Fair Allocation of Resources in Multirate Multicast Trees

Saswati Sarkar and *Leandros Tassiulas*
Dept. of Electrical and Computer Engineering and Institute for Systems Research
University of Maryland, College Park, MD, USA
email addresses: swati@eng.umd.edu, leandros@isr.umd.edu

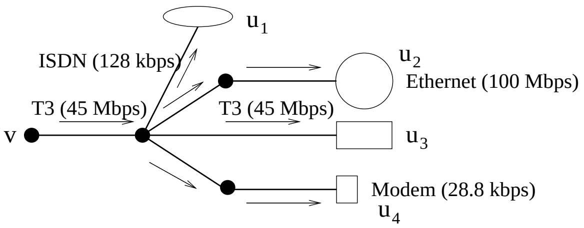

Fig. 1.

Abstract— **We study fairness in arbitrary networks with multicast capabilities. Multicast traffic in internet and ATM provides a motivation for studying these networks. A study of fairness in multicast networks poses several interesting problems e.g., the issue of *intra-session* fairness in addition to that of *inter-session* fairness in unicast networks. We develop a mathematical framework to model the fair allocation of bandwidth in multicast networks with minimum and maximum rate constraints. We present an algorithm for computation of maxmin fair rates allocated to various source-destination pairs. This algorithm may be used to make a fair allocation of layers to various receivers for loss-tolerant traffic. Towards the end of the paper, we outline several interesting open research problems in this area.**

I. Introduction

Multicasting provides an efficient way of transmitting data from a sender to a group of receivers. In general many multicast sessions simultaneously share the network resources. Ideally all sessions should have a fair share of bandwidth. The fair allocation of bandwidth have been studied extensively in the unicast context. Multicasting poses some specific challenges in this regard. This is because a single session may have many destinations, and end systems can have widely varying bandwidth connectivities. On one hand there are fast ethernets (100 Mbps) and on the other hand there are slow modems (28.8 kbps). The paths to different destinations may have different bandwidth capacities, e.g., one may consist of multi-megabit links, such as, $T3$ (45 Mbps) and another may have a 128 kbps ISDN line (Refer to the network shown in figure 1 for an example). Every receiver would like to receive service at a rate commensurate with its capabilities and the capacity of the path leading to it from the source independent of the capabilities of the other receivers of the same session. So multicasting poses the issue of *intra-session* fairness in addition to that of *inter-session* fairness.

The problem of fair allocation of bandwidths to multicast sessions under the constraint that all receivers of the same session must receive service at the same rate has been investigated in [14]. However, intra-session fairness can not be achieved by a single rate of transmission per session. An alternative approach is to have a layered transmission scheme. In this approach, a signal is encoded into a number of layers that can be incrementally combined to provide progressive refinement. The different layers of a multicast group are considered different multicast groups and receivers adapt to congestion by adding and dropping layers, where adding a layer is joining a multicast group and dropping a layer is leaving a group[4]. This layered transmission scheme have been used for video transmission but can be used for other loss-tolerant traffic as well. Layered transmission scheme can be used to attain inter-session and intra-session fairness by having the receivers subscribe to a "fair" number of layers. This idea have not been cultivated very systematically. The two well known network protocols for layered transmission, RLM (Receiver-driven Layered Multicast)[7] and LVMR (Layered Video Multicast with Retransmissions)[5] do not handle inter-session fairness very well[6]. A scheme for fair resource allocation for multi-session layered video multicast has been proposed in [6]. The authors present empirical evidence that the scheme improves inter-session fairness for networks with multiple video sessions sharing only one link. But, there is no experimental or analytical evidence that the scheme works well for more complex networks, with sessions sharing several links with each other. Besides [6] does not make any effort towards the computation of the actual rates or the number of layers allocated to the receivers under some well defined notion of fairness. We believe that a more systematic approach is necessary to study this problem.

