

End-to-End Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless *Ad-Hoc* Networks

Saswati Sarkar, *Member, IEEE*, and Leandros Tassiulas, *Member, IEEE*

Abstract—Sharing the common spectrum among the links in a vicinity is a fundamental problem in wireless *ad-hoc* networks. Lately, some scheduling approaches have been proposed that guarantee fair share of bandwidth among the links. The quality of service perceived by the applications however depends on the end-to-end bandwidth allocated to the multihop sessions. We propose an algorithm that provides provably maxmin fair end-to-end bandwidth to sessions. The algorithm combines a link scheduling that avoids collisions, a fair session service discipline per link, and a hop-by-hop window flow control. All the stages of the algorithm are implementable based on local information, except the link scheduling part that needs some network-wide coordination.

I. INTRODUCTION

LINK transmission scheduling in multihop wireless networks has been investigated over the last twenty years. The earlier work was focused on guaranteeing end-to-end connectivity whenever that was feasible [1], [8], [14]. Transmission scheduling algorithms that provide some guarantees on the rates obtained by each link have lately been proposed [6], [7], [15], [11], [22]. The quality of service perceived by the applications, however, depends on the end-to-end bandwidth allocated to the multihop sessions. We address the objective of providing maxmin fair end-to-end bandwidth to sessions.

Providing end-to-end rate guarantees in wired networks has been studied extensively [4], [13]. Nevertheless, a separate investigation is necessary for wireless *ad-hoc* networks as the scheduling constraints are different from the wireline networks. For example, the strategies in wireline networks rely on the feature that links can be scheduled independent of each other. However, in wireless *ad-hoc* networks links traversing the same node can not simultaneously transmit packets. Different design techniques are required to address these scheduling dependencies. Also, in *ad-hoc* networks the scheduling decisions are taken at the MAC layers of individual nodes that do not have access to network-wide information. However, the fair rate of each session depends on several network wide attributes, e.g., the routes and packet generation rates of different sessions, the congestion in vicinity of each node, etc. It therefore appears that attaining end-to-end fair bandwidth would require cross-layer optimization, knowledge

of network wide information at each node and coordination of the scheduling at different nodes in a session's path.

We combine features of network control approaches for wired networks with wireless link scheduling techniques to design a provably maxmin fair rate allocation algorithm. The algorithm combines a link scheduling that avoids collisions, a fair session service discipline per link and a hop-by-hop window flow control. It does not need the nodes to know any network or transport layer information, e.g., the end-to-end session routes, the packet arrival rates, etc., and does not compute the fair rates apriori. Thus, the optimality result that proves maxmin fairness of the resulting end-to-end rates is an important contribution of this paper, and indicates that cross-layer optimization does not introduce significant additional complexity in this case.

The proposed algorithm is centralized. It is however worthwhile to observe that the only centralized component is the link scheduling that needs to compute a maximum weighted matching of the network graph. We will discuss a possible approach based on a recent result in scheduling [9], for approximating this computation by a distributed policy. This may lead to a fully distributed solution with suboptimal performance. Note that like many other scheduling strategies proposed for wireless networks, e.g., [5], [12], [21], and [22], we need this computation for optimally resolving the link dependencies, and not for sessions being multi-hop.

Recently, Radunovic *et al.* [16] proposed a centralized algorithm for computing the end-to-end maxmin fair rates. It may be possible to use these computed rates to regulate the source's release of packets (e.g., at the transport layer) so as to obtain the maxmin fair rates. The disadvantage of this sequential approach is that the rates must be recomputed every time the topology or the packet arrival rates change. This does not happen with our algorithm since it does not have a precomputation phase. Furthermore, they assume that a node can transmit simultaneously on multiple links, which several transceivers can not do.

Finally, fairness can be defined in many different ways, and maxmin fairness is one of these notions. Another notion for fairness is to maximize the sum of the certain functions of the rates (utilities) of all users. Recently, algorithms have been proposed for attaining this goal in wireless networks [3], [9], [12], [23].

In Section II, we describe the fairness objective and the network model, and present conditions that are necessary and sufficient for a bandwidth allocation to be maxmin fair. In Section III, we present a scheduling strategy that attains maxmin fairness and describe its performance guarantees. In Section IV, we investigate via simulation the performance of the algorithm for different values of certain parameters. In Section V, we discuss several implementation related features

Manuscript received June 25, 2004; revised February 26, 2005. Recommended by Associate Editor R. S. Srikant. The work of S. Sarkar was supported in part by the National Science Foundation under Grants ANI01-06984, NCR02-38340, and CNS04-35306.

S. Sarkar is with the Department of Electrical and Systems Engineering, the University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: swati@ee.upenn.edu).

L. Tassiulas is with the Computer Engineering and Telecommunications Department, the University of Thessaly, Volos, Greece (e-mail: leandros@inf.uth.gr).

Digital Object Identifier 10.1109/TAC.2005.854580

TABLE I
SUMMARY OF THE FREQUENTLY USED SYMBOLS

Symbol	Meaning
N	Number of sessions
r_i	Session i 's maxmin fair rate
ρ_i	Session i 's packet generation rate
σ_i	Burstiness of session i 's packet generation process, $\sigma = \max_i \sigma_i$
W	Window Parameter
L_i	Number of nodes in session i 's path
L	Maximum number of nodes in the path of sessions
\mathcal{R}_n	Set of sessions node n relays
\mathcal{S}_n	Set of sessions that originate at node n
\mathcal{D}_n	Set of sessions that terminate at node n
d_n (degree of node A)	Sum of the number of sessions that originate or terminate at node n and twice the number of sessions that node n relays ($ \mathcal{S}_n + \mathcal{D}_n + 2 \mathcal{R}_n $)
d_{\max}	Maximum degree of a node
F	Number of distinct ranks in the system
$S_{i,n}(t)$	Number of times session i has been sampled at node n in interval $(0, t)$
$C_{i,n}(t)$	Number of tokens of session i generated at node n in interval $(0, t)$
$P_i(t)$	Number of packets of session i generated at its source in interval $(0, t)$
$P_i^{\text{NR}}(t)$	Number of packets of session i at its source at time t that have not been released

of our algorithm. We summarize the frequently used symbols in Table I, and present the proofs in the Appendix.

II. FAIRNESS OBJECTIVE AND NETWORK MODEL

We consider a wireless network with V nodes, E links, and N multihop sessions. Each session has a source, destination and some relay nodes. The session routes are predetermined. Time is slotted. Every node has one radio unit. Thus, in a slot, a node can either transmit one packet, or receive one packet, or remain idle. We assume that all packets have the same number of bits. Every link can transmit 1 packet per slot. Thus, if a session i transmits at the rate of r packets per slot, each of its relays must serve (transmit or receive) i 's packets for $2r$ fraction of total time, and i 's source and destination must serve i 's packets for r fraction of total time. Every node has a frequency that is unique in its two-hop neighborhood. Thus, only the transmissions that have a common node interfere with each other. Hence, the links that are active at any slot must constitute a matching. For example, a bluetooth network satisfies the previous assumptions [10]. We do not consider channel errors.

A bandwidth allocation (r_1, \dots, r_N) is feasible if there exists a scheduling sequence that attains rate r_i for each session i . We examine the conditions for feasibility of a bandwidth allocation. First, assume that the source node for each session

has an infinite supply of packets at all times. Let \mathcal{S}_n (\mathcal{D}_n) be the set of sessions that originate (terminate) at node n , and \mathcal{R}_n be the set of sessions node n relays. Hajek *et al.* [5] showed that if the network is a bipartite graph,¹ then a bandwidth allocation (r_1, \dots, r_N) is feasible if and only if at each node n , $2 \sum_{i \in \mathcal{R}_n} r_i + \sum_{i \in \mathcal{S}_n \cup \mathcal{D}_n} r_i \leq 1$. Intuitively, $2 \sum_{i \in \mathcal{R}_n} r_i + \sum_{i \in \mathcal{S}_n \cup \mathcal{D}_n} r_i$ is the bandwidth consumed by sessions traversing node n as a session i can receive a rate r_i only if each of its relay nodes serve it for $2r_i$ fraction of the total time and its source and destination serve it for r_i fraction of the total time. Thus, a bandwidth allocation (r_1, \dots, r_N) is feasible if and only if at each node n the sum of the bandwidth consumed at n by all sessions traversing n is less than or equal to 1. Many wireless networks, e.g., bluetooth networks are bipartite graphs. For nonbipartite graphs, a sufficient (but not necessary) condition for the feasibility of a bandwidth allocation (r_1, \dots, r_N) is that $2 \sum_{i \in \mathcal{R}_n} r_i + \sum_{i \in \mathcal{S}_n \cup \mathcal{D}_n} r_i \leq 2/3$ for each node n [5]. In practice, bandwidth is allocated so as to utilize the bandwidth capacity of the nodes only partially under normal circumstances, and reserve the rest of the capacity for use during transient overloads. Thus, combining the two cases, we assume that a condition for feasibility of a bandwidth allocation $2 \sum_{i \in \mathcal{R}_n} r_i + \sum_{i \in \mathcal{S}_n \cup \mathcal{D}_n} r_i \leq \alpha$ where α is the desired bandwidth utiliza-

¹A bipartite graph is one where the vertex set can be partitioned in two sets such that there is no edge between the vertices in the same set.

tion factor ($\alpha \leq 2/3$ for nonbipartite graphs and $\alpha \leq 1$ for bipartite graphs). We refer to this constraint as the *node capacity constraint*.

