Maxmin Fair Scheduling in Wireless Networks

L. Tassiulas, S Sarkar
University of Maryland, University of Pennsylvania

August 15, 2001

Abstract

We consider scheduling policies for maxmin fair allocation of bandwidth in wireless adhoc networks. We formalize the maxmin fair objective under wireless scheduling constraints. We propose a fair scheduling which assigns dynamic weights to the flows such that the weights depend on the congestion in the neighborhood and schedule the flows which constitute a maximum weighted matching. It is possible to analytically prove that this policy attains both short term and long term fairness. We consider more generalized fairness notions, and suggest mechanisms to attain these objectives.

1 Introduction

Future generation wireless networks, both civilian and military, are envisioned to provide integrated services via inexpensive low-powered mobile computing devices. End users often expect seamless connectivity from wireline to wireless networks. On the other hand, wireless networks are limited in terms of resources like bandwidth, power, and the transmission medium is error prone. Therefore, meeting stringent quality of service requirements requires efficient resource management. For this purpose, we need to have an understanding of the users requirements and subsequently design the resource allocation techniques to satisfy the specific objectives. We first study several possible quality of service objectives. Each user may specify his individual service requirements, and subsequently the network may allocate resources to meet the desired objectives. However, the pitfall is that there needs to be an elaborate negotiation associated with each session initiation. For wireless networks, where sessions are short-lived, and nodes are continuously on the move, such negotiation may generate significant overhead. An alternate approach is to have users specify their individual utility or "satisfaction" functions, and the network allocate bandwidth such that the sum of the utility functions are maximized. The shortcoming of this approach is that the users need to explicitly know their utility functions. Most often the user satisfaction can not be characterized as precise functions of the allocated bandwidth. For example, satisfaction may depend on perceptual quality of service, and often times it is not possible to express perceptual quality as a function of bandwidth. In these situations, it is better to have a resource allocation objective which do not depend on the exact specifications of the utility functions, but attain good performance for a broad class of these functions. Max-min fair allocation of bandwidth is one such objective.

The basic idea behind maxmin fairness is to first allocate equal bandwidth to all contending users. If a user can not utilize its bandwidth, because of constraint elsewhere, then the residual bandwidth is distributed among others. Thus, no user is penalized excessively, and a certain minimum quality of service is guaranteed to all users. User satisfaction is often a concave function, i.e., satisfaction increases rapidly with increase in bandwidth in the low bandwidth region, and increases slowly with increase in bandwidth in the high bandwidth range. Thus total user satisfaction is often improved if all users obtain an equitable quality of service, rather than some users faring better at the expense of others. More importantly, this can be attained *without assuming specific knowledge about the individual user satisfaction functions. We would provide resource allocation strategies for attaining maxmin fairness in wireless networks.* Refer to Figure 1 for an illustrative example of a maxmin fair allocation in wireless networks. We consider fair allocation of bandwidth in wireless adhoc networks, and adopt the notion of maxmin fairness. The fairness objective is to distribute bandwidth as evenly as possible, without unduly reducing the throughput. A bandwidth allocation is said to be maxmin fair, if it is not possible to increase the allocation of any user without hurting another user with a lower service rate.

Maxmin fair objective has been extensively studied in the wireline context[2, 6]. However, resource allocation constraint significantly differ in wireless networks, and the problem demands a fresh treatment. For example, in wireline networks only the users sharing the same link contend with each other in terms of bandwidth usage. However, in wireless networks, all users in vicinity contend, even if they use different wireless links. This is because transmission from one user reaches every one else in the neighborhood. Now, this contention can be somewhat mitigated if the users in the vicinity use different transmission frequencies in FDMA transmission or different spread-spectrum codes in CDMA transmission. However, even then the same node can not simultaneously transmit or simultaneously receive in more than one link. For example, links L_1 and L_2 can not be simultaneously active in Figure 1. This is because wireless nodes often have single radios and can not conduct multiple transmissions and receptions simultaneously. Also, multiple simultaneous transmissions and receptions increase the power consumption. Hence, we assume throughout that a node can serve only one link at a time, i.e., it can either transmit in one link, or receive in one link, or remain idle. In Figure 1 node N_1 can transmit packets of flow 1 or 2 or receive packet from flow 3 or remain idle.

We conclude that space dependent contention is an inherent feature of wireless networks. Thus, it turns out that the scheduling of different links are interdependent in wireless networks, whereas different links can be scheduled independent of each other in wireline networks. As a result, bandwidth allocations inherently differ in the two. In addition, several flows share the same wireless link and the issue of contention between flows sharing the same link also arises as in wireline networks. Since link scheduling is interdependent, the bandwidth requirement of the flows in one link affects the scheduling of flows in other links as well. The challenge is to obtain a globally fair allocation in spite of this mutual independence. The main contribution of this paper is to propose policies which attain maxmin fair allocation.

Another factor which need to be considered is that the bandwidth allocation should also depend on the bandwidth demands of the individual flows. For example, if a flow does not utilize most of its bandwidth because of low traffic demand, then the excess should be allocated to contending flows. Also, the traffic demands of different flows change with time, and typically the wireless nodes do not have any knowledge about the traffic pattern. Thus the resource allocation policies need to be *online* in order to cater to the changes efficiently. There are several interesting questions in this aspect. A flow may not send packets for sometime, but may like to reclaim its share later by transmitting more packets. The network may wish to allow partial reclaim. The bandwidth allocation policy should have provisions to regulate the amount of reclaim, as also to distribute any bandwidth not utilized by flows with less overall traffic demand. The proposed policy will have parameters to control these, and will not assume any knowledge about arrival traffic pattern. Finally, maxmin fair allocation of bandwidth treats all flows equitably, and gives unequal bandwidth only when flows can

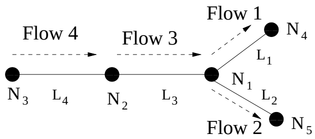

Figure 1: An example wireless network is shown in the figure. We assume that a flow contends with other flows traversing common nodes (e.g. flows 3 and 4). Any two flows which do not share a node can transmit packets simultaneously (e.g., flows 1 and 4). A node can transmit or receive in one flow at one time. Under maxmin fair allocation, every node offers equal bandwidth to all flows traversing the node. Thus flows $1, 2, 3$ are offered $1/3$ each at node N_1 (normalized bandwidth). Flows 3 and 4 are offered $1/2$ each at node N_2. All other nodes serve only one flow each, and hence offer unit bandwidth to the respective flows. Bandwidth attained by a flow depends on the constraint at both the end-points, Thus, flow 3 can not receive more than $1/3$ on account of constraint at node N_1. Flow 4 gets the remaining at node N_2. Thus the maxmin fair shares are $1/3, 1/3, 1/3, 2/3$ for flows $1, 2, 3, 4$ respectively.

not utilize the offered shares. However, in general, it may be necessary to distinguish between flows on the basis of the revenues etc. So, we present a generalized notion of prioritized maxmin fairness, and present scheduling policies for attaining the generalized notion. In this approach, different flows may have different priorities, and distinction is made between the flows on the basis of their priorities within the framework of fairness.

Maxmin fair allocation of bandwidth has not been studied before in the context of wireless networks. Nandagopal *et. al.* discusses scheduling strategies for maximizing the sum of user utilities in wireless networks[11]. They mention that maxmin fair bandwidth allocation can be obtained as a limiting case of utility maximization for a certain choice of utility functions. However, as we argue later, the optimization scheme becomes inefficient in those limiting cases. Maxmin fair allocation need to be addressed separately, and attaining maxmin fairness via utility maximization is not efficient. The other wireless adhoc protocols like IEEE 802.11[13], MACAW[1], CB-FAIR[12] do not attain any well-established notion of fairness.

We first formulate the maxmin fair allocation in presence of wireless specific scheduling constraints, and present necessary and sufficient conditions to check whether a bandwidth allocation is maxmin fair or not. Next, we present online scheduling policies for providing fair allocation of bandwidth.

The scheduling policy does not use any pre-computation of the maxmin fair rates, and also does not assume any knowledge of the arrival traffic pattern. However, the policy can detect whether the traffic demand of a flow is consistently less than its maxmin fair share, and in such cases distribute the excess bandwidth among other flows. At the same time, the policy allows a flow to reclaim a portion of its previous share, and the amount of reclaim can be precisely controlled through parameters. We would address both the fairness objectives, namely maxmin fairness, as also the more generalized notion of prioritized maxmin fairness. The basic idea is to keep an online estimate of the maxmin fair shares of the flows at the nodes. The estimate depends on the congestion at the neighboring nodes, as also the arrival rates of the individual flows. This estimate is used to decide the scheduling of the links.

The rest of this paper is organized as follows. Section 2 defines our network model and fairness objectives. Section 3 presents the scheduling policies for attaining maxmin fair allocations. Section 5 discusses related previous work. Finally, Section 6 concludes the paper.

