Scheduling for Quality of Service based Service Differentiation

Saswati Sarkar[1] and Leandros Tassiulas[2]
[1]Department of Electrical Engineering, University of Pennsylvania, swati@ee.upenn.edu
[2]Department of Electrical and Computer Engineering and Institute for Systems Research University of Maryland, College Park, leandros@isr.umd.edu

*Address all correspondence to Saswati Sarkar

We consider a utility based fairness model for service differentiation among different users. Every user has a utility function and the network allocates bandwidth so as to obtain a fair allocation of the utilities. We present a scheduling policy which attains a provably fair utility allocation. The policy uses a combination of differential sampling, estimation and back pressure based congestion control in order to attain a globally fair allocation via local information based scheduling.

1. Introduction

Service differentiation is a major feature of modern networks. The quality of service received by a user depends on his requirement, the revenue he generates and also the congestion of the network. It is imperative that a service differentiation scheme be associated with a notion of fairness. The underlying principle should be that if two users have similar requirements, revenues and path congestion, then they should receive the same quality of service, and at the same time different values of these quantities should result in different service allocations.

A utility based bandwidth allocation can accommodate this basic principle. Every user associates a value or "utility" with the received bandwidth. The network distributes the bandwidth so as to obtain a fair allocation of utilities. Thus the utilities of the users and the congestion level in the network determine the bandwidth allocation. The network would charge the users in accordance with the utility functions, and the user and the service provider may actually negotiate the utility function before service initiation. The choice of the utility functions and the link capacities together determine the bandwidth allocation of the users. The problem we address in this paper is to obtain a fair allocation of utilities in an arbitrary network subject to network congestion.

We adopt the notion of maxmin fairness. A utility allocation is said to be maxmin fair, if it is not possible to increase the utility of a user without decreasing that of another user which has equal or lower utility[1]. Figure 1 shows an example maxmin fair utility allocation. In this notion of fairness, every link first tries to equalize the user utilities. However, some users can not utilize the offered bandwidth because of bandwidth constraint in other links. In this case, the link distributes the residual bandwidth among the remaining contenders in proportion of their utilities. The objective of this paper is to provide a bandwidth provisioning scheme for attaining the maxmin fair allocation of utilities. We consider general utility functions and do not restrict ourselves to linear or concave utility functions only.

One possibility is to first compute the bandwidth shares which attain the fair utilities and subsequently determine the service order for the packets for allocating the computed bandwidth. Algorithms for computing the requisite bandwidth have been presented in [2, 9]. Now, there are many scheduling policies which can attain any feasible rate allocation, once the rates are known, e.g., fair queuing strategies[3]. But, these scheduling policies work only when the link schedulers know the desired rates, or at least a ratio between the desired rates, and hence using these would involve a pre-computation phase. The two tier approach increases the delay in the overall allocation. The contribution of this paper is to present a scheduling policy which attains the fair utilities without computing the bandwidth beforehand.

Clearly, the fair allocation of utilities is a global optimization problem. This is because fair allocation of resources in a link would depend on the congestion of the other links as well. Consider the network shown in figure 1. Intuitively fair resource allocation in link e_1 would be to equalize the utilities of all 4 sessions and offer bandwidth $4, 3, 2, 1$ to sessions $1, 2, 3, 4$ respectively. However, session 1 can not use more than 1 unit on account of link e_4. So the fair allocation would be $3, 4, 2$ units for sessions $2, 3$ and 4 respectively. Thus the fair allocation in any link can only be determined with the congestion information in other links. The challenge is to obtain a distributed strategy which attains the global optimum. The policy we propose takes scheduling decisions based on local information only, but attains a globally fair allocation of utilities.

We would describe the scheduling policy in greater details later. The basic idea is to sample the sessions in a certain order which depends on the utilities. When a session is sampled, it may or may not transmit a packet. The decision is based on the availability of packets for transmission and the congestion downstream. For example, in Figure 1, session 1 decides to transmit a packet in link e_1 based on the congestion in link e_2 and availability of session 1 packets. The sampling order is crucial in attaining the maxmin fair allocation of utilities. A simple weighted round robin sampling with weights based on the utility functions does not attain the fair allocation of the utilities. We discuss this in details later.

We conclude this section with a brief review of prior work in utility based fairness. Cao *et. al.* presents an algorithm for computing the bandwidth required to attain a maxmin fair allocation of utilities in [2]. Sarkar *et. al.* addresses the same problem in context of multirate, multicast networks in [9]. An alternate notion of fairness is to allocate bandwidth so as to maximize the sum of the total user utilities. Recently, [5,7,8] proposed distributed algorithms for computing the bandwidth required to maximize the sum of the user utilities in unicast networks and [6] addressed this problem in context of multirate, multicast networks. Our objective in this paper is to provide scheduling policy for attaining maxmin fair utilities in networks. This problem has not been addressed before in unicast or multicast networks. In this paper we will address the unicast case only. However, a generalization of this policy attains the same objective in multirate, multicast networks as well. Refer to technical report [11] for details on the scheduling policy for multicast networks.