As a first step towards that direction we give a rigorous mathematical formulation of the problem. We adopt the notion of maxmin fairness[2]. A rate allocation is maxmin fair, if no receiver can be allocated a higher rate without hurting another receiver having equal or lower rate. We present an algorithm for computation of maxmin fair rates. A unified mathematical framework can model the issue of intra-session and inter-session fairness for multicast in both ATM and the internet. The modalities of the rate allocation and the implementational details may differ in the two scenario. As a first step, we do not distinguish between the two. ATM sessions often have minimum rate requirements and maximum rate constraints. These parameters are negotiated during connection establishment phase. Our model is general enough to accomodate these requirements. Internet sessions do not have these requirements because there is no connection establishment phase. Maxmin fairness in multirate multicast networks have been studied recently in [10]. An algorithm for computation of maxmin fair rates is proposed there. This algorithm is centralized, while the algorithm we present here is amenable to distributed implementation.

This paper is organized as follows. Section II describes the problem of maxmin fairness for layered multicast transmission and presents the mathematical framework used to model the problem. Section III presents some interesting properties of the maxmin fair rate allocation when the layers are considered to have a continuum of bandwidth. This section also presents an algorithm for computation of the maxmin fair rates. Section IV discusses the fairness issue when the layers have discrete bandwidth. Section V identifies some directions for future research. This section concludes the paper.

II. Network Model

We consider an arbitrary topology network with N multicast sessions. A multicast session is identified by the pair (v, U), where v is the source node of the session and U is the group of intended destination nodes. We assume that the traffic from node v is transported across a predefined multicast tree to nodes in U. The tree can be established during connection establishment phase if the network is connection oriented or can be established by some well known multicast routing protocol like DVMRP[3], MOSPF[8], CBT[1], PIM[4] and MIP[9] in internet type network. The sessions may have minimum rate constraints. Also some sources may not be able to transmit at a rate higher than a certain threshold. Some receivers may have a low processing ability. In that case, it is useless to allocate higher rates to that session. So rate allocations can have peak rate constraints as well. These parameters are useful for ATM like scenarios, where session establishment is preceded by a negotiation stage and the network can be informed of these requirements during the negotiation stage. In a connectionless network, sessions can not have any such requirement as the network would never know of these, and would have to make rate allocation irrespective of any such requirement. Such a scenario can very well be accomodated in our model, by assuming minimum rate requirement as 0 and maximum rate to be ∞ for for each session.

We would initially assume the layers to be in continuum, or allow the allocation of fractional number of layers. For example, if every layer requires a bandwidth of 2 units, then we can allocate a bandwidth of 2.6 units to a receiver, interpreting it as allocation of 2.3 layers. This assumption would simplify the model and at the same time give a good understanding of the properties of maxmin fair allocation. Discretization of layers introduces certain interesting properties of a fair allocation. We would explore these properties towards the end of this paper and also suggest a possible way to move from the continuous to the discrete case.

We call every source destination pair a virtual session. For example, if a session n has source v_n and destination set U_n, where $U_n = \{u_1, \ldots, u_{n_1}\}$, then this session would correspond to n_1 virtual sessions, $(v_1, u_1), \ldots, (v_1, u_{n_1})$. For example, the network shown in figure 1 has a single session, (v, U), with $U = \{u_1, \ldots, u_4\}$. This session corresponds to 4 virtual sessions, (v, u_1), (v, u_2), (v, u_3), (v, u_4). Our objective would be to achieve a maxmin fair rate allocation for the virtual sessions. We look at rates attained by virtual sessions instead of those attained by the actual sessions because our requirement is that every receiver should get a bandwidth commensurate with its fair share of the capacity of the path between the source and the receiver. So ensuring maxmin fairness of the session rate allocation and allocating the maxmin fair session rate to all the virtual sessions would cause intra-session unfairness. We assume that every virtual session (source-destination pair) has a minimum and a maximum rate.