2 Network model and Fairness Objective Formalization

Wireless nodes transmit through a multiple access medium and all flows in a vicinity contend for use of this medium. We assume that the flows which do not share nodes can transmit or receive packets simultaneously, but flows which share a node can not do so. This happens in systems where nodes in a neighborhood have different frequencies (codes) for transmission using frequency division multiple access (code division multiple access). This is the case for example in the recently emerging bluetooth standard[4]. In bluetooth nodes are organized in groups called piconets. Frequency hopping spread spectrum is used for transmission, and every piconet has a distinct frequency hopping code. So neighboring piconets do not interfere with each others transmission. For any piconet, one node is master and the remaining are the slaves. All communication in a piconet must take place via the master, as slaves can not directly communicate with each other. The master schedules the transmission in a piconet so that only one master-slave or slave-master communication takes place at a time. In this case, only the flows using the common nodes interfere with each other(refer Figure 2) and any two flows not using a common node can transmit simultaneously. We would

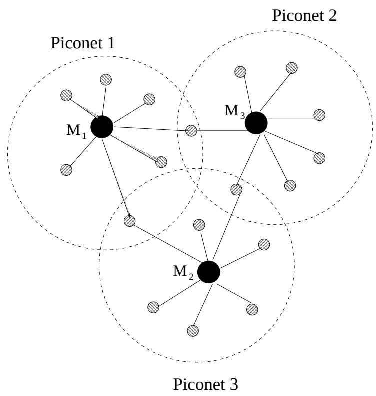

Figure 2: An example bluetooth topology is illustrated. The nodes are organized into 3 piconets. The masters of these piconets are M_1, M_2, M_3 respectively. The remaining nodes are the slave nodes. A master has a link with all its slaves. Some nodes belong to multiple piconets. Simultaneous transmission can take place in any two links which do not share a node.

like to emphasize that our model covers all packet radio networks where only the flows sharing a node contend, and bluetooth is an example application.

The contention between flows sharing a node arises as a node has only one radio unit. So a node can transmit in a flow or receive in another flow, but it can not perform multiple transmissions and/or multiple receptions simultaneously. As a result, packet transmission can take place in only one flow at a node, but flows not sharing a node can transmit packets simultaneously. For example, in Figure 1 flows 1 and 4 can transmit packets simultaneously, but flows 3 and 4 can not do so. Similarly, the contention between flows not using the same node, but within the transmission range of each other can be resolved by using locally distinct frequencies. As [3] argues, this can be attained if every node has a frequency different from all nodes within a distance of $2d$ from it, where d is the maximum distance between any two nodes in the transmission range of each other. If a node has at most p one-hop neighbors (two nodes in transmission range of each other are said to be one-hop neighbors), then at most $p^2 - p + 2$ frequencies are required in all, and the frequency assignment can be computed distributedly[5]. In this case, when a node A transmits to a node B, it transmits using the frequency allotted to node B. It is assumed that every node knows the frequencies of all other nodes in its one-hop neighborhood. For this purpose, every node peri-

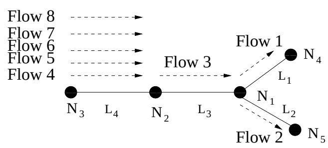

Figure 3: In this example, link L_4 has 5 flows. Any other link has one flow each.

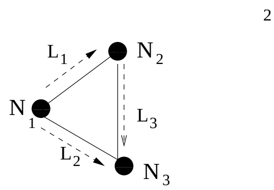

Figure 4: The bandwidth constraint at the nodes can be expressed as $r_1 + r_2 \leq 1$, $r_2 + r_3 \leq 1$ and $r_1 + r_3 \leq 1$. Here, r_i is the bandwidth of flow i. However, these constraints do not define the feasible set completely. Bandwidth allocation $(1/2, 1/2, 1/2)$ satisfies these constraints, but it is not a feasible allocation. This is because only one flow can transmit a packet at a time in this network, and hence the bandwidth of a flow can be at most $1/3$ if all flows get equal bandwidth. We need an additional constraint, $r_1 + r_2 + r_3 \leq 1$ in this case.

odically broadcasts its ID and its frequency. Similar multichannel networks have been studied in [3]. However, the contention between flows sharing a node exists in all of these systems.

2.1 Feasible set characterization:

The system can be modeled by a directed graph. A wireless node represents a vertex in the graph. A directed edge from vertex u to v represents a flow with u as the transmitter node and v as the receiver node. We assume only single link flows in this paper. There can be multiple edges between a pair of nodes, since there can be multiple flows with the same source and destination(refer Figure 3). A flow is said to be *active* if it transmits a packet. The set of active flows must constitute a matching[1] at any time. Let there be N flows. A bandwidth allocation $(r_1, \ldots, r_N)$ is *feasible* if we can find a scheduling of the flows which allocates bandwidth r_i to flow i for

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

every i.

We first characterize the feasible set in this case in terms of normalized bandwidth. We assume that the transmission/reception rate of the radio is the same at every node. Bandwidth of a flow normalized by this uniform transmission/reception rate is the normalized bandwidth. We will denote normalized bandwidth as simply bandwidth. The assumption of uniform transmission/reception rate simplifies our notations, and can easily be generalized. Since no two flows sharing a node can be active simultaneously, clearly sum of the bandwidth of all flows sharing a node must be less than 1 for feasibility. For example, $r_1 + r_2 + r_3 \leq 1$ and $r_3 + r_4 \leq 1$ in Figure 1, if r_i is the bandwidth of flow i. However, these conditions are not sufficient to characterize the feasible set in general. Figure 4 presents a counter-example. We present the sufficiency conditions next.

A graph is bipartite if the set of vertices can be partitioned into two disjoint subsets such that every edge has one end in each subset. In other words there is no edge with both ends in the same subset. A graph is bipartite if and only if it has no odd cycles. For example, the graphs in Figures 1 to 3 are all bipartite. Many wireless networks can be represented as bipartite graphs. For instance, bluetooth networks are often bipartite. In bluetooth, the same node can belong to multiple piconets. If we assume that a master node of one piconet does not belong to another piconet (which is often the case), then the topology can be represented by a bipartite graph. The conditions that the sum of the bandwidth of all flows sharing a node should not exceed 1 are sufficient to characterize the feasible set for bipartite graphs[7]. Hence, the feasibility set of the network in Figure 1 is specified by the inequalities $r_1 + r_2 + r_3 \leq 1$ and $r_3 + r_4 \leq 1$.

For graphs which are not bipartite, the following constraints are required in addition. Let Q be any subset of vertices with odd number of vertices. Let $L(Q)$ be the set of flows with both ends in Q. Then for every odd subset Q the sum of the bandwidth of all flows in $L(Q)$ must be upper bounded by $2/(|Q|-1)$[7]. This characterization of the feasible set is not amenable for computationally efficient bandwidth allocation procedures in general. However, [7] mentions other alternate characterization for feasible sets for non-bipartite graph as well. A sufficient condition for feasibility in any graph is that the sum of the bandwidth of all flows sharing a node is less than $2/3$. For instance, an allocation of $1/3$ to each flow in Figure 4 is feasible. Note that this condition is not necessary but sufficient, i.e., a bandwidth allocation may be feasible even if the sum of the bandwidth of the flows sharing a node is more than $2/3$. However, if the above condition is satisfied, then the allocation is feasible. Summarizing, a bandwidth allocation $(r_1, \ldots, r_N)$ is feasible if

$$\sum_{\substack{i:\text{ flow } i \\ \text{traverses node } j}} r_i \leq \alpha \forall \text{ node } j, \tag{1}$$

where α is the *node utilization factor.* For bipartite graphs, $\alpha = 1$ and $\alpha = 2/3$ for non-bipartite graphs.

Note that the feasibility condition is necessary and sufficient for bipartite graphs. But, the condition is not necessary for non-bipartite graphs, i.e., depending on the topology one may have α up to 1. However, all practical bandwidth allocation schemes leave unutilized bandwidth in every link. This is left so as to prevent buffer overflow during transients, and also to prevent excessive queuing delay at the nodes. Also, all implementations use distributed scheduling, and distributed scheduling normally does not attain the same utilization as the centralized scheduling. So, provisioning for more than 2/3 utilization at any node is neither practical nor advisable in any real system. In fact, even for bipartite graphs, we often choose $\alpha < 1$ to prevent excessive packet delays. The previous discussion implicitly assumes that all flows always have packets for transmission. In general that may not be the case. Hence, we need the demand constraint as well, i.e., if ρ_i is the long term arrival rate of flow i, then

$$r_i \leq \rho_i \tag{2}$$

So, inequalities (1) and (2) together characterize the feasible set. Incidentally, feasibility conditions for a wireline network involve the demand constraints, and the link capacity constraints only. The latter states that the sum of the bandwidth of all flows traversing a link must be less than the link capacity. Note that the flows in different links do not depend on each other and the constraints remain the same irrespective of the nature of the graph (bipartite or otherwise).