This paper is organized as follows. We describe the network model and present our fairness objectives in Section 2. We describe the intuition behind our approach in Section 3. We present our scheduling policy in Section 4. We present our analytical results in Section 5 and conclude in Section 6.

2. Network Model and Fairness Objectives

A session or a data connection is identified by the pair (v, u), where v is the source node of the session and u is the receiver node. We assume that there is a predetermined route between the source and the destination, and do not consider the routing problem in this paper. Let there be N sessions. Rate allocation is an N-dimensional vector $(r_1, \ldots, r_N)$ where r_i is the rate allocated to the ith session. A rate allocation $(r_1, \ldots, r_N)$ is feasible if every link satisfies the capacity condition. The capacity condition is that the total bandwidth consumed by all sessions traversing a link does not exceed the capacity of the link. Formally, for every link l, $\sum_{j \in n(l)} r_j \leq C_l$, where $n(l)$ denotes the set of sessions passing through link l and C_l denotes the capacity of link l. In addition, there is a maximum rate constraint, which requires that $r_i \leq p_i$, where p_i is the maximum rate for session i. Figure 1 shows some capacity and maximum rate constraints.

Every session has a utility function. The utility function for session i is denoted as $U_i(x)$. We assume that a utility function is nonnegative, continuous and strictly increasing. Note that we allow nonlinear and non-concave utility functions. Utility allocation is a N-dimensional vector $(u_1, \ldots, u_N)$, where $u_j = U_j(r_j)$ is the utility allocated to session j. Under the monotonicity assumption on the utility functions, there is a one to one correspondence between a utility allocation vector and a rate allocation. A utility allocation is feasible if and only if it corresponds to a feasible rate allocation.

A feasible utility allocation is maxmin fair if it is not possible to maintain feasibility and increase the utility of a session without decreasing that of any other session which has equal or lower utility. More formally, a feasible utility allocation $\vec{u}^1$ is maxmin fair if it satisfies the following property with respect to any other feasible utility allocation $\vec{u}^2$: if there exists i such that the ith component of $\vec{u}^2$ is strictly greater than that of $\vec{u}^1$ $(u_i^2 > u_i^1)$, then there exists j such that the jth component of $\vec{u}^1$, u_j^1 is less than or equal to the ith component of $\vec{u}^1$, u_i^1 $(u_j^1 \leq u_i^1)$ and the jth component of $\vec{u}^2$ (u_j^2) is strictly less than the jth component of $\vec{u}^1$ $(u_j^2 < u_j^1)$. Refer to the network of figure 1 for an example of maxmin fair utility allocation.

We present the intuition behind our approach in the next section.

3. Back pressure based scheduling for fairness

If the utility of a bandwidth is the bandwidth itself, then maxmin fair allocation of utility attains maxmin fair allocation of bandwidth. Hahne proposed a scheduling policy for attaining maxmin fair bandwidth in [4]. We first discuss the scheduling presented in [4] and then point out its deficiencies in the utility fairness framework, and subsequently present our approach in obtaining a fair allocation of utilities.

In maxmin fair bandwidth allocation, a link offers equal bandwidth to every session traversing the link. If a session can not utilize the bandwidth offered by the link because of constraint in other links, the residual bandwidth is distributed equally among the unconstrained contenders. Thus the scheduling has two components. The first is to

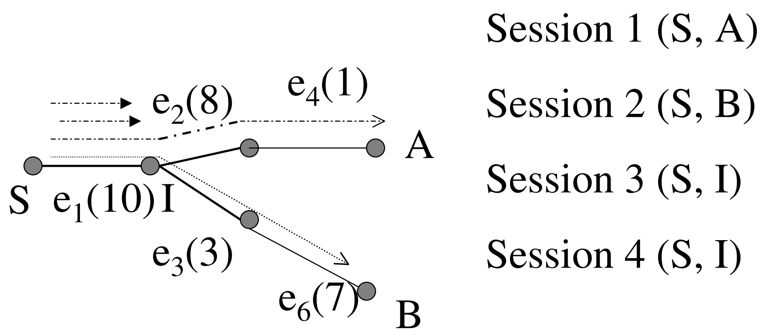

Figure 1. A session is represented by a source destination pair in the figure. The numbers in the brackets () denote the capacities of the respective links. For example, e_1 has capacity 10 units. The utility functions are given by $U_1(x) = x/4$, $U_2(x) = x/3$, $U_3(x) = x/2$ and $U_4(x) = x$. The capacity constraint for link e_1 is is $r_1 + r_2 + r_3 + r_4 \leq 10$. The maximum rate constraint for session 1 is $r_1 \leq 5$. The maxmin fair utility allocation is $(0.25, 1, 2, 2)$. Sessions 1 and 2 are constrained in links e_4 and e_3 respectively. The remaining bandwidth in link e_1 is split between sessions 3 and 4 so as to equalize their utilities.