A source may not have an infinite supply of packets. If a session i generates packets at rate ρ_i , then its bandwidth r_i is upper bounded by ρ_i ; $r_i \leq \rho_i$. We refer to this constraint as the *demand constraint*.

Definition 1: A bandwidth allocation is feasible if and only if it satisfies both the demand and the node capacity constraints.

Definition 2: A feasible bandwidth allocation (r_1, r_2, \dots, r_N) is maxmin fair, if it satisfies the following property w.r.t. any other feasible bandwidth allocation (s_1, \dots, s_N) : if there exists a i such that $r_i < s_i$, then there exists a j such that $r_j \leq r_i$ and $s_j < r_j$.

Maxmin fairness is considered to be a good notion of fairness, as it guarantees equal bandwidth to sessions that traverse paths of similar congestion level and generate packets at equal rates. Note that the maxmin fair rate allocation is unique since the feasible set is compact and convex [17].

For nonbipartite topology graphs the feasible set as defined before (referred to as $\underline{\mathcal{F}}$) is a subset of the actual feasible set (referred to as \mathcal{F}). Thus, the maxmin fair allocation $(\underline{\vec{M}})$ in $\underline{\mathcal{F}}$ is an approximation of that (\vec{M}) in \mathcal{F} . We now comment on the nature of this approximation. We first describe the notion of relative fairness proposed in [17].

Definition 3: A bandwidth allocation \vec{A} is fairer than another bandwidth allocation \vec{B} if: a) $\vec{A} \neq \vec{B}$, and b) existence of an i such that $A_i < B_i$, implies that there exists a j such that $A_j \leq A_i$ and $B_j < A_j$.

This definition leads to another equivalent definition of a maxmin fair bandwidth allocation [19].

Definition 4: A feasible bandwidth allocation \vec{A} is maxmin fair if and only if it is fairer than all other feasible bandwidth allocations.

The feasible set $\bar{\mathcal{F}}$ obtained by selecting $\alpha = 1$ in the node-capacity constraints is a superset of \mathcal{F} . It turns out that the maxmin fair bandwidth allocation \vec{M} in $\bar{\mathcal{F}}$ has the following properties: a) $\vec{M} = 3/2\vec{M}$, and b) \vec{M} is fairer than or equal to \vec{M} . Thus, for nonbipartite graphs the approximate maxmin fair bandwidth allocation \vec{M} is in the worst case $2/3$ times a bandwidth allocation that is fairer than the actual maxmin fair bandwidth allocation \vec{M} . Henceforth, for simplicity, in the nonbipartite topology graphs, we will refer to the approximate maxmin fair bandwidth allocation \vec{M} as the maxmin fair bandwidth allocation.

We now present a necessary and sufficient condition for maxmin fairness.

Definition 5: A node n is a bottleneck node of a session i if session i 's bandwidth is the maximum among the bandwidth of the sessions traversing n and $2\sum_{i \in \mathcal{R}_n} r_i + \sum_{i \in n\mathcal{S}_n \cup \mathcal{D}_n} r_i$ is equal to the bandwidth utilization factor α .

We next present a necessary and sufficient condition for maxmin fairness in multihop wireless networks, which is similar to the bottleneck condition for maxmin fairness in wireline networks [2].

Lemma 1: A feasible bandwidth allocation is maxmin fair if and only if the following holds: for every session i , either the bandwidth allocated to session i is equal to ρ_i , or the session has a bottleneck node.

III. BACK-PRESSURE BASED FAIR BANDWIDTH ALLOCATION ALGORITHM

We propose a modular approach for attaining maxmin fairness for multihop sessions. 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. This modularization allows the use of different algorithms in the first module for different fairness objectives, e.g., weighted maxmin fairness (Section V). The bandwidth shares computed as per the desired objective can now be attained using, in the second module, the existing maximum difference backlog scheduling [21]. This scheduling can stabilize the network for any feasible arrival process. Since the packet release process is fair and, hence, feasible, the overall framework attains the desired fairness objective. We present the basic algorithm that attains maxmin fairness in Subsection III-A, and consider generalizations in Subsection III-B and Section V.

A. Basic Algorithm

We consider the special case that every session is saturated, i.e., its source node always has a packet ($\rho_i = \infty, \forall i$). We present the algorithm in Fig. 1. Here, we describe each part. Fair bandwidth is estimated by a token generation process. Every node generates tokens for all sessions traversing the node. The token generation process is so designed that the tokens are generated for each session at the session's maxmin fair rate. Whenever a new token is generated for a session at the session's source, the source node releases a new packet for transmission. Thus, the packet release process is maxmin fair. Only the released packets are eligible for transmission.

We describe the token generation process for a session i at node m . Node m samples in round robin order the sessions traversing it. Node m samples sessions in at most α fraction of slots. Also, whenever m generates a token for a session it relays, it does not sample in the next slot. Thus, the sum of the token generation rates of sessions originating or terminating at m and twice the token generation rates of sessions m relays does not exceed α . Let l and n be the immediate upstream and downstream (i.e., adjacent) nodes of node m in session i 's path. Let $C_{i,p}(t)$ be the number of tokens generated for session i at node p in the interval $[0, t]$. Let m sample session i in slot t . Then, m generates a token to session i in slot t if and only if $C_{i,m}(t) < \min(C_{i,l}(t), C_{i,n}(t)) + W$. Thus, session i receives a token at m unless the number of tokens for session i at m substantially exceeds that at the adjacent nodes; this prohibitive difference is the window parameter, W . In slot t , m samples the next session in the round robin order if and only if m does not generate a token for session i . Note that the source (destination) node of a session has only one adjacent node for the session, and thus decides whether to generate a token based on the number of tokens at only one adjacent node. Tokens are never removed from a node.

We now explain why for each session the token generation rate equals the maxmin fair rate. The maxmin fair rate of a session is determined by the bandwidth offered by the session's

Procedure Token Generation (node m)
begin

α fraction of slots are designated as sampling slots; m samples in a slot t only if t is a sampling slot and in slot $t - 1$ m did not generate token to a session it relays.

Node m samples sessions in round robin order.

Let session i traverse node m , and

nodes l, n be adjacent to node m in session i 's path.

When session i is sampled in slot t :

if $C_{i,m}(t) < C_{i,l}(t) + W$ and $C_{i,m}(t) < C_{i,n}(t) + W$, **then**

 Generate a token for session i in slot t ($C_{i,m}(t+1) = C_{i,m}(t) + 1$);

else

 Do not generate token for session i ($C_{i,m}(t+1) = C_{i,m}(t)$), and

 Sample the next session in the round robin order.

end**Procedure Packet Release (source i)****begin**

Release a new session i packet for transmission at session i source when a token is generated for the session at the source.

end**Procedure Packet Scheduling For Transmission (link e)****begin**

Let e be the link between the nodes m and n ,

\mathcal{L}_e be the set of sessions traversing link e , and

$P_{i,n}^R(t)$ be the number of released packets of session i waiting at node n at time t .

$W_e(t) = \max_{i \in \mathcal{L}_e} (P_{i,m}^R(t) - P_{i,n}^R(t))$ (*Weight of link is the maximum difference in backlog across the link *).

Schedule the links that constitute a maximum weighted matching.

If link e is scheduled, then

transmit a packet of session j from node m to node n if

$P_{j,m}^R(t) - P_{j,n}^R(t) = \max_{i \in \mathcal{L}_e} (P_{i,m}^R(t) - P_{i,n}^R(t))$ /*session j has the maximum difference of backlog across e */.

end

Fig. 1. Pseudocode of the fair bandwidth allocation algorithm for saturated sessions.

bottleneck node which is the most congested node in the session's path. Intuitively, a session's token generation rate at any node in its path should equal that at its bottleneck node; the difficulty in attaining this equality is that the nodes do not have explicit information about the bottleneck node. A node learns the bottleneck information implicitly by relating the token generation process for a given session to that at the adjacent nodes in the session's path. The number of tokens for a session at two adjacent nodes in the session's path differ by at most W at any time t , and the difference is at most LW for that at any two nodes in the session's path, where L is the maximum number of nodes in the session's path. Thus, the rates of token generation for a session are equal at any two nodes in the session's path. A session's bottleneck node samples it at the least rate in its path. Hence, a session's token generation rate at each node is upper bounded by its sampling rate at its bottleneck node; the token generation rate in fact equals this sampling rate which turns out to be the session's maxmin fair rate.

Lemma 2: Let r_1, r_2, \dots, r_N be the maxmin fair rates of the sessions. Let $C_{i,n}(t)$ be the number of tokens generated for session i at node n in interval $(0, t)$. Then, in any time interval (x, y) , $|C_{i,n}(y) - C_{i,n}(x) - r_i(y - x)| \leq \varrho$, if $W \geq W_0$, where W_0 and ϱ are constants.

