Maxmin Fair Scheduling in Wireless Adhoc Networks

Leandros Tassiulas and Saswati Sarkar

Abstract— **We investigate from an algorithmic perspective the maxmin fair allocation of bandwidth in wireless adhoc networks. We formalize the maxmin fair objective under wireless scheduling constraints, and present a necessary and sufficient condition for maxmin fairness of a bandwidth allocation. We propose an algorithm that assigns weights to the sessions dynamically such that the weights depend on the congestion in the neighborhood, and schedules the sessions that constitute a maximum weighted matching. We prove that this algorithm attains the maxmin fair rates, even though it does not use any information about the statistics of the packet arrival process.**

I. Introduction

We investigate maxmin fair allocation of bandwidth in wireless adhoc networks. A bandwidth allocation is maxmin fair, if it is not possible to increase the bandwidth of a user without reducing that of another user that has a lower bandwidth. 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.

In wireless networks, all users within a certain distance from each other contend for using the transmission medium, even if they use different links. Thus, the schedulings of different links are interdependent. Due to the scheduling dependences, the algorithms that intuitively seem to be fair, do not yield a maxmin fair allocation (Section II). This motivates an investigation of the problem of maxmin fair allocation from an algorithmic perspective. Our contribution is to design a scheduling algorithm that attains a provably maxmin fair bandwidth allocation. The algorithm is expected to provide a basis for developing protocols for fair medium access in different technologies like Bluetooth, IEEE 802.11, etc.; the design of technology specific protocols is however beyond the scope of this paper.

In Section II, we formulate the problem of allocating maxmin fair bandwidth in presence of wireless specific scheduling constraints, and present necessary and sufficient conditions for maxmin fairness of a bandwidth allocation. In Section III, we present a MAC scheduling algorithm for attaining a maxmin fair bandwidth allocation, and describe its analytical performance guarantees. This algorithm maintains estimates of the maxmin fair bandwidth of the sessions at the nodes, and decides the scheduling of the sessions based on the estimates. The estimates at each node are updated based on the arrival of the packets at the node and the congestion at the neighboring nodes. 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. No new computation is necessary when the topology or the arrival rates change. The scheme is therefore robust. In Section IV, we describe different features of the algorithm that are pertinent to the algorithm's implementation in networks, and show that this algorithm attains maxmin fair service rate allocation in input queued switches as well. We present the proofs, which are the paper's important contributions, in the appendix.

We now briefly describe the related work. Fair allocation of bandwidth has been extensively investigated in wireline networks [4], [7], etc. The resource allocation constraints in wireline networks differ from those in wireless networks, e.g., in wireline networks, different links can be independently activated. But, using a different design paradigm, we obtain a MAC scheduling that has the same effect in adhoc networks as Fair Queueing algorithms, e.g., PGPS [14], [3] have in wireline networks. Existing MAC protocols like IEEE 802.11 [13], MACAW [1], CB-FAIR [12] do not provide any fairness guarantees. Huang *et al.* [8] and Fang *et al.* [5] propose the following two-step procedure for attaining the maxmin fair rates in adhoc networks: (a) compute the maxmin fair rates, and subsequently (b) schedule the packets so as to attain the computed rates. These schemes have two main disadvantages. They require re-computation when the packet arrival rates or the topology change. Also, neither is guaranteed to attain the maxmin fair rates. The rate computation algorithm presented by Huang *et al.* [8] generates infeasible rates in several topologies. The scheduling strategy proposed by Fang *et al.* [5] is not guaranteed to attain the computed rates. Nandagopal *et al.* [11] present a heuristic scheduling strategy for attaining proportionally fair bandwidth allocation in wireless networks. Luo *et al.* [10] present a heuristic scheduling strategy to attain an alternate notion of fairness where every session is guaranteed a minimum bandwidth that depends on the predetermined weight of the session.

II. Network model and Fairness Objective

We consider scheduling at the MAC layer of a wireless adhoc network with V nodes, E links and N sessions. At the MAC layer, each session traverses only one link. We introduce the notion of a topology graph. The topology

Parts of this paper have been presented at INFOCOM 2002, New York.

L. Tassiulas is with the Computer Engineering and Telecommunications department in the University of Thessaly, Volos, Greece, and the Department of Electrical and Computer Engineering, University of Maryland, College Park. Email: leandros@inf.uth.gr

S. Sarkar is with the department of Electrical and Systems Engineering in the University of Pennsylvania, Philadelphia, PA, USA. Email: swati@ee.upenn.edu. Her work has been supported by NSF grants ANI01-06984 and NCR02-38340.

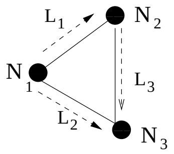

Fig. 1. Session i traverses link i. Bandwidth allocation $(1/2, 1/2, 1/2)$ satisfies the inequalities $r_1 + r_2 \leq 1$, $r_2 + r_3 \leq 1$ and $r_1 + r_3 \leq 1$, where r_i is session i's bandwidth. But, $(1/2, 1/2, 1/2)$ is not a feasible bandwidth allocation. This is because in this network, only one session can transmit a packet in a slot, and hence if all sessions get equal bandwidth, a session's bandwidth can at most be 1/3.

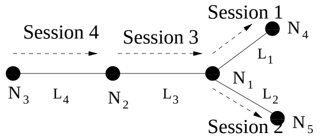

Fig. 2. An example wireless network

graph is an undirected graph where a vertex corresponds to a node in the network, and an edge between two vertices represents a session between the corresponding nodes.

Time is slotted. Every packet has length 1 slot. Every node has one radio. Thus, in a slot, a node can either transmit one packet, or receive one packet, or remain idle. Every node has a locally unique frequency or a transmission code. Thus, transmissions that do not have a common node can simultaneously proceed without any interference. Hence, the sessions that transmit packets in any slot must constitute a matching[1] in the topology graph. For example, a bluetooth network satisfies the above assumptions [6]. The first step is to design an algorithm that attains maxmin fairness in presence of the wireless specific scheduling constraints. Another important distinction between wireless and wireline networks is the presence of location dependent, bursty channel errors in the former. We do not consider channel errors; nevertheless, our rigorous analytical results provide the first step in solving the challenging problem of attaining maxmin fairness in wireless networks, and should facilitate the design of algorithms for the general case with channel errors.

Definition 1: An $N-$dimensional vector of nonnegative real numbers $(r_1, \ldots, r_N)$ is a *feasible bandwidth allocation* if the sessions can be scheduled for packet transmission such that session i attains bandwidth r_i, for each i.

We first formulate the conditions for feasibility of bandwidth allocations. First, assume that a session always has a packet to transmit. Since a node can be involved in a single transmission in a slot, a necessary condition for feasibility of a bandwidth allocation is that the sum of the bandwidth of all sessions traversing a node must be less than or equal to 1. This condition is sufficient for feasibility if the topology graph is bipartite[2] [9], but not otherwise (Figure 1).

For non-bipartite topology graphs, the following condition is both necessary and sufficient for feasibility of any bandwidth allocation [9]. For any arbitrary subset of nodes, Q, with odd cardinality (i.e., odd $|Q|$), the sum of the bandwidth of all sessions that have both source and destination in Q is upper bounded by $(|Q|-1)/2$[9]. Determination of whether a bandwidth allocation satisfies the above condition is computationally complex, as there are exponential number of odd subsets of nodes. A computationally simple sufficient condition for feasibility of a bandwidth allocation in any topology graph is that the sum of the bandwidth of all sessions traversing a node is less than 2/3 [9]. Note that this condition is not necessary, but only sufficient. (Refer to [9] for other sufficent conditions.) No practical bandwidth allocation scheme however utilizes the entire capacity of a link or a node; full utilization causes huge queuing delay and buffer overflow during transients. We thus assume that a bandwidth allocation $(r_1, \ldots, r_N)$ is feasible if

$$\sum_{\substack{i:\ \text{session } i \\ \text{traverses node } j}} r_i \leq \alpha \;\forall \text{ node } j, \quad \text{(node capacity constraint)} \tag{1}$$

where α, the *node utilization factor,* depends on the desired bandwidth utilization and whether the topology graph is bipartite. For bipartite topology graphs, $\alpha \leq 1$, and for non-bipartite topology graphs, $\alpha \leq 2/3$.

