Back pressure based multicast scheduling for fair bandwidth allocation

Saswati Sarkar and Leandros Tassiulas

Abstract— **We study the fair allocation of bandwidth in multicast networks with multirate capabilities. In multirate transmission, each source encodes its signal in layers. The lowest layer contains the most important information and all receivers of a session should receive it. If a receiver's data path has additional bandwidth, it receives higher layers which leads to a better quality of reception. The bandwidth allocation objective is to distribute the layers fairly. We present a computationally simple, decentralized scheduling policy that attains the maxmin fair rates without using any knowledge of traffic statistics and layer bandwidths. This policy learns the congestion level from the queue lengths at the nodes, and adapts the packet transmissions accordingly. When the network is congested, packets are dropped from the higher layers; therefore, the more important lower layers suffer negligible packet loss. We present analytical and simulation results that guarantee the maxmin fairness of the resulting rate allocation, and upper bound the packet loss rates for different layers.**

Keywords: Multicast, Scheduling, Back pressure, Maxmin fairness

I. Introduction

Internet is moving fast from best effort service to class based service, where 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 bandwidth guarantees some minimum quality of service to all users. Attaining a fair allocation in a distributed manner is however a challenging problem, as fair allocation of bandwidth in a link depends on the congestion in the other links as well. In Fig. 1, intuitively, the fair allocation in link e_1 is 2 units for each session. But, session 2 can not use more than 1 unit, since the bandwidth in link e_2 is 1 unit. So, the fair allocations are 3 and 1 units for sessions 1 and 2 respectively.

Allocating fair bandwidth is even more complex in multicast networks, due to network heterogeneity. A multicast session has several receivers, and different receivers have different processing capabilities and different bandwidths in data paths. In Fig. 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.

The diverse bandwidth requirements of receivers can be accommodated by using multirate transmission to serve different receivers of the same session at different rates. The service rate of a session in a link is equal to the maximum of the rates of the session receivers downstream of the link. In multirate transmission, each source hierarchically encodes its signal in several layers. The lowest layer contains the most important information and all receivers of the session should receive it. If a receiver's data path has additional bandwidth, it receives higher layers which leads to a better quality of reception. For example, in Fig. 2, u_4 receives only the lowest layer, whereas u_2 receives many more layers. Hierarchical coding is useful for real time loss tolerant traffic like audio and video. We consider real time traffic in this paper.

We consider the allocation of maxmin fair rates [2] to the receivers. A rate allocation is maxmin fair, if the rate of a receiver can not be increased without reducing the rate of another receiver that has equal or lower rate. Maxmin fairness satisfies many intuitive fairness properties in a multirate multicast network [12], e.g., it distributes bandwidth fairly among different sessions traversing a link, and serves every receiver at a rate commensurate with the fair bandwidth share in its path. The fair bandwidth may be different for different receivers of the same session, e.g., in Fig. 3, the fair rates for u_1 and u_2 are 3.75 and 3.25, respectively. Receiver u_2 can not receive bandwidth higher than 3.25 because link e_5 in its path offers 3.25 units to both sessions 1 and 2.

We now describe the challenges in allocating maxmin fair rates in multirate multicast networks. Topologies of communication networks constantly vary due to periodic failure of links and routers. Composition of multicast groups also change frequently due to joining and leaving of receivers. These in turn require frequent changes in session routes leading to changes in the maxmin fair allocation. Furthermore, the bandwidth available for serving real time traffic also vary depending on the amount of data traffic which change rapidly and may not be readily known at the link schedulers. Finally, a challenge specific to multirate transmission is that the maxmin fair rate allocation may depend on the bandwidth consumed by each layer. This is because the information contained in a higher layer packet is meaningful only if all the lower layer packets have been successfully decoded, and thus the fair rates must be allocated so as to first limit the packet loss for the lower layers, and subsequently use the residual bandwidth to serve the

Parts of this paper have been presented at INFOCOM 2001, Anchorage, Alaska.

S. Sarkar is with the department of Electrical and Systems Engineering in the University of Pennsylvania, Philadelphia, PA, USA. Email: swati@ee.upenn.edu.