Informally speaking a rate allocation for the virtual session is feasible, if the rate for every virtual session is between the minimum and the maximum possible rates for the virtual session. Besides if session n corresponds to virtual sessions $m_1, \ldots, m_{n_l}$ in link l, then the maximum of the rates allocated to the virtual sessions $m_1, \ldots, m_{n_l}$ is the bandwidth consumed by session n in link l. Total bandwidth consumed by all sessions traversing through link l can not exceed the capacity of link l. More formally let there be M virtual sessions. A M-dimensional vector $(r_1, \ldots, r_M)$ is a feasible rate allocation if

- $\mu_i \leq r_i \leq p_i \; \forall i$, where μ_i and p_i are respectively the minimum and maximum rates of virtual session i, $p_i \geq \mu_i \geq 0$,
- Let $n(l)$ denote the set of sessions passing through link l and $m(k, l)$ denote the set of virtual sessions of session k passing through link l and C_l denote the capacity of link l,

$$\sum_{i \in n(l)} \max_{j \in m(i,l)} r_j \leq C_l$$

Refer to Figure 2 for an example.

Informally a feasible rate vector is maxmin fair if it is not possible to maintain feasibility and increase the rate of a virtual session without decreasing that of any other virtual session which has equal or lower rate.

Definition 1
More formally a feasible rate allocation vector $\vec{A}$ is maxmin fair if it satisfies the following property with respect to any other feasible rate allocation vector $\vec{B}$: if there exists i such that the ith component of $\vec{B}$ is strictly

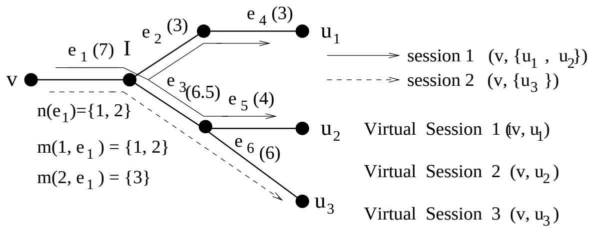

Fig. 2. Session 1 includes virtual sessions, $1,2$. Session 2 includes virtual session 3. The figure shows some sample $n(e_i)$ and $m(i,e_j)$s. The numbers in brackets, () denote the capacities of the respective links. The capacity constraint for link e_1 is $\max(r_1,r_2)+r_3 \leq 7$ and that for link e_3 is $r_2+r_3 \leq 6.5$. The minimum and maximum rate constraints are $2 \leq r_1 \leq 4$, $1 \leq r_2 \leq \infty$ and $0 \leq r_3 \leq 5$.

greater than that of $\vec{A}$, then there exists j such that the jth component of $\vec{A}$, a_j is less than or equal to the ith component of $\vec{A}$, a_i ($a_j \leq a_i$) and the jth component of $\vec{B}$ (b_j) is strictly less than the jth component of $\vec{A}$ ($b_j < a_j$). The components of $\vec{B}$ are more unequal than those of $\vec{A}$ in some sense.

Example II.1: The maxmin fair rate vector in the network of Figure 2 is $(3, 3.25, 3.25)$. It is easy to check that this rate vector is feasible. It is not possible to increase r_1 above 3 because of the capacity constraint for edge e_2. Any increase in r_2 (r_3) will cause a decrease in r_3 (r_2) because of the capacity constraint for edge e_3. Thus $r_2 = r_3 = 3.25$ in the maxmin fair allocation.

Definition 2
A simpler definition of a maxmin fair rate allocation would be that a maxmin fair rate allocation is lexicographically optimal, i.e., its smallest component is the largest amongst the smallest components of all feasible rate allocations, subject to this it has the largest second smallest component and so on.

Example II.2: The lexicographically optimal rate vector in the network of Figure 2 is $(3, 3.25, 3.25)$. The minimum component of every rate vector must be less than or equal to 3, because of the capacity constraint for edge e_2. So the smallest component of this rate vector is the largest amongst the smallest components of all feasible rate vectors. Capacity constraint for edge e_3 forces the second largest component to be 3.25 and subject to this the largest component can only be 3.25 as well.