A session may not always have a packet to transmit. Let ρ_i be session i's packet arrival rate. Then,

$$r_i \leq \rho_i \quad \forall \text{ session } i, \qquad \text{(demand constraint).} \tag{2}$$

An $N-$dimensional vector of nonnegative real numbers $(r_1, \ldots, r_N)$ is a *feasible bandwidth allocation* if and only if it satisfies the node capacity ((1)) and demand ((2)) constraints.

For example, in Figure 2 a bandwidth allocation (r_1, r_2, r_3, r_4) is feasible if and only if it satisfies the following constraints: $r_1 + r_2 + r_3 \leq 1$, $r_3 + r_4 \leq 1$, $r_4 \leq 1$, and $0 \leq r_i \leq \rho_i$, $\forall i$. Here, $\alpha = 1$ as the topology-graph is bipartite.

Definition 2: A feasible bandwidth allocation $\vec{r}^1$ is *maxmin fair* if it satisfies the following property with respect to any other feasible bandwidth allocation $\vec{r}^2$: if there exists i such that the ith component of $\vec{r}^2$ is strictly greater than that of $\vec{r}^1$ $(r_i^2 > r_i^1)$, then there exists j such that the jth component of $\vec{r}^1$, r_j^1 is less than or equal to the ith component of $\vec{r}^1$, r_i^1 $(r_j^1 \leq r_i^1)$ and the jth component of $\vec{r}^2$ (r_j^2) is strictly less than the jth component of $\vec{r}^1$ $(r_j^2 < r_j^1)$.

Consider the algorithm that schedules the matching that consists of the largest number of sessions among those that have received the least service so far, and subject to this the largest number of sessions among those that have received the second minimum service so far, and so on. The algorithm gives absolute priority to sessions that have received the least service, and therefore should intuitively be maxmin fair. The counter example in Figure 3 however shows that this intuition is misleading, even when ses-

[1] Matching is a set of edges such that no two of them have any common vertex.

[2] A graph is bipartite if the set of vertices can be partitioned in two subsets such that there is no edge between vertices in the same subset.

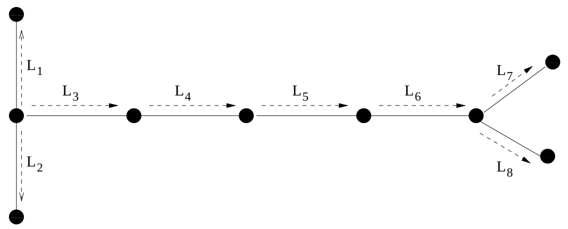

Fig. 3. Every link has a single session. Every session has packets for transmission in every slot. Initially, none of the sessions have transmitted any packet. Thus the sessions that constitute a maximum size matching, e.g., $\{L_1, L_4, L_8\}$ are scheduled for service. In slot 2, the links $\{L_2, L_5, L_7\}$ that constitute a maximum size matching among the sessions that have not transmitted any packet are scheduled for service. In slot 3, the links $\{L_3, L_6\}$ that constitute a maximum size matching among the sessions that have not transmitted any packet are scheduled for service. Note that no other session can transmit simultaneously with those in $\{L_2, L_5, L_7\}$ and $\{L_3, L_6\}$. Now all sessions have transmitted one packet each. Hence, repeating the same procedure, the same sequence will be selected successively. Thus, every session transmits once in every 3 slots, and receives 1/3 units of bandwidth. But, an algorithm that schedules sessions in L_1, L_4, L_6 in one slot, sessions in L_2, L_5, L_7 in the second slot, and sessions in L_3, L_5, L_8 in the third slot, allocates bandwidth 2/3 to the session in link L_5 and 1/3 to each of the other sessions. Hence the bandwidth allocation of 1/3 to each session is not maxmin fair. The maxmin fair allocation is $(1/3, 1/3, 1/3, 1/2, 1/2, 1/3, 1/3, 1/3)$.

sions always have packets for transmission and the topology graph is bipartite. This indicates that innovative techniques will be required for designing an algorithm that attains the maxmin fair rates in presence of the wireless specific scheduling constraints.

We first present a necessary and sufficient condition for maxmin fairness of a bandwidth allocation in wireless networks, which is computationally simple to evaluate.

Definition 3: A session i is *bottlenecked* at a node n it traverses, if the sum of the bandwidth of all sessions traversing n equals α, and i has the maximum bandwidth among all sessions traversing n.

Again, in Figure 2 for a bandwidth allocation of $(1/3, 1/3, 1/3, 2/3)$ sessions 3 and 4 are bottlenecked at N_1 and N_2 respectively. However, N_3 is not a bottleneck node for any session in Figure 2 as the only session (session 4) traversing N_3 has bandwidth less than the node-utilization 1.

Theorem 1: A feasible bandwidth allocation is maxmin fair if and only if every session satisfies at least one of the following conditions: (a) the session has at least one bottleneck node, (b) the session's bandwidth equals its packet arrival rate.

There exists a similar necessary and sufficient condition for maxmin fairness in wireline networks [2].

III. A maxmin fair scheduling algorithm

We first describe the algorithm for the special case that every session always has a packet to transmit, and subsequently present the intuition behind the design. We next motivate the changes required for the general case when all sessions may not always have packets to transmit. In Figure 4, we describe the algorithm in this general case. In Table III, we illustrate the algorithm using the network in Figure 2. At the end of the section, we present the performance guarantees.

We now describe the algorithm for the special case that every session always has a packet to transmit.

1. Each node allocates service tokens to the sessions traversing the node in a *round-robin-like* fashion. We later describe the token allocation procedure.
2. Weight of an edge in the topology graph in a slot is the minimum of the number of tokens of the corresponding session at the session's source and destination. Recall that each session corresponds to an edge in the topology graph.
3. In each slot, the sessions that constitute a maximum weighted matching[3] in the topology graph are scheduled for service.
4. Whenever a session is served, a token is removed from both its source and destination.

Now we describe the service token allocation procedure.

1. In slot t, a session i with end nodes m, n, is *eligible* to receive a service token at node m if the number of tokens for i at m does not exceed that at n by W or more, where W is a parameter of the algorithm.
2. Each node n allocates a service token to the first eligible session in a round-robin order.

Note that a session may be eligible for receiving tokens at one of its end nodes and in-eligible at the other.

We will show that a session receives tokens at each of its end nodes at a rate that equals the session's maxmin fair rate. Note that the weight of an edge in the topology graph is the difference between the number of tokens generated for the corresponding session at one end node of the session and the number of packets transmitted for the session. Thus, the weight is a measure of how much the service received by the session lags behind the session's maxmin fair rate. A maximum weighted matching gives priority to sessions that correspond to edges with large weights, but not absolute priority though. For example, if scheduling a session i with the maximum weight prevents the scheduling of two sessions j, k with lower weights such that the total weight of j, k exceed i's weight, then j and k are preferred to i. The scheduling thus reduces the difference between the sums of the service rates received by all sessions and the maxmin fair rates of the sessions; this attains the maxmin fair rates.

