} - 1. \end{aligned}$$

Thus, every session associated with bucket n is sampled at least $\sum_{k=1}^{Q} (y_k - x_k)/|\mathcal{X}| - Q$ times for any arbitrary sequence of nondecreasing times $x_1, y_1, \ldots, x_Q, y_Q$, and any arbitrary Q. Since $|\mathcal{X}| \leq L$, $\hat{d}_1 \leq 1/|\mathcal{X}|$. Thus, (31) holds with $\varsigma_1(1) = 0, \varepsilon_1(1) = 1$.

Since $\varepsilon_F(1) \geq \varepsilon_1(1)$, $W \geq 3^{L-1}(\varepsilon_1(1) + \sigma)/2$. Hence, (32) follows from Lemma 3 with $\varsigma_2(1) = 2^{L-1}\varsigma_1(1)$ and $\varepsilon_2(1) = 3^{L-1}(\varepsilon_1(1) + \sigma)$.

Now, we prove (33) for $p = 1$. Consider a session i with rank 1 and $d_i^* = \lambda_i$. Thus, $\hat{d}_1 = \lambda_i$. Let n be the bucket at the source node of i.

$$\begin{aligned} A_i^{\text{NR}}(t) &= A_i(t) - C_{i,n}(t) \\ &\leq (\lambda_i - \hat{d}_1)t + \sigma_i + \varsigma_2(1) + \varepsilon_2(1) \\ &\quad \text{(from (2) and (32) for } p = 1) \\ &= \sigma_i + \varsigma_2(1) + \varepsilon_2(1) \text{ (since } \hat{d}_1 = \lambda_i). \end{aligned}$$

Thus, (33) follows for $p = 1$.

Now, we prove (34) for $p = 1$. Consider a session i with rank 1. Let n be a bucket associated with i. Consider a sequence of non-decreasing times $x_1, y_1, \ldots, x_K, y_K$.

$$\begin{aligned} &\sum_{k=1}^{K} (C_{i,n}(y_k) - C_{i,n}(x_k)) \\ = \ & C_{i,n}(y_K) - C_{i,n}(x_1) - \sum_{k=1}^{K-1} (C_{i,n}(x_{k+1}) - C_{i,n}(y_k)) \\ \leq \ & C_{i,n}(y_K) - C_{i,n}(x_1) - \hat{d}_1 \sum_{k=1}^{K-1} (x_{k+1} - y_k) \\ & + \varsigma_2(1) + (K-1)\varepsilon_2(1) \text{ (from (32) for } p = 1). \end{aligned} \tag{35}$$

Since $\hat{d}_1 = d_i^*$ and $d_i^* \leq \lambda_i$, $\hat{d}_1 \leq \lambda_i$. First, let $\hat{d}_1 < \lambda_i$. Thus, from Lemma 1, i has a bottleneck constraint and hence a bottleneck bucket, B. Let $\mathcal{X}$ be the set of sessions associated with B. Since i has rank 1, $|\mathcal{X}| = L$, rank$(j) = 1 \ \forall \ j \in \mathcal{X}$, and $\hat{d}_1 = 1/L$.

$$\begin{aligned} & C_{i,B}(y_K) - C_{i,B}(x_1) \\ \leq \ & y_K - x_1 - \sum_{m \in \mathcal{X} \setminus \{i\}} (C_{m,B}(y_K) - C_{m,B}(x_1)) \\ \leq \ & y_K - x_1 \\ & -(L-1)\left(\hat{d}_1(y_K - x_1) - \varsigma_2(1) - \varepsilon_2(1)\right) \\ & \text{(from (32) since rank}(j) = 1, \ \forall \ j \ \in \mathcal{X}) \\ = \ & \hat{d}_1(y_K - x_1) + (L-1)\left(\varsigma_2(1) + \varepsilon_2(1)\right) \\ & \text{(since } \hat{d}_1 = 1/L). \end{aligned} \tag{36}$$

Now, let $\hat{d}_1 = \lambda_i$. Let B be the bucket at the source of i.

$$\begin{aligned} & C_{i,B}(y_K) - C_{i,B}(x_1) \\ \leq \ & A_i^{\text{NR}}(x_1) + A_i(y_K) - A_i(x_1) \\ \leq \ & \sigma_i + \varsigma_2(1) + \varepsilon_2(1) + \lambda_i(y_K - x_1) + \sigma_i \\ & \text{(from (33) and (2))} \\ = \ & \hat{d}_1(y_K - x_1) + 2\sigma_i + \varsigma_2(1) + \varepsilon_2(1) \text{ (since } \hat{d}_1 = \lambda_i). \end{aligned} \tag{37}$$