These two definitions are not entirely equivalent. The first is stronger than the other. In general, if we consider maxmin fairness of finite dimensional vectors in closed and bounded feasible sets, a maxmin fair vector always exists as per the second definition, but may not exist if we go by the first definition. This would exactly be the case if we do not allow allocation of fractional layers, as we would discuss later. However if we consider the layers in continuum then both definitions are equivalent and maxmin fair rate allocation exists as per both definitions.

Example II.3: Let layers be in continuum. Then as per both definitions, $(3, 3.25, 3.35)$ is maxmin fair in the network of Figure 2. Let layers not be in continuum. A layer has a bandwidth of .5 units and fractional layers can not be allocated. Then no virtual session can get a bandwidth of 3.25 units. It can get either 3 or 3.5 units. So an allocation of $(3, 3.25, 3.25)$ does not exist. Rate vectors $(3, 3.5, 3)$ and $(3, 3, 3.5)$ are both lexicographically optimal and hence maxmin fair as per Definition 2. Neither is maxmin fair as per Definition 1. In fact there does not exist a maxmin fair vector as per Definition 1.

Henceforth we shall ignore the maximum rate constraints. This does not cause any loss in generality because maximum rate constraints can be incorporated by adding artificial links between receivers with maximum rate constraints and the rest of the network, with capacities of the artificial links equal to the maximum rates of the respective receivers. The size of the augmented network is comparable to that of the given network. So complexity of any algorithm for computation of the maxmin fair rates in a network should remain the same, if we use the augmented network instead. We will present a constructive algorithm to obtain the maxmin fair rate allocation for layers in continuum in the next section.

III. Properties of maxmin fair rate allocation

We first present an important property of the maxmin fair rate vector. We conclude this section with an algorithm for computation of maxmin fair rate allocation. We eliminate the proofs because of lack of space. The proofs have been presented in technical report[13].

We first introduce the concept of bottleneck links. A link l is said to be *bottlenecked* with respect to a virtual session k traversing across it if the following conditions are met:

- Capacity of the link is fully utilized, i.e., the sum of the rates allocated to the sessions travelling across the link must be equal to the capacity of the link
$$\sum_{i \in n(l)} \max_{j \in m(i,l)} r_j = C_l$$
where $n(l)$, $m(i,l)$, C_l, r_j are as defined before.
- The virtual session has the maximum rate amongst all virtual sessions of the same session travelling through the link, i.e.,
$$r_k = \max_{j \in m(\chi(k),l)} r_j$$
where $\chi(k)$ is the session corresponding to virtual session k.
- If any other virtual session j traversing through link l, has a rate higher than that of virtual session k, then

rate of virtual session j is equal to the minimum rate of some virtual session in $m(\chi(j), l)$ (set of virtual sessions travelling across link l and belonging to the same session as virtual session j).

Example III.1: Refer to the network of Figure 2. Consider the rate vector $(3, 3.25, 3.25)$. Link e_2 is bottlenecked w.r.t. virtual session (v, u_1). Link e_3 is bottlenecked w.r.t. virtual sessions (v, u_2) and (v, u_3). Link e_1 is not bottlenecked w.r.t. any virtual session because its capacity is not fully utilized. Consider the rate vector $(2, 3.25, 3.25)$. This is also a feasible rate vector. As before e_3 is bottlenecked w.r.t. virtual sessions (v, u_2) and (v, u_3). Now no link is bottlenecked w.r.t. virtual session (v, u_1).

Lemma 1: A feasible rate vector is maxmin fair iff every virtual session has a bottleneck link.

We would present an algorithm for computing the maxmin fair virtual session rates shortly. The proof of the correctness of the algorithm will use this lemma. We prove this lemma in technical report[13].

Now we present an algorithm for computation of maxmin fair rates. First we introduce certain terminologies.