We now explain why each session receives tokens at each of its end nodes at a rate that equals the session's maxmin fair rate. Since the objective is to attain the maxmin fair allocation, each node A must first try to allocate equal bandwidth to all sessions. If a session can not utilize the allocated share at A on account of constraints at the other node, then A must distribute the residual bandwidth among the un-congested sessions. Tokens are generated according to a similar principle. A node generates tokens to sessions in a round robin manner, excluding only the sessions that have a small number of tokens at the other

[3]Weight of a matching is the sum of the weights of the edges included in the matching. A maximum weighted matching is the matching with the maximum weight.

node. If a session has a small number of tokens at a node, then the node is congested; thus, the session's transmission opportunities at its other end node should be distributed among other sessions.

Consider Figure 2. Let all sessions always have packets to transmit. The maxmin fair rates of sessions $1, 2, 3, 4$ are $1/3, 1/3, 1/3, 2/3$ respectively. As Table I(A) shows, N_1 samples sessions $1, 2, 3$ at rate $1/3$ each. Initially node N_2 generates tokens to sessions 3 and 4 at rate $1/2$ each. But, from slot 15 onwards it can not generate tokens to session 3 several times it samples, because session 3 receives tokens from node N_1 at the rate of only $1/3$, and the number of tokens of session 3 accumulated at N_1 and N_2 can not differ by more than the constant W. So, N_2 generates tokens for session 4 in these residual slots. Node N_3 samples session 4 every slot, and generates tokens to session 4 whenever the number of tokens of session 4 at N_2 is greater than that at N_3 minus W. So, sessions 3 and 4 receive tokens at rates $1/3$ and $2/3$ respectively from both ends. Note that these are the maxmin fair rates of these sessions. Similarly, sessions 1 and 2 also receive tokens at their maxmin fair rates from both end points.

The token generation for a session at the session's source must be modified when sessions do not always have packets to transmit. The rest of the algorithm remains the same. When a session's packet is generated at its source, the packet is considered "unmatched." A session receives a token in round robin order at its source if it has an unmatched packet, and if its number of tokens at its destination is not too few. When a session receives a token at its source, one unmatched packet becomes matched. The modification ensures that a session that generates packets at a low rate, receives tokens at a low rate as well, and thus few transmission opportunities. Thus, the residual bandwidth can be distributed among sessions that generate packets at high rates.

Let the arrival rate for session 3 in Figure 2 be $1/6$. The other sessions always have packets to transmit. The maxmin fair allocation is $(5/12, 5/12, 1/6, 5/6)$. As Table I(B) shows, session 3's source N_2 samples session 3 at the rate $1/2$, but can generate tokens at the rate $1/6$ only, because it does not find sufficient unmatched packets. The number of tokens accumulated at 3's destination N_1 can be at most W more than that at N_2. Hence, although N_1 samples session 3 at the rate $1/3$, it generates tokens at rate $1/6$ only. Thus session 3 receives tokens at the rate of $1/6$ from both end-points. Both N_1 and N_2 generate tokens to other sessions in the residual slots. For example, N_1 generates tokens at the rate $5/12$ to sessions 1 and 2 each, and N_2 generates tokens at the rate $5/6$ to session 4.

Note that the number of packets waiting for transmission for a session is greater than or equal to the number of tokens of the session at the session's source. Thus, when a session does not have a packet to transmit, the weight of the corresponding edge in the topology graph is 0, and therefore the session is not scheduled for service.

We present performance guarantees that hold for the following arrival process. Number of packets generated for session i in any interval of length t differs from $\rho_i t$ by at most σ_i, where ρ_i is i's packet arrival rate and σ_i is i's burstiness. A session i always has a packet to transmit if $\rho_i > 1$ and $\sigma_i = 0$. Assume that a session's source node has infinite buffer.

Theorem 2: Let $(r_1, \ldots, r_N)$ be the maxmin fair bandwidth allocation. If $W > \gamma_N$, then the number of tokens generated at each end node of a session i in any interval of length t differs from $r_i t$ by at most a constant ν. Here, constants γ_N and ν do not depend on t.

Theorem 3: Let $(r_1, \ldots, r_N)$ be the maxmin fair bandwidth allocation. If $W > \gamma_N$, the number of packets served in any interval of length t for any session i differs from $r_i t$ by at most a constant κ. Here, constants γ_N and κ do not depend on t.

The constants γ_N, ν and κ depend on the topology and traffic parameters. We formulate γ_N, ν in appendix -B and κ in appendix -C respectively.

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, γ_N, needed to guarantee the convergence results in Theorems 2 and 3, can be computed only with explicit knowledge of the network topology. This motivates an 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 16 nodes and 14 sessions that is shown in Figure 5. 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 [17]. 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 $|1 - \frac{C_{i,n}(t)}{r_i t}|$. We plot the maximum and average relative errors over all sessions as a function of t in Figure 5.

We observe the following from Figure 5. The average relative error decays fast, e.g., it is less than 0.05 within 100 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 γ_N for guaranteed convergence exceeds 13^{28}! 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.

IV. Discussion and Conclusion

Our contribution is primarily theoretical since the scheduling policy we propose is the first, and till date the

(A)

T	Sessions sampled by					Sessions receiving tokens from					Number of tokens of session								Sessions served
	N_1	N_2	N_3	N_4	N_5	N_1	N_2	N_3	N_4	N_5	1 at		2 at		3 at		4 at		
											N_1	N_4	N_1	N_5	N_1	N_2	N_2	N_3	
1	1	3	4	1	2	1	3	4	1	2	1	1	0	1	0	1	0	1	1
2	2	4	4	1	2	2	4	4	1	2	0	1	1	2	0	1	1	2	2,4
3	3	3	4	1	2	3	3	4	1	2	0	2	0	2	1	2	0	2	3
4	1	4	4	1	2	1	4	4	1	2	1	3	0	3	0	1	1	3	1,4
5	2	3	4	1	2	2	3	4	1	-	0	3	1	3	0	2	0	3	2
6	3	4	4	1	2	3	4	-	-	2	0	3	0	3	1	2	1	3	3
7	1	3	4	1	2	1	3	4	-	-	1	3	0	3	0	2	1	4	1,4
8	2	4	4	1	2	2	4	-	1	-	0	3	1	3	0	2	1	3	2,4
9	3	3	4	1	2	3	3	4	-	2	0	3	0	3	1	3	0	3	3
10	1	4	4	1	2	1	4	-	-	-	1	3	0	3	0	2	1	3	1,4
11	2	3	4	1	2	2	3	4	1	-	0	3	1	3	0	3	0	3	2
12	3	4	4	1	2	3	4	-	-	2	0	3	0	3	1	3	1	3	3
13	1	3	4	1	2	1	3	4	-	-	1	3	0	3	0	3	1	4	1,4
14	2	4	4	1	2	2	4	-	1	-	0	3	1	3	0	3	1	3	2,4
15	3	3,4	4	1	2	3	4	4	-	2	0	3	0	3	1	3	1	3	3

(B)

T	Sessions sampled by					Sessions receiving tokens from					Number of tokens of session								Sessions served
	N_1	N_2	N_3	N_4	N_5	N_1	N_2	N_3	N_4	N_5	1 at		2 at		3 at		4 at		
											N_1	N_4	N_1	N_5	N_1	N_2	N_2	N_3	
1	1	3,4	4	1	2	1	4	4	1	2	1	1	0	1	0	0	1	1	1,4
2	2	3	4	1	2	2	3	4	1	2	0	1	1	2	0	1	0	1	2
3	3	4	4	1	2	3	4	4	1	2	0	2	0	2	1	1	1	2	3
4	1	3,4	4	1	2	1	4	4	1	2	1	3	0	3	0	0	2	3	1,4
5	2	3,4	4	1	2	2	4	4	1	-	0	3	1	3	0	0	2	3	2,4
6	3	3,4	4	1	2	3	4	4	-	2	0	3	0	3	1	0	2	3	4
7	1	3,4	4	1	2	1	4	4	-	-	1	3	0	3	1	0	2	3	1,4
8	2	3	4	1	2	2	3	4	1	-	0	3	1	3	1	1	1	3	2,4
9	3	4	4	1	2	3	4	4	-	2	0	3	0	3	2	1	1	3	3
10	1	3,4	4	1	2	1	4	4	-	-	1	3	0	3	1	0	2	4	1,4
11	2	3,4	4	1	2	2	4	4	1	-	0	3	1	3	1	0	2	4	2,4
12	3	3,4	4	1	2	3	4	4	-	2	0	3	0	3	2	0	2	4	4

TABLE I

We demonstrate the token generation and the service process in the network of Figure 2. Here, T represents the number of slots, and $W = 3$. In (a), all sessions always have packets to transmit. In (b), Session 3 generates packets in slots $1, 7, 13, ...$, but Other sessions always have packets to transmit.

only provably max-min fair in adhoc networks. We now discusss several aspects that make the algorithm amenable for implementation in adhoc networks. Future research is however necessary to realize this promise.