2.2 Fairness Objective Formalization

A bandwidth allocation is said to be maxmin fair if the bandwidth allotted to a flow can not be improved without decreasing that of any other flow having equal or less bandwidth. More formally, a feasible rate allocation $\vec{r}^1$ is maxmin fair if it satisfies the following property with respect to any other feasible rate 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 $\vec{r}^1$ ($r_j^2 < r_j^1$). The bandwidth allocations according to $\vec{r}^2$ are less even than those according to $\vec{r}^1$ in some sense. For example, $(1/3, 1/3, 1/3, 2/3)$ is the maxmin fair allocation in Figure 1. Note that the network can be represented by a bipartite graph, and we take $\alpha = 1$. On account of the bandwidth constraint at node N_1, the bandwidth allocated to flows $1, 2, 3$ can be increased only by hurting other flows which have equal rates. Also, on account of the constraint at node N_1, the bandwidth allocated to flow 4 can be increased by hurting flow 3 which has a lower rate.

Clearly, maxmin fair allocation gives absolute priority to the flows which receive the worst quality of service. However, depending on the service requirements, it may be necessary to have a more generalized notion of fairness. Network may need to discriminate among flows on the basis of their requirements and the pricing schemes. So, we consider the concept of prioritized maxmin fairness. Here, every flow i has a priority w_i, which is a positive real number. Greater the priority of a flow, better the service it should receive, subject to bandwidth availability. A bandwidth allocation is prioritized maxmin fair if it is not possible to increase the bandwidth of any flow i without hurting another flow j for which $r_j/w_j \leq r_i/w_i$. More formally, a feasible rate allocation $\vec{r}^1$ is maxmin fair if it satisfies the following property with respect to any other feasible rate 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 $r_j^1/w_j \leq r_i^1/w_i$ and the jth component of $\vec{r}^2$ (r_j^2) is strictly less than the jth component of $\vec{r}^1$ ($r_j^2 < r_j^1$). For example, if flow 4 has priority 4 and other flows have priority 1 each in Figure 1, then the prioritized maxmin fair allocation is $(2/5, 2/5, 1/5, 4/5)$. Bandwidth of flow 4 increases from $2/3$ to $4/5$ by increasing its priority. Clearly different qualities of service can be enforced within the framework of fairness by introducing different priorities.

2.3 A necessary and Sufficient Condition for Maxmin Fairness

We present a necessary and sufficient condition for maxmin fairness in wireless networks. A flow i is said to be *bottlenecked* at a node n if the sum of the bandwidth of all flows traversing the node equals the node utilization, α and the bandwidth of flow i is the maximum amongst those of other flows traversing the node n. For example, nodes N_1 and N_2 are the bottlenecks for flows 3 and 4 respectively in Figure 1 ($\alpha = 1$ as this is a bipartite graph). However, node N_3 is not the bootleneck for any flow in Figure 1 as the only flow (flow 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 flow satisfies at least one of the following conditions: (a)the flow has at least one bottleneck node (b)the bandwidth allocated to the flow equals its long term arrival rate.*

Proof of Theorem 1: We first show that if a feasible bandwidth allocation satisfies the above properties, then it is maxmin fair. Consider any flow i. If it satisfies condition (b) then its bandwidth can not be increased while maintaining feasibility. Let flow i satisfy condition (a). Let its bottleneck node be denoted n. If the bandwidth of flow i is increased, the bandwidth of some other flow j which traverses n must be decreased in order to maintain feasibility as the total bandwidth of all flows traversing n equals its bandwidth utilization α. Bandwidth of any flow traversing node n is either less than or equal to that of i. Thus any increase in bandwidth of flow i will decrease that of some flow j which has bandwidth less than or equal to that of i. Thus the bandwidth allocation is maxmin fair.

Now consider a maxmin fair bandwidth allocation. We will show that the bandwidth allocation satisfies the conditions of the theorem. Consider any flow i. Suppose it does not satisfy conditions (a) and (b). Thus its bandwidth is less than its arrival rate. Also, at each end node, either there exists a flow which has bandwidth greater than that of i or the total bandwidth of all flows traversing the node is less than the node utilization α. In either case, the bandwidth of i can be increased without decreasing that of any other flow which has bandwidth less than or equal to that of i, and without violating the feasibility conditions. This contradicts the fact that the bandwidth allocation is maxmin fair. □

Note that this “bottleneck” condition is similar to analogous conditions for maxmin fairness in wireline networks, and is particularly convenient for testing the maxmin fairness of a bandwidth allocation. The idea behind this result is that if the bandwidth allotted to a flow is less than its arrival rate, then it must be constrained at one of its nodes, and that node should be “fully utilized”. Also, for fairness, that node should not allot greater service to any other flow.

A similar necessary and sufficient condition exists for weighted maxmin fairness. A flow i is said to be *prioritized bottlenecked* at a node n if the sum of the bandwidth of all flows traversing the node i equals the node utilization, α and $r_i/w_i \geq r_j/w_j$ for any other flow j traversing node n. For example, node N_2 is the prioritized-bottleneck for flow 3 in Figure 1 under the priorities $(1,1,1,4)$ and allocation $(2/5, 2/5, 1/5, 4/5)$ for the flows. Note that in absence of priorities, node N_1 is the bottleneck node for flow 3.

Theorem 2 *A feasible bandwidth allocation is prioritized maxmin fair if and only if every flow satisfies at least one of the following conditions: (a)the flow has at least one prioritized bottleneck node (b)the bandwidth allocated to the flow equals its long term arrival rate.*

Theorem 2 is a generalization of Theorem 1. The proof is similar to that of Theorem 1 and is omitted for brevity.

3 A scheduling policy for generating maxmin fair bandwidth allocation

Here, we will present scheduling policies for attaining maxmin fairness, as well as its prioritized generalization. We first show that designing a scheduling policy for attaining maxmin fairness is nontrivial. Several intuitively appealing scheduling policies do not attain this objective. A promising approach is to give priority to flows which have not been scheduled many times in the recent past. That is, we will first schedule as many flows as possible which have received the least service so far, subject to this we will schedule as many flows as possible which have received second minimum service and so on. Figure 5 provides a counter-example where such a policy does not attain maxmin fairness even for the simple case of a bipartite graph with all flows having large arrival rates (i.e., demand constraints do not exist). Now, we will present policies which attain maxmin fairness.

For simplicity we consider slotted system only. Initially, we also assume that every packet is of duration 1 slot. At first we do not consider any priorities. We will discuss the generalizations later. As discussed before, the set of flows scheduled every slot must be a matching. The challenge is to choose the matching such that the flows get maxmin fair bandwidth. Figure 5 illustrates that giving absolute priority to flows which received less service so far, does not attain our purpose. We are going to use a

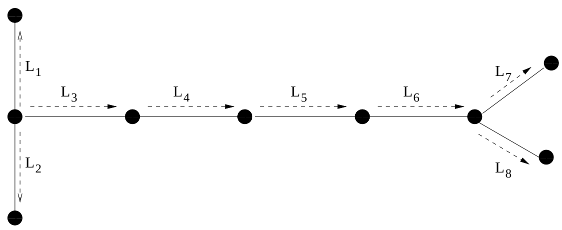

Figure 5: Every link has a single flow in this example. Assume that every flow has packets for transmission in every slot. Initially, none of the flows have transmitted any packet. The suggested policy selects the maximum number of flows which form a matching. One such selection consists of flows in links L_1, L_4, L_8. These flows transmit one packet each in the first slot. Next slot, at first only the flows which have not transmitted any packet are considered, and a maximum matching is selected from this set. Thus, a possible selection consists of flows in links L_2, L_5, L_7. Note that no other flows can transmit packet once these are selected. Third slot, the only flows which have not transmitted any packet are those in links L_3 and L_6. These flows are selected, and no other flow can transmit packet simultaneously with these flows. Now all flows have transmitted one packet each. Hence, repeating the same selection procedure, one would repeatedly select the same sequence successively. Thus, in the long run every flow transmits once every 3 slots. Thus, every flow receives 1/3 units of bandwidth. But, it is possible to allocate bandwidth 2/3 to the flow in link L_5 while allocating 1/3 to each of the other flows. Hence the bandwidth allocation of 1/3 to each flow is not maxmin fair. The increase is possible by sequentially choosing flows in L_1, L_4, L_6 in one slot, flows in L_2, L_5, L_7 next slot, and flows in L_3, L_5, L_8 next slot. In fact, the maxmin fair allocation is $(1/3, 1/3, 1/3, 1/2, 1/2, 1/3, 1/3, 1/3)$.

maximum weighted matching[2] where weight of every flow depends on an estimate of the maxmin fair bandwidth of the flow, previous service received by the flow, and the traffic demand of the flow. We prove that this scheme attains maxmin fairness. A node updates the weight of every flow every slot, based on the new maxmin fair estimate, traffic availability and the scheduling in the previous slot. The scheduling of the flows in the slot uses this newly updated weights. Thus, scheduling and the estimation of the maxmin fair bandwidth proceed in parallel. No prior knowledge of traffic pattern is assumed. The weight update procedure uses local information only. However, the scheduling uses a maximum weighted matching and hence uses global information.