L_s is the set of links traversed by virtual session s

A virtual session s is *saturated* if there exists a link l in L_s such that the capacity of the link is fully utilized and s has the maximum rate amongst all virtual sessions of $m(\chi(s), l)$, i.e.,

$$\sum_{i \in n(l)} \max_{j \in m(i,l)} r_j = C_l \text{ and } r_s = \max_{j \in m(\chi(s),l)} r_j$$

A session is *saturated* on a link l if all the virtual sessions of the session travelling through the link l are saturated.

$S(k)$ denotes the set of unsaturated virtual sessions at the end of the kth iteration

$\Xi_l(k)$ denotes the set of unsaturated sessions passing through link l at the end of the kth iteration

$\eta_l(k)$ denotes the link control parameter of link l at the end of the kth iteration. Link control parameter is the iterate which would be used in computation of the maxmin fair rates. It is an estimate of the fair share of the bandwidth of the link which can be allocated to the unsaturated virtual sessions traversing the link. This bandwidth would have been allocated to the unsaturated virtual sessions traversing the link, if there were no bandwith constraints on other links and also if the virtual sessions did not have any minimum rate constraints. As we shall see later, if a virtual session is saturated, then its rate is already determined.

$\eta_{il}(k)$ denotes the session link parameter of session i traversing through link l. It is the bandwidth assigned to session i if there were no bandwidth constraints for any of its virtual sessions on other links.

$\lambda_{il}(k)$ is the rate allocated to the session i on link l at the end of the kth iteration. It is actually the maximum of the rates allocated to the virtual sessions in $m(i, l)$ at the end of the kth iteration

$F_l(k)$ denotes the total bandwidth consumed by the saturated sessions passing through link l at the end of the kth iteration.

The following algorithm computes the maxmin fair rates for the virtual sessions.

1. $k = 0$ $\eta_l(0) = 0$, $F_l(0) = 0$, $S(0) = \{1, \ldots, M\}$, $\Xi_l(0) = n(l)$ $\forall$ link l,
2. $k \rightarrow k + 1$
3. For every link l in the network compute the link control parameter. If $\Xi_l(k-1) \neq \phi$, then $\eta_l(k)$ is the maximum possible θ, which satisfies the equation, $F_l(k-1) + \sum_{i \in \Xi_l(k-1)} \max(\theta, \mu_{il}) = C_l$ else $\eta_l(k) = \eta_l(k-1)$. $\eta_{il}(k) = \max(\eta_l(k), \max_{j \in m(i,l)} \mu_j)$
4. Compute $r_s(k)$ for all virtual sessions $s \in S$, where $r_s(k) = \min_{l \in L_s} \eta_{\chi(s)l}(k)$, if $s \in S(k-1)$, else $r_s(k) = r_s(k-1)$.
5. For every link l in the network compute the session rate in link l, for every session in $n(l)$, $\lambda_{il}(k) = \max_{s \in m(i,l)} r_s(k)$
6. Compute the set of virtual sessions unsaturated after the kth iteration, $S(k) = \{s : \forall l \in L_s$, either $\sum_{i \in n(l)} \lambda_{il}(k) < C_l$ or $r_s(k) < \lambda_{\chi(s),l}\}$
7. If $S(k) = \phi$, i.e., all virtual sessions are saturated, the algorithm terminates else go to the next step.
8. For every link l, compute the set of unsaturated sessions passing through the link l at the end of the kth iteration: $\Xi_l(k) = \{n : n \in \{1, \ldots, N\}$, $m(n, l) \cap S(k) \neq \phi\}$
9. For every link l, for which $\Xi_l(k) \neq \phi$, compute the bandwidth consumed by the saturated sessions passing through link l, $F_l(k) = \sum_{i \in n(l) \setminus N_l(k)} \lambda_{il}(k)$.
10. Go to step(2).