The algorithm is adaptive as it does not need any information about the network and the statistics of the arrival process. But, the lower bound γ_N on the threshold W needed to guarantee Theorems 2 and 3 depends on the topology. Simulation results however indicate that the bandwidth attained by the algorithm converges to the maxmin fair bandwidth even if W is below γ_N.

The token generations and the packet transmissions operate in parallel.

For deciding whether to generate a token to a session, a node needs to know the number of tokens of the session at the session's other node. Nodes can communicate the number of tokens in the headers of data and acknowledgement packets. The convergence results in Theorems 2 and 3 hold even if a node uses a delayed estimate of the number of tokens at other nodes in the token generation process, as long as the delay is finite. We have shown in [16] that in wireline networks the rates obtained by a similar algorithm converges to the maxmin fair rates irrespective of the feedback delay.

The computation of the weights of the edges and the token generation can be exceuted at the nodes with local information. We assume that a centralized processor computes the maximum weighted matching, and broadcasts the results. Each maximum weighted matching computation has complexity $O(VE)$. Theorems 2 and 3 hold even if *the matching is computed at regular intervals, rather than every slot,* as long as the intervals are finite. Also, the computation in each interval can use weights of the edges in a previous slot; the convergence is still guaranteed if the delays are finite. The algorithm becomes fully distributed if the maximum weighted matching is computed in a distributed manner. We plan to investigate the design of distributed algorithm for computing an approximate maximum weighted matching. Randomized scheduling algorithms may be useful in this context [19].

The scheduling problem in an input queued switch is mathematically equivalent to that in wireless networks. The nodes and links in an input queued switch form a bipartite graph, and the links scheduled must constitute a matching. Let the switch have $\sqrt{N}$ input nodes, $\sqrt{N}$ output nodes and N links. A vector of nonnegative real numbers $(r_1 \dots r_N)$ is a feasible service rate allocation for links if it satisfies inequality (1) with $\alpha = 1$. Our algorithm attains maxmin fairness of service rate allocation in any system where the scheduled links must be a matching and the feasible set of rate allocations is defined by inequality (1). Therefore, the algorithm attains maxmin fairness in input queued switches.

We believe that the scheduling policy can be generalized to attain other notions of fairness, e.g., proportional fairness [15], if the tokens generation procedure is modified. Radunovic *et al.* [15] have argued that proportional fairness attains a better tradeoff between fairness and throughput optimization than maxmin fairness. Their conclusion is primarily based on their observation that maxmin fairness allocates equal rates to all sessions in the network. But, this happens because they assume that a node can transmit simultaneously on multiple links, which several current day transceivers can not do. When this capability is not assumed, maxmin fairness may allocate different rates to different sessions (Figure 3).

We have considered fair allocation of bandwidth to MAC layer sessions. At the MAC layer every session spans only one link, whereas sessions at higher layers traverse multiple links. Fair allocation of end-to-end bandwidth will require cross-layer optimizations involving the network and MAC layers, or the transport, network and MAC layers. Radunovic *et al.* [15] have considered this optimization in the setting described above. In our case, this remains an open problem. The network we consider is however multihop as all nodes need not be in each others transmission range.

Appendix

A. Proof of Theorem 1

We first show that if a feasible bandwidth allocation satisfies condition (a) or (b), then it is maxmin fair. Consider any session i. If i satisfies condition (b) then its bandwidth can not be increased while maintaining feasibility. Let i satisfy condition (a), and n be its bottleneck node. If i's bandwidth is increased, then to maintain feasibility, the bandwidth of some other session j that traverses n must be decreased, as the total bandwidth of all sessions traversing n equals α. 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 the bandwidth of some session j that has bandwidth less than or equal to i's bandwidth. Thus, the bandwidth allocation is maxmin fair.

Now, assume that the bandwidth allocation is maxmin fair. We show that each session i satisfies condition (a) or (b). Suppose session i does not satisfy conditions (a) and (b). Thus, i's bandwidth is less than its arrival rate. Also, at each node n in i's path, either there exists a session that has bandwidth greater than that of i, or the total bandwidth of all sessions traversing n is less than α. In both cases, i's bandwidth can be increased without decreasing the bandwidth of any other session that has bandwidth less than or equal to i's bandwidth and without violating the feasibility conditions. This contradicts the definition of maxmin fairness.

B. Proof of Theorem 2

We prove Theorem 2 for the case that all sessions have packets for transmission at all times. We first introduce some terminologies. Let $P_{i,n}(s,t)$ be the number of tokens generated at end node n of session i during interval (s,t), d_n be the number of sessions traversing a node n, and $d = \max_n d_n$. Let $\alpha = p/q$ where p and q are relatively prime positive integers.

Theorem 4: Let $(r_1, \dots, r_N)$ be the maxmin fair bandwidth allocation. Let

$$\begin{aligned}
\mathcal{S}_k &= \{n : \text{node } n \text{ is in sessions } k\text{'s path,} \\
&\qquad \text{but is not } k\text{'s bottleneck node}\}. \\
p_1 &= \frac{\alpha}{max_{n:d_n<d} d_n}, \\
p_k &= \min_{n \in \mathcal{S}_k} \frac{\alpha - \sum_{i:r_i<r_k, i \text{ traverses } n} r_i}{|\{i : r_i \geq r_k, i \text{ traverses } n\}|}, \qquad k > 1, \\
\gamma_0 &= 0, \\
\beta_0 &= 0, \\
\gamma_k &= \sum_{i:r_i<r_k} \beta_i + q + 1, \;\; k > 0, \\
\beta_k &= 2 \sum_{i:r_i \leq r_k} \gamma_i + q + \frac{2\sum_{i:r_i \leq r_k} \gamma_i + q + \gamma_k}{p_k - r_k}, \;\; k > 0.
\end{aligned}$$

Let $W > \gamma_N$. Then, there exists a constant T, such that in any interval $(s,t) \subset (T, \infty)$, for each session i, $|P_{i,n}(s,t) - r_i(t-s)| \leq \max(\beta_N, 2\gamma_N)$.

Theorem 2 follows from Theorem 4 with $\nu = \max(\beta_N, 2\gamma_N) + T$.

