Back pressure based multicast scheduling for fair bandwidth allocation

Saswati Sarkar[1] and Leandros Tassiulas[2]
[1]Department of Electrical Engineering
University of Pennsylvania
[2]Department of Electrical and Computer Engineering and Institute for Systems Research
University of Maryland, College Park

Abstract—**We study fair allocation of resources in multicast networks with multirate capabilities. In multirate transmission, the session source hierarchically encodes its signal and the receivers subscribe to the appropriate number of layers. The objective of the network is to distribute the layers fairly. This can be attained either by computing the fair rates first, and then using a scheduling policy to attain the fair rates, or by using a scheduling policy which allocates the fair rates without computing them explicitly. The first requires knowledge of system parameters like link bandwidth, which are not generally known to the link schedulers. The second approach is more realistic. We present a scheduling policy which allocates the fair rates without computing them beforehand. We have presented analytical and experimental results demonstrating the fairness of the resulting rate allocation. In addition to guaranteeing the fair rates, this policy confines the packet losses to enhancement layers, and protects the more important base layers, when there is shortage of bandwidth. Furthermore, this policy does not require any knowledge of traffic statistics, is computationally simple, and is essentially local information based.**

I. Introduction

Present day internet is moving fast from best effort service to class based service. Different classes of users get different quality of service and are charged differently. Internet service providers would like to provide fair quality of service in the same class. Also, fair allocation of resources guarantees some minimum quality of service to all users. However, attaining a fair allocation of resources is a challenging problem in the current day networking scenario. 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 2 units for each session. However, session 2 can not use more than 1 unit on account of link e_1. So the fair allocation would be 3 units for session 1 and 1 unit for session 2. Thus the fair allocation in any link can only be determined with the congestion information in other links.

The complications increase in presence of multicast capabilities. This is because of network heterogeneity. In a multicast network, a single session has several receivers, and different receivers have different processing capabilities. Data paths for different receivers have different bandwidth. In figure 2, receiver u_3 receives information through a $T3$ (45 Mbps) link, whereas another receiver of the same session, u_1 is served by a 128 kbps ISDN line. Receiver u_4 is a 28.8 Kbps modem, whereas receiver u_2 is a 100 Mbps ethernet. Service rate of a receiver should not decrease because of the presence of other slow receivers in the same session. Again, a receiver should not receive service at a rate higher than it can sustain. Also, bandwidth should be split fairly between different sessions traversing the same link.

Multirate transmission can be used to accommodate these diverse requirements. It is possible to serve different receivers of the same session at different rates, if multirate transmission is used. The service rate of a session in a link is equal to the maximum of the session receivers downstream of the link. Thus the same session receives service at different rates in different links on its path. For this purpose, a source hierarchically encodes its signal in several layers. The lowest layer consists of the most important information and all receivers of the session should receive it. Receivers can subscribe to higher layers for successive refinement of reception quality at the expense of additional bandwidth. If the path to a receiver is congested, then it receives the base layer only, whereas a receiver subscribes to a large number of layers if its data path has a lot of bandwidth. For example, in figure 2 receiver u_4 receives only the base layer, whereas receiver u_2 receives many more additional layers. Incidentally, hierarchical coding is useful for real time loss tolerant traffic like audio and video. This is because, unlike data, these real time transmissions provide intelligible reception in presence of packet loss. Only, reception quality gradually degrades with increasing information loss. So the base layer provides crude information, while reception of higher layers enhance the reception quality. We consider real time traffic in this paper.

Our objective is to split bandwidth fairly among different sessions traversing a link, and also serve every receiver at a rate commensurate with the fair bandwidth share along its path. The fair share may be different for different receivers of the same session. Consider figure 3 for example (ignore the notations like $n(e_i)$ and $m(i, e_j)$ for the time being). Fair rate allocation is 3.75 for u_1 and 3.25 for receivers u_2 and u_3 each. The data path of receiver u_2 consists of links e_1, e_3, e_5 and e_7. Link e_5 is the bottleneck link in this case, as it has a bandwidth of 6.5 units to be shared between sessions 1 and 2. Fair share in this bottleneck link is 3.25 for both the sessions. Link e_3 is the bottleneck for receiver u_1 of session 1 and the fair share for session 1 in this link is 3.75 after allocating 3.25 to session 2. 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. Maxmin fairness is a good notion of fairness and as [14] points out, maxmin fairness satisfies many intuitive fairness properties in a multirate multicast network.

The usual approach is to compute the fair shares of the receivers and then determine the service order for packets in the links so as to actually serve packets as per the fair rates. We have presented distributed algorithms for computing fair allocations in [15]. Now, there are many scheduling policies which can attain any feasible rate allocation, once the rates are known, e.g., fair queuing strategies[3], [4], [7], [9], [**?**]. However, these scheduling policies work only when the link schedulers know the desired rates, or at least a ratio between the desired rates. But, computing the fair rates has several problems. Firstly, computation algorithms require exact knowledge of system parameters, like link bandwidth. Now, bandwidth available for real time traffic varies from time to time, depending on the service contracts between the network and the data traffic senders. In general, the schedulers at the nodes may not have exact knowledge of this available link capacity. It is also necessary to exchange messages between neighboring nodes. This adds to the overhead, and this overhead is often non-negligible.

To overcome these limitations we present a *scheduling policy which attains the maxmin fair rates in a multirate, multicast network, without computing them beforehand*. No such scheduling policy is known for multirate multicast networks. As opposed to the fair queuing strategies, this policy can operate without any knowledge about the actual maxmin fair rates. Our scheduling policy is geared towards attaining maxmin fair rates, whereas the fair queuing strategies can attain any feasible rates *once the desired rates are known*. Our scheduling policy does not need to know the link capacities, and does not assume centralized coordination or global information at any computing processor. Also, this policy works in a multirate scenario where the source hierarchically encodes its signal into several layers. Information contained in a higher layer packet is meaningful only if all the lower layer packets have been successfully decoded. So, any multirate scheduling policy should first strive to attain low packet loss for the lower layers, and use the residual bandwidth to serve the higher layers. Our scheduling policy not only attains the maxmin fair rates for the individual receivers, but also offers different quality of service to different layers. More precisely, loss rates of the lower layers are negligible while those of the higher layers are somewhat higher. It attains this distinction without any knowledge of the layer bandwidth.

We would describe the scheduling policy in greater details later. The basic idea is to serve the sessions in a round robin manner in each link. 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. Since we consider multicast sessions, congestion of all links on the path of the session originating from the destination node of the link must be considered. For example in figure 3, the decision to transmit a session 1 packet in link e_3 during its round robin turn depends on the congestion in links e_4 and e_5. The same decision for session 2 is based on the congestion in link e_5 only. Also, a session always gives priority to a lower layer packet over a higher layer packet.