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

higher layers. Now, the layer bandwidths dynamically vary depending on the coding strategy and characteristics of the traffic, and will not be known at the intermediate routers in the path of a session. Thus, the rate allocation strategies that assume static topologies and static congestion levels, or assume knowledge of traffic statistics, layer bandwidths and available link bandwidths are not adequate. This motivates the design of adaptive scheduling strategies that attain the maxmin fair rates by gradually learning the network dynamics. There are several key challenges in designing such mechanisms. The learning strategies must be (a) decentralized as no scheduler knows the state of the entire network, (b) computationally simple as the routers can only devote limited processing cycles towards these computations and (c) may not know the values of many crucial parameters such as layer bandwidths and link bandwidths available for transmitting real time traffic. We solve these key challenges by exploiting scheduling dynamics.

Our contribution is to design a computationally simple, decentralized, adaptive scheduling policy that attains the maxmin fair rates without knowing the link capacities, global network topology, or traffic statistics. This policy adapts the packet transmissions in accordance with network conditions and congestion levels which it learns from the queue lengths at the nodes. Specifically, the policy samples sessions in each link in a round robin manner for transmitting a packet in the 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 queue lengths downstream of the link. Here, higher queue lengths downstream of a link indicates higher congestion levels downstream and prevents transmission of the packets of a session in upstream links (i.e., in links closer to the source) - such policies are denoted as "back-pressure" policies [15]. For example, in Fig. 3, the decision to transmit a session 1 packet in link e_3 during its round robin turn depends on the congestions in links e_4 and e_5. The decision for session 2 is based only on the congestion in link e_5. Also, a session always gives priority to a lower layer packet over a higher layer packet. This in turn confines all the packet losses to the highest layer it serves, even though the scheduling does not use any knowledge of the layer bandwidths. The techniques used for learning the congestion level and the scheduling among the layers eliminate the need for any new schedule computation when the topology or the traffic characteristics or the layer bandwidths change. The scheme is therefore robust. We analytically prove that the policy attains the maxmin fair rates.

Maxmin fair rates can also be allocated by first computing the fair rates and subsequently determining the service order for packets in the links so as to serve the packets as per the fair rates. We have presented distributed algorithms for computing the maxmin fair rates in [13]. Now, there exists scheduling policies that can attain any feasible rate allocation, once the feasible rates or at least the ratios between the feasible rates are known, e.g., fair queuing strategies [3][7]. But, computing the fair rates has several problems. For example, computation algorithms must

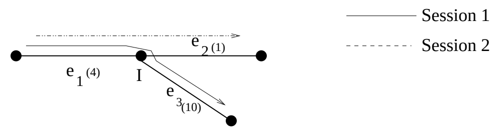

Fig. 1. An example network demonstrating that the fair bandwidth share in a link depends on congestion in other links. The numbers in brackets, (), denote the capacities of the respective links. For example, e_1 has capacity 4 units.

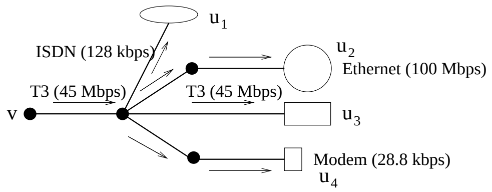

Fig. 2. A sample network showing network heterogeneity. The network has one session with four receivers.

know the bandwidth available for real time traffic which varies depending on the amount of data traffic and is not readily known at the link schedulers. Computation of fair rates also requires exchange of messages between neighboring nodes, leading to considerable additional traffic. Also, the fair rates must be re-computed when the packet arrival rates or the topology change. The scheduling strategy we propose in this paper eliminates the requirement for this pre-computation by using adaptive learning techniques, and thereby removes the above disadvantages.