B.1 Proof of Theorem 4

We first state the lemmas that we use in proving Theorem 4. Let $P_{i,A}(t)$ be the number of tokens generated at end node A of session i during interval $(0,t)$. Note that $P_{i,A}(t)$ differs from the number of tokens of i at A in slot t, $A'_i(t)$, since the number of tokens decrease with packet transmission. Also, equal number of tokens are removed from both end nodes of i, A and B. Thus, $P_{i,A}(t) - P_{i,B}(t) = A'_i(t) - B'_i(t)$.

$$\text{Thus, } |P_{i,A}(t) - P_{i,B}(t)| \leq W \quad \forall \quad t. \tag{3}$$

Lemma 1: Let in interval (u_1, u_2), session i be sampled at each end node A, B at least $r(u_2 - u_1) - \gamma$ times. Let $W > \gamma \geq 0$. Then,

$$\begin{aligned} P_{i,A}(u_2) &\geq P_{i,A}(u_1) + r(u_2 - u_1) - 2\gamma, \quad (4)\\ P_{i,B}(u_2) &\geq P_{i,B}(u_1) + r(u_2 - u_1) - 2\gamma. \quad (5)\end{aligned}$$

Lemma 2: Let in any interval $(s,t) \subset (t_0, \infty)$ a session i be sampled at least $r_A(t-s) - \gamma_A$ times at its end node A. Let at i's other end-node B, $P_{i,B}(s,t) \leq r_B(t-s) + \gamma_B$. Let $r_A > r_B$ and $\min(\gamma_A, \gamma_B) \geq 0$. Then, there exists $t_1 \geq t_0$ such that in any interval $(s,t) \in (t_1, \infty)$,

$$\max\left(P_{i,A}(s,t), P_{i,B}(s,t)\right) \leq r_B(t-s) + \gamma_B + \frac{\gamma_A + \gamma_B}{r_A - r_B}.$$

The proof of Theorem 4 follows.

Without loss of generality, assume that $r_i \leq r_{i+1}$ for all i.

We first prove the following. For each session k, there exists a time t_k such that $t_k \leq t_{k+1}, k \geq 1$, and the following hold in any interval (s,t) in (t_k, ∞).

1. (a) Session k is sampled at least $r_k(t-s) - \gamma_k$ times at both its end points.
(b) Let session k traverse node n, but n is not k's bottleneck node. Node n samples session k at least $p_k(t-s) - \gamma_k$ times.
2. For any node n in k's path, $P_{k,n}(s,t) \geq r_k(t-s) - 2\gamma_k$.
3. For any node n in k's path, $P_{k,n}(s,t) \leq r_k(t-s) + \beta_k$.

Theorem 4 follows from statements 2 and 3, with $T = t_N$.

Let $R(i) = \{j : r_j = r_i\}$. Clearly, $\cup_{i=1}^{N} R(i) = \{1, \ldots, N\}$. Using induction, we prove the above statements progressively for $k \in R(1), R(2), \ldots$.

Base case: We prove statements $1, 2, 3$ for $k \in R(1)$.

Statement 1

*(*a) Every node n samples each session traversing the node at least $(\frac{\alpha}{d_n})(t-s) - q$ times in any interval (s,t). Recall that d_n sessions traverse node n. The result follows since $r_k = \frac{\alpha}{d}$ and $d_n \leq d$.

*(*b) If session k traverses node n, but is not bottlenecked at n, then $d_n < d$. Thus $\alpha/d_n > \alpha/d$. Statement $1(b)$ follows.

Statement 2

We have shown that in any interval (s,t), session k is sampled at least $r_k(t-s) - q$ times at each of its end nodes. Also, $W > q$. Now, statement 2 follows from Lemma 1.

Statement 3

Let n be a bottleneck node of session k. Let $Y = \{i : i \text{ traverses } n, i \neq k\}$. Clearly, $Y \subset \{i : r_i \leq r_k\}$. Since $r_k = r_1$, $Y \subset R(1)$.

$$P_{k,n}(s,t) \leq \alpha(t-s) + q - \sum_{y \in Y} P_{y,n}(s,t).$$

From statement 2, $\forall\ y \in Y$, $P_{y,n}(s,t) \geq r_y(t-s) - 2\gamma_y$. Thus,

$$\begin{aligned} P_{k,n}(s,t) &\leq \alpha(t-s) + q - \sum_{y \in Y}(r_y(t-s) - 2\gamma_y) \\ &\leq \left((\alpha - \sum_{y \in Y} r_y)(t-s)\right) + q + 2\sum_{y: r_y \leq r_k} \gamma_y \\ &\leq r_k + q + 2\sum_{y: r_y \leq r_k} \gamma_y. \end{aligned} \tag{6}$$

The last inequality follows since $r_k = \alpha - \sum_{y \in Y} r_y$, as n is k's bottleneck node. Thus, statement 3 follows in this case. Let m be a non-bottleneck node of session k. From statement $1(b)$, m samples k at least $p_k(t-s) - \gamma_k$ times in any interval (s,t) and $p_k > r_k$. Thus, statement 3 follows from (6) and Lemma 2.

Induction hypothesis: Let statements $1, 2, 3$ hold for all sessions in $R(1) \cup \ldots \cup R(j-1)$. We now prove statements $1, 2, 3$ for session $k \in R(j)$.

Statement 1

Let n be a node in session k's path.

(a) Let $Y = \{i : i \text{ traverses } n, r_i < r_k\}$, and $Z = \{i : i \text{ traverses } n, r_i \geq r_k\}$. Let $C_x(s,t)$ be the number of times session x is sampled at node n in interval (s,t).

Since every node samples sessions in round robin order,

$$\begin{aligned} C_k(s,t) &\geq \max_{z \in Z} C_z(s,t) - 1 \\ &\geq \frac{1}{|Z|}\sum_{z \in Z} C_z(s,t) - 1 \\ &\geq \frac{1}{|Z|}[\alpha(t-s) - \sum_{y \in Y} P_{y,n}(s,t) - q] - 1. \end{aligned}$$

The last step follows since n samples the sessions in Z in each slot in which sessions in Y do not receive tokens. Clearly, $Y \subseteq R(1) \cup \ldots \cup R(j-1)$. Now, we use statement 3 of the induction hypothesis for upper bounding $\sum_{y \in Y} P_{y,n}(s,t)$.

$$\begin{aligned} &C_k(s,t) \\ &\geq \frac{1}{|Z|}\left(\alpha(t-s) - q - \sum_{y \in Y}(r_y(t-s) + \beta_y)\right) - 1 \\ &\geq \frac{1}{|Z|}(t-s)(\alpha - \sum_{y \in Y} r_y) - (q + 1 + \sum_{y: r_y < r_k} \beta_y) \quad (7) \\ &\geq \frac{1}{|Z|}(t-s)\sum_{z \in Z} r_z - (q + 1 + \sum_{y: r_y < r_k} \beta_y) \\ &\geq r_k(t-s) - (q + 1 + \sum_{y: r_y < r_k} \beta_y). \end{aligned}$$

The last step follows since $r_z \geq r_k$ for all $z \in Z$. Statement $1(a)$ follows.

(b) If n is not a bottleneck node of session k, statement $1(b)$ follows from (7).

Statement 2

Since $W > \gamma_N \geq \gamma_k \geq 0$, statement 2 follows from Lemma 1 and Statement $1(a)$.

Statement 3

Let n be a bottleneck node of session k. Let $Y = \{i : i \text{ traverses } n, i \neq k\}$. Note that i traverses n, if $i \in R(1) \cup \ldots \cup R(j)$.

$$P_{k,n}(s,t) \leq \alpha(t-s) + q - \sum_{y \in Y} P_{y,n}(s,t).$$

From statement 2, $\forall\ y \in Y$, $P_{y,n}(s,t) \geq r_y(t-s) - 2\gamma_y$. Thus,

$$\begin{aligned} P_{k,n}(s,t) &\leq \alpha(t-s) + q - \sum_{y \in Y}(r_y(t-s) - 2\gamma_y) \\ &= \left((\alpha - \sum_{y \in Y} r_y)(t-s)\right) + q + 2\sum_{y \in Y}\gamma_y \\ &\leq r_k + 2\sum_{i: r_i \leq r_k}\gamma_y + q. \end{aligned} \tag{8}$$

The last inequality follows since $r_k = \alpha - \sum_{y \in Y} r_y$, as n is k's bottleneck node.