We define W_0 and ϱ in the proof of Lemma 2, in terms of N, L, d_{\max}, α (refer to Table I and Fig. 1 for definitions of these quantities). Note that the values of W_0 and ϱ do not depend on the interval (x, y) .

The implicit discovery of the bottleneck information from the bandwidth allocation process at neighboring nodes has been motivated by fair bandwidth allocation algorithms in wireline networks [4]. This is commonly termed as "back-pressure." The mechanisms used in this discovery, e.g., the sampling and the token generation, and also the link scheduling are however significantly different due to the dependence between the scheduling of different links in wireless networks which does not arise in wireline networks.

```

Procedure Token Generation at Source Node (source  $i$ )
begin
  Let node  $n$  be the source of session  $i$ .
  Let  $P_i^{\text{NR}}(t)$  be the number of packets of session  $i$  at time  $t$  that have been generated at  $n$  but not been released.
  Sampling procedure is the same as that in Figure 1 for all sessions traversing  $n$ .
  Token generation procedure for all sessions other than  $i$  is similar to that in Figure 1.
  Let node  $l$  in session  $i$ 's path be the immediate downstream node of node  $n$ .
  When session  $i$  is sampled in slot  $t$ :
  if  $C_{i,n}(t) < C_{i,l}(t) + W$  and  $P_i^{\text{NR}}(t) > 0$ , then
    generate token for session  $i$  ( $C_{i,n}(t+1) = C_{i,n}(t) + 1$ );
  else
    Do not generate token for session  $i$  ( $C_{i,n}(t+1) = C_{i,n}(t)$ ), and
    Sample the next session in the round robin order.
end

```

Fig. 2. Pseudocode of the token generation process at the source nodes for systems with some unsaturated sessions.

Whenever the source node of a session generates a new token for the session, it releases a new packet. The maximum difference in backlog scheduling [21] transmits the released packets along the prespecified routes to the desired destinations. The maximum difference in backlog scheduling assigns a weight to each link as follows. The difference in backlog of a session in a link is equal to the difference between the number of released packets of the session waiting at the source node of the link and that at the destination node of the link. The weight of a link is the maximum difference in backlog of the sessions traversing the link. Note that only a session's source may have packets that have not been released. The links that constitute a maximum weighted matching are scheduled for service. When a link is scheduled, a released packet of the session that has the maximum difference in backlog in the link is served. The maximum difference in backlog scheduling stabilizes a network if the packet arrival process is feasible [20]. The packet arrival process in the current network is the packet release process. Tokens and, hence, packets are released for each session at the maxmin fair rate (Lemma 2) which is feasible by definition. The network is thus stable, and hence the rate of delivery of packets of a session to the session's destination equals the session's packet release rate which is the session's maxmin fair rate.

Theorem 1: Let r_1, r_2, \dots, r_N be the maxmin fair rates of the sessions. Let $D_i(t)$ be the number of packets for session i that have reached session i 's destination by time t . Then, in any time interval (x, y) , $|D_i(y) - D_i(x) - r_i(y - x)| \leq \kappa$, if $W \geq W_0$, where, W_0 and κ are constants.

Note that κ , which we define in the following proof, does not depend on the interval (x, y) .

Proof of Theorem 1: From Lemma 2, the number of packets of session i released from i 's source in any interval (x, y) differs from $r_i(y - x)$ by at most ϱ . Since (r_1, \dots, r_N) is feasible, results in [20] show that the packet queue length at any time in any node is bounded by a constant κ_1 , if the

maximum difference in backlog scheduling is used. Thus, the number of i 's packets that reach i 's destination in any interval (x, y) differs from $r_i(y - x)$ by at most a constant κ , where $\kappa = L\kappa_1 + \varrho$. Note that κ_1 and ϱ depend on N, L, d_{\max}, α and not on the interval (x, y) . \square

Theorem 1 shows that in any interval the number of packets of a session delivered to the destination differs from the maxmin fair number by at most a constant. Thus, the long term rates are maxmin fair.

B. Generalization for Addressing the Unsaturated Case

A session i is unsaturated if its source does not always have packets for transmission, i.e., $\rho_i < \infty$. When some or all sessions are unsaturated, the maxmin fair rates can be attained by altering the token generation procedure at the source node in the basic algorithm in Section 1; refer to Fig. 2 for the modification. The source node of a session now does not generate a new token for the session if it does not have a packet that has not been released. The rest of the algorithm remains the same. 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.

If a session has a low rate of packet generation, then it receives fewer tokens at the source and subsequently at other nodes as well, since back-pressure upper bounds by a constant LW the difference between the number of tokens for a session in any two nodes. Thus, the session obtains fewer transmission opportunities, and hence, less bandwidth as required for maxmin fairness.

We prove Lemma 2 in the Appendix for pseudodeterministic (ρ, σ) arrival processes in which the number of packets generated for any session i in any interval of length t is upper bounded by $t\rho_i + \sigma_i$ and lower bounded by $t\rho_i - \sigma_i$ for any t . Here, ρ_i is the long term arrival rate of a session i , and σ_i is the burstiness. Similar to the saturated case, Theorem 1 can be proved using Lemma 2 and, thus, the modified algorithm attains the maxmin

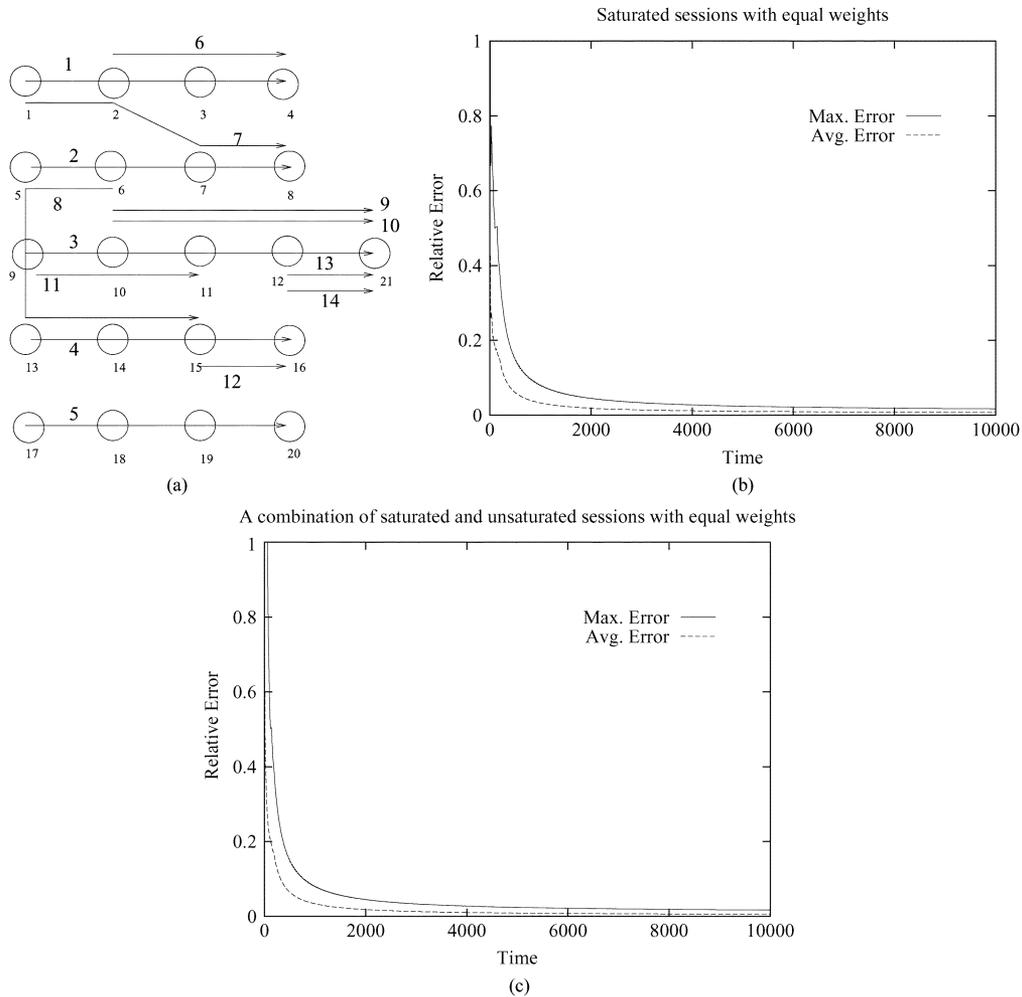


Fig. 3. First figure shows the topology with 21 nodes and 14 sessions that is used in the simulations. The second figure considers the case when all the sessions are saturated. The third figure considers the case when all the sessions are saturated, except session 7 which receives packets at the rate 0.1 per unit time.

fair rates. Let $\sigma = \max_i \sigma_i$. Now, W_0, ρ, κ_1 , and κ depend on N, L, d_{\max}, α , and σ .

IV. PERFORMANCE EVALUATION

We now examine using simulations: a) the time required for convergence of the token generation rates to the maxmin fair rates, and b) how the convergence depends on the choice of the window parameter W . Note that we do not have a tight analytical bound on the convergence time. The lower bound on W, W_0 , needed to guarantee the convergence results in Lemma 2 and Theorem 1, depends on N, L, d_{\max}, α , and σ . Thus, this bound is impossible to compute without explicit knowledge of the network topology. This motivates the investigation of the impact of different choices of W on the convergence of the token generation rates to the maxmin fair values.