We now review the related research in fairness in multicast networks. Tzeng *et al.* study the problem of fair allocation of bandwidth to multicast sessions under the constraint that all receivers of the same session must receive packets at the same rate [17]. But, if all receivers of a session are served at the same rate, then the slow receivers can be overwhelmed and the fast receivers starved. Rubenstein *et al.* have shown that fairness properties of a multicast network improve if multi-rate transmission is used instead of single rate transmission, and have presented a centralized algorithm for computing the maxmin fair rates [12]. Well-known network protocols for multirate multicast transmission, RLM (Receiver-driven Layered Multicast) [11] and LVMR (Layered Video Multicast with Retransmissions) [9] do not provide fairness among sessions [10]. Li *et al.* propose a scheme for fair allocation of bandwidth in layered video multicast that strives to rectify this defect in RLM and LVMR [10]. There is no analytical guarantee that the scheme attains fairness; the empirical evidence is for networks with only one link. Our research is complementary since we present a scheduling strategy that is guaranteed to attain maxmin fair rates in multirate, multicast networks.

The paper is organized as follows. In Section II, we describe our network model. In Section III, we motivate our policy. In Section IV, we describe our policy. In Section V, we evaluate the performance of our policy. In Section VI, we discuss the salient features of our policy. We outline the proofs in the appendix and refer to the technical report [14] for details.

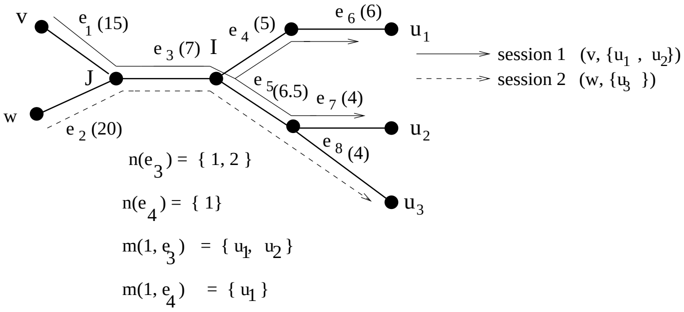

Fig. 3. Session 1 consists of receivers u_1 and u_2, and session 2 has receiver u_3. The capacity constraints for links e_3 and e_5 are $\max(r_1, r_2) + r_3 \leq 7$ and $r_2 + r_3 \leq 6.5$ respectively. The maximum rates are 5 and 20 for sessions 1 and 2 respectively. Hence, $r_1 \leq 5$, $r_2 \leq 5$ and $r_3 \leq 20$. The maxmin fair rates are $3.75, 3.25, 3.25$ for receivers u_1, u_2 and u_3 respectively.

II. Network Model

We consider a network with N sessions and M receivers. A session may have one or more receivers, and is identified by the pair (v, U), where v is the source and U is the set of receivers. The traffic from v is transmitted across a predetermined multicast tree to nodes in U. The tree can be established during connection establishment if the network is connection oriented, or can be established by a multicast routing protocol like DVMRP[4], CBT[1], etc. in a connectionless network like internet.

To ensure fairness in a multirate network, we must consider fair rate allocation for the receivers separately, instead of those for the overall sessions. Every source i has a maximum rate p_i; p_i is infinity if the source always has a packet to transmit. Rate allocation is an M-dimensional vector $(r_{11}, \ldots, r_{1m_1}, \ldots, r_{i1}, \ldots, r_{im_i}, \ldots, r_{N1}, \quad \ldots, r_{Nm_N})$, where r_{ij} is the rate of the jth receiver of the ith session. For simplicity, we will use a single index, henceforth.

Definition 1: A rate allocation $\vec{r} = (r_1, \ldots r_M)$ is a feasible rate allocation if

1. the rate of each receiver j is less than or equal to the maximum rate of its session i, i.e., $r_j \leq p_i$, and
2. the total bandwidth of all sessions traversing a link is less than or equal to the capacity of the link; the bandwidth of a session in a link is equal to the maximum of the bandwidths of the session's receivers downstream of the link. Thus, $\sum_{i \in n(l)} \max_{j \in m(i,l)} r_j \leq C_l$ (capacity constraint), where $n(l)$ is the set of sessions traversing link l, $m(k, l)$ is the set of receivers of session k downstream of link l and C_l is the capacity of link l in packets per unit time*.

Fig. 3 illustrates an example network with a few capacity and maximum rate constraints.

A feasible rate vector is maxmin fair if it is not possible to maintain feasibility and increase the rate of a receiver without decreasing the rate of any other receiver that has equal or lower rate. The formal definition follows.