From (36) and (37), there exists a bucket B associated with i such that

$$\begin{aligned} & C_{i,B}(y_K) - C_{i,B}(x_1) \\ \leq \ & \hat{d}_1(y_K - x_1) + 2\sigma_i \\ & + \max(L-1, 1)\left(\varsigma_2(1) + \varepsilon_2(1)\right). \end{aligned} \tag{38}$$

$$\text{Now, } |C_{i,n}(t) - C_{i,B}(t)| \leq b_i W \ \forall \ t. \tag{39}$$

$$\begin{aligned} & C_{i,n}(y_K) - C_{i,n}(x_1) \\ \leq \ & C_{i,B}(y_K) - C_{i,B}(x_1) + 2b_i W \text{ (from (39))} \\ \leq \ & \hat{d}_1(y_K - x_1) + \max(L-1, 1)\left(\varsigma_2(1) + \varepsilon_2(1)\right) \\ & + 2b_i W + 2\sigma_i \text{ (from (38))}. \end{aligned} \tag{40}$$

From (35) and (40),

$$\begin{aligned} & \sum_{k=1}^{K} (C_{i,n}(y_k) - C_{i,n}(x_k)) \\ \leq \ & \hat{d}_1 \sum_{k=1}^{K} (y_k - x_k) + \max(L, 2)\left(\varsigma_2(1) + \varepsilon_2(1)\right) \\ & + 2b_i W + 2\sigma_i + K\varepsilon_2(1). \end{aligned} \tag{41}$$

Thus, for $p = 1$, (34) follows from (41) with $\varsigma_3(1) = \max(L,2)\left(\varsigma_2(1)+\varepsilon_2(1)\right) + 2LW + 2\sigma$ and $\varepsilon_3(1) = \varepsilon_2(1)$.

Now, we assume (31) to (34) for $1,\ldots,p$, and show that (31) to (34) hold for $p+1$.

We first prove (31). Consider a session i with rank greater than or equal to $p+1$. Consider a bucket n associated with i. Let $\mathcal{Y}= \{w : w \text{ is associated with } n, \text{rank}(w) \leq p\}$ and $\mathcal{Z}= \{w : w \text{ is associated with } n, \text{rank}(w) \geq p+1\}$. In any interval $(x_k, y_k]$,

$$\sum_{j\in\mathcal{Z}} \left(S_{j,n}(y_k) - S_{j,n}(x_k)\right) + \sum_{j\in\mathcal{Y}} \left(C_{j,n}(y_k) - C_{j,n}(x_k)\right) \geq y_k - x_k.$$

Since sessions are sampled in round robin order, $S_{i,n}(y_k) - S_{i,n}(x_k) \geq S_{j,n}(y_k) - S_{j,n}(x_k) - 1$ for any two sessions i,j associated with n. Thus,

$$\begin{aligned} & |\mathcal{Z}|\left(S_{i,n}(y_k) - S_{i,n}(x_k) + 1\right) \\ \geq\ & y_k - x_k - \sum_{j\in\mathcal{Y}} \left(C_{j,n}(y_k) - C_{j,n}(x_k)\right). \end{aligned}$$

Thus,

$$\begin{aligned} & \sum_{k=1}^{K} \left(S_{i,n}(y_k) - S_{i,n}(x_k)\right) \\ \geq\ & \frac{1}{|\mathcal{Z}|}\left(\sum_{k=1}^{K}(y_k - x_k) - K|\mathcal{Z}| \right. \\ & \left. - \sum_{j\in\mathcal{Y}}\sum_{k=1}^{K}\left(C_{j,n}(y_k) - C_{j,n}(x_k)\right)\right) \\ \geq\ & \frac{\left(1 - \sum_{j\in\mathcal{Y}} d_j^*\right)\sum_{k=1}^{K}(y_k - x_k)}{|\mathcal{Z}|} \\ & - \frac{|\mathcal{Y}|}{|\mathcal{Z}|}\varsigma_3(p) - K\frac{|\mathcal{Z}| + |\mathcal{Y}|\varepsilon_3(p)}{|\mathcal{Z}|}. \end{aligned}$$