The algorithm proceeds as follows. Initially all the link control parameters are assigned zero values. All sessions and virtual sessions are unsaturated. Next the algorithm computes the link control parameters as per step (3). If the minimum rates were 0 for all virtual sessions, then the link control parameter for link l would be $C_l/|n(l)|$. In presence of nonzero minimum rates, computation takes place as per step (3). If a virtual session had no bandwidth constraint on other links, then it would have got a rate equal to the maximum of the link control parameter and the minimum allowable rates (μ_ss) of the virtual sessions of the same session and passing through the link. On account of the bandwidth constraint in other links, the virtual session gets a rate equal to the minimum of the above quantity, the minimum taken over all links in the path of the virtual session. At least one virtual session is saturated every iteration[13]. The bandwidth consumed by the saturated sessions, if any, are computed. This bandwidth is subtracted from the link capacity, and the link control parameters are recomputed at the beginning of every iteration. Since at least one virtual session is saturated every iteration, the algorithm terminates in at most $|S|$ iterations (S is the set of virtual sessions). We illustrate the operation of the algorithm with an example.

Example III.2: Consider the network of Figure 2. The minimum rate for virtual session (v, u_1) is now 4. The capacities of edges e_2, e_4 are now 4. The maximum rate constraints do not exist. The rest of the constraints remain the same. Virtual sessions (v, u_1), (v, u_2) and (v, u_3) are named virtual sessions $1, 2, 3$ respectively. $L_1 = \{e_1, e_2, e_4\}$, $L_2 = \{e_1, e_3, e_5\}$, $L_3 = \{e_1, e_3, e_6\}$ $\eta_{e_1}(1) = 3$, $\eta_{e_2}(1) = 4$, $\eta_{e_3}(1) = 3.25$, $\eta_{e_4}(1) = 4$, $\eta_{e_5}(1) = 4$, $\eta_{e_6}(1) = 6$. Now, $\eta_{1e_1}(1) = 4, \eta_{2e_1}(1) = 3$, $\eta_{1e_2}(1) = 4$, $\eta_{1e_3}(1) = 3.25$, $\eta_{2e_3}(1) = 3.25$, $\eta_{1e_4}(1) = 4$, $\eta_{1e_5}(1) = 4$, $\eta_{2e_6}(1) = 6$. Computing the $r_s(1)$s as per step 4, we have $r_1(1) = 4$, $r_2(1) = 3.25$, $r_3(1) = 3$. Observe that virtual sessions 1 and 3 are saturated, while virtual session 2 is not. $S(1) = \{2\}$. Thus session 2 is saturated on all links. Session 1 is unsaturated on only those links which are on the path of virtual session 2. $\Xi_{e_1}(1) = \Xi_{e_3}(1) = \Xi_{e_5}(1) = \{1\}$, $\Xi_l(1) = \phi$, if $l \notin \{e_1, e_3, e_5\}$. $F_{e_1}(1) = F_{e_3}(1) = 3$, and $F_{e_5}(1) = 0$. Doing the iteration once again, $\eta_{e_1}(2) = 4$, $\eta_{e_3}(2) = 3.5$, $\eta_{e_5}(2) = 4$. $\eta_l(2) = \eta_l(1)$ for the rest of the links. $\eta_{1e_1}(2) = 4$, $\eta_{1e_3}(2) = 3.5$, $\eta_{1e_5}(2) = 4$. Thus $r_2(2) = 3.5$. $r_s(2) = r_s(1)$, $s \in \{1,3\}$. Now virtual session 2 is also saturated. So $S(2) = \phi$ and the algorithm terminates. The rates obtained upon termination are $(4, 3.5, 3)$.

The rates allocated to the virtual sessions, upon termination of the algorithm are the maxmin fair rates. We first show that when the algorithm terminates every virtual session has a bottleneck link[13]. Maxmin fairness follows from Lemma 1.