3.1 Description of the policy

Since the objective is maxmin fairness, every node would initially try to give the same number of transmission chances to all its flows. But, some of the flows may not have sufficient packets to transmit. Also, when a flow transmits a packet, it uses resource at both ends. So, a flow may not be able to utilize all the chances offered by a node on account of congestion in the other node. In these cases, a node should devote less time in these constrained flows, and serve unconstrained flows in the residual time. When a flow is scheduled it is as if it receives service from both its end nodes. Each node is viewed as an independent server that allocates service to the flows traversing the node. The challenge is that a flow can be served only if its two nodes are synchronized to provide their service at the same time.

The basic stages of the policy are as follows

1. Each node allocates service tokens to the flows traversing the node in a *round-robin-like* fashion. We describe the **Service Token Allocation** procedure in detail later.

2. Each flow maintains two service token buckets, one for each end node, where it stores the tokens received by the corresponding end node. The service credit of the flow equals the minimum of the two service token buckets.

3. The collection of non-conflicting flows with maximum service credit is selected for service at each slot. More specifically, a maximum weighted matching of the flows is scheduled for transmission. Here, weight of each flow is its service credit.

[2] 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.

4. Whenever a flow is served one token is deducted from each one of its token buckets.

Now we describe the **service token allocation** procedure. The service token allocation process has some subtleties that are better explained if we distinguish first the following special case.

Service token allocation: Saturated System

Assume that each flow has an infinite packet supply.

1. A flow (i,j) (i.e., a flow with end nodes i, j) is *eligible* to receive a service token at slot t from node i if the size of the token bucket i of the flow does not exceed the size of the token bucket j of the flow by more than W service tokens.

2. A node n allocates service tokens in α fraction of slots (α is the desired utilization factor), and idles in $1-\alpha$ fraction. It allocates the service tokens in a round-robin fashion in the slots it does not idle, considering only the eligible flows at that slot.

Note that the eligibility of a flow is node dependent, i.e. a flow may be eligible for service by one of its end nodes and ineligible by the other.

For the saturated system and the policy as described above it holds that the vector of the time average service rates received by the flows converges to the maxmin feasible service rate vector.

Lets consider now the more general case of a system where flows will may not have infinite packet supply. The packets in flow (i,j) are generated according to an arrival process with rate a_{ij}. the scheduling policy is as above with the following differences.

Service token allocation: System with arrivals

The service token allocation is done following the same round-robin mechanism described in the saturated system with the difference that if the packet buffer of a flow is empty then the flow is ineligible.

The collection of non-conflicting links with non empty packet queues and with maximum aggregate service credit is selected for service at each slot. Note that flow scheduling relies on service credits and not on queue lengths.

For the above policy it holds that the vector of the time average service rates received by the flows converges to the maxmin feasible service rate vector. Furthermore the flows for which the arrival rate equals the service rate the packet length process is stable. Note that the service rate as defined above coincides with the departure rate. In figure 6 we have a pseudocode description of the algorithms for this most general case.

The scheduling policy has been described in the pseudo-code in Figure 6.

Every node maintains the following for each flow i traversing the node :
(a)number of unmatched packets (new-packet-num$_i$)
(b)number of tokens at this end (token-num-transmitting-end$_i$/token-num-receiving-end$_i$)
(c)estimate of the number of tokens at the other end
(token-num-estimate-transmitting-end$_i$/token-num-estimate-receiving-end$_i$)
Packet Arrival:
When an exogeneous packet arrives for flow i, new-packet-num$_i$ $\rightarrow$ new-packet-num$_i$ $+ 1$
Flow Sampling:
A node idles in $1 - \alpha$ fraction of slots
A node samples flows in round robin order in the slots it does not idle.
Token-generation:
When node n samples flow i and node n is the transmitter node for flow i, then node n generates a token to flow i, if

- new-packet-num$_i$ > 0 and
- token-num-transmitting-end$_i$ $-$ token-num-estimate-receiving-end$_i$ $\leq W$

If token is generated to flow i,

$$\begin{aligned}\text{new-packet-num}_i &\rightarrow \text{new-packet-num}_i - 1\\ \text{token-num-transmitting-end}_i &\rightarrow \text{token-num-transmitting-end}_i + 1\end{aligned}$$

When node n samples flow i and node n is the receiver node for flow i, then node n generates a token to flow i, if

$$\text{token-num-receiving-end}_i - \text{token-num-estimate- transmitting-end}_i \leq W$$

If token is generated to flow i,

$$\text{token-num-receiving-end}_i \rightarrow \text{token-num- receiving-end}_i + 1$$

Packet Transmission:
Weight of a flow i = $\min\left(\text{token-num-transmitting-end}_i, \text{token-num-receiving-end}_i\right)$
Compute a maximum weighted matching
Transmit packets from the flows selected in the maximum weighted matching
If a flow i transmits a packet,

$$\begin{aligned}\text{token-num-transmitting-end}_i &\rightarrow \text{token-num-transmitting-end}_i - 1\\ \text{token-num-receiving-end}_i &\rightarrow \text{token-num-receiving-end}_i - 1\\ \text{token-num-estimate-transmitting-end}_i &\rightarrow \text{token-num-estimate-transmitting-end}_i - 1\\ \text{token-num-estimate-receiving-end}_i &\rightarrow \text{token-num-estimate-receiving-end}_i - 1\end{aligned}$$

Figure 6: Pseudo-Code for Scheduling Policy

3.2 Intuitive Justification

We prove analytically that this policy attains maxmin fairness in the appendix. Here, we intuitively argue that every flow receives tokens at the maxmin fair rate. Note that the number of tokens accumulated at either end of a flow can never differ by more than a constant. As a result, both ends generate tokens at the same rate to a flow. Since every node samples flows in round-robin order, a node offers the same token rate to all flows traversing the node. However, the token generation rate may have to slow down for a flow because of reduced traffic demand or bandwidth constraint at the other end. The node then generates tokens to other unconstrained flows in the residual slots. This is as per the maxmin fair objective.

We use the example in Figure 1 for further elucidation. First consider the case when all flows are saturated i.e., all flows always have packets to transmit. In this case, every node samples all flows in round-robin order, and generates a token to a flow, unless the other end has too few tokens (traffic demand is not really a constraint). Now, node N_1 samples flows $1, 2, 3$ at rate $1/3$ each. Initially node N_2 generates tokens to flows 3 and 4 at rate $1/2$ each. But, it soon can not generate tokens to flow 3 several times it samples, because the other end of flow 3, node N_1 is generating tokens at the rate of $1/3$ only, and the number of tokens accumulated at both ends can not differ by more than W, a constant number. So, the node generates tokens for flow 4 in these residual slots. The other end of flow 4, node N_3 samples flow 4 every slot, and can potentially generate a token every slot, token generation rate of node N_2 permitting. So, flow 3 gets tokens at the rate $1/3$ from both ends, and flow 4 gets tokens at the rate $2/3$ at both ends. Note that these are the maxmin fair rates of these flows.

Next, consider the case when all flows do not always have packets to send. It is essential that both ends of the flow detect this condition and allocate the residual bandwidth to other flows which have enough packets. The transmitter node monitors this by matching tokens with packets. Whenever a token is generated, it is matched with a hitherto unmatched packet. If there is no such packet, a new token is not generated. The receiver end does not generate tokens if there are too few tokens at the other end, and thus implicitly takes care of the traffic demand as well. Consider the case where arrival rate for flow 3 in Figure 1 is only $1/6$. Now, node N_2 samples flow 4 at rate $1/2$, but can generate token at the rate $1/6$ only, because it does not find sufficient unmatched packets. Number of tokens accumulated at receiver node N_1 can not be W more than that at transmitter node

N_2. Hence, even though node N_1 samples flow 3 at rate $1/3$, it generates tokens at rate $1/6$ only. Both the nodes N_1 and N_2 distribute the remaining chances to other contending flows. For example, node N_1 gives tokens at the rate $5/12$ to flows 1 and 2 respectively, and node N_2 gives token at the rate $5/6$ to flow 4. Incidentally, the maxmin fair rates in this case are $(5/12, 5/12, 1/6, 5/6)$. Since, tokens are generated at the maxmin fair rate at both ends of a flow, and the weight of a flow is the minimum of the number of tokens at both sides, weight of a flow increases at the maxmin fair rate. Every time a flow is served its weight decreases by 1. Thus the weight of a flow actually represents the difference between the maxmin fair service and the service actually received. A maximum weighted matching gives priority to flows with high weights, i.e., the flows whose service differ most from the respective maxmin fair bandwidth. So higher the weight of a flow, better is its chance to be scheduled in the next slot, and thus the flows ultimately receive the maxmin fair bandwidth. However, even though a maximum weighted matching gives priority to flows with high weights, it does not give absolute priority. For example, if choosing a flow with the maximum weight prevents the choice of two lower weight flows such that the total weight of these flows exceed the weight of the maximum weight flow, then the two lower weight flows are preferred. So, a maximum weighted matching also considers the global perspective. It is interesting to note that giving absolute priority to the maximum weight flows does not attain the maxmin fair objective.