The last inequality follows since $\text{rank}(w) \leq p$, and $d_w^* = \hat{d}_{\text{rank}(w)}$, $\forall\, w \in\mathcal{Y}$. Also, $\varsigma_3(j) \geq \varsigma_3(j-1)$, $\varepsilon_3(j) \geq \varepsilon_3(j-1)$, $\forall j$. Thus, induction hypothesis (inequality (34)) applies. Now,

$$\begin{aligned} & \sum_{k=1}^{K} \left(S_{i,n}(y_k) - S_{i,n}(x_k)\right) \\ \geq\ & \frac{\sum_{j\in\mathcal{Z}} d_j^* \sum_{k=1}^{K}(y_k - x_k)}{|\mathcal{Z}|} - \frac{|\mathcal{Y}|}{|\mathcal{Z}|}\varsigma_3(p) \\ & - K\frac{|\mathcal{Z}| + |\mathcal{Y}|\varepsilon_3(p)}{|\mathcal{Z}|} \ \text{(since } \sum_{w\in\mathcal{Z}} d_w^* + \sum_{w\in\mathcal{Y}} d_w^* \leq 1) \\ \geq\ & \hat{d}_{p+1}\sum_{k=1}^{K}(y_k - x_k) - \frac{|\mathcal{Y}|}{|\mathcal{Z}|}\varsigma_3(p) \\ & - K\frac{|\mathcal{Z}| + |\mathcal{Y}|)\varepsilon_3(p)}{|\mathcal{Z}|}. \end{aligned} \tag{42}$$

The last step follows since $\text{rank}(w) \geq p+1$, and hence $d_w^* \geq \hat{d}_{p+1}$, $\forall\, w \in\mathcal{Z}$. Thus, from (42), (31) holds for $p+1$, with $\varsigma_1(p+1) = (L-1)\varsigma_3(p)$, and $\varepsilon_1(p+1) = (L-1)\varepsilon_3(p)+1$.

Consider a session i with rank greater than or equal to $p+1$. Note that $\lambda_i \geq \hat{d}_{p+1}$, and $W \geq 3^{L-1}(\varepsilon_1(p+1)+\sigma)/2$. Thus, (32) follows from Lemma 3, with $\varsigma_2(p+1) = 2^{L-1}\varsigma_1(p+1)$ and $\varepsilon_2(p+1) = 3^{L-1}(\varepsilon_1(p+1)+\sigma)$.

The proof for (33) is similar to that in the base case.

Now, we prove (34) for $p+1$. The argument is similar to that for the base case. We point out the differences. Consider a session i with rank $p+1$. Let n be a bucket associated with i. Consider any sequence of non-decreasing times $x_1, y_1, \ldots, x_K, y_K$.

$$\begin{aligned} & \sum_{k=1}^{K} \left(C_{i,n}(y_k) - C_{i,n}(x_k)\right) \\ =\ & C_{i,n}(y_K) - C_{i,n}(x_1) - \sum_{k=1}^{K-1}\left(C_{i,n}(x_{k+1}) - C_{i,n}(y_k)\right) \\ \leq\ & C_{i,n}(y_K) - C_{i,n}(x_1) - \hat{d}_{p+1}\sum_{k=1}^{K-1}(x_{k+1} - y_k) \\ & + \varsigma_2(p+1) + (K-1)\varepsilon_2(p+1). \end{aligned} \tag{43}$$

The last inequality follows from (32) for $p+1$.

Since $\hat{d}_{p+1} = d_i^*$ and $d_i^* \leq \lambda_i$, $\hat{d}_{p+1} \leq \lambda_i$. Now, first let $\hat{d}_{p+1} < \lambda_i$. Since $d_i^* = \hat{d}_{p+1}$, $d_i^* < \lambda_i$. Thus, from Lemma 1, i is associated with a bottleneck constraint, and hence a bottleneck bucket, B. Let $\mathcal{X}$ be the set of sessions associated with B. Since i has rank $p+1$, ranks of all sessions associated with B are less than or equal to $p+1$.

$$\begin{aligned} & C_{i,B}(y_K) - C_{i,B}(x_1) \\ \leq\ & y_K - x_1 - \sum_{m\in\mathcal{X}\setminus\{i\}}\left(C_{m,B}(y_K) - C_{m,B}(x_1)\right) \\ \leq\ & y_K - x_1 \\ & - \sum_{m\in\mathcal{X}\setminus\{i\}}\left(d_m^*(y_K - x_1) - \varsigma_2(p+1) - \varepsilon_2(p+1)\right) \\ & \text{(from (32))} \\ =\ & \hat{d}_{p+1}(y_K - x_1) \\ & + (|\mathcal{X}| - 1)(\varsigma_2(p+1) + \varepsilon_2(p+1)). \end{aligned} \tag{44}$$

The last step follows since $\hat{d}_{p+1} + \sum_{m\in\mathcal{X}\setminus\{i\}} d_m^* = 1$.