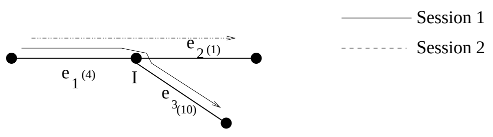

Fig. 1. An example network demonstrating that fair share in a link depends on congestion in other links. The numbers in brackets, () denote the capacities of the respective links. For example, e_1 has capacity 4 units.

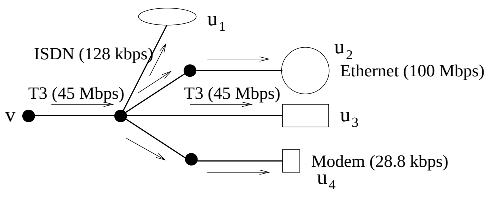

Fig. 2. A sample network showing network heterogeneity. The network has one session with four receivers. The paths to the receivers have widely varying bandwidth.

We conclude this section with a review of prior work in multicast fairness. Tzeng *et al* studies the problem of fair allocation of bandwidth to multicast sessions under the constraint that all receivers of the same session must receive service at the same rate[18]. However this has the potential of overwhelming the slow receivers and starving the fast ones. Rubenstein *et. al* have formally shown that fairness properties of a multicast network improves if multi-rate transmission is used instead of single rate transmission and have presented a centralized algorithm for computing the maxmin fair rates[14]. Well known network protocols for multirate multicast transmission, RLM (Receiver-driven Layered Multicast)[13] and LVMR (Layered Video Multicast with Retransmissions)[11] do not provide fairness among sessions[12]. Li *et al* proposes a scheme for fair resource allocation for multi-session layered video multicast which strives to rectify this defect in RLM and LVMR[12]. The authors present empirical evidence that the scheme improves fairness among sessions for networks with multiple video sessions sharing only one link. But, there is no experimental or analytical evidence that the scheme attains fair allocation for more complex networks, with sessions sharing several links with each other. In absence of further mechanisms, like elaborate scheduling policies, it may not be possible to establish conclusively that the scheme attains fair allocation of rates as per some well defined notion of fairness, like maxmin fairness for example.

II. Network Model

We consider an arbitrary topology network with N multicast sessions and M receivers in all. 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 or receivers. 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

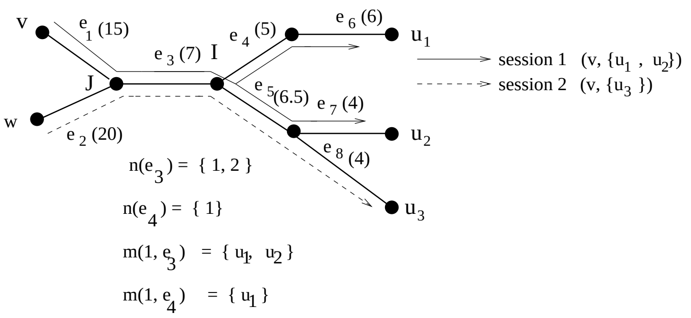

Fig. 3. Session 1 consists of receivers u_1 and u_2, and session 2 has receiver u_3. The capacity constraint for link e_3 is $\max(r_1, r_2) + r_3 \leq 7$ and that for link e_5 is $r_2 + r_3 \leq 6.5$. The maximum rates are 5 for session 1 and 20 for session 2. Hence, $r_1 \leq 5$, $r_2 \leq 5$ and $r_3 \leq 20$. The maxmin fair allocation is $(3.75, 3.25, 3.25)$ for receivers u_1, u_2 and u_3 respectively. Under this allocation, $\lambda_{1e_3} = 3.75$ and $\lambda_{1e_5} = 3.25$.

some well known multicast routing protocol like DVMRP[5], CBT[1], etc. in an internet type network.

To ensure fairness in a multirate network, we need to consider fair rate allocation for the receivers separately, instead of those for the overall sessions. We assume that every source i has a maximum rate p_i. This maximum rate can be infinity, if the source is greedy and desires to transmit traffic whenever possible. Rate allocation is an M-dimensional vector $(r_{11}, \ldots, r_{1m_1}, \ldots, r_{i1}, \ldots, r_{im_i}, \ldots, r_{N1}, \ldots, r_{Nm_N})$, with r_{ij} being the rate allocated to the jth receiver of the ith session. For simplicity we will use a single index, henceforth. A rate allocation $(r_1, \ldots, r_M)$ is feasible, if the rate for every receiver is less than or equal to the maximum rate for its source. Besides, total bandwidth consumed by all sessions traversing a link can not exceed the capacity of the link. Bandwidth consumed by a session in a link is equal to the maximum of the bandwidth allocated to its receivers downstream of the link. Bandwidth consumed by a session in one link may be different from that in another link, as different receivers of the same session may have different bandwidth. Formally, a rate allocation $\vec{r} = (r_1, \ldots r_M)$ is a feasible rate allocation if

1. Rate allocated to any receiver j is less than or equal to the maximum rate of its session i, i.e., $r_j \leq p_i$, if j is a receiver of session i,
2. Also, $\sum_{i \in n(l)} \max_{j \in m(i,l)} r_j \leq C_l$ (capacity condition),
   where $n(l)$ denotes the set of sessions passing through link l, $m(k,l)$ denotes the set of receivers of session k downstream of link l and C_l denotes the capacity of link l.

Figure 3 illustrates an example network with a few capacity and maximum rate constraint.

A feasible rate vector is maxmin fair if it is not possible to maintain feasibility and increase the rate of a receiver without decreasing that of any other receiver which has equal or lower rate. 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. Refer to the network of figure 3 for an example of maxmin fair allocation.

As we have discussed before, loss rates should be different for different layers under hierarchical encoding. This is because lower layers contain more important information than higher layers. Let layer i emitted by a source consume b_i units of bandwidth. Let the bandwidth r allocated to a receiver be sufficient to serve all packets of the first k layers and a portion of the packets of the $k+1$th layer, i.e., $\sum_{i=1}^{k} b_i < r < \sum_{i=1}^{k+1} b_i$. In the ideal scenario, the receiver should receive all packets of the first k layers and at least $\frac{r - \sum_{i=1}^{k} b_i}{b_{k+1}}$ fraction of packets of the $k+1$th layer and possibly no packet from the higher layers. We will show that our scheduling policy not only delivers packets in accordance with the maxmin fair rates, but also attains the above loss rates for the layers, i.e., the first k layers suffer negligible packet loss and the loss rate for the $k+1$th layer is $1 - \frac{r - \sum_{i=1}^{k} b_i}{b_{k+1}}$ in this case. We assume that receiving a portion of the packets of a layer improves the reception quality as compared to receiving no packet of the layer. This assumption is realistic as packet loss causes a graceful degradation in signal quality in many coding schemes[6]. For example, in [6] signal exhibits a reasonable enhancement in quality above the lowest layer for 10% packet loss in the next higher layer. This transition is gradual up until 100% packet loss in the next higher layer. Real time transmission in internet uses these coding schemes, because packet losses are frequent. If the fair rates can be computed, and the layer bandwidth are known at all intermediate nodes, then well known scheduling policies like fair queuing strategies[4][3][7][9], MMRS[**?**] can attain the fair rates and serve the appropriate layers. However, layer bandwidth may not be known at the intermediate nodes. More importantly rate computation algorithms require knowledge of system parameters like link capacities etc.[15]. Also, rate computation strategies need frequent exchange of computation related messages between nodes, and this adds to the overhead. Depending on the bandwidth constraint, this additional overhead may be unacceptable. It would be useful to have a scheduling policy which serves the appropriate layers and attains the fair rates in multirate multicast networks, without computing the fair rates beforehand or assuming knowledge of layer bandwidth. We present the intuition behind the policy we develop in the next section.