Outline of analytical results: The properties of the policy that were outlined above for the saturated and the system with arrivals follow from the formal result stated here. For the purpose of analysis, we assume that the arrival process for any flow can be characterized as follows: Number of packets arriving for flow i in any interval of length t differs from the number $\rho_i t$ by at most σ_i, where ρ_i is the long term arrival rate and σ_i is the burstiness which does not depend on the interval size. Note that these does not preclude "greedy" flows which always have packets. A greedy flow i can be modeled by choosing $\rho_i = 1$ and $\sigma_i = 0$.

Theorem 3 *If the threshold W and the buffers are sufficiently large, then the number of tokens generated at each end of a flow i in any interval of length t differs from $r_i t$ by at most a constant ν_i, where r_i is the maxmin fair rate of flow i, and ν_i is a constant which does not depend on the length of the interval.*

The constant ν_i depends on the topology, traffic parameters and the threshold size. It follows that the minimum of the number of tokens generated at the two ends of a flow i in any interval of length t differs from $r_i t$ by at most a constant ν_i. We prove Theorem 3 in the appendix. The constants ν_i can be found from the proof (from statement of Theorem 5 more precisely).

Theorem 4 *The number of packets served in any interval of length t for any flow i differs from $r_i t$ by at most a constant κ_i, where r_i is the maxmin fair rate of flow i, and κ_i is a constant which does not depend on the length of the interval t but only on the topology, traffic parameters and the threshold size.*

Proof of Theorem 4: The packet transmission process is logically equivalent to the following setup. Every time the minimum of the number of tokens accumulated at both ends increases, a new packet is "released" for transmission. A "released" packet is transmitted only when the flow is selected in the maximum weighted matching of flows. It follows from Theorem 3 that the number of packets released in any interval differs from the maxmin fair number of packets by at most a constant, and the value of this constant is independent of the interval size and depends only on the topology of the system. The packet release process is maxmin fair and hence feasible. Note that the weight of a flow is the number of released packets waiting for transmission. It follows from results in [14, 15] that the number of packets of any flow transmitted in an interval by a maximum weighted matching based scheduling differs from the number of released packets by at most a constant. The value of this constant is independent of the interval size and depends only on the topology of the system. The result follows. □

We conclude this section with the generalization for obtaining the prioritized maxmin fair rates. The scheduling process essentially remains the same. The difference is in the sampling of flows. Intuitively, now we need a weighted round robin scheduling where the weights depend on the priorities. However, the generalization is not straight forward, as a flow may not be able to generate token to a flow it samples, and in this case may need to sample another flow. The sampling order in slots "missed" by flows is not clear. We will use a different sampling in this case. Every node now maintains the total number of times it sampled a flow right from the start. Let this be $C_i(t)$ at time t for flow i. At any time a node n needs to sample a flow, it samples flow i such that $C_i(t)/w_i = \min_{j:\text{ flow } j \text{ traverses node } n} C_j(t)/w_j$, where

w_j is the priority of flow j. The rest of the scheduling i.e., the conditions for generating token to a flow, and also serving the flows as per a maximum weighted matching remains the same in this case. Intuitively, traffic demand permitting, a flow with a high priority gets larger number of tokens from both ends, and is thus scheduled more often. It can be analytically shown that this policy attains the prioritized maxmin fairness. Both theorems 3 and 4 hold in this case, with r_i being the prioritized maxmin fair rate of flow i. Finally, the quantity $C_i(t)/w_i$ increases continuously with time, and may become arbitrarily large. Instead, a node n can maintain the quantity $C_i(t)/w_i - \min_{j:\text{ flow } j \text{ traverses node } n} C_j(t)/w_j$ for each flow i traversing node n, and sample the flow with 0 value of this quantity. This does not affect the scheduling order in any way, but keeps the storage space bounded.

4 Discussion

We point out certain salient features of this policy in the discussion section. The token generation and the weight computation part is completely distributed. It is true that every node needs information regarding the number of tokens at the other ends of the flows. This can be communicated to neighbors. Transmitter nodes can communicate this information in the data packet headers, and receiver nodes can communicate this in the acknowledgement packets or in the headers of packets of flows in reverse direction. Clearly, the nodes may not know the exact number of tokens at the other ends, but will know the number of tokens at a previous time. That is this information will reach with a delay. However, the scheduling does not assume the exact knowledge, but only uses an estimate, and the estimate can be the number of tokens at a previous time. The analytical results hold as long as the delay is finite. The intuition is that as long as the delay is finite the estimate will differ from the exact value by at most a constant, and the constants do not affect the long term average throughput. Experimental results support this fact. As Theorem 3 indicates, the tokens are generated at the maxmin fair rate. Hence, the token generation can be used just for maxmin fair rate computation as well.

The only centralized part of the policy is the computation of the maximum weighted matching. The envisioned architecture here is that a centralized processor computes the maximum weighted matching, and broadcasts the computation to every node. Now, the centralized processor may not know the latest weights. Again, if the maximum weighted matching is computed with weight estimates, where the estimates differ from the actual weights by constants, then the long term throughput do not change. In other words, it can be shown analytically that both Theorems 3 and 4 hold if the maximum weighted matching is computed with delayed estimates of weights. The scheduling does not assume any knowledge about the statistics of the arrival process, e.g., the arrival rates etc. Flows may generate few packets in one interval, and may wish to reclaim a part of their lost share later. The current version of the policy does not allow this. This can be incorporated with a minor modification in the description in Figure 6. Now, the transmitter node does not generate any token for a flow, if it does not have an unmatched packet. The modification would be to generate up to K tokens for a flow, which are not matched with a packet, and then wait till further packet arrival. The number K can be decided apriori and determines the amount of reclaim. Also, during the matching computation the scheduler does not consider flows without any packet waiting for transmission (now weight of a flow may exceed the number of packets waiting for transmission by K), and the maximum weighted matching is computed from the rest. It may be shown analytically that maxmin fair bandwidth is attained with this modification as well, as long as K is finite.

We would like to discuss the choice of the parameters like the threshold W and the buffer sizes. The analytical results hold for large but finite values of both. Also, the analytical bounds depend on the topology and the traffic parameters. Refer to the proof for Theorem 3 for exact expressions of these bounds. However, the analytical bounds are overly pessimistic. All our experimental results indicate that fairly modest values of these quantities are required. Also, greater the delay in receiving the number of tokens at the other end, greater is the threshold W required for convergence to maxmin fair rates, but still the values are moderate for reasonable delays.

We guarantee both long term fairness and short term fairness. The service received by any flow in any interval differs from the maxmin fair number of packets by at most a constant(Theorem 3). This is an important advantage as many flows can be short-lived.

Flows can suffer from packet losses, particularly if the arrival rate is greater than the maxmin fair bandwidth. If the buffers are sufficiently large (but finite), and no more than $\rho_i t + \sigma_i$ packets arrive for a flow i in any interval of length t, then the packet loss for the flow i in any interval of length t is at most $t \max(\rho_i - r_i, 0) + \phi_i$, where ϕ_i is a constant which do not depend on the interval size. Thus, long term loss rate depends on the difference between the long term arrival rate and the maxmin fair rate. Thus there is no congestion related packet loss if the arrival rate is less than the maxmin fair rate and the burstiness is bounded. Again, the analytical bounds for buffer sizes can be found in the appendix. But, from experimental observations, the result holds for much smaller buffer sizes. We have assumed a slotted system with equal length packets. Now, slots are actually "logical slots", i.e., one slot represents one packet duration. In general, packets may have different lengths. In this case fairness notion may be in terms of bandwidth consumed or the number of packets transmitted, and the exact policy guaranteeing maxmin fairness may need some modification depending on the fairness notion. However, this assumption of equal length packets simplifies the policy, and at the same time illustrates the essential intuition.

5 Review of Relevant Research

We briefly review relevant research in this section. Several bandwidth allocation policies have been proposed in the context of packet-radio networks. However, none of these present a mechanism for attaining maxmin fairness. Hajek and Sasaki characterize the feasible set for packet radio networks which consider interference between flows sharing common vertices only[7]. The same paper also presents an offline scheduling policy for attaining any feasible bandwidth allocation in such networks. However, being offline this strategy can not respond to dynamic changes in traffic demand. Tassiulas *et. al.* presents online scheduling policies for attaining any feasible allocation in wireless networks[14, 15]. These policies attain the desired throughput as long as the arrival rates are feasible. The bandwidth allocation may not be fair if the traffic arrival rates are not feasible.

Maxmin fair bandwidth allocation for wireline networks has received significant attention[2, 6]. The allocation constraints are significantly different for wireless networks. Hahne presents a scheduling policy for attaining maxmin fair bandwidth allocation in wireline networks[6]. There is some similarity between the scheduling in [6] and the token generation policy in our paper, namely a link does not serve a packet if the congestion downstream is high in [6]. The packet scheduling needs to be completely different in our case on account of the interdependence between the links. For example, we schedule flows selected in a maximum weighted matching, where the weight of a flow depends on the traffic demand and the congestion of the neighbors. A back-pressure based round robin scheduling is used in [6]. Also, the scheduling of flows traversing different links are independent.