We present simulation results for a network of 21 nodes and 14 sessions that is shown in Fig. 3. Here, $W = 5$. We simulate the token generation procedure in C. We do not simulate the maximum difference in backlog scheduling, as it has been known to attain any feasible rate if the packet arrival process is

feasible [20]. We consider the relative difference between the long term token generation rate for each session i at its source ($C_{i,n}(t)/t$) and the maxmin fair rate (r_i). The relative difference, which we call relative error, at time t for session i is $|1 - (C_{i,n}(t)/r_i t)|$. We plot the maximum and average relative errors over all sessions as a function of t in Fig. 3. In Fig. 3, the second figure considers the case when all the sessions are saturated; the third figure considers the case when all sessions are saturated except session 7 that receives packets at the rate 0.1 per unit time.

We observe the following from Fig. 3. The average relative error decays fast, e.g., it is less than 0.05 within 500 slots. The maximum relative error decays slower indicating that a few sessions experience slower convergence. The token generation rates converge to the maxmin fair rates even though $W = 5$; the lower bound W_0 for guaranteed convergence is 2^{70} . We observed similar trends for several other topologies. We conclude that on an average, the token generation rate converges rapidly to the maxmin fair bandwidth. Also, in practice, convergence is not sensitive to the choice of W and moderate values of W , e.g., $W \approx 5$, ensure convergence. Thus, small window sizes can be used to control the delay and buffer requirements.

V. DISCUSSION AND CONCLUSION

The token generation and the scheduling can operate in parallel. A sequential operation increases the overall delay in attaining the desired bandwidth allocation.

The scheme need not restart if new sessions join or old sessions leave. The analytical performance guarantees hold even in this dynamic scenario.

The analytical performance guarantees hold even when a node knows the number of tokens at its neighbors after a delay, as long as the delay is upper bounded. We have shown in [18] that the rates obtained by a similar back-pressure technique converges to the maxmin fair rates irrespective of the feedback delay. The guarantees also hold when the scheduling decisions are taken, and/or tokens are generated at intervals, as long as the intervals are upper bounded. When tokens are generated at intervals, multiple tokens must be generated to each session simultaneously.

The algorithm is adaptive as it does not need the statistics of the arrival process.

The algorithm requires per-flow states at the nodes. However, this is not likely to substantially increase the complexity in wireless networks as the number of sessions traversing a node is limited by the available bandwidth rather than by the computational complexity.

A node can execute the token generation and the packet release processes with the knowledge of the status of only its one-hop neighbors. The maximum difference backlog scheduling is however a centralized procedure as it requires the computation of a maximum weighted matching. A maximal matching² can however be computed in a distributed manner [9]. Recently, performance guarantees have been obtained for maximal matching based schedulings [9]. It may be possible to use these results to prove that the combination of the current token generation scheme, packet release process and a scheduling that uses maximal matching instead of the maximum backlog scheduling attain approximately maxmin fair rates. This is a topic of future investigation.

The system does not remove any token. Thus, the register storing the number of tokens may overflow. The performance guarantees hold if equal number of tokens of a session are removed from each node in the session's path. The removal process can be executed by periodic exchange of synchronization information. The additional system overhead is small as the periods are long.

We now consider some important generalizations. We first consider the more generalized notion of weighted maxmin fairness. Let the weight of session i be G_i . A feasible bandwidth allocation (r_1, r_2, \dots, r_N) is weighted maxmin fair, if it satisfies the following property w.r.t. any other feasible bandwidth allocation (s_1, \dots, s_N) : If there exists a i such that $r_i < s_i$, then there exists a j such that $r_j/G_j \leq r_i/G_i$ and $s_j < r_j$. Unequal weights allow allocation of bandwidth on the basis of the quality of service requirements. A session with a higher weight can have a bandwidth higher than another even if both sessions travel the same path and generate packets at the same

rate. The sampling procedure in the basic algorithm presented in Section I must be altered to attain the weighted maxmin fair rates. In any slot t , node n samples the session that has the minimum weighted number of tokens, i.e., the minimum value of $C_{i,m}(t)/G_i$ among all sessions i traversing the node. Thus, the sessions with higher weights are sampled more often. The rest of the algorithm remains the same. Theorem 1 and Lemma 2 hold.

We have so far assumed that every link can transmit 1 packet per unit time. We now consider that link l can transmit c_l packets per unit time.³ First, the node capacity condition for feasibility changes because of this generalization. Let $\mathcal{L}_{i,n}$ be the set of links in session i 's path that are incident on node n . Note that $\mathcal{L}_{i,n}$ consists of two links if n relays i and one link if n is the source or destination of i . Now, the node capacity constraint is that at each node n $\sum_{i: i \text{ traverses } n} \sum_{l \in \mathcal{L}_{i,n}} r_i/c_l \leq \alpha$ [23]. A modified token generation and scheduling algorithm will attain the maxmin fair rates. The token generation process differs in that a node now generates fractional number of tokens. More specifically, when a node n samples a session i it generates $(1/\sum_{l \in \mathcal{L}_{i,n}} (1/c_l))$ tokens and it samples the sessions in all sampling slots (nodes no longer idle in slots succeeding the token generation slots since they now generate fractional tokens). The source of each session maintains a queue of packets that it has generated but not released for transmission, and releases a packet from this queue only when the difference between the amount of tokens and the number of released packets is 1 or more. The source does not generate any token if this queue is empty. The rest of the token generation process is the same. We assume that c_l is rational for each l . Let $c_l = a_l/b_l$ where a_l, b_l are relatively co-prime integers. Let J be the least common multiple of b_l s for all links. The scheduling process differs in that the scheduling must be computed at the beginning of intervals of J slots and persisted with during the interval. This is necessary since the links can not transmit fractional packets.

We now outline a scheduling algorithm that would attain the maxmin fair rates in the actual feasible set in nonbipartite graphs. For simplicity, we again assume that $c_l = 1$ for each l . Let $\mathcal{M}(i, Q)$ be the set of links that are in session i 's path and have both end points in a set Q of nodes. A rate allocation vector is feasible in a nonbipartite graph if and only if for each subset of nodes, Q , with odd cardinality (i.e., odd $|Q|$), $\sum_{i=1}^N r_i |\mathcal{M}(i, Q)| \leq (|Q| - 1)/2$ [5]. There exists a token-generator for each constraint, which will be denoted by the corresponding subset. Each token generator samples in round-robin order all the sessions involved in the constraint. Let $M_{i,Q}(t)$ be the number of tokens generated by the token generator for constraint Q for session i till time t . When the generator for Q samples i , it generates $1/|\mathcal{M}(i, Q)|$ token to i if and only if the total number of tokens generated by a generator corresponding to another constraint involving i is at least $M_{i,Q}(t) - W$ where W is a parameter. Note that if $\mathcal{M}(i, Q) = \phi$, then session i is not involved in constraint Q . Session i releases a packet for transmission whenever $\min_Q M_{i,Q}(t)$ increases, and the released packets are transmitted as per the maximum

²A set of edges is a maximal matching if it is a matching and if every edge not in the matching shares an end-point with some edge in the matching.

³For simplicity, we still assume that all packets have the same number of bits, but this can be relaxed easily.

difference in backlog scheduling described before. The rate of packet delivery at each destination is maxmin fair in the actual feasible set. This token generation can be implemented in distributed manner by maintaining separate token generators for different constraints at each node involved in the constraint, and relating the generation processes in the adjacent nodes as described before. The token generation will still require exponential number of generators per node and therefore consume exponential computation complexity per slot. This limits its practical utility. Some open research problems are to determine whether there exists a polynomial complexity token generation scheme for attaining the exact maxmin fair rates in nonbipartite topology graphs, and designing one if one such exists.

We now summarize the contributions of this paper. We consider the objective of attaining maxmin fairness in multihop wireless *ad-hoc* networks. We present conditions that are necessary and sufficient for a bandwidth allocation to be maxmin fair. We present an adaptive scheduling strategy that attains maxmin fairness for sessions with different traffic demands. The strategy can be generalized to attain other fairness objectives such as weighted maxmin fairness.

APPENDIX PROOF OF LEMMA 1

Consider a feasible bandwidth allocation (r_1, \dots, r_N) . Assume that for each i , either $r_i = \rho_i$, or session i has a bottleneck node. We show that (r_1, \dots, r_N) is maxmin fair. Consider any session i . If $r_i = \rho_i$, then i 's bandwidth can not be increased while maintaining feasibility. Let i have a bottleneck node, n . If i 's bandwidth is increased, the bandwidth of some other session j that traverses n must be decreased to satisfy the node capacity constraint. Bandwidth of any session traversing n is either less than or equal to that of i . Thus, any increase in i 's bandwidth will decrease that of some session j that has bandwidth less than or equal to i 's bandwidth. Thus, (r_1, \dots, r_N) is maxmin fair.