IV. Allocation of layers: from continuum to discrete

In the previous section we discussed the allocation of maxmin fair bandwidth to the virtual sessions. We assumed that the layers are in continuum. This assumption holds when fractional layers can be allocated and there are ceratin situations when fractional layers can be allocated. Again, fractional layers can not be allocated in certain cases[13]. In these cases, we can only allocate full layers to the source destination pair (virtual sessions), and layers have discrete bandwidth. Let us consider an example to see how fractional layers can be allocated. Let 2.9 layers be allocated to a receiver as per a rate allocation. This means the receiver subscribes to 3 layers, but 10% of the packets of the 3rd layer are dropped on account of resource limitations. Certain coding schemes can recover a small percentage of lost packets. So there may be some quality improvement if the receiver subscribes to 2.9 layers instead of 2 layers. This is the idea behind fractional allocation of layers. However coding schemes can recover packets only if a small percentage of packets are lost. So subscription to 2 layers and 2.1 layers may produce the same quality, because if 90% of the packets of the 3rd layer are dropped, then subscription to the 3rd layer would be meaningless. Also sometimes fractional layers can not be allocated, because the coding scheme may not be able to recover lost packets. We would discuss fair allocation of layers to source destination pairs in this section, under the assumption that fractional layers can not be allocated. A layer allocation vector is a $M-$dimensional vector, with nonnegative integral components. A layer allocation is feasible, if the bandwidths consumed by the layers allocated to the receivers form a feasible rate vector.

An interesting question is whether a maxmin fair allocation of layers exists as per the Definitions 1 and 2. Since layer allocation is a finite dimensional vector and the feasible set is closed and bounded, a lexicographically optimal layer allocation exists. This lexicographically optimal layer allocation is the maxmin fair layer allocation as per Definition 2. However in general a maxmin fair layer allocation may not exist as per Definition 1. Example II.3 demonstrated this. Consider the following simpler example for more insight:

Example IV.1: A network has a single link and two sessions passing through the link. Every layer consumes 1 unit of bandwith. The capacity of the link is 1 unit. A maxmin fair rate vector allocates .5 unit of bandwidth to each of the sessions, i.e., .5 layer to each. We can not allocate fractional layers. So possible allocation vectors are $(0, 1)$, $(1, 0)$, $(0, 0)$. None of these vectors are maxmin fair as per Definition 1. For each of these vectors, it is possible to maintain feasiblity and increase the number of layers of one session, possibly hurting another session, but the other session has higher number of layers. For example, consider the allocation $(0, 1)$. It is possible to increase the number of layers of session 1, by hurting session 2. However session 2 has higher number of layers than session 1 in this allocation.

We can look at *maximally fair* layer allocations instead. We first use the concept of *relative fairness* introduced in [12]. A layer allocation vector $\vec{A}$ is *fairer* than another layer allocation vector $\vec{B}$, if for every virtual session i which has higher number of layers under $\vec{B}$ than under $\vec{A}$, there is some other virtual session j whose number of layers was already no more than that of i under $\vec{A}$, and has been decreased further by $\vec{B}$. A more formal definition of relative fairness follows.

A layer allocation vector $\vec{A}$ is *fairer* than another layer allocation vector $\vec{B}$ if

- $\vec{A} \neq \vec{B}$ and
- if there exists an i such that $a_i < b_i$, then there exists a j such that $b_j < a_j$ and $a_j < a_i$. (a_k is the kth component of $\vec{A}$ and b_k is the kth component of $\vec{B}$)

A feasible layer allocation $\vec{A}$ is maximally fair if no other feasible layer allocation is fairer than $\vec{A}$. There can be more than one maximally fair layer allocation vectors. One or more of these maximally fair layer allocation vectors are lexicographically optimal. In the network of Example IV.1 both the layer allocation vectors $(1, 0)$ and $(0, 1)$ are maximally fair and lexicographically optimal. One objective could be to allocate layers as per some maximally fair rate allocation. An algorithm for computation of maximally fair rate allocation is a topic for future research.