Let m be a non-bottleneck node of k. From statement $1(b)$, m samples k at least $p_k(t-s) - \gamma_k$ times in an interval (s,t) and $p_k > r_k$. Thus, statement 3 follows from (8) and Lemma 2. Note that $t_k \geq t_{k-1}$.

B.2 Proof of Lemma 1

Consider a time interval (u_1, u_2). If $|P_{i,A}(u) - P_{i,B}(u)| < W\ \forall\ u \in (u_1, u_2)$, then $|A_i'(u) - B_i'(u)| < W$, $\forall\ u \in (u_1, u_2)$. Thus, in interval (u_1, u_2), i receives a token at each end node every time it is sampled at the end node; therefore, (4) and (5) follow from the lower bound on the sampling rates.

Now, let $|P_{i,A}(u) - P_{i,B}(u)| = W$ for some $u \in (u_1, u_2)$. Let u' be the smallest time in (u_1, u_2) at which $|P_{i,A}(u) - P_{i,B}(u)| = W$. Without loss of generality, let $P_{i,A}(u') = P_{i,B}(u') + W$. First, assume that $P_{i,B}(u) = P_{i,A}(u) + W$ for some $u \in (u_1, u_2)$. Consider the slots t in which $|P_{i,A}(t) - P_{i,B}(t)| = W$ in the following arbitrary sample path. Here, the symbol $A(B)$ at a time t denotes $P_{i,A}(t) = P_{i,B}(t) + W (P_{i,B}(t) = P_{i,A}(t) + W)$.

$u_1\ A\ A\ A\ A(t_1)\ B\ B\ B\ B\ B\ B\ B(t_2)\ A\ A\ A(t_3)$
$B(t_4)\ A(t_5)\ B\ B\ B(t_6)\ u_2$

In interval (u_1, t_1), i receives a token at B every time it is sampled at B, since $P_{i,B}(t) < P_{i,A}(t) + W\ \forall t \in (u_1, t_1)$.

$$P_{i,B}(t_1) - P_{i,B}(u_1) \geq r(t_1 - u_1) - \gamma. \tag{9}$$

$$P_{i,A}(t_1) = P_{i,B}(t_1) + W. \tag{10}$$

$$P_{i,B}(t_2) = P_{i,A}(t_2) + W. \tag{11}$$

In interval (t_1, t_2), i receives a token at A every time it is sampled at A, since $P_{i,A}(t) < P_{i,B}(t) + W$, $\forall t \in (t_1, t_2)$.

$$\text{Thus, } P_{i,A}(t_2) - P_{i,A}(t_1) \geq r(t_2 - t_1) - \gamma. \tag{12}$$

From (10), (11), (12),

$$P_{i,B}(t_2) - P_{i,B}(t_1) \geq r(t_2 - t_1) - \gamma + 2W. \tag{13}$$

Adding (9), (13),

$$P_{i,B}(t_2) - P_{i,B}(u_1) \geq r(t_2 - u_1) + 2(W - \gamma). \tag{14}$$

Similarly,

$$P_{i,B}(t_4) - P_{i,B}(t_2) \geq r(t_4 - t_2) + 2(W - \gamma). \tag{15}$$

$$P_{i,B}(t_6) - P_{i,B}(t_4) \geq r(t_6 - t_4) + 2(W - \gamma). \tag{16}$$

$$\vdots$$

Let t_l be the greatest time t in (u_1, u_2) at which $P_{i,B}(t) = P_{i,A}(t) + W$.

$$P_{i,B}(t_l) - P_{i,B}(t_{l-2}) \geq r(t_l - t_{l-2}) + 2(W - \gamma). \tag{17}$$

Note that $P_{i,B}(t) < P_{i,A}(t) + W$ in (t_l, u_2). Thus,

$$P_{i,B}(u_2) - P_{i,B}(t_l) \geq r(u_2 - t_l) - \gamma. \tag{18}$$

Adding (14) to (18),

$$P_{i,B}(u_2) - P_{i,B}(u_1) \geq r(u_2 - u_1) - \gamma + 2k_B(W - \gamma). \tag{19}$$

where k_B is the number of times A sequences are followed by B sequences.

$$\text{Since } W > \gamma, P_{i,B}(u_2) - P_{i,B}(u_1) \geq r(u_2 - u_1) - \gamma. \tag{20}$$

Now, (5) follows from (20).

We now prove (4).

$$\begin{aligned} P_{i,A}(t_1) &= P_{i,B}(t_1) + W. \\ P_{i,A}(t_1) &\geq P_{i,B}(u_1) + W + r(t_1 - u_1) - \gamma \text{ (from (9))}, \\ &\geq P_{i,A}(u_1) + r(t_1 - u_1) - \gamma \text{ (from (3))}. \end{aligned}$$

$$P_{i,A}(t_1) - P_{i,A}(u_1) \geq r(t_1 - u_1) - \gamma. \tag{21}$$

In the sample path under consideration, in each interval (t_{2p+1}, t_{2p+3}), $p = 0, 1, \ldots$, a sequence of A symbols follows a sequence of B symbols. In (u_1, t_2), a sequence of B symbols follows a sequence of A symbols. Using similar arguments as in the proof of the lower bound of $P_{i,B}(t_2) - P_{i,B}(u_1)$, we can show that

$$P_{i,A}(t_3) - P_{i,A}(t_1) \geq r(t_3 - t_1) + 2(W - \gamma). \tag{22}$$

$$P_{i,A}(t_5) - P_{i,A}(t_3) \geq r(t_5 - t_3) + 2(W - \gamma). \tag{23}$$

$$\vdots$$

Let t_m be the greatest time t in (u_1, u_2) at which $P_{i,A}(t) = P_{i,B}(t) + W$.

$$P_{i,A}(t_m) - P_{i,A}(t_{m-2}) \geq r(t_m - t_{m-2}) + 2(W - \gamma). \tag{24}$$

Note that $P_{i,A}(t) < P_{i,B}(t) + W$ in (t_m, u_2). Thus,

$$P_{i,A}(u_2) - P_{i,A}(t_m) \geq r(u_2 - t_m) - \gamma. \tag{25}$$

Adding (21) to (25),

$$P_{i,A}(u_2) - P_{i,A}(u_1) \geq r(u_2 - u_1) - 2\gamma + 2k_A(W - \gamma),$$

where k_A is the number of times A sequences follow B sequences.

$$\text{Since } W > \gamma, P_{i,A}(u_2) - P_{i,A}(u_1) \geq r(u_2 - u_1) - 2\gamma. \tag{26}$$

Thus, (4) follows from (26).

B.3 Proof of Lemma 2

We will show that in any interval $(s,t) \subset (t_0,\infty)$, $P_{i,A}(t) - P_{i,A}(s) \le r_B(t-s) + \gamma_B + \frac{\gamma_A+\gamma_B}{r_A-r_B}$. First assume that there exists a time $t_1 \ge t_0$ s.t.

$$P_{i,A}(t) \ge P_{i,B}(t) + W - \frac{\gamma_A+\gamma_B}{r_A-r_B} \quad \forall \quad t \in (t_1,\infty). \quad (27)$$

Consider an interval $(s,t) \in (t_1,\infty)$.

$$P_{i,A}(t) \le P_{i,B}(t) + W.$$

Thus, from (27),

$$\begin{aligned} P_{i,A}(t) - P_{i,A}(s) &\le P_{i,B}(t) - P_{i,B}(s) + \frac{\gamma_A+\gamma_B}{r_A-r_B}. \\ &\le r_B(t-s) + \gamma_B + \frac{\gamma_A+\gamma_B}{r_A-r_B}. \end{aligned}$$

The lemma follows.