offer equal bandwidth to all sessions, and the second is to reduce the bandwidth of the constrained sessions and distribute it among the unconstrained ones. The first can be attained by a simple round robin scheduling in every link, and the second can be attained by a back pressure based flow control. Back pressure based flow control decides whether or not to transmit a packet when a session is sampled, depending on the congestion experienced by the session in its downstream links. If the downstream link is congested when a session is sampled, the session does not transmit a packet and passes its chance to the next session in the round robin order. For example, if session 1 experiences congestion in link e_2 in Figure 1 then session 1 does not transmit a packet in e_1 even when it is sampled. The congestion condition is inferred from the queue lengths at the corresponding nodes. Typically, if the queue lengths exceed a certain threshold, then the links are considered congested. Congestion conditions of links e_2 and e_3 can be inferred from the queue lengths at node I in Figure 1. Summarizing, sessions are sampled in round robin order in every link. When a session is sampled, it may or may not transmit a packet. The decision is based on the congestion of the downstream links.

For fair allocation of utilities, bandwidth offered to sessions in a link should be such that their utilities are equalized. If a session can not utilize the offered bandwidth because of constraint in other links, then the residual bandwidth should be distributed so as to equalize the utilities of the unconstrained sessions. Thus different amount of bandwidth need to be offered to different sessions in the first place, and a round robin sampling can not attain this. An intuitively appealing approach is to use a weighted round robin sampling. However, the determination of the weights is not obvious. Intuitively, if $b_1, b_2, \ldots, b_p$ are the required bandwidth of sessions $1, 2, \ldots, p$ sharing a link l, then the weights of the sessions $w_1, w_2, \ldots, w_p$ should satisfy $w_1 : w_2 : \ldots : w_p = b_1 : b_2 : \ldots : b_p$. The complication is that the ratio between the bandwidth attaining the maxmin fair utilities is not known since there is no computation phase. These bandwidth depend on the utility functions and the congestion condition condition of the bottleneck links of the sessions ("bottleneck link" of a session is the most congested link on its path). The simple back pressure based round robin scheduling of [4] does not detect the bottleneck congestion of the sessions.

We use an estimation procedure to detect the bottleneck conditions. We use a back pressure based weighted scheduling, where weights of the sessions are adjusted in accordance with the output of the estimation procedure. Initially, all sessions are sampled in a weighted round robin order, where the weights are ratios of the bandwidth required to equalize their utilities. A session sends a packet when it is sampled if it has a packet to send, and if the number of packets of the session at the destination node of a link is less than a threshold W. The violation of the former condition detects congestion upstream (closer to the source) and the violation of the latter condition detects congestion downstream (closer to the destination). The sessions which are constrained elsewhere can not utilize many of their slots because of one or both of the above transmission conditions. If the bottleneck link is downstream, then the destination node often has W packets and the session misses its chance. If the bottleneck link is upstream, then the session often does not have packets to send and thus misses its chance. The sessions which do not utilize many of their chances receive less utility than others. These sessions are considered bottlenecked. The bottleneck bandwidth of such a session is estimated to be the bandwidth it utilized so far. The residual bandwidth is computed from this estimate, and the shares of the non-bottlenecked sessions are determined. Now, the slots missed by the bottlenecked sessions are offered to the non-bottlenecked ones in proportion of the computed bandwidth shares. This is in addition to the regular slots offered to these sessions. Incidentally, we have observed that a back pressure based weighted round robin scheme with static configuration of the weights does not attain the fair allocation of utilities, for nonlinear utility functions. Technical report[11] presents an example scenario where an intuitive setting of the weights fails to allocate the maxmin fair utilities.

Note that estimation error is always possible. For example, a session may not be constrained but may not temporarily utilize its offered share on account of some transients. In this case, the estimation procedure will incorrectly conclude the session to be bottlenecked. Also, a session may be actually constrained, but again may still utilize more bandwidth than its bottleneck link offers in certain intervals on account of low utilization of other sessions in the bottleneck link. In this case, the estimation procedure will conclude that the session is not bottlenecked, even if it is actually bottlenecked. Again, the bottleneck bandwidth may be different from the utilized bandwidth on account of transients, and this also causes estimation error. So, the estimations are repeated at regular intervals, and the weights are recomputed if the estimation changes. Again this gives rise to the possibility of oscillation. The challenge is to develop an estimation procedure which recovers from the errors and leads to the maxmin fair allocation of utilities. We present such an estimation procedure in the next section.