Now, let $\hat{d}_{p+1} = \lambda_i$. Let B be the bucket at the source node of i. Like in the base case, using (32) and (2), we can prove that

$$\begin{aligned} C_{i,B}(y_K) - C_{i,B}(x_1) \ \leq\ & \hat{d}_{p+1}(y_K - x_1) + 2\sigma_i \\ & + \varsigma_2(p+1) + \varepsilon_2(p+1). \end{aligned} \tag{45}$$

From (44) and (45), there exists a bucket B associated with i such that,

$$\begin{aligned} & C_{i,B}(y_K) - C_{i,B}(x_1) \\ \leq\ & \hat{d}_{p+1}(y_K - x_1) + 2\sigma_i \\ & + \max(L-1,1)\left(\varsigma_2(p+1) + \varepsilon_2(p+1)\right). \end{aligned} \tag{46}$$

From (46), like in the base case,

$$\begin{aligned} & C_{i,n}(y_K) - C_{i,n}(x_1) \\ \leq \; & \hat{d}_{p+1}(y_K - x_1) + 2\sigma_i + 2b_i W \\ & + \max(L-1,1)\left(\varsigma_2(p+1) + \varepsilon_2(p+1)\right). \end{aligned} \quad (47)$$

From (43) and (47),

$$\begin{aligned} & \sum_{k=1}^{K} \left(C_{i,n}(y_k) - C_{i,n}(x_k)\right) \\ \leq \; & \hat{d}_{p+1} \sum_{k=1}^{K} (y_k - x_k) + 2b_i W + 2\sigma_i + K\varepsilon_2(p+1) \\ & + \max(L,2)\left(\varsigma_2(p+1) + \varepsilon_2(p+1)\right). \end{aligned} \quad (48)$$

Thus, (34) follows from (48) with $\varsigma_3(p+1) = \max(L,2)\left(\varsigma_2(p+1) + \varepsilon_2(p+1)\right) + 2LW + 2\sigma$ and $\varepsilon_3(p+1) = \varepsilon_2(p+1)$. Thus, (31) to (34) hold in the induction case.

Note that $\varsigma_i(x), \varepsilon_i(x)$ are increasing in both i and x. Thus, from (31), (32) and (34), Lemma 4 holds with $\beta = \varsigma_3(F)$ and $\gamma = \varepsilon_3(F)$. ■

D. Proof of Theorem 1

We present the proof for arbitrary $\vec{\lambda}$ and when each session spans one link. 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 2, there exists constants ϱ, W_0, such that when $W \geq W_0$, $\forall\, t$, $|\frac{A_i^{\mathrm{R}}(t)}{t} - d_i^*| \leq \frac{\varrho}{t}$. 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.

E. Proof of Theorem 2

We prove Theorem 2 using a supporting lemma, Lemma 5, which we state and prove first. The lemma and its proof do not assume that the sessions are single-hop, and therefore hold for multi-hop sessions as well.

Lemma 5: If $(\lambda_1, \ldots, \lambda_N) \in \Lambda$, *then* $(\lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N/\tilde{\beta}_N(\mathcal{N}))$ *satisfies (4).*

Proof: Let $\left(\lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N/\tilde{\beta}_N(\mathcal{N})\right)$ not satisfy (4). We will show that $\vec{\lambda} \notin \Lambda$.