III. Back pressure based flow control for fairness

Intuitively, a simple round robin scheduling in every link should split link bandwidth fairly among all sessions traversing a link. However, this scheme has a deficiency, which we explain below. A session traverses multiple links, and different links offer different bandwidth to the session. The link which offers minimum bandwidth to a session is denoted as the bottleneck link of the session. For example, links e_1 and

e_2 are the bottleneck links of sessions 1 and 2 respectively in figure 1. A session should not be served at a rate higher than that offered by its bottleneck link, in any link on its path. This would cause congestion and packet loss in the bottleneck link. Also, a significant portion of the bandwidth of non-bottleneck links will be wasted in serving packets which do not reach the destination. A simple round robin scheduling does not ensure that the service rate of a session in any link on its path is equal to that in its bottleneck link. Credit based flow control can be used for conveying the bottleneck information implicitly.

Hahne[10] used credit flow control for attaining fairness in unicast networks. A credit value (W) is decided apriori. The basic idea is to keep track of the number of session packets waiting for transmission at the destination node of a link. For example, the flow control strategy must keep track of the number of session 2 packets waiting at node I of figure 1 for controlling the flow of session 2 in link e_1. If this number is less than the credit value and the session has packets for transmission, then the session transmits a packet in the link, when it is sampled. If this number is equal to the credit value, then the session does not transmit even if it is sampled, and has packets for transmission. For example, session 2 will not transmit packets in link e_1 if there are W session 2 packets at I. Consider figure 1 to see how credit based flow control works. Simple round robin offers 2 units of bandwidth to both sessions 1 and 2 in e_1. Now, e_2 serves session 2 at a lower rate (1 per unit time). Thus there will be an accumulation of session 2 packets at node I and hence session 2 will often not transmit packet over e_1 even when it is sampled. Thus link e_1 will serve the session at a rate lower than it otherwise would. Now link e_3 can transmit session 1 packets at a higher rate than e_1. So node I will not have session 1 packets often, and this will reduce the transmission rate for session 1 packets in e_3. In fact, any link l serves any session i at the same rate as the bottleneck link of the session.

Credit flow control presents some inherent complications for multirate multicast networks. We would first explain the difficulties and then present our approach in overcoming the complications. The path of a session may consist of multiple links originating from the same node. For example, session 1 traverses through links e_4 and e_5 originating from node I in the network of figure 3. These links serve the session at different rates. As a result, the number of packets of the same session, waiting at a node for transmission in different links are different. This number may be less than the credit value for one link, but may be greater than the credit value in another link. Consider figure 3 for example. It can happen that the number of session 1 packets waiting at node I for transmission in link e_4 is 2 and the same number for link e_5 is 10, and the credit value is 3. Thus it is not clear how credit flow control can be used to determine when a link should serve a session, and when it should not. Also the choice of the flow control scheme should be such that the rate of a session in a link l is equal to the maximum of that in the links originating from the destination node of l. Rate of session 1 in link e_3 should be equal to the maximum of that in links e_4 and e_5 in figure 3. We show that this can be attained by allowing a

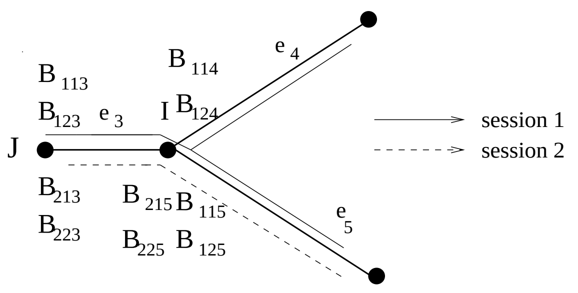

Fig. 4. We show a section of the network shown in figure 3(B_is). We assume that both sessions transmit two layers only. We show the different queues at the nodes. Here, B_{ijk} is the queue of session i layer j packets waiting for transmission in link e_k. The associated queues are shown in figure 5.

link l to serve a session if the number of packets of the session in at least one of the links originating from the destination of l is less than the credit value W. For example in figure 3, the scheduler for e_3 keeps track of the number of session 1 packets waiting at I for transmission in e_4 and the number waiting for transmission in e_5. If at least one of these is less than W, the scheduler transmits a session 1 packet in e_3 in its round robin turn. For the time being, we assume that the queue lengths at destination node of a link are known at the link scheduler, which is normally at the origin of the link. We discuss this assumption later.

Since service rate of a session in a link is equal to the maximum of the service rates of the links downstream, the source of a link may receive packets at a rate higher than the link can serve. For example, if link e_4 serves session 1 faster than link e_5 does in figure 3, then link e_3 will serve packets at a rate equal to that of e_4 and consequently, e_5 receives packets at a rate higher than it can serve. Thus, there will be packet loss at intermediate nodes (node I in this example), since the node buffers are finite. Let the credit value be W. In the unicast case, a link l does not serve a packet if the destination node has W packets. So there is no packet loss in the intermediate nodes, if the sizes of node buffers are W, or greater. In our case there will be packet loss as long as the buffers are finite. So, our task is to attain the maxmin fair rates in presence of packet loss, and also to regulate the loss so that packets are lost from higher layers only. Again, this is because hierarchical transmission has the property that a layer yields useful information, only if packets from all lower layers have been successfully decoded. We attain this objective by using different priorities for different layers.