Now consider a maxmin fair bandwidth allocation (r_1, \dots, r_N) . We will show that for each i , either $r_i = \rho_i$, or session i has a bottleneck node. Consider any session i . Let $r_i < \rho_i$ and let i not have a bottleneck node. Thus, at each node in i 's path, either there exists a session j such that $r_j > r_i$, or the sum of the bandwidth of all sessions traversing node n is less than α . In either case, i 's bandwidth can be increased without decreasing that of any other session that has bandwidth less than or equal to i 's bandwidth and without violating the feasibility conditions. This contradicts the fact that (r_1, \dots, r_N) is maxmin fair.

APPENDIX PROOF OF LEMMA 2

We prove Lemma 2 for a system that consists of both saturated and unsaturated sessions. Let, $P_i(t)$ be the number of packets generated for session i at i 's source in interval $(0, t)$.

$$\text{Then, } |P_i(t) - \rho_i t| \leq \sigma_i \quad \forall t. \quad (1)$$

Let $\alpha = \mu/\eta$, where μ and η are relatively prime positive integers. Let each node sample the sessions in μ slots in every contiguous η slots.

We prove using the following steps. a) 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 node in its path, then it receives tokens at rate r or higher from every node in its path (Lemma 3). b) We next show that a session's sampling rate at any node in its path equals its maxmin fair rate (Lemma 4). By definition, a session's maxmin fair rate is less than or equal to its packet generation rate. The result follows.

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 node n in the interval $(0, t)$, L_i be the number of nodes in session i 's path, $L = \max_i L_i$, and β, γ are constants that will be formulated later. We prove Lemmas 3 and 4 in Sections C and D.

Lemma 3: Consider an arbitrary K and a sequence of K disjoint intervals, $(t_l, w_l), l = 1, \dots, K$, that satisfies the following property for session i , for every positive integer M and every sequence of subintervals $(x_m, y_m), m = 1, \dots, M, (x_m, y_m) \subset (t_l, w_l)$, for some l . At every node n in i 's path

$$\sum_{m=1}^M (S_{i,n}(y_m) - S_{i,n}(x_m)) \geq r \sum_{m=1}^M (y_m - x_m) - e - Mf \quad (2)$$

where e and f are constants that do not depend on M and the sub intervals $(x_m, y_m), m = 1, \dots, M$. Let $\rho_i \geq r$ and $W \geq 3^{L_i-1}(f + \sigma_i)/2$. Then, at every node n in i 's path

$$\begin{aligned} & \sum_{l=1}^K (C_{i,n}(w_l) - C_{i,n}(t_l)) \\ & \geq r \sum_{l=1}^K (w_l - t_l) - 2^{L_i-1}e - K3^{L_i-1}(f + \sigma_i). \quad (3) \end{aligned}$$

Lemma 4: Let r_1, \dots, r_N be the maxmin fair rates of sessions $1, \dots, N$. Consider any positive integer K , and an arbitrary nondecreasing sequence of times $x_1, y_1, \dots, x_K, y_K$. Let $W \geq 3^{L-1}(\varepsilon_1(F) + \sigma)/2$, where $\varepsilon_1(F)$ is defined in (7) to (12). For each node A and for each session i traversing A

$$\sum_{k=1}^K (S_{i,A}(y_k) - S_{i,A}(x_k)) \geq r_i \sum_{k=1}^K (y_k - x_k) - \beta - K\gamma \quad (4)$$

$$\sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \geq r_i \sum_{k=1}^K (y_k - x_k) - \beta - K\gamma \quad (5)$$

$$\sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \leq r_i \sum_{k=1}^K (y_k - x_k) + \beta + K\gamma. \quad (6)$$

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

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{r}_p , the p th

lowest among the maxmin fair rates of different sessions. Let F be the number of distinct ranks $F \leq N$

$$\varsigma_1(1) = 0. \quad (7)$$

$$\varepsilon_1(1) = \eta + 1. \quad (8)$$

$$\varsigma_2(p) = 2^{L-1}\varsigma_1(p) \quad (9)$$

$$\varepsilon_2(p) = 3^{L-1}(\varepsilon_1(p) + \sigma) \quad (10)$$

$$\begin{aligned} \varsigma_3(p) &= 2\sigma + \max(d_{\max}, 2)(\varsigma_2(p) \\ &\quad + \varepsilon_2(p)) + 2LW + \eta - 1 \end{aligned} \quad (11)$$

$$\varepsilon_3(p) = \varepsilon_2(p) \quad (12)$$

$$\varsigma_1(p+1) = (d_{\max} - 1)\varsigma_3(p) \quad (13)$$

$$\varepsilon_1(p+1) = (d_{\max} - 1)\varepsilon_3(p) + \eta + 1. \quad (14)$$

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

Proof of Lemma 2: Lemma 2 follows from (5) and (6) of Lemma 4 with $\varrho = \beta + \gamma$ and $W_0 = 3^{L-1}(\varepsilon_1(F) + \sigma)/2$. \square

APPENDIX PROOF OF LEMMA 3

We first present the intuition behind the proof. The proof is by induction on the number of nodes in a session's path. The sessions with one node form the base case (allowing such sessions simplify the proof). Note that any such session receives a token every time it is sampled and has a packet that has not been released, since no adjacent node 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 nodes, and then prove the lemma for sessions with $p+1$ nodes. Consider a session with $p+1$ nodes and adjacent nodes A and B in its path (Fig. 4). Node B does not prevent the generation of any token at A unless the number of tokens at A is W more than that at B . If the number of tokens at A is W more than that at B , A does not prevent any token generation at B , and the nodes from B to the destination generate tokens as though they constitute a session with fewer nodes, oblivious to the presence of the nodes from A to the source. By induction hypothesis, and from the sampling and packet generation rates, the session receives tokens at rate r or higher at B in these intervals. In all these slots, the number of tokens at A exceed that at B by W . Thus, A 's token generation rate is lower bounded by B 's token generation rate which is at least r . In other slots, B does not prevent the generation of any token at A . Thus, the token generation at the nodes from the source to A resembles that for a session with fewer nodes. Thus, by induction hypothesis and the assumption on the sampling rate, in all slots, A generates tokens at rate r or higher for the session. Note that a session spanning 2 nodes can also be taken as the base case, but then the proof becomes longer. Also, sessions of all path lengths need not be present in the system.

Proof of Lemma 3: We prove by induction on the number of nodes p in a session's path.

First, consider a session i with one node n . Let n not be the source of i . The lemma holds from the assumption on the sampling rate [condition (2)]. Now, let n be the source of i . Let z_l

be the last slot in interval (t_l, w_l) in which i does not have a packet that has not been released. Also, recall that $P_i^{\text{NR}}(t)$ is the number of packets of session i at its source at time t that have not been released.

$$\begin{aligned} C_{i,n}(z_l) - C_{i,n}(t_l) &= P_i(z_l) - P_i(t_l) + P_i^{\text{NR}}(t_l) \\ &\geq P_i(z_l) - P_i(t_l) \\ &\geq \rho_i(z_l - t_l) - \sigma_i \quad [\text{from (1)}] \\ &\geq r(z_l - t_l) - \sigma_i \quad (\text{since } r \leq \rho_i). \end{aligned} \quad (15)$$

Node n generates a token for session i every time it samples i in $(z_l, w_l), \forall l$

$$\begin{aligned} &\sum_{l=1}^K (C_{i,n}(w_l) - C_{i,n}(z_l)) \\ &= \sum_{l=1}^K (S_{i,n}(w_l) - S_{i,n}(z_l)) \\ &\geq r \sum_{l=1}^K (w_l - z_l) - e - Kf \quad [\text{from (2)}]. \end{aligned}$$

$$\begin{aligned} &\sum_{l=1}^K (C_{i,n}(w_l) - C_{i,n}(t_l)) \\ &= \sum_{l=1}^K (C_{i,n}(w_l) - C_{i,n}(z_l)) \\ &\quad + \sum_{l=1}^K (C_{i,n}(z_l) - C_{i,n}(t_l)) \\ &\geq r \sum_{l=1}^K (w_l - t_l) - e - K(f + \sigma_i) \\ &\quad [\text{from (15) and (16)}]. \end{aligned} \quad (16)$$

Thus, (3) holds in the base case.

We assume that (3) holds for all sessions with p or fewer nodes, and prove (3) for an arbitrary session i with $p+1$ nodes. Consider an arbitrary node A in i 's path. If the number of tokens of i at A does not exceed that at nodes adjacent to A in i 's path by W or more in the intervals $(t_l, w_l), l = 1, \dots, K$, then the token generation process for i at A is not affected by back-pressure, and the proof is similar to the base case. Thus, we assume that there exists in i 's path a node B that is adjacent to A , and $C_{i,A}(t) = C_{i,B}(t) + W$ at some time t in these intervals. Let a slot t where $C_{i,A}(t)$ exceeds $C_{i,B}(t)$ by W be a type 1 slot, and a slot t where $C_{i,B}(t)$ exceeds $C_{i,A}(t)$ by W be a type 2 slot; a slot may neither be type 1 nor type 2. Consider each $[t_i, w_i]$ 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 m th "u-slot" ("v-slot") of the l th interval is u_{lm} (v_{lm}) (Fig. 5). Note that

$$C_{i,A}(u_{lm}) = C_{i,B}(u_{lm}) + W \quad \forall l, m \quad (17)$$

$$C_{i,B}(v_{lm}) = C_{i,A}(v_{lm}) + W \quad \forall l, m \quad (18)$$

$$C_{i,A}(t) \leq C_{i,B}(t) + W \quad \forall t. \quad (19)$$

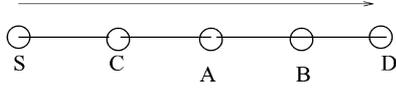


Fig. 4. We show the path of a session between source S and destination D . The A -set consists of nodes S, C, A and the B -set consists of nodes B, D .