The scheduling policy needs to operate in tandem with the estimation procedure. It is important to design a scheduling policy which allows recovery from estimation errors. The following scenario shows that it is often nontrivial to design such a policy. Consider figure 1. Assume that the capacity of link e_3 is 10 units instead of 3 units as shown in the figure. The maxmin fair utility allocation is $(0.25, 1.5, 1.5, 1.5)$. Session 1 is constrained by link e_4. The remaining bandwidth of link e_1 is split among the remaining three sessions. All of these can utilize the offered bandwidth. The corresponding bandwidth allocation is $(1, 4.5, 3, 1.5)$. Suppose, the estimation procedure incorrectly concludes that session 1 is bottlenecked with a bottleneck bandwidth of 0.5 unit. In reality, the bottleneck bandwidth is 1 unit. Subsequently, session 1 is sampled at the rate 0.5 in link e_1, and the remaining bandwidth is split between the other three sessions. Session 1 is sampled at rates 8 and 1 units respectively in links e_2 and e_4, and receives an overall bandwidth of 0.5 unit from the path. As a result, its utility is 0.125. The other three sessions receive a utility of 19/12 each. Thus all subsequent estimations indicate that session 1 has the least utility amongst all the sessions in link e_1 and it is bottlenecked at 0.5 unit of bandwidth. The same sampling proportions continue as before, and the utility allocation converges to $(1/8, 19/12, 19/12, 19/12)$, whereas the maxmin fair utility allocation is $(1/4, 3/2, 3/2, 3/2)$. Thus, the system never recovers from the initial estimation error. To avoid such a scenario, one should never decrease the bandwidth offered to a session below that required to equalize its utility with the other sessions, even if the estimation indicates it to be bottlenecked at a lower bandwidth. Rather, other sessions should be offered transmission opportunity in additional slots only when a bottlenecked session misses these slot (a slot is missed by a session if it does not send a packet when it is sampled in the slot). Thus, session 1 would continue to be sampled at rates respectively $4, 8$ and 1 in links e_1, e_2 and e_4 even though the erroneous estimation indicates that it is bottlenecked at 0.5 units of bandwidth. It can not utilize more than 1 unit. So, its attained utility converges to 0.25, and it misses many of its assigned slots in link e_1 on account of the congestion at link e_4. When session 1 misses a slot, one of the other three sessions are sampled in the slot. The share of these sessions depend on their utility functions. Thus these sessions are sampled at the rates $4.5, 3, 1.5$ respectively in link e_1. Thus the sessions $1, 2, 3$ and 4 receive utilities $(1/4, 3/2, 3/2, 3/2)$ respectively, and these are the maxmin fair utilities. We present the details of the policy in the next section.

4. Description of the Policy

The scheduling consists of two steps. One is the sampling procedure of sessions in a link, and the second is the actual packet transmission procedure when a session is sampled. The link scheduler samples the sessions for packet transmission in accordance with the sampling procedure. When a session is sampled, it decides whether or not to transmit a packet according to the packet transmission procedure. We first describe the packet transmission procedure. Also, the policy has two parameters, window parameter, W and the group granularity parameter ϵ. We discuss the choice of these parameters later.

Packet Transmission Decision:

A predefined window parameter W is selected apriori. When a session i is sampled in a link l at time t it transmits a packet if both the following conditions are satisfied:
(a) there are session i packets waiting for transmission in link l and
(b) the number of session i packets waiting at the destination node of link l is less than

W.

If a session can not send a packet, it passes its chance to a different session, which is determined by the sampling procedure. We assume that the number of packets waiting at the destination of the session is always 0, i.e., a packet is removed from the network as soon as it reaches its destination. This transmission policy ensures that the difference between the number of packets of a session transmitted in any two links on its path differs by a constant independent of the size of the interval. Thus the long term transmission rate for a session in any two links on its path are equal. Hence, the overall transmission rate for any session is determined by that of the most congested link on its path.

Sampling Procedure:

We describe the general sampling procedure here. Sessions are grouped depending on the average bandwidth utilization, and sessions of different groups are sampled at different rates. We describe the grouping first.

The average bandwidth utilization of the individual sessions in any link are measured at regular intervals. The measurement epochs are called the estimation epochs. Let the average bandwidth utilization of a session i in link l between times 0 and estimation epoch t be $\hat{r}_{i,l}(t)$. The utility received by a session is the utility of its average bandwidth. The sessions are freshly grouped in each link at every estimation epoch t. At $t = 0$, all sessions are in group 1 in every link. Consider link l and estimation epoch t. The sessions with minimum utility in link l constitute group 1, $G_l(1, t)$. This group also includes sessions which have utility within ϵ of the minimum utility. Here, ϵ is the group granularity parameter. The ith group, $G_l(i, t)$ consists of sessions which receive the ith minimum utility among the sessions traversing link l. More precisely, this group consists of all sessions which receive the minimum utility or at most ϵ more than the minimum utility where the minimum is taken in a set excluding the members of groups $G_l(1, t), \ldots, G_l(i-1, t)$. Clearly, the grouping changes from time to time depending on the bandwidth consumption. Note that ϵ affects the group granularity. If ϵ is large, then sessions with widely different utilities will be in the same group and hence treated similarly. If ϵ is too small, then at least initially there will be frequent group changes because of the estimation errors. We discuss the choice of ϵ later.