IV. Description of the Policy

We propose a scheduling policy based on prioritized round robin with window flow control for multirate multicast networks. Here, $B_{(i,k,l)}(t)$ denotes the number of layer k packets of session i waiting for transmission in link l at time t. Packets of the same session waiting for transmission in multiple links originating from the same node, need not be stored in separate memory location. So, the quantities $B_{(i,k,l)}$s represent logical rather than physical buffers. A node needs to keep track of $B_{(i,k,l)}(t)$s for all layers k of all sessions i traversing through any link l originating from the node. Refer to figures 4, 5 and 6

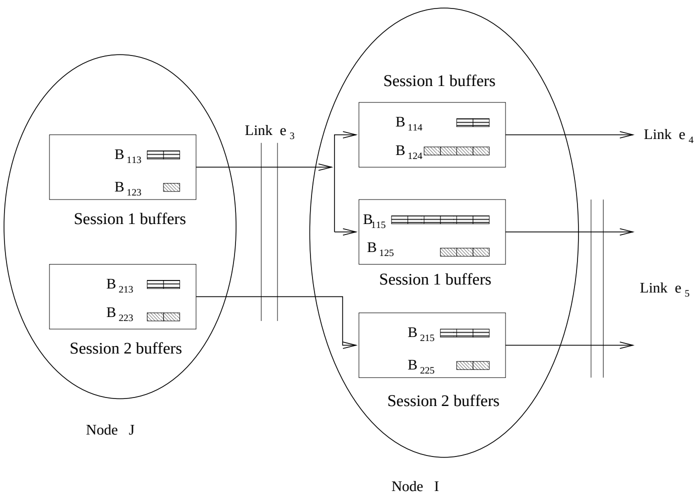

Fig. 5. We show the logical buffers of figure 4. Note that a session 1 packet may wait for transmission in both links e_4 and e_5. It is not necessary to have separate copies of the packet waiting at the same node, as this figure shows. This figure represents the logical queues only. The logical queues may be maintained by pointers. Physical queue corresponding to these logical queues is shown in figure 6.

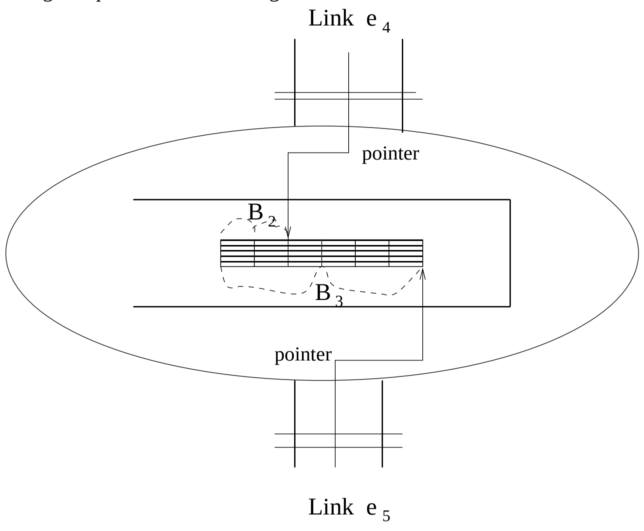

Fig. 6. We show a physical buffer at node I of figures 4 and 5. We assume that the switches are input queued. This physical buffer holds layer 1 packets of session 1 transmitted via link e_3, and corresponds to logical buffers, B_{114} and B_{115}. Packets are replicated only at the transmission epoch. So, the buffer holds 6 packets in all. All 6 need to be transmitted in link e_5 and only the last 2 need to be transmitted in link e_4. The first 4 have already been transmitted in link e_4. Hence, $B_2(B_{(114)})$ contains 2 packets and $B_3(B_{(115)})$ contains 6 packets in figure 5. Every link maintains a pointer at the first packet it needs to transmit.

for examples. We assume finite size physical buffers. Hence, memory limitations force $B_{(i,k,l)}(t)$s to be less than or equal to a quantity G, for all sessions i, layers k, link l and time t. We assume credit value W, and $G > W$.

We assume that a session source is connected to the network through an access link. No other session traverses the access link. For example, link e_1 is the access link of session 1 and link e_2 is the access link of session 2 in figure 3. We first describe the scheduling for access links. Whenever the source of a session i desires to transmit a packet, the access link checks whether at least one of the logical buffers at the destination of the link has less than W session i packets of the corresponding layer. If so, the access link transmits the packet, else it does not transmit the packet. If any logical buffer at the destination of the link has G session i packets of the corresponding layer (i.e., the logical buffer is "full"), the packet is not added to the queue of packets waiting to be transmitted in the link and this packet is lost for this link. More formally, let session i source wish to transmit a layer k packet at time t over access link l. Access link l transmits the packet if $\min_{l' \in \kappa(i,l)} B_{(i,k,l')}(t) < W$, where $\kappa(i,l)$ is the set of links on session i path originating from the destination node of access link l. When access link l transmits a session i packet of layer k, it joins the queue for transmission in all links in $\kappa(i,l)$, except those which already have G session i packets of layer k. If $B_{(i,k,l')}(t) = G$, for some link $l' \in \kappa(i,l)$, the transmitted packet is not added to $B_{(i,k,l')}(t)$, i.e.,the packet is lost for link l' and receivers downstream of l'. If $\min_{l' \in \kappa(i,l)} B_{(i,k,l')}(t) = W$, then access link l does not transmit the packet. (Note that $\min_{l' \in \kappa(i,l)} B_{(i,k,l')}(t)$ can not exceed W at any time t.) Refer to figure 3 for an example. Here, $\kappa(1,e_1) = \{e_3\}$. So, access link e_1 transmits a layer j packet of session 1 at time t if $B_{(1,j,e_3)}(t) < W$. Here, no packet transmitted over e_3 is lost as packet transmission takes place only when $B_{(1,j,e_3)}(t) < G$ as $G > W$. Packet loss can take place only when $\kappa(i,l)$ contains more than one links. We assume that the scheduler for link l knows $\min_{l' \in \kappa(i,l)} B_{(i,k,l')}(t)$, for all sessions i traversing link l and all times t. We will argue in sections V and VI that this assumption can be relaxed.

Now we consider the scheduling of non-access links. When a link l is ready to transmit a packet (which happens when it has finished transmitting the previous packet), it samples all sessions traversing the link starting from the one after the session last served. When a session i is sampled, it finds out whether it can send a lowest layer (layer 1) packet. It sends a lowest layer packet if at the sampling time t,

1. there are session i layer 1 packets waiting for transmission in link l i.e., if $B_{(i,1,l)}(t) > 0$, and
2. if the number of session i layer 1 packets waiting for transmission in link l' is less than W for at least one link l' originating from the destination of l, i.e., if $\min_{l' \in \kappa(i,l)} B_{(i,1,l')}(t) < W$.

If session i can not transmit layer 1 packet because of the violation of either of the above conditions, it tries to send a next higher layer packet. If it can not send the next higher layer packet, it tries the next and so on. In general, session i sends a layer k packet, if

1. it can not send a layer $1, \ldots, k-1$ packet,
2. $B_{(i,k,l)}(t) > 0$, and
3. $\min_{l' \in \kappa(i,l)} B_{(i,k,l')}(t) < W$.