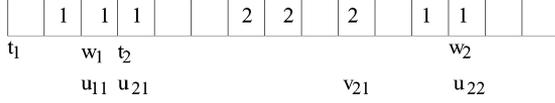


Fig. 5. We show two intervals $[t_1, w_1]$ and $[t_2, w_2]$, and some type 1 and 2 slots. We also show the corresponding u and v slots. Here, $[t_1, u_{11}]$, $[t_2, u_{21}]$, $[v_{21}, u_{22}]$ are example sub-intervals that end in u -slots and start from the nearest v -slot or t_i -slot.

If the link between A and B is removed, then the nodes in i 's path that remain connected to $A(B)$ is the A -set (B -set) (Fig. 4). Let A -set (B -set) consist of $a(b)$ nodes. Note that $a + b = p + 1, a > 0, b > 0$. Thus, $a \leq p, b \leq p$. Consider a sub-interval that ends at a u slot and starts from a t_j (inclusive) or a v -slot (not inclusive), whichever is the nearest to the u -slot (Fig. 5). Let there be J_l such subintervals 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, A does not prevent any session i token generation at B in these subintervals. Hence, in these sub-intervals, the token generation for i in the nodes in the B -set resembles that in the nodes of a session of length b . Condition (2) holds for i in every node in the B -set for every set of subintervals of these I_1 sub-intervals, since any such subinterval is in $[t_l, w_l]$ for some l . Thus, the number of tokens generated for i in these I_1 subintervals in each node of the B -set 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 Fig. 5; the subintervals 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

$$\begin{aligned} & \sum_{l=1}^K \left((C_{i,B}(u_{l1}) - C_{i,B}(t_l)) \right. \\ & \quad \left. + \sum_{m=2}^{J_l} (C_{i,B}(u_{lm}) - C_{i,B}(v_{lm-1})) \right) \\ & \geq r \sum_{l=1}^K \left((u_{l1} - t_l) + \sum_{m=2}^{J_l} (u_{lm} - v_{lm-1}) \right) \\ & \quad - 2^{p-1}e - I_1 3^{p-1}(f + \sigma_i) \end{aligned} \quad (20)$$

$$\begin{aligned} & C_{i,A}(u_{l1}) - C_{i,A}(t_l) \\ & \geq C_{i,B}(u_{l1}) + W - C_{i,B}(t_l) - W \\ & \quad \text{[from (17) and (19)]} \\ & = C_{i,B}(u_{l1}) - C_{i,B}(t_l) \end{aligned} \quad (21)$$

$$\begin{aligned} & C_{i,A}(u_{lm}) - C_{i,A}(v_{lm-1}) \\ & = C_{i,B}(u_{lm}) - C_{i,B}(v_{lm-1}) + 2W \\ & \quad \text{[from (17) and (18)]} \end{aligned} \quad (22)$$

$$\begin{aligned} & \sum_{l=1}^K \left((C_{i,A}(u_{l1}) - C_{i,A}(t_l)) \right. \\ & \quad \left. + \sum_{m=2}^{J_l} (C_{i,A}(u_{lm}) - C_{i,A}(v_{lm-1})) \right) \\ & \geq \sum_{l=1}^K \left((C_{i,B}(u_{l1}) - C_{i,B}(t_l)) \right. \\ & \quad \left. + \sum_{m=2}^{J_l} (C_{i,B}(u_{lm}) - C_{i,B}(v_{lm-1})) \right) + 2W(I_1 - K) \\ & \quad \text{[from (21) and (22)]} \\ & \geq r \sum_{l=1}^K \left((u_{l1} - t_l) \right. \\ & \quad \left. + \sum_{m=2}^{J_l} (u_{lm} - v_{lm-1}) \right) - 2^{p-1}e - K 3^{p-1}(f + \sigma_i) \\ & \quad + (I_1 - K)(2W - 3^{p-1}f - 3^{p-1}\sigma_i) \\ & \quad \text{[from (20)].} \end{aligned} \quad (23)$$

Now, consider the subintervals obtained after removing these I_1 subintervals from $\cup_{l=1}^K (t_l, w_l)$. These new sub-intervals do not contain any type 1 slot. Thus, B does not prevent any session i token generation at A . Hence, the session i token generation in the nodes in the A -set resembles that of a session of length $a, a \leq p$. The number of session i tokens generated at A 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 $v_{l1} > u_{l1} \forall l$

$$\begin{aligned} \text{Thus, } & \sum_{l=1}^K \left((C_{i,A}(w_l) - C_{i,A}(u_{J_l})) \right. \\ & \quad \left. + \sum_{m=1}^{J_l-1} (C_{i,A}(v_{lm}) - C_{i,A}(u_{lm})) \right) \\ & \geq r \sum_{l=1}^K \left((w_l - u_{J_l}) + \sum_{m=1}^{J_l-1} (v_{lm} - u_{lm}) \right) \\ & \quad - 2^{p-1}e - (I_1 - K) 3^{p-1}(f + \sigma_i) \\ & \quad - 2K 3^{p-1}(f + \sigma_i). \end{aligned} \quad (24)$$

Adding (23) and (24)

$$\begin{aligned} & \sum_{l=1}^K (C_{i,A}(w_l) - C_{i,A}(t_l)) \\ & \geq r \sum_{l=1}^K (w_l - t_l) - 2^p e - K 3^p (f + \sigma_i) \\ & \quad + (I_1 - K)(2W - 3^p(f + \sigma_i)). \end{aligned} \quad (25)$$

Note that $p + 1 \leq L_i$ 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 (20). Under this assumption, $I_1 \geq K$. Hence, (3) holds for session i at node A . If there is no type-1 slot in $[t_l, w_l]$ for some l , then the summation in (20) 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 $I_1 \geq K_1$, (3) holds at all nodes in i 's path. \square

APPENDIX
PROOF OF LEMMA 4

We outline the proof for the special case that all sessions are saturated. 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{r}_1 or higher at every node. Now, (5), the lower bound on the token generation rate follows from Lemma 3. Next, we show (6), i.e., the token generation rates are upper bounded by \hat{r}_1 for all sessions with rank 1. This follows because the sampling and, hence, the token generation rate is upper bounded by \hat{r}_1 at the bottleneck node, and due to back-pressure the token generation rates for a session are equal at different nodes 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{r}_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 .

We consider both saturated and unsaturated sessions in the formal proof.

Proof of Lemma 4: We prove the following for ranks $p = 1, \dots, F$, by induction on p .

For each node A , for each session i that traverses A and has rank greater than or equal to p , for any positive integer K , and for any nondecreasing sequence of times $x_1, y_1, \dots, x_K, y_K$

$$\begin{aligned} & \sum_{k=1}^K (S_{i,A}(y_k) - S_{i,A}(x_k)) \\ & \geq \hat{r}_p \sum_{k=1}^K (y_k - x_k) - \varsigma_1(p) - K\varepsilon_1(p). \quad (26) \end{aligned}$$

For each node A , for each session i that traverses A and has rank greater than or equal to p , for any positive integer K , and for any nondecreasing sequence of times $x_1, y_1, \dots, x_K, y_K$

$$\begin{aligned} & \sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \\ & \geq \hat{r}_p \sum_{k=1}^K (y_k - x_k) - \varsigma_2(p) - K\varepsilon_2(p). \quad (27) \end{aligned}$$

If a session i has rank p , and $r_i = \rho_i$

$$P_i^{\text{NR}}(t) \leq \sigma_i + \varsigma_2(p) + \varepsilon_2(p) \quad \forall t. \quad (28)$$

For each node A , for each session i that traverses A and has rank p , for any positive integer K , and for any nondecreasing sequence of times $x_1, y_1, \dots, x_K, y_K$

$$\begin{aligned} & \sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \\ & \leq \hat{r}_p \sum_{k=1}^K (y_k - x_k) + \varsigma_3(p) + K\varepsilon_3(p). \quad (29) \end{aligned}$$

We first prove (26)–(29) for $p = 1$. Note that $\hat{r}_1 = \min(\alpha/d_{\max}, \min_i \rho_i)$. Consider a node A . Let \mathcal{X} be the set of sessions traversing A and \mathcal{R}_A be the set of sessions A relays. Note that $\mathcal{R}_A \subseteq \mathcal{X}$ and $d_A = |\mathcal{X}| + |\mathcal{R}_A|$. From the sampling process, in any interval (x_k, y_k)

$$\begin{aligned} & \sum_{j \in \mathcal{X}} (S_{j,A}(y_k) - S_{j,A}(x_k)) \\ & \quad + \sum_{j \in \mathcal{R}_A} (C_{j,A}(y_k) - C_{j,A}(x_k)) \geq \alpha(y_k - x_k) - \eta. \end{aligned}$$

Now, using $S_{j,A}(y_k) - S_{j,A}(x_k) \geq C_{j,A}(y_k) - C_{j,A}(x_k)$

$$\begin{aligned} & \sum_{j \in \mathcal{X}} (S_{j,A}(y_k) - S_{j,A}(x_k)) \\ & \quad + \sum_{j \in \mathcal{R}_A} (S_{j,A}(y_k) - S_{j,A}(x_k)) \geq \alpha(y_k - x_k) - \eta. \end{aligned}$$

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

$$\begin{aligned} d_A(S_{i,A}(y_k) - S_{i,A}(x_k) + 1) & \geq \alpha(y_k - x_k) - \eta \\ S_{i,A}(y_k) - S_{i,A}(x_k) & \geq \alpha/d_A(y_k - x_k) - \eta - 1. \end{aligned}$$

Thus, every session traversing node A is sampled at least $\alpha/d_A \sum_{k=1}^Q (y_k - x_k) - Q(\eta + 1)$ times for any arbitrary sequence of nondecreasing times $x_1, y_1, \dots, x_Q, y_Q$, and any arbitrary Q . Since $d_A \leq d_{\max}$, $\hat{r}_1 \leq \alpha/d_A$. Thus, (26) holds with $\varsigma_1(1) = 0$ and $\varepsilon_1(1) = \eta + 1$.