Now, we describe the bandwidth share of different groups. We introduce a notation here. The utility offered by link l to sessions in group j at time t is denoted as $\phi_l(j, t)$. The bandwidth share of session i in group j is $U_i^{-1}(\phi_l(j, t))$, where $\phi_l(j, t)$ can be computed as follows:

$$\sum_{i \in n(l)} U_i^{-1}(\phi_l(1, t)) = C_l \tag{1}$$

$$\sum_{i \in n(l) \setminus \cup_{k=1}^{j-1} G_l(k,t)} U_i^{-1}(\phi_l(j, t)) = C_l - \sum_{i \in \cup_{k=1}^{j-1} G_l(k,t)} \hat{r_{i,l}}(t), j > 1 \tag{2}$$

Next, we discuss how the sampling reflects these bandwidth shares. For this, we introduce the notion of the level of a slot. Initially, level of every slot is 1. Any session i is sampled at the rate $U_i^{-1}(\phi_l(1, t))$ in the level 1 slots irrespective of the grouping. For this purpose, the scheduling computes integers $x_{il1}(t)$ for each session i, where $x_{il1}(t)/x_{jl1}(t) = U_i^{-1}(\phi_l(1, t)) / U_j^{-1}(\phi_l(1, t))$. All slots are divided in frames of length

$\sum_{i \in n(l)} x_{il1}(t)$ and session i is sampled in $x_{il1}(t)$ slots in each frame[†].

If a group j session is sampled in a slot, and if it does not transmit a packet in the slot, then the level of the slot is updated to $j+1$. We clarify that the level of a slot changes as more and more sessions are sampled in the slot. A session i can be sampled in a slot if the sessions who have been offered the slot before i do not utilize the slot. Thus, before any session is sampled in a slot, the slot has level 1. Then, if the first session sampled in a slot is from group j and the session does not transmit any packet in the slot, the level of the slot is updated to $j+1$. Subsequently, if a session in group $j+4$ is sampled in the slot, and it does not utilize the slot either, then the level of the slot becomes $j+5$, etc.

Only sessions which are in groups higher than j can be sampled in a level $j+1$ slot. Any such session i is sampled at the rate $U_i^{-1}(\phi_l(j+1,t)) - U_i^{-1}(\phi_l(j,t))$ in level $j+1$ slots. For this purpose, the scheduling computes integers $x_{ilj+1}(t)$ for each session i in group $j+1$ or above, where $x_{ilj+1}(t)/x_{klj+1}(t) = U_i^{-1}(\phi_l(j+1,t)) - U_i^{-1}(\phi_l(j,t)) / U_k^{-1}(\phi_l(j+1,t)) - U_k^{-1}(\phi_l(j,t))$. The level $j+1$ slots are divided in frames of length $\sum_{i \in n(l) \setminus \cup_{k=1}^{j} G_l(k,t)} x_{ilj+1}(t)$. If a session i is in group $j+1$ or above, it is sampled in $x_{ilj+1}(t)$ slots in each level $j+1$ frame. We present an example to illustrate the complete scheduling procedure.

Illustrative Example:

Consider the network in figure 1. We illustrate the scheduling in link e_1. All 4 sessions traverse this link. Figure 2 illustrates the scheduling. Let the average utility attained over the past for sessions $1,2,3,4$ be 0.25, 1.008, 1.995 and 2 respectively[‡]. Let the group granularity parameter ϵ be 0.01. The group computation scheme divides these sessions in 3 groups. Group 1 consists of session 1, group 2 of session 2 and group 3 of sessions 3 and 4. Note that $\phi_{e_1}(1,t)=1$, $\phi_{e_1}(2,t)=1.5$ and $\phi_{e_1}(3,t)\approx 2$. Using $\phi_{e_1}(1,t)=1$, $x_{1e_11}=4$, $x_{2e_11}=3$, $x_{3e_11}=2$, $x_{4e_11}=1$. Using the values of $\phi_{e_2}(2,t)$ and $\phi_{e_3}(3,t)$, $x_{2e_12}=3$, $x_{3e_12}=2$, $x_{4e_12}=1$. Similarly, for group 3, $x_{3e_13}=2$, $x_{4e_13}=1$. The entire time axis is divided into level 1 frames of 10 slots each. Sessions $1,2,3,4$ are sampled in $4,3,2,1$ slots in each of the frames respectively. In this example, session 1 can not send packets in the first 3 of its 4 slots it is sampled in each level 1 frame, on account of the bottleneck in link e_4. Other sessions utilize their chances in the level 1 slots offered to them. So 3 level 1 slots of each level 1 frame become level 2 slots. Level 2 slots are divided into level 2 frames of 6 slots. Sessions 2, 3 and 4 are sampled in 3, 2 and 1 slots respectively, in each level 2 frame. Session 2 misses most of the slots on account of its bottleneck in link e_3. These slots become level 3 slots, sessions 3 and 4 utilize all the offered slots. Each level 3 frame consists of 3 slots each, and these are offered to group 3 sessions, sessions 3 and 4 in proportion $2:1$. Note that session 1 is still sampled at rate 4 and offered a utility of 1 unit ($\phi_{e_1}(1,t)=1$) in spite of the bottleneck estimate. However, session 1 consistently misses many of these slots on account of the bottleneck in e_4, and the other sessions utilize the slots missed by session 1.