When link l transmits a session i packet of layer k, it joins the queue for transmission in all links in $\kappa(i,l)$, except those which already have G session i packets of layer k. If $B_{(i,k,l')}(t) = G$, for some link $l' \in \kappa(i,l)$, the transmitted packet is not added to $B_{(i,k,l')}(t)$, and the packet is lost for link l' and receivers downstream of l'.

If session i can not send any packet, the link l samples the next session. If none of the sessions can send a packet, the link idles for some time, before starting the sampling once again. However, if some session transmits a packet, after transmitting the packet, the link samples the sessions, starting from the session after the one last served. We explain the policy in the following example.

Example IV.1: Consider the network shown in figure 4. It is a part of the network of figure 3. The initial logical buffer contents are shown in figure 5. The quantities B_{ijk} of the figure denote $B_{(i,j,e_k)}$. We assume $W = 3$ and $G = 6$. Here $\kappa(1, e_3) = \{e_4, e_5\}$ and $\kappa(2, e_3) = e_5$. Session 2 was served last. So the link samples session 1 for packets. Note that $B_{(1,1,e_3)} > 0$ and $\min_{l' \in \kappa(1,e_3)} B_{(1,1,l')} = B_{(1,1,e_4)} = 2 < W$. So link e_3 transmits a session 1 layer 1 packet. The packet is added to the queue for transmission in link e_4, but not in link e_5 because $B_{(1,1,e_5)} = 6 = G$. This packet is lost for link e_5 and the downstream receiver u_2.[*] The modified buffer contents are $B_{(1,1,e_3)} = 1$, and $B_{(1,1,e_4)} = 3$. The rest of the buffer contents remain the same as before.

Next the link samples session 2 for packets. Session 2 can not transmit a layer 1 packet because $\min_{l' \in \kappa(2,e_3)} B_{(2,1,l')} = B_{(2,1,e_5)} = 3 = W$. However, it can transmit a layer 2 packet because $B_{(2,2,e_3)} > 0$, and $B_{(2,2,e_5)} = 2 < W$. So it transmits a layer 2 packet. This packet is added to the queue for transmission in link e_5. The modified buffer contents are $B_{(2,2,e_3)} = 1$ and $B_{(2,2,e_5)} = 3$. The rest of the buffer contents remain the same as before.

Next the link samples session 1 for packets. Session 1 can not send any layer 1 packet because $\min_{l' \in \kappa(1,e_3)} B_{(1,1,l')} = 3 = W$. It can not send any layer 2 packet because $\min_{l' \in \kappa(1,e_3)} B_{(1,1,l')} = 3 = W$. So the link samples session 2 for packets. Session 2 can not send any layer 1 packet because $\min_{l' \in \kappa(2,e_3)} B_{(2,1,l')} = 3 = W$. It can not send any layer 2 packet because $\min_{l' \in \kappa(2,e_3)} B_{(2,1,l')} = 3 = W$. So the link idles for some time. It starts serving only when links e_4 and e_5 serve some packets and reduce $B_{(i,k,l)}$ for $i \in \{1, 2\}$, $k \in \{1, 2\}$ and $l \in \{e_4, e_5\}$.

V. Performance Evaluation

We discuss the performance of this scheduling policy. We would present analytical and simulation results. The following theorem indicates that this scheduling policy allocates maxmin fair rates to all receivers. Also, the packet loss is concentrated in the highest layer served. We have omitted all proofs throughout, on account of space restrictions. Refer to technical report[16] for proofs and details.

Let session i source transmit γ_i layers in all. Bandwidth of the kth layer of the ith session is $b_{i,k}$. Also, $\sum_{k=1}^{\gamma_i} b_{i,k} = p_i$, where p_i is the maximum rate of source i. Session i source is "well-behaved" if it attempts to transmit at most $b_{i,k}(v - u) + \xi_{i,k}$ packets and at least $b_{i,k}(v - u) - \xi_{i,k}$ packets of the kth layer in any interval $[u, v)$. The parameters $\xi_{i,k}$ are "transmission jitters." Many sources are well-behaved in practice.

[*]It may happen that the new packet is added to the queue but another old packet in the queue $B_{(1,2,e_5)}$ is dropped. The bottomline is that one packet is lost. The lost packet may either be the new one or an old one.

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.

Theorem 1: Let all sources be well behaved. Let maxmin fair rate of receiver j be r_j. Then receiver j receives at most $r_j(v - u) + \delta_j$ packets and at least $r_j(v - u) - \delta_j$ packets in any interval $[v, u)$, if $G \geq G^*$ and $W \geq W^*$. The constants $\delta_j, j = 1, \ldots, M, G^*, W^*$ depend only on the system parameters, e.g., path lengths, transmission jitters etc., and not on the length of the interval $[v, u)$ or on the time instants u, v.

Theorem 1 implies that the long term average rates of the receivers are equal to the maxmin fair rates. Also, the number of packets delivered to the receivers in any interval differs from the maxmin fair rates by at most a constant, and the value of this constant does not depend on the length of the interval. This shows that the policy is fair in short intervals as well, in some sense. The next theorem shows that packet loss is concentrated in the highest layer served. We introduce another notation. The quantity $(u)^+$ stands for $\max(u, 0)$.

Theorem 2: Let all sources be well behaved. Let $G \geq G^*$ and $W \geq W^*$. Consider any receiver j with maxmin fair rate r_j. Let receiver j belong to session i. The number of layer k packets lost in the path of receiver j in any interval $[v, u)$ is at most $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - r_j, b_{i,k}\right)\right)^+ (v - u) + \Delta_{j,k}$. Here, $\Delta_{j,k}$ is a finite constant which depends only on system parameters and not on the length of the interval $[v, u)$, or time instants u, v.

Let receiver j belong to session i. If the sum of the transmission rates of the first k layers of session i is less than or equal to the maxmin fair rate of receiver j (i.e., $\sum_{w=1}^{k} b_{i,w} \leq r_j$), then ideally receiver j should receive all layer k packets sent by the source. In this case, $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - r_j, r_j\right)\right)^+ = 0$, and hence by Theorem 2, the number of layer k packets lost in the path of receiver j in any interval $[v, u)$, is at most $\Delta_{j,k}$. Thus receiver j observes a long term loss rate of 0 for layer k packets in this case. Also, the number of layer k packets lost in any interval is upper bounded by a constant $\Delta_{j,k}$, which does not depend on the size of the interval. If the maxmin fair rate is between the sum of the transmission rates of the first $k - 1$ layers and the sum of the transmission rates of the first k layers (i.e., $\sum_{w=1}^{k-1} b_{i,w} \leq r_j < \sum_{w=1}^{k} b_{i,w}$), then receiver j should receive packets of layer k partially. Theorem 2 says that in this case, the packet loss in any interval $[v, u)$, is upper bounded by $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - r_j, b_{i,k}\right)\right) (v - u) + \Delta_{j,k}$. So the long term loss rate is $\min\left(1, \frac{\sum_{w=1}^{k} b_{i,w} - r_j}{b_{i,k}}\right)$. Thus, the residual bandwidth left after serving the first $k - 1$ layer packets is used to serve the kth layer.