Recently, bandwidth allocation strategies have received considerable attention for wireless adhoc networks. Existing protocols like IEEE 802.11[13], MACAW[1] and CB-fair[12] do not attain any well-defined notion of fairness. Nandagopal *et. al.* present scheduling strategies for maximizing the total user utilities in wireless networks with flows spanning one link each[11]. They specifically address the case where the user utilities are logarithmic and the output allocation is *proportionally fair.* They mention that if the utility function of every user is $-(-logx)^n$, where n is a positive integer, and x is a normalized bandwidth with $0 < x < 1$, then the output converges to maxmin fair allocation in the limiting case of infinitely large n. Now, the utility maximization schemes operate on successive update of bandwidth allocations[8, 11], and the magnitude of the updates may become very large, if n is very large. For example, if the bandwidth of a user is x, then x is updated by $\alpha - \frac{\beta x p(x)}{n(-logx)^{n-1}}$, where α, β are constants defining different system tradeoff, and $p(x)$ is the loss rate at bandwidth x. Now unless the attempt rates are very small (which leads to poor channel utilization), packet collision is inevitable, and there is always an associated non-negligible packet-loss. As n becomes arbitrarily large, $n(-logx)^{n-1}$ becomes very small if $x < 1$. Hence, since there is non-negligible loss rate, the updates become very large if x is close to 1. If the optimal value of the bandwidth is close to 1, then the updates are large even when the bandwidth allocation is close to the optimal value. Thus, there is rapid oscillation around the optimal point. So, it is not a good idea to use large n, but on the other hand the approximation is poor for small n. Consequently, we do not recommend utility based maximization approaches for attaining maxmin fairness, even though the utility maximization approach is ideally suited for attaining proportional fairness. Also, [9] suggests a technique for selecting the utility functions such that the output of the utility maximization in wireline networks is the maxmin fair allocation. However, in their approach, the appropriate utility functions can be selected *only when the maxmin fair allocation is known.* So, to use the approach of [9] we first need to compute the maxmin fair allocation, and then use a centralized policy to compute the utility functions, and finally use utility maximization to obtain the maxmin fair allocation. This is clearly not efficient. Luo *et. al.* explores a tradeoff between fairness and throughput maximization in [10]. Incidentally, the prioritized maxmin fairness can attain a similar tradeoff, by selecting the appropriate weights.

An alternate notion of fairness is considered in [10]. Here, every flow has a weight, and the network guarantees certain minimum bandwidth to every flow, the minimum bandwidth depending on the weights of the flows in a certain neighborhood of the flow. Subject to this minimum bandwidth allocation, the system utilization is maximized. While it is possible to obtain maxmin fairness, by appropriately selecting the weights for the flows in [10], again the appropriate weights can be computed *only after the maxmin rates are known.* So, a pre-computation phase is necessary if maxmin fairness is to be obtained via the weighted fairness approach of [10]. Finally, Ozugur *et. al.* presents a scheduling policy for fair allocation of rates in [12]. However, [12] do not attain fair allocations under a well-defined notion of fairness like maxmin fairness, proportional fairness, etc.

6 Conclusion

We summarize the contribution of this paper. We presented a formal study of maxmin fairness in wireless networks. We proposed a scheduling policy for attaining maxmin fairness. It can be shown analytically that this scheduling policy attains maxmin fair bandwidth allocation in the long-run and has good short term fairness properties as well. In addition, this policy can attain several notions of fairness, e.g., prioritized maxmin fairness etc. with minor modifications. Ongoing work is directed towards designing distributed scheduling policies. We are also considering fair bandwidth allocation for flows spanning several links. More generalized interference models with flows in a neighborhood interfering with each other is under investigation. It would also be interesting to study fairness notions in presence of packet loss due to channel error.

A Proof of Theorem 3

For simplicity, we will prove Theorem 3 under the assumption that all sessions have packets for transmission at all times. The generalization to the case when all sessions may not always have packets for transmission is straightforward.

We introduce some terminologies for this purpose. Consider a flow with end-points A, B respectively. See figure 1 below. Let $A_i(t)$, $B_i(t)$ be the number of tokens generated for end-points A, B respectively for session i in time interval $(0, t)$.

Notice that $A_i(t)$ differs from the actual number of tokens $A'_i(t)$ at A at time t, since the scheduling process will decrease the number of generated

Figure 1.

tokens in interval $(0, t)$ for both ends A and B. Notice also that the scheduling process will decrease the number of tokens in B whenever it decreases for A, and vice versa. Thus, $A_i(t) - B_i(t) = A'_i(t) - B'_i(t)$.

We state some lemmas which we use in the proof for Theorem 3.

Lemma 1 *Suppose that both ends of a session i get at least $rt - \gamma$ token chances in an interval of length t. Let W be the window size.*

If $W > \gamma$, $A_i(t)$ and $B_i(t)$ increases at least by $rt - 2\gamma$ in the same interval.

Proof

Note: For simplicity, we are going to omit index i for end-points A, B in the proof.

Consider a time interval (u_1, u_2).

The problem is divided in 2 cases:

Case 1:

Suppose that $|A(u) - B(u)| < W \; \forall \; u \; \epsilon \; (u_1, u_2)$. This also means that $A'(u) - B'(u) < W$.

Thus, every token chance is utilized to generate a token. We can deduce the following formulas:

$$A(u_2) \geq A(u_1) + r(u_2 - u_1) - \gamma \tag{3}$$

$$B(u_2) \geq B(u_1) + r(u_2 - u_1) - \gamma \tag{4}$$

The result follows from the conditions in the lemma.

Case 2:

Suppose that $|A(u) - B(u)| = W$ for some $u \; \epsilon \; (u_1, u_2)$.

Let u' be the least time at which this happens.

Without loss of generality consider:

$$A(u') = B(u') + W$$

- Let $B(u) = A(u) + W$ for some $u \in (u_1, u_2)$.

  Without loss of generality we have a run of $A(k') = B(k') + W$ and $B(k'') = A(k'') + W$, which represents the instants where one end is greater than the other by W tokens.

Let's examine the following sample path:

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

We explain it as follows: u_1 denotes the start time of the sample period. The symbol $A(B)$ at a time t denotes $A(t) = B(t) + W (B(t) = A(t) + W)$.

Note that A is greater than B by W in some time points in interval (u_1, t_1). B fully utilizes token chances in this interval.

$$B(t_1) - B(u_1) \geq r(t_1 - u_1) - \gamma \tag{5}$$

It also holds:

$$A(t_1) = B(t_1) + W \tag{6}$$

$$B(t_2) = A(t_2) + W \tag{7}$$

Arguing similarly, A utilizes all token chances in interval (t_1, t_2).

$$A(t_2) \geq A(t_1) + r(t_2 - t_1) - \gamma \tag{8}$$

$$(\ 7),\ (\ 8) \Rightarrow B(t_2) \geq A(t_1) + r(t_2 - t_1) - \gamma + W \tag{9}$$

$$(\ 6),\ (\ 9) \Rightarrow B(t_2) \geq B(t_1) + r(t_2 - t_1) - \gamma + 2W \tag{10}$$

$$(\ 5),\ (\ 10) \Rightarrow B(t_2) - B(u_1) \geq r(t_2 - u_1) + 2(W - \gamma) \tag{11}$$

Similar argument shows:

$$\begin{aligned} B(t_4) - B(t_2) &\geq r(t_4 - t_2) + 2(W - \gamma) && (12) \\ B(t_6) - B(t_4) &\geq r(t_6 - t_4) + 2(W - \gamma) && (13) \end{aligned}$$

$$\vdots$$

$$B(u_2) - B(t_l) \geq r(u_2 - t_l) - \gamma \tag{14}$$

We consider a generalization of the sample path, where t_l is the last time point where $B(t_l) = A(t_l) + W$.

$$(11),\ (12),\ (13),\ (14) \Rightarrow$$

$$B(u_2) - B(u_1) \geq r(u_2 - u_1) - \gamma + 2k(W - \gamma) \tag{15}$$

where k is defined as the number of times A sequences are followed by B sequences.

Assuming $W > \gamma \Rightarrow$

$$2k(W - \gamma) \geq 0 \tag{16}$$

$$\begin{aligned} (15),\ (16) \Rightarrow B(u_2) - B(u_1) &\geq r(u_2 - u_1) - \gamma \\ &\geq r(u_2 - u_1) - 2\gamma \end{aligned} \tag{17}$$

It must be shown that the lemma also holds for end-point A:

We have:

$$A(t_1) = B(t_1) + W$$

$$B(t_1) - B(u_1) \geq r(t_1 - u_1) - \gamma$$

This means:

$$\begin{aligned} A(t_1) &\geq B(u_1) + W + r(t_1 - u_1) - \gamma \\ &\geq A(u_1) - W + W + r(t_1 - u_1) - \gamma \\ &\geq A(u_1) + r(t_1 - u_1) - \gamma \end{aligned} \tag{18}$$

Following similar arguments with the proof of the lower bound of $B(t_2) - B(u_1)$ where A sequences are followed by B sequences (in this case B sequences are followed by A sequences) :

$$A(t_3) \geq A(t_1) + r(t_3 - t_1) + 2(W - \gamma)$$

$$A(t_5) \geq A(t_3) + r(t_5 - t_3) + 2(W - \gamma)$$

$$\vdots$$

$$A(t_k) \geq A(t_{k-2}) + r(t_k - t_{k-2}) + 2(W - \gamma)$$

Time point t_k is the last time point where $A(t_k) = B(t_k) + W$.