[†]Here, we assumed that $U_i^{-1}(\phi_l(1,t))$ are rational for all i,l,t. If these quantities are irrational, then rational approximation can be used, with an allowable error of δ.

[‡]Note that these are close to the maxmin fair utilities as shown in Figure 1. However, initially the estimated utility may differ from the maxmin fair utilities on account of estimation error. We illustrate a specific case here.

Intuitive Justification of the Policy:

We present the intuition behind the design of the policy. For this purpose, we need to study the computation of the maxmin fair utility allocation first. The maxmin fair utility allocation can be computed by the following centralized computation algorithm proposed in [2]. Every link offers bandwidth to the sessions such that their utilities are equalized, and the overall link capacity is fully utilized. Initially, the utility offered by link l to any session traversing l is $\eta_l(1)$, and the bandwidth offered to a session i traversing link l is $U_i^{-1}(\eta_l(1))$, where

$$\sum_{i \in n(l)} U_i^{-1}(\eta_l(1)) = C_l \tag{3}$$

Here, $n(l)$ is the set of sessions traversing link l. The bandwidth received by a session equals the minimum bandwidth offer in its path, and the link which offers the minimum bandwidth is its bottleneck link. Thus the bandwidth received by session i is $\min_{l \in L_i} U_i^{-1}(\eta_l(1))$, where L_i is the set of links session i traverses. Also, the utility received by session i is $\min_{l \in L_i} \eta_l(1)$. All sessions receiving the minimum utility are removed from the network. Residual bandwidth in each link is the difference between the link capacity and the bandwidth consumed by the sessions removed from the network. Link utilities $\eta_l(2)$ are recomputed in the residual network, and the sessions with minimum utility are removed once again. The process continues till all sessions are finally removed, and the output is the maxmin fair utility allocation[2].

The scheduling policy was emulates the output of the centralized algorithm using local information only. Note that $\eta_l(1) = \phi_l(1, t)$ at all t (from equations (1) and (3)). Thus any session i receives a utility offer of $\eta_l(1)$ in any link l in its path at all t. Now, the backpressure based transmission policy ensures that the bandwidth received by a session equals the minimum bandwidth offer on its path. Thus every session i receives a bandwidth of at least $\min_{l \in L_i} U_i^{-1}(\eta_l(1))$ in the long run, and the sessions removed in the first iteration of the computation algorithm of [2] receive exactly this amount again in the long run. Also, these sessions receive the overall minimum utility. However, these results hold for the long term averages, but the estimations also consider the transients. Analytical results show that the estimation recovers from errors due to transient, and ultimately the sessions removed in the first iteration of the centralized algorithm are the ones in group 1 ($G_l(1, t)$) of the links[11]. Also, the estimated utilized bandwidth for these sessions converge to the bandwidth allocated to these sessions by the centralized policy.

Note that while computing utility $\phi_l(i, t)$ we consider only the residual bandwidth left over after removing the estimated bandwidth consumption of groups $1, 2, \ldots, i-1$ sessions (equation (2)). This is similar to the computation of the utility $\eta_l(i)$ in the centralized algorithm. In fact, it can be analytically shown that $\phi_l(i, t)$ closely approximates $\eta_l(i)$ after a finite number of estimation epochs[11].

The sampling procedure assigns levels to the slots and decides which sessions to sample in a slot depending on the level of the slot and the group of the sessions. We argue that this procedure actually samples a session i of group j at the rate $U_i^{-1}(\phi_l(j, t))$ in any link l on its path. Since $\phi_l(j, t)$ closely approximates $\eta_l(j)$ and the utility functions are continuous, a session i of group j ultimately receives a bandwidth offer of $U_i^{-1}(\eta_l(j))$ in any link on its path. On account of the back pressure based transmission, session i