As the theorems indicate, the above results are guaranteed only when the credit value and the buffer size exceed certain lower bounds. These lower bounds depend on system parameters. In practice, it may not be possible to compute these lower bounds and decide the credit and buffer sizes accordingly. Systems have buffers of fixed size, and scheduling policies need to function within the existing memory constraint. The proposed scheduling policy shows a gradual performance improvement with increase in buffer and credit sizes.

We introduce the notation of the *rank* of a receiver. Let there be J distinct maxmin fair receiver rates. If the maxmin fair rate of receiver j is the mth smallest among the distinct maxmin fair rates, then the rank of receiver j is m.

Theorem 3: There exists a sequence of constants, $W_1, W_2, \ldots, W_J$, and $G_1, G_2, \ldots G_J$ such that, if $W \geq W_m$, and $G \geq G_m$, then all receivers of rank m and above, receive service at a rate greater than or equal to the mth smallest maxmin fair rate. All receivers of rank smaller than m, receive service at their maxmin fair rates. Also, the entire packet loss is concentrated in the highest layer served, for all receivers.

The analytical lower bounds on window and buffer sizes, $W_1, W_2, \ldots, W_J$, $(W^* = W_J)$, $G_1, G_2, \ldots G_J$ $(G^* = G_J)$ are pessimistic bounds. Simulations indicate that, in practice, service rates converge to the maxmin fair rates, and packet loss is concentrated in the highest layer for much smaller credit and buffer sizes.

The scheduling policy we described so far assumes that a link scheduler has complete information about queue lengths at the destination node of the link. More precisely, the assumption is that at all times t the scheduler for link l knows $\min_{l' \in \tau(i,l)} B_{(i,j,l')}(t)$ for all layers j of all sessions i traversing link l. This is a reasonable assumption in networks with hop by hop congestion feedback and negligible propagation delay. Though many current day networks employs end to end congestion control only, many others use hop by hop back pressure based congestion control. In fact, in the LAN context, recent simulation results show that hop by hop back pressure can be better than TCP for dealing with short-lived congestion[17]. Though terrestrial networks have negligible propagation delay, propagation delay is significant for some other networks, e.g., those with satellite links. In these networks, the feedback will typically be delayed, i.e., at time t the scheduler will have information about destination queue lengths at some previous time t'. In this case, the policy remains the same, except that the scheduler takes decisions based on the information it has currently. The service rates of the receivers converge to the maxmin fair rates as before, even in the presence of propagation delays. Packet loss is concentrated in the highest layer served as before. Theorems 1 to 3 hold in this case as well, as long as the propagation delays are bounded[16]. The window and buffer thresholds, W_i^* and G_i^*s depend on link propagation delays also. Refer to [16] for formal proof. The intuitive reason is as follows. The available information can differ from the actual queue lengths at the destination node by at most a constant which depends on the propagation delay and link capacities. This constant will increase the constants $\Delta_{i,j}$s but the long term throughputs are independent of these($\Delta_{i,j}$s). Our simulation studies further confirm this observation.

Now we present the results of our experimental studies. We considered a 15 session 400 node random network for experimental evaluation. Nodes are points on a $20X20$ grid. There

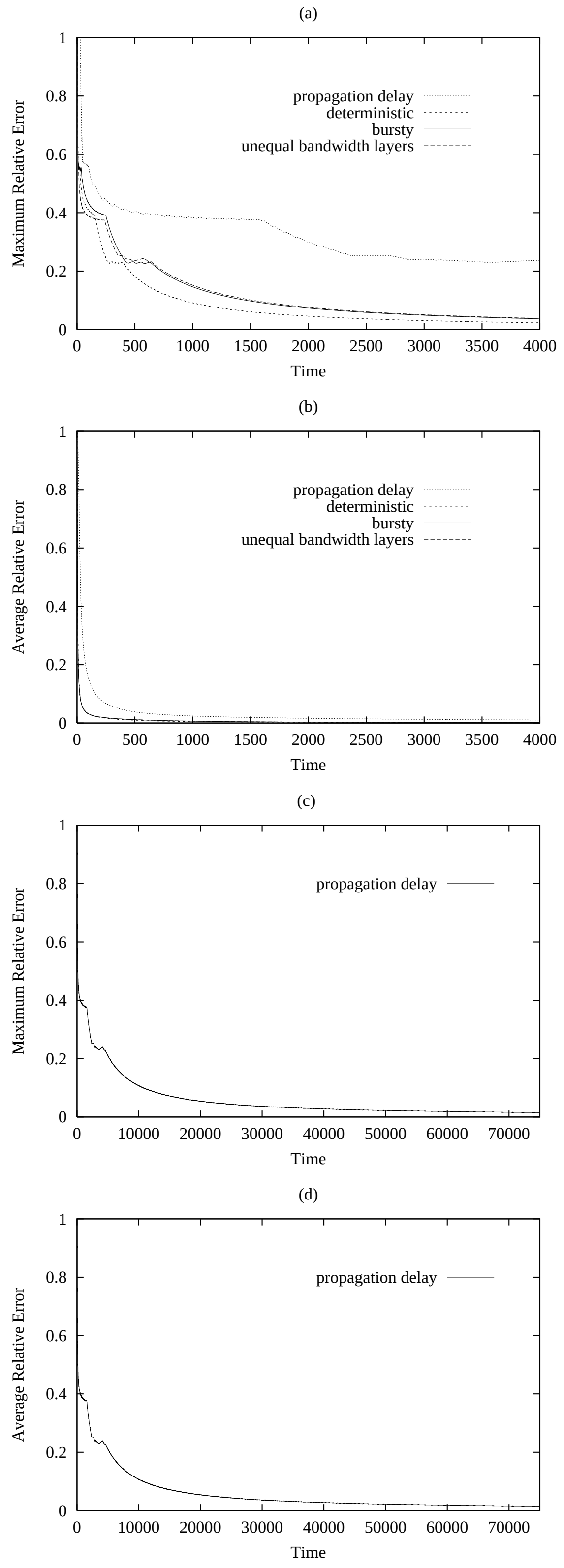

Fig. 7. These figures demonstrate the convergence of the packet delivery rates attained by the proposed scheduling policy to the maxmin fair rates. We have plotted the convergence errors as a function of time for different traffic models.