Definition 2: 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 $\vec{r}^1$ ($r_j^2 < r_j^1$).

Refer to Fig. 3 for an example maxmin fair allocation.

As discussed before, under hierarchical encoding, loss rates should be different for different layers, since 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+1th 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. Our scheduling policy satisfies this objective. 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 justified as in many coding schemes signal quality gradually degrades with increase in packet loss ("graceful degradation") [5].

III. Back pressure based flow control for fairness

We now present the intuition behind our policy. We first explain why a simple round robin scheduling in every link does not attain the maxmin fair rates. A session traverses multiple links, and different links offer different bandwidths to the session. Assume that the session has only one receiver and hence only one source-destination path. The link that offers minimum bandwidth to a session is the session's bottleneck link. In Fig. 1, e_1 and e_2 are the bottleneck links of sessions 1 and 2 respectively. If a session is served in any link in its path at a rate higher than that offered by its bottleneck link, there will be congestion and packet loss in the bottleneck link, and a significant portion of the bandwidths of nonbottleneck links will be wasted in serving packets that do not reach the destination. A simple round robin scheduling does not ensure that the service rate of a session in any link in its path is equal to that in its bottleneck link. Credit based flow control can be used for conveying the bottleneck information implicitly.

Hahne [8] used credit flow control for attaining fairness in networks that have only unicast sessions†. A credit value (W) is decided apriori. The sessions in each link are sampled in round robin manner. When a session is sampled in a link, if the number of packets of a session waiting for transmission at the destination node of a link is less than W and the session has packets for transmission, then the session transmits a packet in the link. If the number of packets of the session waiting for transmission at the destination node of a link is equal to W, then the session does

*Capacity of a link is the number of packets it can transmit per unit time. We assume that all packets have the same number of bits.

†A unicast session has only one receiver.

not transmit, even if it is sampled and has packets for transmission. We explain credit based flow control using Fig. 1. Round robin sampling offers 2 units of bandwidth to both sessions 1 and 2 in e_1. Now, e_2 serves session 2 at a rate of 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 packets in e_1 even when it is sampled. Thus, link e_1 will serve session 2 at a rate lower than 2. Now, link e_3 can transmit session 1 packets at a rate higher than the rate at which e_1 can transmit session 1 packets. So, often node I will not have session 1 packets, and this will reduce the transmission rate for session 1 packets in e_3. It turns out that any link l serves a session i traversing l at the same rate as i's bottleneck link.

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. A session's route may consist of multiple links originating from the same node, and these links serve the session at different rates. Thus, the number of packets of a session waiting at a node for transmission in different links are different. For one link, this number may be less than W, but for another link this number may be greater than W. In Fig. 3, session 1 traverses links e_4 and e_5 originating from node I. Now, W can be 3, and the number of session 1 packets waiting at node I for transmission in links e_4 and e_5 can be 2 and 10 respectively. Thus, it is not clear how credit flow control can be used to determine when a link should serve a session. Also, the flow should be controlled so 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. In Fig. 3, rate of session 1 in link e_3 should be equal to the maximum of that in links e_4 and e_5. We show that this can be attained by allowing a link l to serve a session if the number of packets of the session waiting for transmission in at least one of the links originating from the destination of l is less than W. In Fig. 3, the scheduler for e_3 considers the number of session 1 packets waiting at I for transmission in e_4 and the number of session 1 packets waiting at I 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.

Since service rate of a session in a link is equal to the maximum of the service rates of the session in the links downstream, the source of a link may receive packets at a rate higher than the rate at which the link can serve. In Fig. 3, if link e_4 serves session 1 faster than link e_5 does, then link e_3 will serve packets at a rate equal to that of e_4 and consequently, e_5 will receive packets at a rate higher than the rate at which it can serve. Thus, there will be packet loss at intermediate nodes (node I in this example), since the node buffers are finite. 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 at least W. But, in the multicast case, there will be packet loss as long as the buffers are finite. So, the goal is to attain the maxmin fair rates in presence of packet loss, and also to regulate the loss so that the packets are lost only from the higher layers. We attain the latter objective by using different priorities for different layers.