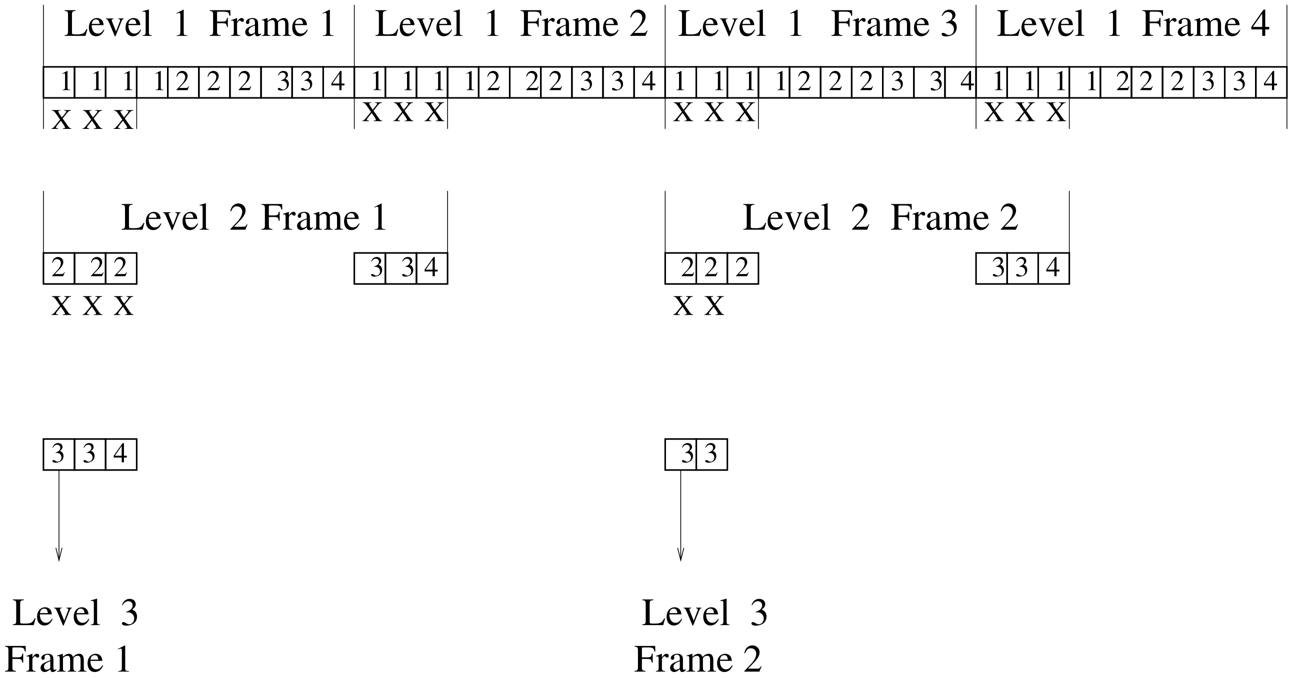

Figure 2. The number in a slot indicates the session which is sampled in the slot in the corresponding level. For example sessions $1, 2, 3$ are sampled in the first slot in levels $1, 2, 3$ respectively. The X sign indicates that the session sampled in the slot does not utilize its chance, and hence the slot passes to the next level. For example, sessions 1 and 2 do not utilize their chances in the first slot in levels 1 and 2 respectively. So this slot passes to the third level. Session 1 does not utilize its chance in the eleventh slot in level 1. This slot passes to level 2. Session 3 is sampled and utilizes its chance in this slot in level 2. Level of the first slot in the first level 1 frame changes from 1 to 2 and subsequently to 3.

receives a bandwidth of $\min_{l \in L_i} U_i^{-1}(\eta_l(j))$ in every link, and this attains its maxmin fair utility as per the computation algorithm of [2].

Any session i is sampled at the rate $U_i^{-1}(\phi_l(1,t))$ in the level 1 slots. The sampling rate for a group j session (session i) in level k slots, is $\frac{U_i^{-1}(\phi_l(k,t)) - U_i^{-1}(\phi_l(k-1,t))}{\sum_{q \in n(l) \setminus \cup_{p=1}^{k-1} G_l(p,t)} U_q^{-1}(\phi_l(k,t)) - U_q^{-1}(\phi_l(k-1,t))}$, if $k \leq j$. The denominator equals $\sum_{q \in G_l(k-1,t)} \left(U_q^{-1}(\phi_l(k-1,t)) - \hat{r}_{q,l}(t) \right)$. Also, the level k slots are generated at the rate $\sum_{q \in G_l(k-1,t)} \left(U_q^{-1}(\phi_l(k-1,t)) - \hat{r}_{q,l}(t) \right)$ (note that all slots do not have level k). Thus, the sampling rate of any session i of group j in level k slots is $U_i^{-1}(\phi_l(k,t)) - U_i^{-1}(\phi_l(k-1,t))$, if $k \leq j$. Thus, any session i receives a total bandwidth of $U_i^{-1}(\phi_l(j,t))$ if it is a group j session, in any link l in its path.

5. Analytical Performance Guarantees

The previous intuitive argument justifies the design of this algorithm. Now we present our analytical results. We have omitted all proofs throughout, on account of space restrictions. Refer to technical report[11] for proofs and details.

Session i source is "well-behaved" if it generates at most $p_i(v-u)+\xi_i$ packets and at least $p_i(v-u)-\xi_i$ packets in any interval $[u,v)$. The parameters ξ_i are the "transmission jitters." Many sources are well-behaved in practice. Oftentimes, a leaky bucket shaper is placed between a traffic source and a network. The output of the leaky bucket is well-behaved. Analytical results will be presented for "well behaved" sources only. However we expect these results to hold in a more general scenario. Let the average bandwidth received by session in interval $(0,t)$ be $r_i(t)$, where
$r_i(t) = \frac{1}{t}$Total number of packets received by session i in interval$(0,t)$.