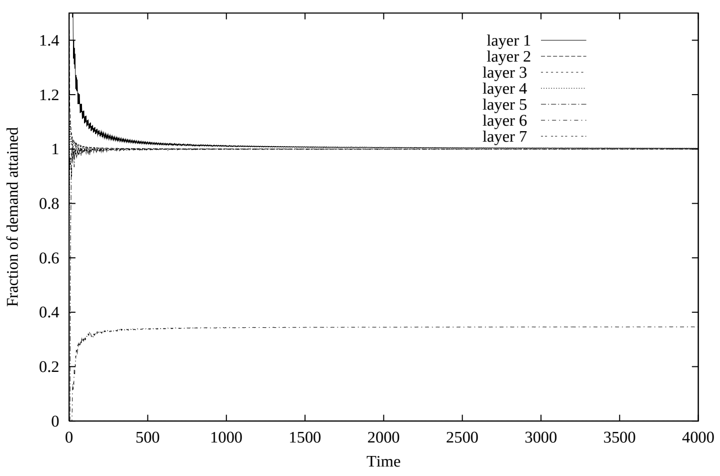

Fig. 8. The figure shows the fraction of packets of different layers delivered to one particular receiver in the random network.

exists an edge between any two nodes with a probability (p) that depends on the euclidean distance between the nodes (d) ($p = \exp(\alpha(1 - d))$), where α is the decay constant. We assumed $\alpha = 2$. We adopted this edge probability model because distant nodes are less likely to have an edge between them. Source and receivers of every session have been selected randomly. There are 96 receivers in all, i.e., average session size is 6.4. The session route consists of shortest paths between the source and the destinations. All sources transmit 20 layers.

Figure 7 demonstrates the convergence of the service rates attained by different receivers to the respective maxmin fair rates for different traffic patterns. We studied the difference between the actual rate of packet delivery for any receiver and its maxmin fair rate. The attained rate at time t for receiver s is the total number of packets delivered to s in the interval $[0, t)$ divided by the time t. If maxmin fair rate of a receiver s is r_s^m, and the attained rate at time t is $r_s^a(t)$, then error for receiver s is $|1 - \frac{r_s^a(t)}{r_s^m}|$ at time t. Figures $7(a)$ and $7(c)$ plot the maximum relative error, and figures $7\,(b)$ and $7(d)$ plot the average relative error, the maximum and average are taken over all receivers. We considered deterministic traffic patterns and traffic patterns with jitters. A traffic pattern is deterministic at rate d, if packets are generated once every $1/d$ seconds. A jittery (d, ξ) traffic pattern transmits traffic at a long term rate of d per unit time with a jitter of ξ. In this traffic model, at most $dt + \xi$ and at least $dt - \xi$ packets are generated in any interval of length t. In the deterministic case (curve labeled "deterministic" in figures 7(a) and 7(b)), packets of all layers of all sources are generated periodically at rate 1 per unit time. For the traffic model with jitters (curve labeled "bursty" in figures 7(a) and 7(b)), every source transmits packets of every layer as per a jittery$(1, 3)$ model. For unequal bandwidth layer case (curve labeled "unequal bandwidth layers" in figures 7(a) and 7(b)), source i generates layer j packets as per a jittery$(b_{i,j}, \xi_{i,j})$ model. The layer bandwidth $b_{ij}s$ and transmission jitters $\xi_{i,j}$s have been chosen randomly, and are different for different i, js. For example, the first 3 layers of session 1 have bandwidth and jitter of $(.21, 2)$, $(2.78, 3)$ and $(1.81, 0)$ respectively, whereas the first 3 layers of session 2 have bandwidth and jitters of $(3.5, 2)$, $(3.21, 0)$ and $(3.30, 1)$ respectively. All these curves ignore propagation delays. Convergence is fast for all these cases. The average error converges to 0 much faster than the maximum error, indicating that the attained rates of most of the receivers converge to the maxmin fair rates very fast, whereas convergence is relatively slow for a few others. Convergence is fastest for the deterministic traffic model. The results for the deterministic traffic model were obtained with credit value of 5 and buffer size of 10 units only. We used credit and buffer sizes of 8 and 16 for the other two traffic models with jitters. Thus, the attained rates converge to the maxmin fair rates for small credit and buffer sizes.

We also considered effect of propagation delays and subsequently delayed feedback. In our model, propagation delay of a link is equal to the euclidean distance between the end points of the link. The "propagation delay" curves in figures 7(a) to figures 7(d) study the errors for the "unequal bandwidth layer" traffic model with propagation delay. Figure 7(c) and 7(d) shows the errors for this case for a longer range of time. We used credit and buffer sizes of 100 and 200 respectively in presence of propagation delays. As expected, long term rates still converge to the maxmin fair rates, but the convergence is slower than when propagation delay is ignored. Delayed feedback increases buffer and credit size requirements. However, these requirements are still reasonable.

Figure 8 demonstrates the different loss rates suffered by different layers and exhibits that packet losses are confined to the highest layer served in the case we studied. It actually shows the fraction of packets delivered to one particular receiver in the random network. This fraction for layer i is the ratio between the number of packets of layer i delivered to the receiver in $[0, t)$ and the product of the layer bandwidth b_i and time t. The maxmin fair service rate of this receiver is 9 packets per unit time. The traffic model is the same as the "unequal bandwidth case" discussed before. We have ignored propagation delays in this case. The source for this receiver transmits packets of the first 6 layers at rate (b_is) 0.21, 2.78, 1.81, 3.01, 0.84 and 1 per unit time, respectively. The transmission jitters (ξ_is) for these layers are 2, 3, 0, 3 0 and 1 respectively. Ideally the receiver should receive all packets of the first 5 layers, 36% packets of layer 6 and possibly no packet of any higher layer. Analytical results guarantee delivery of packets of layer i at rate b_i, if $i \in \{1, \ldots, 5\}$ and rate $0.36b_i$ if $i = 6$. As the figure shows, for the first 5 layers, the fraction of packets delivered to the receiver converges to 1 very fast. In fact, it is greater than 1 for some lower layers initially. This is because the source sends an initial burst of packets for every layer on account of the transmission jitters. The network is able to deliver some of these bursts as well, for the lower layers. Hence the ratio between the number of packets delivered and b_it is initially greater than 1 for the lower layers. The fraction of packets of layer 6 delivered to the receiver converges to 0.36

as well. Very few packets of other layers reach the receiver. This shows that packet loss is confined to the highest layer served (layer 6 in this case). Also note that the long term rates are always higher for a lower layer as compared to that for a higher layer. Window and buffer sizes are 8 and 16 respectively for this simulation.

VI. Conclusion

We discuss some salient features of this scheduling policy in this concluding section.