We now show (27). We first show that $P_{i,A}(t) = P_{i,B}(t) + W$ infinitely often in (t_0,∞). Let there exist $t_2 \ge t_0$ s.t. $P_{i,A}(t) < P_{i,B}(t) + W \ \forall\ t \ge t_2$. Consider a $t_3 \ge t_2 + \frac{2W+\gamma_A+\gamma_B}{r_A-r_B}$. Thus, $P_{i,A}(t) < P_{i,B}(t) + W \ \forall\ t \in [t_2,t_3]$. Thus, in $[t_2,t_3]$, i receives a token at A every time it is sampled at A.

$$\text{Thus, } P_{i,A}(t_3) \ge P_{i,A}(t_2) + r_A(t_3-t_2) - \gamma_A.$$

$$\text{Since } t_2 \ge t_0, P_{i,B}(t_3) \le P_{i,B}(t_2) + r_B(t_3-t_2) + \gamma_B.$$

$$\begin{aligned} \text{Thus, } P_{i,A}(t_3) - P_{i,B}(t_3) &\ge P_{i,A}(t_2) - P_{i,B}(t_2) \\ &\quad +(r_A-r_B)(t_3-t_2) \\ &\quad -(\gamma_A+\gamma_B). \end{aligned}$$

$$\text{Since } P_{i,A}(t_2) - P_{i,B}(t_2) \ge -W,$$

$$P_{i,A}(t_3) - P_{i,B}(t_3) \ge -W + (r_A-r_B)(t_3-t_2) - (\gamma_A+\gamma_B).$$

$$\begin{aligned} &\text{Since } P_{i,A}(t_3) < P_{i,B}(t_3) + W, \\ &W > -W + (r_A-r_B)(t_3-t_2) - (\gamma_A+\gamma_B). \\ &\Rightarrow t_3 < t_2 + \frac{2W+\gamma_A+\gamma_B}{r_A-r_B}. \end{aligned}$$

This contradicts the fact that $t_3 \ge t_2 + \frac{2W+\gamma_A+\gamma_B}{r_A-r_B}$. Thus, $P_{i,A}(t) = P_{i,B}(t) + W$ infinitely often in (t_0,∞).

We now show (27), with t_1 being the smallest t in (t_0,∞) s.t. $P_{i,A}(t) = P_{i,B}(t) + W$. We next show that $P_{i,A}(t) \ge P_{i,B}(t) + W - \frac{\gamma_A+\gamma_B}{r_A-r_B}$ for all $t \in [t_1,\infty)$. Consider a $t_3 \ge t_1$. Let $P_{i,A}(t_3) < P_{i,B}(t_3) + W - \frac{\gamma_A+\gamma_B}{r_A-r_B}$. Then, $t_3 > t_1$. Thus, $[t_1,t_3)$ is a nonempty interval. There exists $t \in [t_1,t_3)$ s.t. $P_{i,A}(t) = P_{i,B}(t) + W$. Let t_2 be the largest $t \in [t_1,t_3)$ s.t. $P_{i,A}(t) = P_{i,B}(t) + W$. Thus, for all $t \in (t_2,t_3]$, $P_{i,A}(t) < P_{i,B}(t) + W$. Thus, in $(t_2,t_3]$ i receives a token at A every time it is sampled at A.

$$\begin{aligned} \text{Thus, } P_{i,A}(t_3) - P_{i,A}(t_2) &\ge r_A(t_3-t_2) - \gamma_A. \quad (28) \\ \text{Also, } P_{i,A}(t_3) - P_{i,A}(t_2) &< P_{i,B}(t_3) + W \\ &\quad -P_{i,B}(t_2) - W. \\ &\le r_B(t_3-t_2) + \gamma_B. \quad (29) \end{aligned}$$

From (28) and (29),

$$\begin{aligned} r_A(t_3-t_2) - \gamma_A &< r_B(t_3-t_2) + \gamma_B. \\ \text{Thus, } t_3 - t_2 &< \frac{\gamma_A+\gamma_B}{r_A-r_B}. \end{aligned}$$

Since i can receive at most 1 token at any node in each slot, $P_{i,B}(t_3) < P_{i,B}(t_2) + \frac{\gamma_A+\gamma_B}{r_A-r_B}$. Since $P_{i,A}(t)$ is the number of tokens generated for i at A in $(0,t)$, $P_{i,A}(t_3) \ge P_{i,A}(t_2)$.

$$\begin{aligned} \text{Thus, } P_{i,A}(t_3) - P_{i,B}(t_3) &> P_{i,A}(t_2) - P_{i,B}(t_2) \\ &\quad - \frac{\gamma_A+\gamma_B}{r_A-r_B}. \\ &= W - \frac{\gamma_A+\gamma_B}{r_A-r_B}. \quad (30) \end{aligned}$$

This contradicts the assumption that $P_{i,A}(t_3) < P_{i,B}(t_3) + W - \frac{\gamma_A+\gamma_B}{r_A-r_B}$. Thus, (27) holds.

C. *Proof of Theorem 3*

Consider session i. Let edge e in the topology graph correspond to session i. In a slot t, the weight of edge e is the difference between the number of tokens generated for i at one of i's end node in interval $(0,t)$, and the number of packets transmitted for i in interval $(0,t)$. We will show that e's weight is bounded by a constant ϑ. Since $W > \gamma_N$, from Theorem 2, the number of tokens generated at each end node of i in any interval of length t differs from $r_i t$ by at most a constant ν. The theorem follows with $\kappa = \nu + \vartheta$.

Now we show that the weight of any edge e is upper-bounded by a constant ϑ. Consider a fictitious network that has the same topology graph as the actual network. The fictitious and the actual networks schedule the same sessions every slot. A session generates a packet in the fictitious network when the weight of the corresponding edge in the topology graph of the actual network increases. The weight of any edge increases when the corresponding session receives a token at one of its end nodes. Since $W > \gamma_N$, from Theorem 2, the number of tokens generated at each end node of i in any interval of length t differs from $r_i t$ by at most a constant ν. It follows that the increase in the weight of the edge corresponding to i differs from $r_i t$ by at most a constant 2ν. Thus, the number of packets generated by i in the fictitious network, in any interval of length t differs from $r_i t$ by at most 2ν, and therefore, the arrival rates in the fictitious network satisfy inequality (1). Also, the number of packets waiting for transmission for a session in the fictitious network equals the weight of the corresponding edge in the topology graph of the actual network. Therefore, the fictitious network schedules the sessions that correspond to the edges in a maximum weighted matching in its topology graph, where the weight of an edge is the number of packets waiting for transmission in the corresponding session. From results in [17], [18], the fictitious network is stable, and a session in the fictitious network has at most ϑ packets waiting for transmission in any slot, where ϑ depends on the topology. Thus, the weight of any edge e is upper-bounded by a constant ϑ.

References

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LANs,” *Proceedings of ACM Sigcomm, 1994,* University College, London, UK, September, 1994, pp. 212-225.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ:Prentice-Hall, 1987.

[3] J. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms *Proceedings of ACM SIGCOMM'* 96, *pp* $143-156$, Palo Alto, CA, August $'96$

[4] A. Charny, An Algorithm for Rate Allocation in a Packet-Switching Network with Feedback. *M.S. thesis*, Massachusetts Institue of Technology, May 1994.

[5] Z.Y. Fang and B. Bensaou, “Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks for Wireless Ad-hoc Networks,” *Proceedings of INFOCOM*, 2004, March $9-12$, Hong Kong, China

[6] B. Miller, C. Bisdikian, A. Edlund, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications,* Prentice-Hall, 2000.