Theorem 1 *Let maxmin fair utility of a session i be u_i. Then $\lim_{t\to\infty} U_i\left(r_i(t)\right) = u_i$, if group granularity parameter $\epsilon < \epsilon^*$ and window parameter $W > W^*$. The constants ϵ^*, W^* depend only on the system parameters, e.g., path lengths, transmission jitters etc.*

Theorem 1 implies that the utility of the long term average rates of the users are equal to the maxmin fair utilites. As we observe here, Theorem 1 holds for certain ranges of the group granularity and window parameters. We present exact values for ϵ^* and W^* in technical report [11]. In general, the analytical bounds are overly conservative, and our experimental results indicate that convergence takes place for moderate values of both ϵ and W. We discuss the choice of group granularity and window parameters later.

It may appear intriguing that the message generation rates p_i do not appear explicitly in Theorem 1. However, note that the message generations rates are the same as the maximum transmission rates in the feasibility condition of page 3. Thus, the p_is affect the feasible set of rate vectors, and hence also affect the maxmin fair utility allocation $(u_1,\ldots,u_N)$. Similarly, the transmission jitters affect the thresholds, ϵ^* and W^*.

6. Conclusion and Discussion

We first discuss some salient features of our policy. We presented a globally optimum fully distributed policy which is essentially local information based. Link schedulers need the congestion information of the neighbors. However, this information can be piggy backed in data or acknowledgement packets. Also, the congestion information needs to indicate whether the queue lengths have crossed a threshold or not, and this consumes only one bit. The policy works even if there is feedback delay. In this case, the packet transmission decisions are taken on the basis of the current knowledge about the queue lengths of the neighboring nodes. The sampling procedure uses only local information and is thus unaffected by the feedback delay. All the analytical convergence results hold for any bounded feedback delay. The intuition is as follows. The delayed information can differ from the actual queue lengths at the destination node by at most a constant which depends on the propagation delay and the link capacities. This constant does not affect the long term throughputs[11]. However, the choice of the buffer and window thresholds depend on the value of the delay.

Now, we discuss the choice of the parameters. The group granularity parameter needs to be small enough so that the scheduling policy can distinguish between sessions with different maxmin fair utilities. For example, if ϵ is not small enough, then two sessions with different maxmin fair utilities will be in the same group. At the same time, the grouping changes very frequently initially if the granularity parameter is very small, and the convergence is slow. The parameter ϵ can be determined adaptively.

Next, the window parameter needs to be large enough to guarantee convergence. However, the analytical bounds are pessimistic, and experimental results indicate that the convergence is not sensitive to the value of W. The size of the buffer at any intermediate node must be PW where P is the number of sessions traversing the node.

Finally, on account of space constraints we did not describe the generalization of the policy that addresses the multicast case. The generalized policy has the same sampling procedure, but the packet transmission procedure is more complex. The packet transmission procedure is the same as that in [10]. It is interesting to note that the generalization from unicast to multicast for maxmin fair allocation of bandwidth differs in the packet transmission procedure only[4,10]. Technical report[11] describes the generalization in detail.

Summarizing, we have studied utility based fairness notions in this paper. We presented provably fair distributed, local information based scheduling policies for attaining maxmin fair allocation of utilities.

REFERENCES

1. D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ:Prentice-Hall, 1987
2. Z. Cao and E. Zegura. Utility Max-Min: An Application-Oriented Bandwidth Allocation Scheme *Proceedings of INFOCOM'99*
3. P. Goyal, H. Vin and H. Chen. Start-time Fair Queueing: A scheduling algorithm for integrated services. *Proceedings of the ACM-SIGCOMM* 96 *pp* 157 – 168, Palo Alto, CA, August '96
4. 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
5. F. Kelly, A. Maulloo, and D. Tan. "Rate control for communication networks: Shadow prices, proportional fairness and stability" *Journal of Operations Research society,* 49(3):237 – 252, March 1998.
6. K. Kar, S. Sarkar, L. Tassiulas. "Optimization Based Rate Control for Multirate Multicast Sessions," *Proceedings of INFOCOM'*2001, Alaska, April 2001.
7. K. Kar, S. Sarkar, L. Tassiulas. "A Simple Rate Control Algorithm for Maximizing Total User Utility", *Proceedings of INFOCOM'*2001, Alaska, April 2001.
8. S. Low and D. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence, *IEEE/ACM Transactions on Networking,* 7(6), Dec. 1999.
9. S. Sarkar and L. Tassiulas. "Fair Allocation of Utilities in Multirate Multicast Networks," *Proceedings of 37th Annual Allerton Conference on Communication, Control and Computing,* October 1999.
10. S. Sarkar and L. Tassiulas. "Back pressure based multicast scheduling for fair bandwidth allocation," *Proceedings of INFOCOM'*2001, *Alaska*
11. S. Sarkar and L. Tassiulas. "Scheduling for Quality of Service based Service Differentiation," *Technical Report,* Electrical Engineering Department, University of Pennsylvania, http://www.seas.upenn.edu/~swati/publication.htm