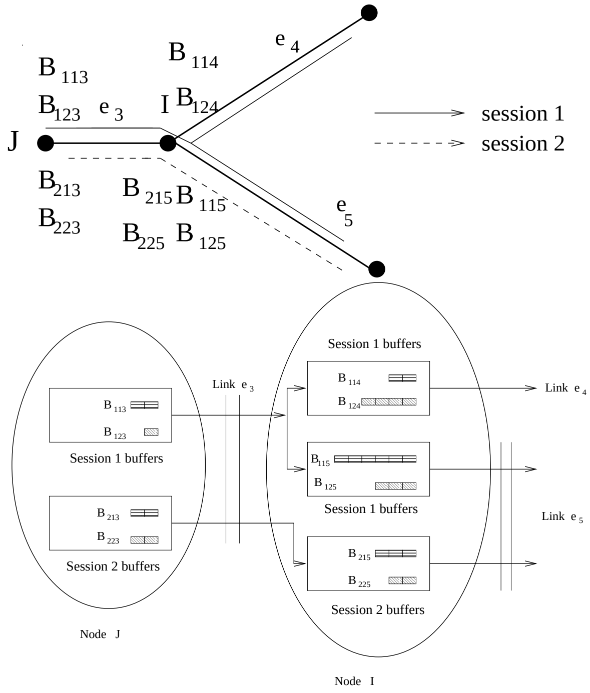

Fig. 4. In the left figure, we show a section of the network shown in Fig. 3. We assume that each session transmits two layers. Here, B_{ijk} is the queue of session i layer j packets waiting for transmission in link e_k. In the right figure, we show the logical buffers and the logical queues at the nodes. The logical queues suggest that if a session 1 packet waits at I for transmission in both links e_4 and e_5, then I maintain two separate copies of the packet; this however is not necessary. The logical queues are variables used in scheduling, and can be maintained by pointers. In Fig.5, we show the physical queues corresponding to these logical queues.

IV. Description of the policy

We propose a scheduling policy based on prioritized round robin with credit flow control for multirate multicast networks. We first introduce some terminologies. Let the credit value be W, $D(i,l)$ be the set of links that originate from the destination node of link l and are in session i's routing tree, and $B_{(i,k,l)}(t)$ be 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 locations. So, the quantities $B_{(i,k,l)}$s represent logical rather than physical buffers. Every node maintains $B_{(i,k,l)}(t)$s for all layers k of all sessions i traversing any link l originating from the node. Refer to Fig. 4 and 5 for examples. Since physical buffers have finite sizes, for all sessions i, layers k, link l and time t, $B_{(i,k,l)}(t)$s must be less than or equal to a quantity G. Let $G > W$. We assume that at time t the scheduler for link l knows $\min_{l' \in D(i,l)} B_{(i,k,l')}(t)$ for all sessions i traversing link l. In Section V, we discuss how to relax this assumption.

We have stated the scheduling algorithm in each link in Fig. 6. We now explain each step. Link l samples all ses-

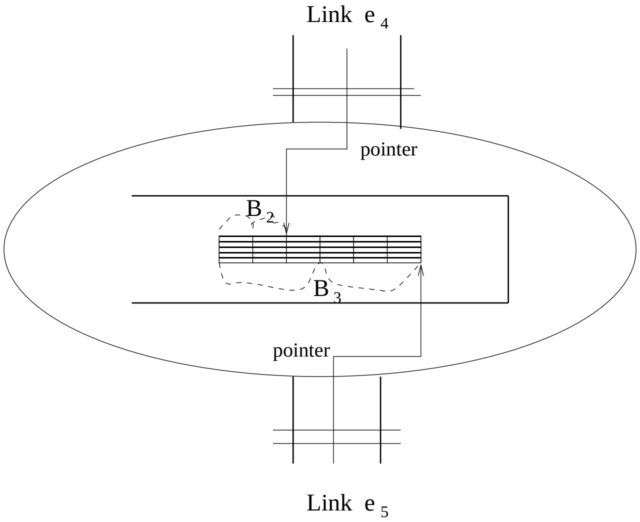

Fig. 5. We show a physical buffer at node I of Fig. 4. We assume that the switches are input queued. This physical buffer stores layer 1 packets of session 1 transmitted via link e_3, and corresponds to logical buffers, B_{114} and B_{115}. Packets are replicated only at the transmission epoch. So, the buffer stores 6 packets. 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, in Fig.4, $B_2(B_{(114)})$ contains 2 packets and $B_3(B_{(115)})$ contains 6 packets. Every link maintains a pointer at the first packet it needs to transmit.