[7] E. Hahne. “Round-Robin Scheduling for Max-Min Fairness in Data Networks,” *IEEE Journal on Selected Areas in Communications,* Vol. 9, No. 7, Sept. 1991, pp $1024-1039$.

[8] Xiao Long Huang and Brahim Bensaou, “On Max-min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation,” *Proceedings of MOBIHOC,* 2001, October $4-5$, Long Beach, CA, USA

[9] B. Hajek and G. Sasaki, “Link Scheduling in Polynomial Time”, *IEEE Transactions on Information Theory,* Vol 34, No. 5, 1988.

[10] H. Luo, S. Lu, V. Bharghavan, “ A New Model for Packet Scheduling in Multihop Wireless Networks.” ACM Mobicom 2000, Boston, MA, August 2000, pp. $76-86$.

[11] T. Nandagopal, T. Kim, X. Gao and V. Bharghavan, “ Achieving MAC Layer Fairness in Wireless Packet Networks,” pp. $87-98$. *Proceedings of ACM Mobicom 2000,* Boston, MA, August 2000.

[12] T. Ozugur, M. Naghshineh, P. Kermani C. Olsen, B. Rezvani, J. Copeland, “ Balanced Media Access Methods for Wireless Networks,” *Proceedings of ACM Mobicom 1998,* Dallas, Tx October 1998, pp. $21-32$.

[13] IEEE, “Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications,” IEEE Standard 802.11, 1999.

[14] A. K. Parekh and R. G. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: The single-node case,” *IEEE/ACM Trans. on Networking*, Vol. 1, No. 3, pp. 344–357, Jun. 1993.

[15] B. Radunovic, J. Le Boudec, “Rate Performance Objectives of Multi-hop Wireless Networks,” *Proceedings of IEEE INFOCOM 2004,* March, 2004, Hong Kong, China.

[16] S. Sarkar, L. Tassiulas, “Back pressure based multicast scheduling for fair bandwidth allocation,” *Proceedings of INFOCOM'*2001, Alaska, April 2001, pp. $1123-1133$

[17] L. Tassiulas, “Adaptive back-pressure congestion control based on local information,” *IEEE Transactions on Automatic Control,* Vol. 40, No. 2, pp. 236-250, February 1995.

[18] 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, VOL 37, No. 12, December 1992.

[19] L.Tassiulas. “ Linear complexity algorithms for maximum throughput in radio networks and input queued switchs, ” *Proceedings of INFOCOM'1998*, San Francisco, CA, April 1998.

Leandros Tassiulas (S'89, M'91) was born in 1965, in Katerini, Greece. He obtained 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 is Professor in the Dept. of Computer and Telecommunications Engineering, University of Thessaly, Greece and Research Professor in the Dept. of Electrical and Computer Eng and the Institute for Systems Research, University of Maryland College Park since 2001. He has held positions as Assistant Professor at Polytechnic University New York (1991-95), Assistant and Associate Professor University of Maryland College Park (1995-2001) and Professor 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 and a Bodosaki Foundation award in 1999 and the INFOCOM ‘94 best paper award.

Saswati Sarkar (S'98, M'00) received Master of Engineering in Electrical Communication Engineering from the Indian Institute of Science in 1996 and Phd 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 in University of Pennsylvania. Her research interests are in resource allocation and performance analysis in communication networks. She 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 of IEEE Transaction on Wireless Communications from 2001.

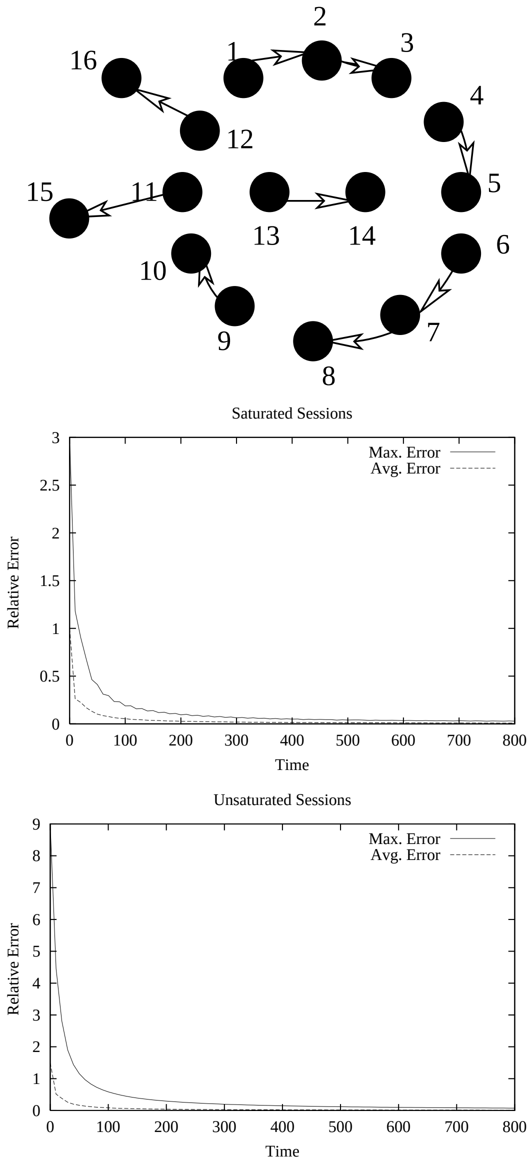

Fig. 5. The first figure shows the topology with 16 nodes and 14 sessions that is used in the simulations. There are 2 sessions in links $(1,2)$, $(4,5)$, $(6,7)$ $(7,8)$ and $(12,16)$, and one session in $(2,3)$, $(9,10)$, $(13,14)$ and $(11,15)$. When all the sessions are saturated, the maxmin fair allocation is $(0.33, 0.33, 0.5, 0.5, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.33, 1.0, 1.0, 1.0)$. The second figure demonstrates the convergence of the token generation rates to the maxmin fair rates in this case. When the first session in $(1,2)$ receives packets at the rate 0.1 per unit time, and other sessions are saturated, the maxmin fair allocation is $(0.1, 0.45, 0.5, 0.5, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.45, 1.0, 1.0, 1.0)$. The third figure demonstrates the convergence of the token generation rates to the maxmin fair rates in this case.

Fig. 4. Pseudo-Code for scheduling algorithm

```
Information at a node n for each session i traversing n
begin
  number of tokens for i at i's source and destination (token-num-source_i, token-num-destination_i),
  if  n is i's source,  then
     number of unmatched packets (new-packet-num_i).
end

Procedure Packet Arrival (Session i)
begin
  When i generates a packet, new-packet-num_i → new-packet-num_i + 1
end

Procedure Token generation (node n)
begin
  Sample sessions in α fraction of slots; sample sessions in round robin order.
  When n samples i
  if n is i's source  then
     if new-packet-num_i > 0 and token-num-source_i − token-num-destination_i < W  then
        generate a token for i ( token-num-source_i → token-num-source_i + 1 ),
        match an unmatched packet with a token (new-packet-num_i → new-packet-num_i − 1),
        do not sample any other session in the slot;
     else
        Sample the next session in round robin order, unless all sessions have been sampled in the slot.
  else
     if  token-num-destination_i − token-num-source_i < W  then
        generate a token for i ( token-num-destination_i → token-num-destination_i + 1 );
     else
        Sample the next session in round robin order, unless all sessions have been sampled in the slot.
end

Procedure Packet Transmission
begin
  Weight of the edge in the topology graph corresponding to session i = min (token-num-source_i, token-num-destination_i)
  Transmit packets from the sessions selected in the maximum weighted matching.
  if i transmits a packet, then
     token-num-source_i → token-num-source_i − 1,
     token-num-destination_i → token-num-destination_i − 1.
end
```