A lexicographically optimal layer allocation is the best one can hope for, when the layers do not have a continuum bandwidth. It is not known whether there exists a polynomial complexity algorithm for computation of lexicographically optimal layer allocation. This is also a topic of future research.

V. Conclusion

A major contribution of this paper is to develop a mathematical framework to model the fair allocation of bandwidth in the multicast scenario, while taking care of both intra-session and inter-session fairness. This framework can model fairness in both internet and ATM like networks. We have presented an algorithm for computation of the maxmin fair rates. This algorithm can be used to make maxmin fair layer allocation, if fractional layers can be allocated. This algorithm is amenable towards distributed implementation. This is because the computation of the session link control parameters of any link needs only information about the saturation status of sessions traversing the link and the previous iteration rates of the saturated sessions traversing the link. Also a virtual session (receiver) can determine its rate if it knows its session link control parameters of the links on its path. To know whether it is saturated or unsaturated, it needs to know only the session link rates on its path. We have described a distributed implementation of this algorithm in [13]. We have discussed the issue of fairness when fractional layers can not be allocated and pointed out several interesting differences, like the nonexistence of a maxmin fair layer allocation in general.

This is only a first step in a systematic study in this direction. Development of a lexicographically optimal layer allocation, computation of the maxmin fair rates and a maximally fair allocation of layers in a distributed manner and finally devising a dynamic scheduling algorithm for attaining the maxmin fair rates without actually computing them could be topics of future research.

References

[1] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees: an architecture for scalable inter-domain multicast routing, *Proceedings of ACM SIGCOMM,* Ithaca, NY, Sept. 1993, pp. 85-95

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

[3] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs, ACM Transactions on Computer Systems, vol 8, no. 2, pp. 54-60, Aug. 1994.

[4] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. An architecture for wide area multicast routing, *Proceedings of ACM SIGCOMM,* London, UK, Aug. 1994, pp. 126-135.

[5] X. Li, S. Paul and M. H. Ammar. Layered Video Multicast with Retransmission (LVMR): Evaluation of Hierarchical rate control, *Proceedings of IEEE Infocom' 98*, San Francisco, CA, March 1998

[6] X. Li, S. Paul and M. H. Ammar. Multi-session Rate Control for Layered Video Multicast *Technical Report* GT-CC-98 − 21, College of Computing, Georgia Institute of Technology, 1998

[7] S. McCanne, V. Jacobson and M. Vetterli. Receiver-Driven Layered Multicast, *Proceedings of ACM SIGCOMM '96*, Stanford, CA, September 1996

[8] J. Moy, Multicast routing extensions for OSPF, *Comm, ACM* vol 37, no. 8, pp 61-66, Aug 94.

[9] M. Parsa, J.J.Garcia-Luna-Aceves. A protocol for Scalable Loop-Free Multicast Routing, *IEEE Journal on Selected Areas In Communications*, Vol 15, No. 3, 1997.

[10] D. Rubenstein, J. Kurose and D. Towsley. The Impact of Multicast Layering on Network Fairness *Proceedings of ACM SIGCOMM '99, Cambridge, MA, September,* 1999

[11] M. Satyanarayanan and F. Siegal. MultiRPC: A parallel remote procedure call mechanism. Tech Rep. CMU-CS-86-139, Carnegie-Mellon Univ., Aug. 1986

[12] S. Sarkar and K. N. Sivarajan: Fairness in Cellular Mobile Networks *Proceedings of the 34th Annual Allerton Conference on Communication, Control and Computing,*1996, *pp* 457 − 469

[13] S. Sarkar and L. Tassiulas: Distributed Algorithms for Computation of Fair Rates in Multirate Multicast Trees *Technical Report* TR 99 − 42, Institute for Systems Research and University of Maryland, 1999.

[14] H. Y. Tzeng and K. Y. Siu. On Max-Min Fair Congestion Control for Multicast ABR Service in ATM, *IEEE Journal on Selected Areas In Communications*, Vol 15, No. 3, 1997.