Now, since $\left(\lambda_1/\tilde{\beta}_1(\mathcal{N}), \ldots, \lambda_N/\tilde{\beta}_N(\mathcal{N})\right)$ not satisfy (4), there exists a session-link i such that

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

$$\text{Since } \beta_j \leq \tilde{\beta}_{q(j)}, \quad \sum_{j \in S_i \cup \{i\}} \frac{\lambda_{q(j)}}{\beta_j(\mathcal{N})} > 1.$$

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

$$\begin{aligned} \sum_{j \in S_i \cup \{i\}} \frac{\lambda_{q(j)}}{K_i(\mathcal{N})} &> 1. \\ \Rightarrow \sum_{j \in S_i \cup \{i\}} \lambda_{q(j)} &> K_i(\mathcal{N}). \end{aligned} \quad (49)$$

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.

$$\begin{aligned} \text{Thus, } \liminf_{n\to\infty} \sum_{j \in S_i \cup \{i\}} \frac{D_j(n)}{n} &\leq K_i(\mathcal{N}) \\ \Rightarrow \sum_{j \in S_i \cup \{i\}} \liminf_{n\to\infty} \frac{D_j(n)}{n} &\leq K_i(\mathcal{N}) \\ &< \sum_{j \in S_i \cup \{i\}} \lambda_{q(j)} \text{ (from (49)).} \\ \Rightarrow \liminf_{n\to\infty} \frac{D_j(n)}{n} &< \lambda_{q(j)} \text{ for some } j \in S_i \cup \{i\} \\ \Rightarrow \liminf_{n\to\infty} \frac{D_{L_j}(n)}{n} &< \lambda_{q(j)}. \end{aligned}$$

The last inequality follows since $D_{L_j}(n) \leq D_j(n)$ for all j, n. Thus, if $\lim_{n\to\infty} \frac{D_{L_j}(n)}{n}$ exists, then its value is less than $\lambda_{q(j)}$. 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. ■

Note that for the special case of single-hop sessions, sessions and session-links are identical, and for any single-hop session j, $\tilde{\beta}_j = \beta_j$. Thus, Theorem 2 follows from Lemma 5 since whenever any $\vec{\lambda}$ satisfies (4), $\vec{\lambda} \in \Lambda^{\mathrm{MS}}$.

F. Proof of Theorem 3

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, $(\lambda_{i_1}/K_{i_1}(\mathcal{N}), \lambda_{i_2}/K_{i_2}(\mathcal{N}), \lambda_{i_3}/K_{i_3}(\mathcal{N})) \notin \Lambda^{\mathrm{MS}}$. ■

G. Proof of Theorem 4

We prove Theorem 4 using Lemma 5 and another supporting lemma, Lemma 6, which we state and prove next.

Lemma 6: Let $\vec{\lambda}'$ *strictly satisfy (4) (i.e., the inequalities are strict). 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 $\vec{\lambda}'$ strictly satisfy (4). Let $\alpha_j(t)$ and $\tilde{D}_j(t)$ 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 $\widehat{n} = |\mathcal{X}_j|$. First, we obtain relations among these parameters. If session-link j satisfy $Q_j(\nu) > 0$ for every $\nu \in [t, t+\tau]$, then for every $\nu \in [t, t+\tau]$,

$$\sum_{k \in \mathcal{X}_j} \tilde{D}_k(\nu) \geq 1. \tag{50}$$

$$\begin{aligned} Q_j(t) + \sum_{\nu=t+1}^{t+\tau} \alpha_j(\nu) &\leq \sum_{\nu=1}^{t+\tau} A_{q(j)}(\nu) \\ &\leq t\alpha_{\max} + \sum_{\nu=t+1}^{t+\tau} A_{q(j)}(\nu). \end{aligned} \tag{51}$$

Now we have,

$$\begin{aligned} &\mathbb{P}\{B_{j,t} > \tau\} \\ &\leq \mathbb{P}\left\{\bigcap_{v=t}^{t+\tau}\left\{\left[\sum_{k\in\mathcal{X}_j} Q_k(t) + \sum_{\nu=t+1}^{v}\sum_{k\in\mathcal{X}_j}\alpha_k(\nu) \right.\right.\right. \\ &\qquad\qquad \left.\left.\left. - \sum_{\nu=t+1}^{v}\sum_{k\in\mathcal{X}_j}\tilde{D}_k(\nu) > 0\right]\right\}\right\} \\ &\leq \mathbb{P}\left\{\bigcap_{v=t+1}^{t+\tau}\left\{\sum_{k\in\mathcal{X}_j} Q_k(t)\right.\right. \\ &\qquad \left.\left. + \sum_{\nu=t+1}^{v}\left(\sum_{k\in\mathcal{X}_j}\alpha_k(\nu) - 1\right) > 0\right\}\right\} \text{ (from (50))} \\ &\leq \mathbb{P}\left\{\sum_{k\in\mathcal{X}_j} Q_k(t) + \sum_{\nu=t+1}^{t+\tau}\sum_{k\in\mathcal{X}_j}\alpha_k(\nu) - \tau > 0\right\} \\ &\leq \mathbb{P}\left\{\frac{t\widehat{n}\alpha_{\max}}{\tau} + \frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau}\sum_{k\in\mathcal{X}_j} A_{q(k)}(\nu) - 1 > 0\right\} \\ &\qquad \text{(from (51))} \\ &= \mathbb{P}\left\{\frac{t\widehat{n}\alpha_{\max}}{\tau} + \sum_{k\in\mathcal{X}_j}\left(\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{q(k)}(\nu) - \lambda'_{q(k)}\right)\right. \\ &\qquad\qquad \left. > 1 - \sum_{k\in\mathcal{X}_j}\lambda'_{q(k)}\right\}. \end{aligned}$$