There are no A's between t_k and u_2. Thus:

$$A(u_2) \geq A(t_k) + r(u_2 - t_k) - \gamma$$

Summing up the above inequalities:

$$\Rightarrow A(u_2) \geq A(u_1) + r(u_2 - u_1) - 2\gamma + 2k(W - \gamma)$$

where k is defined as the number of times B sequences are followed by A sequences. Assuming $W > \gamma$,

$$\Rightarrow A(u_2) \geq A(u_1) + r(u_2 - u_1) - 2\gamma \tag{19}$$

Note that the case $B(u) < A(u) + W \;\forall\; u\; \epsilon(u_1, u_2)$ is incorporated in the previous case. Obviously, B always utilizes tokens in this interval. Thus,

$$B(u_2) - B(u_1) \geq r(u_2 - u_1) - \gamma$$

Now, we have a sequence of A's which is a special case of the previous paradigm. Following similar arguments as before, we can easily deduce that $A(u_2) \geq A(u_1) + r(u_2 - u_1) - 2\gamma$.

The result follows. □

Lemma 2 *Suppose that one end,B, of a session i gets at most* $r_B(t-s)+\gamma_B$ *token chances and its other end,A, gets at least* $r_A(t-s)-\gamma_A$ *token chances in any interval of length* (s,t) *in* (t_0,∞). *Assume that* $r_A > r_B$. *Then, there exists* $t_1 \geq t_0$ *such that* $A_i(t) - A_i(s)$ *and* $B_i(t) - B_i(s)$ *is upper bounded by* $r_B(t-s) + \gamma_B + \frac{\gamma_A+\gamma_B}{r_A-r_B}$ *in any interval* $(s,t) \in (t_1,\infty)$.

Proof

The case of $B_i(t)$ is simple. Since the number of token chances in B is upper bounded by $r_Bt + \gamma_B$, the same upper bound holds for the number of tokens generated in an interval of length t. This upper bound is smaller than $r_Bt + \gamma_B + \frac{\gamma_A+\gamma_B}{r_A-r_B}$.

We drop the index i as before for simplicity, and will refer to $A_i(t)$ as $A(t)$ and $B_i(t)$ as $B(t)$. We will first show that $A(t) = B(t) + W$ infinitely often in the interval (t_0, ∞). Using this fact, we will show that there exists a time $t_1 \geq t_0$ s.t. $A(t) \geq B(t) + W - \frac{\gamma_A+\gamma_B}{r_A-r_B}$ for all $t \in (t_1, \infty)$. Since $A(t) \leq B(t)+W$ at any time t, it would follow that in any interval of length τ the difference between the increase in $A(t)$ and $B(t)$ is upper bounded by $r_B\tau + \gamma_B + \frac{\gamma_A+\gamma_B}{r_A-r_B}$. We proceed to show that $A(t) = B(t) + W$ infinitely often in the interval (t_0, ∞).

Let there exist $t_2 \geq t_0$ s.t. $A(t) < B(t) + W \ \forall\, t \geq t_2$. Consider any time t_3 in (t_2, ∞). This implies that A utilizes every token chance it gets during (t_2, t_3):

$$A(t_3) \geq A(t_2) + r_A(t_3 - t_2) - \gamma_A$$

The following upper bound holds for B:

$$B(t_3) \leq B(t_2) + r_B(t_3 - t_2) + \gamma_B$$

From the above relations,

$$A(t_3) - B(t_3) \geq A(t_2) - B(t_2) + (r_A - r_B)(t_3 - t_2) - (\gamma_A + \gamma_B)$$

Using the fact that $A(t_2) - B(t_2) \geq -W$,

$$A(t_3) - B(t_3) \geq -W + (r_A - r_B)(t_3 - t_2) - (\gamma_A + \gamma_B)$$

Since $A(t_3) < B(t_3) + W$,

$$\begin{aligned} W &> -W + (r_A - r_B)(t_3 - t_2) - (\gamma_A + \gamma_B) \\ \Rightarrow t_3 &< t_2 + \frac{2W + \gamma_A + \gamma_B}{r_A - r_B} \end{aligned}$$

Thus, if $A(t) < B(t) + W$ for all t in any interval (t_2, t_3) in (t_0, ∞), then $t_3 < t_2 + \frac{2W+\gamma_A+\gamma_B}{r_A-r_B}$. It follows that $A(t) = B(t) + W$ infinitely often in (t_0, ∞). Let t_1 be the first time t in (t_0, ∞) s.t. $A(t) = B(t) + W$.

Next, we will show that $A(t) \geq B(t) + W - \frac{\gamma_A+\gamma_B}{r_A-r_B}$ for all $t \in (t_1, \infty)$. Let $A(t_3) < B(t_3) + W$ at some $t_3 \in (t_1, \infty)$. There exists $t_2 \in (t_1, t_3)$ s.t. $A(t_2) = B(t_2) + W$ by the definition of t_1. Consider t_2 to be the last such time. So, $A(t) < B(t) + W$ for all $t \in (t_2, t_3]$.

$$\begin{aligned} \text{Thus } A(t_3) - A(t_2) &< B(t_3) + W - B(t_2) - W \\ &\leq r_B(t_3 - t_2) + \gamma_B \qquad (20) \end{aligned}$$

During this interval, A utilizes every token chance since $A(t) < B(t)+W$.

$$\text{Thus } A(t_3) - A(t_2) \geq r_A(t_3 - t_2) - \gamma_A \qquad (21)$$

From (20) and (21)

$$\begin{aligned}
r_A(t_3 - t_2) - \gamma_A &< r_B(t_3 - t_2) + \gamma_B && (22)\\
\text{Thus } t_3 - t_2 &< \frac{\gamma_A + \gamma_B}{r_A - r_B}\\
A(t_3) &\geq A(t_2) \text{ from definition}\\
B(t_3) &\leq B(t_2) + \frac{\gamma_A + \gamma_B}{r_A - r_B} \text{ from (23)}\\
\text{Thus } A(t_3) - B(t_3) &\geq A(t_2) - B(t_2) - \frac{\gamma_A + \gamma_B}{r_A - r_B}\\
&= W - \frac{\gamma_A + \gamma_B}{r_A - r_B} && (23)
\end{aligned}$$

Thus, we showed that $A(t_3) - B(t_3) \geq W - \frac{\gamma_A + \gamma_B}{r_A - r_B}$ for all t_3 in (t_1, ∞) such that $A(t_3) < B(t_3) + W$. It follows that $A(t) \geq B(t) + W$ for all $t \in (t_1, \infty)$.

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

$$\begin{aligned}
A(s) &\geq B(s) + W - \frac{\gamma_A + \gamma_B}{r_A - r_B}\\
A(t) &\leq B(t) + W\\
A(t) - A(s) &\leq B(t) - B(s) + \frac{\gamma_A + \gamma_B}{r_A - r_B}\\
&\leq r_B(t - s) + \gamma_B + \frac{\gamma_A + \gamma_B}{r_A - r_B}
\end{aligned}$$

The result follows. □

Let the maxmin fair rate of session i be r_i. Recall that α is the sampling rate for any node.

Theorem 5 *Let $P_{in}(s, t)$ be the total number of tokens generated at end-point n of session i during interval (s, t). Let*

$$\begin{aligned}
\gamma_0 &= 0\\
\beta_0 &= 0\\
\gamma_k &= (N - 1)\beta_{k-1} + 1\\
\beta_k &= 2N\gamma_k + \frac{2N\gamma_k + \gamma_k}{p_k - r_k}
\end{aligned}$$

The term p_k will be defined in the proof, and depend on the network topology only. Suppose $W > \gamma_N$. For each session i, $|P_{i_n}(s, t) - r_i(t - s)| \leq \max(\beta_N, \gamma_N)$, in any interval (s, t).

Proof of Theorem 4 We argue as follows:

Rank the sessions into increasing order of their max-min fair shares. Denote these shares by $r_1, r_2, \ldots, r_N$.

We first prove the following:

For every k, $\exists$ time t_k such that:

1. (a) Any session i of rank greater than or equal to k, is sampled at least $r_k(t-s) - \gamma_k$ times at both ends in any interval (s,t) in (t_k, ∞).

   (b) Let node n not be a bottleneck node for session i. There exists a number $p_k > r_k$ which satisfies the following property. If session i has rank greater than or equal to k, node n samples session i at least $p_k(t-s) - \gamma_k$ times in any interval (s,t) in (t_k, ∞).

2. For each session i of rank $\geq k$, $P_{i_n}(s,t) \geq r_k(t-s) - 2\gamma_k$ in any interval (s,t) in (t_k, ∞).

3. For each session i of rank k, $P_{i_n}(s,t) \leq r_k(t-s) + \beta_k$ in any interval (s,t) in (t_{k+1}, ∞).

The proof is by induction on the rank k of the sessions, for the 3 statements above.