- If the maximum rates of sessions form a feasible rate allocation, then the maximum rates are maxmin fair. Thus the proposed scheduling policy attains the maximum rates, as long as these rates are feasible. In other words, resource limitation permitting, the proposed scheduling policy satisfies all users.
- As we have mentioned before, the scheduling policy is adaptive as it does not require any knowledge of the maximum rates of the users, layer bandwidth, or the user traffic statistics. The results are independent of the hierarchical structure of signals and unequal bandwidth layers are permitted. Also, a malicious session can not increase the throughput of its receivers, by choosing layer bandwidth suitably. Maxmin fairness of receiver throughputs are guaranteed irrespective of the layer bandwidth. This is useful because in real networks these parameters are not generally known.
- The proposed scheduling policy is simple to implement. No intensive computation is involved anywhere.
- A link scheduler takes scheduling decisions whenever it is free to transmit a packet. It need not synchronize with schedulers for other link.
- Note that this policy offers different quality of service to different layers. Layered traffic is a special case of priority traffic, with the lowest layer traffic having the highest priority, and the higher layers lower priority. It is possible to generalize this scheduling policy to attain maxmin fairness with priorities, by considering different priority sessions instead of considering different layers. This would allow service differentiation within the framework of fairness.
- The scheduling policy does not assume any particular drop policy. Physical buffers can follow any drop pattern. When a packet arrives and finds the buffer full, it is not necessary that the new packet is dropped, but an old packet may also be dropped to make place for the new packet. Dropping an old packet may be a better option for real time transmission, because packets arriving after a certain delay become useless. The routers may follow a drop tail (drop the new packet) or random drop (drop a randomly chosen packet in the queue) or drop head (drop the oldest packet) policy. The allocated rates will converge to the maxmin fair rates in all these cases.
- While describing the policy, we assumed that the access links are dedicated to the individual sessions. In general, access links can also be shared by several users. In that case, access link scheduling will be the same as scheduling for other links, i.e., the access link will use prioritized round robin based on window flow control. The layer priorities are also imposed on the access link scheduling similar to the priorities in other links. The assumption that an access link is used by only one session has been made for simplicity, and has no impact on the performance.
- The credit/buffer thresholds can be selected statically or dynamically. Initially an adhoc choice can be made for these values, and if the system does not show convergence after some time the values can be increased. For our experiments, we made an adhoc choice for these values. We selected sizes much lower than the computed thresholds, and found that even the large size networks we consider in this paper converge to maxmin fair rates for reasonably small buffer and credit sizes.

The disadvantages of this policy are as follows. Link schedulers need congestion information of the neighbors. Specifically, it needs to know whether the number of packets of a session at the destination node of the link is less than the credit value or not. Whenever, queue length of a session crosses the threshold of W, a message indicating the same can be sent to the source node. This message can be piggy backed in information packets as well. Thus overhead is low. As we have mentioned earlier, this hop by hop congestion feedback has certain advantages over end to end congestion control only, and is used in LAN based networks[17]. Also, we have shown analytically and experimentally that feedback delays do not alter the throughput, as long as the delays are bounded. This makes it suitable for implementation in networks where propagation delay is significant, e.g., networks with satellite links. It is worthwhile to point out that even though this is a back pressure based policy, it does not suffer from any deadlock since the back pressure on a session is exerted by the accumulation of the packets of the same session. Also, livelock can be avoided by using first come first serve scheduling among packets of the same layer of the same session. The other disadvantage is that this scheduling policy requires per flow states in the routers. The complexity increase is not as drastic as it seems, though. Arguing in the lines of Grossglausser and Bolot[8], implementing a multicast/multilayer service requires per-flow state in the routers anyway. So, the incremental cost of maintaining some more information for each flow and using this additional information in the scheduling policy is much smaller than that in the unicast case. However, if these additional flow states become an issue, then this policy can be used in the VPNs and intranets, and state aggregation may be resorted to in the backbones. In a nutshell, this scheduling policy is suitable for use in large, dynamic, high speed networks that have access to decentralized, delayed and partial information only. Finally, this scheduling policy assumes that intermediate routers are able to make class based distinction in multicast networks. Current day multicast capable routers are not able to do so. However, this is likely to happen in near future, especially as multicast communication becomes more popular. Newly suggested internet policies like diffserv and intserv already propose class based distinction in unicast networks to guarantee requisite quality of services. It is likely that these would be extended to multicast in future.

Summarizing, we have presented a scheduling policy which attains maxmin fair rates in multirate multicast networks, without computing the rates beforehand. The scheduling policy concentrates the packet loss in the highest layer served. This improves the quality of service in view of the characteristic of the hierarchical coding that reception of higher layer conveys useful information, only when all the lower layers have been successfully decoded. As discussed, this scheduling policy has several other attractive properties, which render it suitable for use in large, dynamic, high speed networks that have access to decentralized, delayed and partial information only. It allows the network to enforce fair allocations, without trusting the receivers to attain fairness. This scheme is computationally simple, has low message overhead and hence does not unduly overload the network. Also, we have assumed multirate transmission. Class based distinction is more difficult for multirate transmission than for unirate transmission. With a minor modification, this scheduling policy will work for unirate networks. In fact an extension of the policy in [10] will attain fairness in unirate multicast networks, as unirate multicast networks are not significantly different from unicast networks.

REFERENCES

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

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

[3] J. Bennett and H. Zhang. WF^2Q: Worst-cast Fair Weighted Fair Queuing *Proceedings of IEEE INFOCOM'* 96, *pp* $120-128$, San Francisco, CA, March 1996

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

[5] 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

[6] M. Ghanbari. Two-Layer Coding of Video Signals for VBR Networks. *IEEE Journal on Selected Areas in Communications*, VOL. 7. No. 5. June 1989

[7] S. Golestani A self-clocked fair queueing scheme for broadband applications. *Proceedings of IEEE' INFOCOM'* 94, *pp* $636-646$,INFOCOM, Toronto, CA, April 1994.

[8] M. Grossglausser and J. Bolot. On service models for Multicast Transmission in Heterogeneous Environments. *Proceedings of IEEE' INFOCOM'* 2000, Tel-Aviv, Israel, March, 2000

[9] 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 $'96$

[10] 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 $1024-1039$

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

[12] 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

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

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

[15] S. Sarkar and L. Tassiulas: Distributed Algorithms for Computation of Fair Rates in Multirate Multicast Trees *Proceedings of IEEE INFOCOM'* 2000, Tel Aviv, Israel, March 2000

[16] S. Sarkar: Fairness and Congestion Control in multirate multicast networks. *Phd Thesis*, University of Maryland, College Park, July 2000

[17] F. A. Tobagi and W. K. Noureddine, "Back-Pressure Mechanisms in Switched LANs Carrying TCP and Multimedia Traffic," *submitted to IEEE GLOBECOM'* 99, *Symposium on High-Speed Networks*, December, 1999

[18] 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