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

$$\begin{aligned} &\mathbb{P}\{B_{j,t} > \tau\} \\ &\leq \mathbb{P}\left\{\left\{\frac{t\widehat{n}\alpha_{\max}}{\tau} > \frac{\delta}{\widehat{n}+1}\right\}\right. \\ &\qquad \left.\bigcup_{k\in\mathcal{X}_j}\left\{\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{q(k)}(\nu) - \lambda'_{q(k)} > \frac{\delta}{\widehat{n}+1}\right\}\right\} \\ &\leq \mathbb{P}\left\{\frac{t\widehat{n}\alpha_{\max}}{\tau} > \frac{\delta}{\widehat{n}+1}\right\} \\ &\qquad + \sum_{k\in\mathcal{X}_j}\mathbb{P}\left\{\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{q(k)}(\nu) - \lambda'_{q(k)} > \frac{\delta}{\widehat{n}+1}\right\} \\ &= \sum_{k\in\mathcal{X}_j}\mathbb{P}\left\{\frac{1}{\tau}\sum_{\nu=t+1}^{t+\tau} A_{q(k)}(\nu) - \lambda'_{q(k)} > \frac{\delta}{\widehat{n}+1}\right\} \\ &\qquad \text{if } \tau > \frac{t\alpha_{\max}}{\delta}. \end{aligned}$$

Now, from (1), the packet queue of every session-link will almost surely become empty infinitely often. Also,

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

■

Theorem 4 follows from Lemmas 5 and 6.

References

[1] D. Bertsekas and R. Gallager. *Data Networks (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[2] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu. Joint asynchronous congestion control and distributed scheduling for multihop wireless networks. In *Proceedings of INFOCOM*, Barcelona, Spain, April 2006.

[3] P. Chaporkar, K. Kar, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in multihop wireless networks. Technical report, Univ. of Pennsylvania, Philadelphia, PA, Jul 2005. http://www.seas.upenn.edu/∼swati/publication.htm.

[4] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in multihop wireless networks. In *Proceedings of 43d Annual Allerton Conference on Communication, Control and Computing*, Allerton, Monticello, Illinois, September 28-30 2005.

[5] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proceedings of INFOCOM*, Miami, FL, Mar 2005.

[6] H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *Proceedings of MOBICOM*, pages 76–86, Boston, MA, August 2000.

[7] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving mac layer fairness in wireless packet networks. In *Proceedings of MOBICOM*, pages 87–98, Boston, MA, August 2000.

[8] M. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proceedings of INFOCOM*, Miami, Florida, March 2005.

[9] D. Peleg. *Distributed Computing: A Locality-sensitive Approach*. Society of Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[10] S. Sarkar and L. Tassiulas. End-to-end bandwidth guarantees through fair local spectrum share in wireless ad-hoc networks. *IEEE Transactions on Automatic Control*, September 2005.

[11] L. Tassiulas and A. Ephremidis. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, Dec 1992.

[12] L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless ad hoc networks. *IEEE Journal for Selected Areas in Communications, special issue on Ad Hoc Networks, Part I*, 23(1), January 2005.

[13] X. Wu and R. Srikant. Regulated maximal matching: a distributed scheduling algorithm for multihop wireless networks with node-exclusive spectrum sharing. In *Proceedings of IEEE CDC-ECC'05*, Seville, Spain, Dec 2005.

[14] X. Wu and R. Srikant. Bounds on the capacity region of multihop wireless networks under distributed greedy scheduling. In *Proceedings of INFOCOM*, Barcelona, Spain, April 2006.