Since $\varepsilon_F(1) \geq \varepsilon_1(1)$, $W \geq 3^{L-1}(\varepsilon_1(1) + \sigma)/2$. Hence, (27) 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 (28) for $p = 1$. Consider a session i with rank 1 and $r_i = \rho_i$. Thus, $\hat{r}_1 = \rho_i$. Let n be the source of i

$$\begin{aligned} P_i^{\text{NR}}(t) & = P_i(t) - C_{i,n}(t) \\ & \leq (\rho_i - \hat{r}_1)t + \sigma_i + \varsigma_2(1) + \varepsilon_2(1) \\ & \quad \text{(from (1) and (27) for } p = 1) \\ & = \sigma + \varsigma_2(1) + \varepsilon_2(1) \quad \text{(since } \hat{r}_1 = \rho_i \text{ and } \sigma_i \leq \sigma). \end{aligned}$$

Thus, (28) follows for $p = 1$.

Now, we prove (29) for $p = 1$. Consider a session i with rank 1. Let A be a node in i 's path. Consider a sequence of nondecreasing times $x_1, y_1, \dots, x_K, y_K$

$$\begin{aligned} & \sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \\ & = C_{i,A}(y_K) - C_{i,A}(x_1) - \sum_{k=1}^{K-1} (C_{i,A}(x_{k+1}) - C_{i,A}(y_k)) \\ & \leq C_{i,A}(y_K) - C_{i,A}(x_1) - \hat{r}_1 \sum_{k=1}^{K-1} (x_{k+1} - y_k) \\ & \quad + \varsigma_2(1) + (K-1)\varepsilon_2(1). \quad (30) \end{aligned}$$

The last step follows from (27) for $p = 1$. First, let $\hat{r}_1 < \rho_i$. Thus, i has a bottleneck node B . Let \mathcal{X} be the set of sessions traversing node B . Since i has rank 1, $d_B = d_{\max}$, rank(j) = 1, $\forall j \in \mathcal{X}$, and $\hat{r}_1 = \alpha/d_{\max}$. Recall that $S_B \cup \mathcal{D}_B$ is the set of sessions that originate or terminate at B and \mathcal{R}_B is the set of

sessions that B relay. Let $\mathcal{X}_1 = \mathcal{S}_B \cup \mathcal{D}_B$. First, let B be i 's source or destination.

$$\begin{aligned}
& C_{i,B}(y_K) - C_{i,B}(x_1) \\
& \leq \alpha(y_K - x_1) - \sum_{m \in \mathcal{X}_1 \setminus \{i\}} (C_{m,B}(y_K) - C_{m,B}(x_1)) \\
& \quad + 2 \sum_{m \in \mathcal{R}_B} (C_{m,B}(y_K) - C_{m,B}(x_1)) + \eta - 1 \\
& \leq \alpha(y_K - x_1) - (d_{\max} - 1)(\hat{r}_1(y_K - x_1) - \varsigma_2(1) \\
& \quad - \varepsilon_2(1)) + \eta - 1 \\
& \quad \text{(from (27) since rank}(j) = 1, \forall j \in \mathcal{X}) \\
& = \hat{r}_1(y_K - x_1) + (d_{\max} - 1)(\varsigma_2(1) + \varepsilon_2(1)) + \eta - 1 \\
& \quad \text{(since } \hat{r}_1 = \alpha/d_{\max}). \quad (31)
\end{aligned}$$

Now, let B relay i

$$\begin{aligned}
& 2(C_{i,B}(y_K) - C_{i,B}(x_1)) \\
& \leq \alpha(y_K - x_1) - 2 \sum_{m \in \mathcal{R}_B \setminus \{i\}} (C_{m,B}(y_K) - C_{m,B}(x_1)) \\
& \quad + \sum_{m \in \mathcal{X}_1} (C_{m,B}(y_K) - C_{m,B}(x_1)) + \eta - 1 \\
& \leq \alpha(y_K - x_1) - (d_{\max} - 2)(\hat{r}_1(y_K - x_1) \\
& \quad - \varsigma_2(1) - \varepsilon_2(1)) + \eta - 1 \\
& \quad \text{(from (27) since rank}(j) = 1, \forall j \in \mathcal{X}) \\
& = 2\hat{r}_1(y_K - x_1) + (d_{\max} - 2)(\varsigma_2(1) + \varepsilon_2(1)) + \eta - 1 \\
& \quad \text{(since } \hat{r}_1 = \alpha/d_{\max})
\end{aligned}$$

$$\begin{aligned}
& C_{i,B}(y_K) - C_{i,B}(x_1) \\
& \leq \hat{r}_1(y_K - x_1) + (d_{\max} - 2)(\varsigma_2(1) \\
& \quad + \varepsilon_2(1))/2 + (\eta - 1)/2. \quad (32)
\end{aligned}$$

Now, let $\hat{r}_1 = \rho_i$. Let B denote the source of i

$$\begin{aligned}
& C_{i,B}(y_K) - C_{i,B}(x_1) \\
& \leq P_i^{\text{NR}}(x_1) + P_i(y_K) - P_i(x_1) \\
& \leq \sigma_i + \varsigma_2(1) + \varepsilon_2(1) + \rho_i(y_K - x_1) + \sigma_i \\
& \quad \text{[from (28) and (1)]} \\
& = \hat{r}_1(y_K - x_1) + 2\sigma_i + \varsigma_2(1) + \varepsilon_2(1) \quad \text{(since } \hat{r}_1 = \rho_i). \quad (33)
\end{aligned}$$

From (31)–(33), there exists a node B in i 's path such that

$$\begin{aligned}
& C_{i,B}(y_K) - C_{i,B}(x_1) \\
& \leq \hat{r}_1(y_K - x_1) + 2\sigma_i \\
& \quad + \max(d_{\max} - 1, 1)(\varsigma_2(1) + \varepsilon_2(1)) + \eta - 1 \quad (34)
\end{aligned}$$

$$\begin{aligned}
& \text{Now, } |C_{i,A}(t) - C_{i,B}(t)| \\
& \leq L_i W \forall t \quad (35)
\end{aligned}$$

$$\begin{aligned}
& C_{i,A}(y_K) - C_{i,A}(x_1) \\
& \leq C_{i,B}(y_K) - C_{i,B}(x_1) + 2L_i W \quad \text{[from (35)]} \\
& \leq \hat{r}_1(y_K - x_1) + \max(d_{\max} - 1, 1) \\
& \quad \times (\varsigma_2(1) + \varepsilon_2(1)) \\
& \quad + 2L_i W + 2\sigma_i + \eta - 1 \quad \text{[from (34)].} \quad (36)
\end{aligned}$$

From (30) and (36)

$$\begin{aligned}
& \sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \\
& \leq \hat{r}_1 \sum_{k=1}^K (y_k - x_k) + \max(d_{\max}, 2) \\
& \quad \times (\varsigma_2(1) + \varepsilon_2(1)) + 2L_i W + 2\sigma_i \\
& \quad + \eta - 1 + K\varepsilon_2(1). \quad (37)
\end{aligned}$$

Thus, for $p = 1$, (29) follows from (37) with $\varsigma_3(1) = \max(d_{\max}, 2)(\varsigma_2(1) + \varepsilon_2(1)) + 2LW + 2\sigma + \eta - 1$ and $\varepsilon_3(1) = \varepsilon_2(1)$.

Now, we assume (26)–(29) for $1, \dots, p$, and show that (26)–(29) hold for $p + 1$.