Base case : Rank $k = 1$

Statement 1

*(*a) Consider a session of rank equal to 1. Such a session has at least one end-point of max-degree d in the graph. The max-min fair rate r_1 of this session equals the sampling rate of tokens it is given to its bottleneck node by the round-robin scheduling. More precisely, $r_1 = \frac{\alpha}{d}$.

Every node n with degree d_n ($d_n \leq d$) offers at least $(\frac{\alpha}{d_n})(t-s) - 1$ token chances in any interval (s,t), to each of its sessions. Since the sampling rate for a session is at least the minimum number of token chances that are offered at both ends, this rate is lower bounded by $\frac{\alpha}{d} = r_1$, as $d_n \leq d$. □

*(*b) If a session is not bottlenecked at a node n then $d_n < d$. Thus $\alpha/d_n > \alpha/d$. Statement $1(b)$ follows with $p_1 = \frac{\alpha}{max_{n:d_n<d} d_n}$.

Statement 2

In the base case of the previous statement (a), it was proved that all sessions of rank equal to 1 and above, get at least $r_1(t-s) - 1$ token chances at both ends in any interval (s,t) in (t_1, ∞). Using Lemma 1, it holds that $P_{i_n}(s,t)$ for all sessions of rank ≥ 1 increases at least by $r_1(t-s) - 2$ in any interval (s,t) in (t_1, ∞).

Statement 3

A session of rank 1 has at least one end-point of max-degree in the graph. A node of maximum degree samples any session traversing it at most $\frac{\alpha}{d}(t-s)+1$ times in any interval (s,t), and is the bottleneck node for each session. We have also seen that $r_1 = \alpha/d$. If both end points have maximum degree, then the result follows from Lemma 2. Otherwise, one end-point is not a bottleneck node and according to the base case of statement $1(b)$, it gets at least $p_1(t-s) - \gamma_1$ token chances in any interval (s,t). Thus, the result follows from Lemma 2.

Induction hypothesis: Let the 3 statements hold for ranks $1, 2, \ldots k-1$.

The induction step will be proved for rank k.

First consider *Statement* $1(a)$:

Let Y be the set of sessions y traversing node n with rank less than k, and Z is the set of sessions traversing node n not included in Y. Let x be any session with rank $\geq k$.

Let q be the number of token chances in (s,t) in (t_k, ∞) that are not used by sessions in Y. Let $C_x(s,t)$ denote the number of times session x is sampled at node n in interval (s,t).

$$q = \alpha(t-s) - \sum_{y \in Y} P_{yn}(s,t) \tag{24}$$

These sessions in Z get at least q token chances in interval (s,t).

$$q \leq \sum_{z \epsilon Z} C_z(s,t) \leq |Z| \; max_{z \in Z} C_z(s,t) \tag{25}$$

According to round-robin scheduling, a session x in Z will receive as many token chances as any other session over a node n during (s,t):

$$C_{xn}(s,t) \geq \max_{z \epsilon Z} C_{zn}(s,t) - 1 \tag{26}$$

$$\text{From } (25), (26) \;\; C_x(s,t) \quad \geq \quad \frac{1}{|Z|} q - 1$$

$$\text{From } (24), \;\; C_x(s,t) \quad \geq \quad \frac{1}{|Z|}[\alpha(t-s) - \sum_{y \epsilon Y} P_{y_n}(s,t)] - 1$$

By the induction hypothesis, part 3 of Theorem 1 holds for $1, \ldots, k-1$. Let $q(y)$ denote the rank of session y.

$$\begin{aligned}
C_x(s,t) &\geq \frac{1}{|Z|}[\alpha(t-s) - \sum_{y \epsilon Y}(r_y(t-s) + \beta_{q(y)})] - 1 \\
&= \frac{1}{|Z|}(t-s)(\alpha - \sum_{y \epsilon Y} r_y) - \frac{1}{|Z|}\sum_{y \epsilon Y}\beta_{q(y)} - 1 \\
&\geq \frac{1}{|Z|}(t-s)\sum_{z \epsilon Z} r_z - (\frac{1}{|Z|}\sum_{y \epsilon Y}\beta_{q(y)} + 1) \qquad (27) \\
&\geq \frac{1}{|Z|}(t-s)\sum_{z \epsilon Z} r_k - (\frac{1}{|Z|}\sum_{y \epsilon Y}\beta_{q(y)} + 1) \qquad (28) \\
&= r_k(t-s) - (\frac{1}{|Z|}\sum_{y \epsilon Y}\beta_{q(y)} + 1) \\
&\geq r_k(t-s) - (\frac{|Y|}{|Z|}\beta_{k-1} + 1) \\
&\geq r_k(t-s) - ((N-1)\beta_{k-1} + 1) \\
&= r_k(t-s) - \gamma_k
\end{aligned}$$

Now we prove statement $1(b)$. Let n not be the bottleneck node of session x. Either (27) or (28) or both are strict inequalities in this case. Thus statement $1(b)$ follows with $p_k = \min(\frac{\alpha - \sum_{y \in Y} r_y}{|Z|}$, where the minimum is taken over all nodes which have sessions of rank k or greater and is not the bottleneck of at least one such node.

Statement 2:

It follows directly from Lemma 1 and Statement 1. It has been proved in Lemma 1 that if both ends of a session i get at least $r_k t - \gamma_1$ token chances in any interval of length t, then $P_{i_n}(t)$ increases by at least $r_k t - 2\gamma$ in any interval of length t. Statement 1 also has been proved. Thus, the proof follows from these results.

Statement 3:

It is known that under a maxmin fair allocation scheme, every session has at least one *bottleneck node*, where the sum of the rates of the sessions over that node equals its capacity α, and the rate of that session is greater than or equal to all other sessions traversing that node.

Let x be any session with rank k. Let node n be the bottleneck node (in case of two bottleneck nodes, one of them is selected arbitrarily). Let Y

denote the set of sessions $y \neq x$ traversing node n. Thus all sessions in Y have rank at most k. It holds:

$$P_{x_n}(t_1, t_2) \leq \alpha(t_2 - t_1) - \sum_{y \epsilon Y} P_{y_n}(t_1, t_2)$$

From Statement 2, $P_{y_n}(t_1, t_2) \geq r_y(t_2 - t_1) - 2\gamma_{p(y)}$.
From these 2 relations,

$$P_{x_n}(t_1, t_2) \leq \alpha(t_2 - t_1) - \sum_{y \epsilon Y} (r_y(t_2 - t_1) - 2\gamma_{p(y)})$$

$$\Rightarrow P_{x_n}(t_1, t_2) \leq (\alpha - \sum_{y \epsilon Y} r_y)(t_2 - t_1) + \sum_{y \epsilon Y} 2\gamma_{p(y)} \tag{29}$$

Node n is a bottleneck node:

$$\sum_{k \ \epsilon \ n} r_k = \alpha$$

$$\Rightarrow r_x + \sum_{y \epsilon Y} r_y = \alpha \tag{30}$$

$$\begin{aligned} (29), (30) \Rightarrow P_{x_n}(t_1, t_2) &\leq r_x(t_2 - t_1) + \sum_{y \epsilon Y} 2\gamma_{p(y)} \\ &\leq r_x(t_2 - t_1) + 2|Y|\gamma_k \\ &\leq r_x(t_2 - t_1) + 2(N - 1)\gamma_k \qquad (31) \end{aligned}$$

In case that both nodes of session x are bottlenecks, statement 3 follows from (31).

If the other node of session x is not a bottleneck node, according to statement $1(b)$ the number of token chances it gets is at least $p_k(t - s) - \gamma_k$ in an interval (s, t), $p_k > r_k$. Then using (31) and Lemma 2

$$\begin{aligned} P_{xn} &\leq r_k(t - s) + 2N\gamma_k + \frac{2N\gamma_k + \gamma_k}{p_k - r_k} \\ &\leq r_k(t - s) + \beta_k \end{aligned}$$

Theorem 3 follows from Theorem 5.

References

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," *Proceedings of ACM Sigcomm,* September, 1994

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

[3] R. Garces and J.J. Garcia-Luna-Aceves, "Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks", *Proceedings of IEEE INFOCOM 2000,* Tel-Aviv, 2000

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

[5] J.J. Garcia-Luna-Aceves and J. Raju, "Distributed Assignment of Codes for Multihop Packet-Radio Networks", *Proceedings of MILCOM, 1997,* Monterey, California, 1997.

[6] 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 $1024 - 1039$.

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

[8] S. Kunniyur, R. Srikant: "End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks *Proceedings of INFOCOM 2000*, Tel-Aviv, 2000

[9] S. Low and D. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence, *IEEE/ACM Transactions on Networking,* 7(6), Dec. 1999.

[10] H. Luo, S. Lu, V. Bhargh avan, " A New Model for Packet Scheduling in Multihop Wireless Networks." ACM Mobicom 2000, Boston, MA, August 2000.

[11] T. Nandagopal, T. Kim, X. Gao and V. Bharghavan, " Achieving MAC Layer Fairness in Wireless Packet Networks." *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,* October 1998.

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

[14] 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.

[15] 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.