We first prove (26). Consider a session i with rank greater than or equal to $p + 1$. Consider a node A in i 's path. Let $\mathcal{Y} = \{w : w \text{ traverses } A, \text{rank}(w) \leq p\}$, $\mathcal{Z} = \{w : w \text{ traverses } A, \text{rank}(w) \geq p + 1\}$, $\mathcal{Y}_1 = \{w : w \in \mathcal{Y}, A \text{ relays } w\}$ and $\mathcal{Z}_1 = \{w : w \in \mathcal{Z}, A \text{ relays } w\}$. Now, $\mathcal{Y}_1 \subseteq \mathcal{Y}$, $\mathcal{Z}_1 \subseteq \mathcal{Z}$ and \mathcal{Z} includes i . From the sampling process, in any interval (x_k, y_k)

$$\begin{aligned}
& \sum_{j \in \mathcal{Z}} (S_{j,A}(y_k) - S_{j,A}(x_k)) + \sum_{j \in \mathcal{Z}_1} (C_{j,A}(y_k) - C_{j,A}(x_k)) \\
& \quad + \sum_{j \in \mathcal{Y}} (C_{j,A}(y_k) - C_{j,A}(x_k)) + \sum_{j \in \mathcal{Y}_1} (C_{j,A}(y_k) - C_{j,A}(x_k)) \\
& \geq \alpha(y_k - x_k) - \eta.
\end{aligned}$$

Now, using $S_{j,A}(y_k) - S_{j,A}(x_k) \geq C_{j,A}(y_k) - C_{j,A}(x_k)$ for each j

$$\begin{aligned}
& \sum_{j \in \mathcal{Z}} (S_{j,A}(y_k) - S_{j,A}(x_k)) + \sum_{j \in \mathcal{Z}_1} (S_{j,A}(y_k) - S_{j,A}(x_k)) \\
& \geq \alpha(y_k - x_k) - \sum_{j \in \mathcal{Y}} (C_{j,A}(y_k) - C_{j,A}(x_k)) \\
& \quad - \sum_{j \in \mathcal{Y}_1} (C_{j,A}(y_k) - C_{j,A}(x_k)) - \eta.
\end{aligned}$$

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

$$\begin{aligned}
& (|\mathcal{Z}| + |\mathcal{Z}_1|)(S_{i,A}(y_k) - S_{i,A}(x_k) + 1) \\
& \geq \alpha(y_k - x_k) - \sum_{j \in \mathcal{Y}} (C_{j,A}(y_k) - C_{j,A}(x_k)) \\
& \quad - \sum_{j \in \mathcal{Y}_1} (C_{j,A}(y_k) - C_{j,A}(x_k)) - \eta.
\end{aligned}$$

Thus

$$\begin{aligned}
& \sum_{k=1}^K (S_{i,A}(y_k) - S_{i,A}(x_k)) \\
& \geq \frac{1}{|\mathcal{Z}| + |\mathcal{Z}_1|} \left(\alpha \sum_{k=1}^K (y_k - x_k) - K|\mathcal{Z}| - K|\mathcal{Z}_1| \right) \\
& \quad - K\eta - \sum_{j \in \mathcal{Y}} \sum_{k=1}^K (C_{j,A}(y_k) - C_{j,A}(x_k))
\end{aligned}$$

From (43), like in the base case

$$C_{i,A}(y_K) - C_{i,A}(x_1) \leq \hat{r}_{p+1}(y_K - x_1) + 2\sigma_i + \max(d_{\max} - 1, 1)(\varsigma_2(p+1) + \varepsilon_2(p+1)) + 2L_iW + \eta - 1. \quad (44)$$

From (39) and (44)

$$\begin{aligned} & \sum_{k=1}^K (C_{i,A}(y_k) - C_{i,A}(x_k)) \\ & \leq \hat{r}_{p+1} \sum_{k=1}^K (y_k - x_k) + \max(d_{\max}, 2)(\varsigma_2(p+1) + \varepsilon_2(p+1)) + 2L_iW + 2\sigma_i + \eta - 1 + K\varepsilon_2(p+1). \end{aligned} \quad (45)$$

Thus, (29) follows from (45) with $\varsigma_3(p+1) = \max(d_{\max}, 2)(\varsigma_2(p+1) + \varepsilon_2(p+1)) + 2LW + 2\sigma + \eta - 1$ and $\varepsilon_3(p+1) = \varepsilon_2(p+1)$. Thus, (26)–(29) hold in the induction case.

Note that $\varsigma_i(x), \varepsilon_i(x)$ are increasing in both i and x . Thus, from (26), (27), and (29), Lemma 4 holds with $\beta = \varsigma_3(F)$ and $\gamma = \varepsilon_3(F)$. \square

REFERENCES

- [1] D. J. Baker and A. Ephremides, "The architectural organization of a packet radio network via a distributed algorithm," *IEEE Trans. Commun.*, vol. COM-29, no. 11, pp. 1694–1701, Nov. 1981.
- [2] D. Bertsekas and R. Gallager, *Data Networks*. Upper Saddle River, NJ: Prentice-Hall, 1987.
- [3] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005.
- [4] E. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, pp. 1024–1039, Sep. 1991.
- [5] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 910–917, Sep. 1988.
- [6] X. L. Huang and B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation," in *Proc. IEEE/ACM MobiHoc*, Long Beach, CA, Oct. 2001, pp. 221–231.
- [7] N. Johansson, U. Korner, and L. Tassiulas, "A distributed scheduling algorithm for a bluetooth scatternet," in *Proc. 17th Int. Teletraffic Congr.*, Salvador da Bahia, Brazil, Dec. 2001, pp. 61–72.
- [8] J. Ju and V. O. K. Li, "An optimal topology transparent scheduling method in multihop packet radio networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 298–306, Jun. 1998.
- [9] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," in *Proc. INFOCOM*, Miami, FL, Mar. 2005.
- [10] B. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*. Upper Saddle River, NJ: Prentice-Hall, 2000.
- [11] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving mac layer fairness in wireless packet networks," in *Proc. ACM Mobicom*, Boston, MA, Aug. 2000, pp. 87–98.
- [12] M. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. INFOCOM*, Miami, FL, Mar. 2005.
- [13] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control—the single node case," in *IEEE/ACM Trans. Netw.*, vol. 1, Jun. 1993, pp. 344–357.
- [14] M. Post, P. Sarachik, and A. Kershenbaum, "A biased greedy algorithm for scheduling multihop radio networks," in *Proc. 19th Annu. Conf. Information Sciences and Systems*, Johns Hopkins University, Mar. 1985.
- [15] A. Racz, G. Miklos, F. Kubinszky, and A. Valko, "A pseudo random coordinated scheduling algorithm for bluetooth scatternets," in *Proc. IEEE/ACM MobiHoc*, Long Beach, CA, Oct. 2001, pp. 193–203.
- [16] B. Radunovic and J. Le Boudec, "Rate performance objectives of multi-hop wireless networks," in *Proc. INFOCOM*, Hong Kong, China, Mar. 2004.
- [17] S. Sarkar and K. Sivarajan, "Fairness in wireless mobile networks," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2412–2426, Aug. 2002.
- [18] S. Sarkar and L. Tassiulas, "Back pressure based multicast scheduling for fair bandwidth allocation," in *Proc. INFOCOM*, Anchorage, AK, May 2001, pp. 1123–1133.
- [19] —, "Fair bandwidth allocation for multicasting in networks with discrete feasible set," *IEEE Trans. Comput.*, vol. 53, no. 7, pp. 785–797, Jul. 2004.
- [20] L. Tassiulas, "Adaptive back-pressure congestion control based on local information," *IEEE Trans. Autom. Control*, vol. 40, no. 2, pp. 236–250, Feb. 1995.
- [21] L. Tassiulas and A. Ephremidis, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 1, pp. 165–168, Jan. 1992.
- [22] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," *Proc. INFOCOM*, pp. 763–772, Jun. 2002.
- [23] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," in *Proc. INFOCOM*, Hong Kong, China, Mar. 2004.

Saswati Sarkar (S'98–M'00) received the M.Eng. in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1996, and the Ph.D. degree in electrical and computer engineering from University of Maryland, College Park, in 2000.

She is currently an Assistant Professor in the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia. Her research interests are in resource allocation and performance analysis in communication networks.

Dr. Sarkar received the Motorola Gold Medal for the Best Masters Student in the Division of Electrical Sciences at the Indian Institute of Science and a National Science Foundation (NSF) Faculty Early Career Development Award in 2003. She has been an Associate Editor for the *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS* since 2001 and for the *Journal of Computer Networks* since 2004.

Leandros Tassiulas (S'89–M'91) was born in Katerini, Greece, in 1965. He received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1989 and 1991, respectively.

He has been a Professor in the Department of Computer and Telecommunications Engineering, the University of Thessaly, Greece, and a Research Professor in the Department of Electrical and Computer Engineering and the Institute for Systems Research, the University of Maryland, College Park, since 2001. He has held positions as Assistant Professor at Polytechnic University, Brooklyn, NY (1991–1995), Assistant and Associate Professor at the University of Maryland, College Park (1995–2001), and Professor at the University of Ioannina, Greece (1999–2001). His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols of wireless systems, sensor networks, high-speed internet, and satellite communications.

Dr. Tassiulas received a National Science Foundation (NSF) Research Initiation Award in 1992, an NSF CAREER Award in 1995, an Office of Naval Research Young Investigator Award in 1997, a Bodosaki Foundation Award in 1999, and the INFOCOM'94 Best Paper Award.