sions traversing l in round robin order. When session i is sampled, l first examines whether there exists any layer j that satisfies both the packet-availability and the next-hop-congestion conditions. The packet-availability condition for layer j of session i examines whether there are session i layer j packets waiting for transmission in link l; if not, then clearly layer j packets of session i can not be transmitted. The next-hop congestion condition for layer j of session i examines whether the number of session i layer j packets waiting for transmission in link l' is less than W for at least one link l' originating from the destination of l; if not, then the destination of l has a high congestion level for layer j packets of session i and hence l does not serve additional layer j session i packets. If there exists a layer j of session i that satisfies both the above conditions, then l transmits a packet from the lowest layer of i, k, that satisfies these conditions; this ensures that lower layers suffer less packet loss than higher layers. This layer k packet joins the queue for transmission in all links l' in $D(i,l)$, except those that have G session i packets of layer k (i.e., except if $B_{(i,k,l')}(t) = G$). If $B_{(i,k,l')}(t) = G$ for some link $l' \in D(i,l)$, the packet is lost for link l' and the receivers downstream of l'. If the packet-availability and the next-hop-congestion conditions are not satisfied by any layer of i, l does not transmit any packet of i, and samples the session that is next to i in the round robin order.

We now elucidate the policy using the following example.

Example IV.1: In Fig. 4, we show a part of the network of Fig. 3. The quantities B_{ijk} in the figure denote $B_{(i,j,e_k)}(t)$. Let $W = 3$ and $G = 6$. Here, $D(1,e_3) = \{e_4,e_5\}$ and $D(2,e_3) = \{e_5\}$. We assume that links e_4, e_5 do not transmit any packet in the interval $[t, t+2]$. Link e_3

Fig. 6. Pseudo-Code for scheduling at each link

Information at the scheduler for link l for each session i traversing l

begin

Credit value (W),

Set of links that originate from the destination node of link l and are in session i's routing tree ($D(i,l)$),

Number of layer k packets of session i waiting for transmission in link l at time t ($B_{(i,k,l)}(t)$)

end

Procedure Scheduling at link l

begin

When l finishes transmission of a packet, it samples in round robin order all sessions traversing l starting from the one after the session last served, until it samples a session that transmits a packet.

When l samples session i,

if for some layer j, $B_{(i,j,l)}(t) > 0$, (**packet-availability condition**) and $\min_{l' \in D(i,l)} B_{(i,j,l')}(t) < W$, (**next-hop-congestion condition**)

then

i transmits the packet from the lowest layer k for which $B_{(i,k,l)}(t) > 0$ and $\min_{l' \in D(i,l)} B_{(i,k,l')}(t) < W$, and

after the transmission the packet joins the queue for transmission in all links l' in $D(i,l)$ for which $B_{(i,k,l')}(t) < G$,

else

i does not transmit a packet.

end

samples session 1 at time t. Now, $B_{(1,1,e_3)}(t) > 0$ and $\min_{l' \in D(1,e_3)} B_{(1,1,l')}(t) = B_{(1,1,e_4)}(t) = 2 < W$. So, e_3 transmits a session 1 layer 1 packet which is added to the queue for transmission in link e_4, but not in link e_5 as $B_{(1,1,e_5)}(t) = 6 = G$. The transmitted packet is lost for link e_5 and the receiver u_2.‡ Assume that the transmission of each packet in e_3 consumes 1 unit of time. Now, $B_{(1,1,e_3)}(t+1) = 1$, and $B_{(1,1,e_4)}(t+1) = 3$. The rest of the buffer contents remain the same at time $t+1$ as at time t.

At time $t+1$, e_3 samples session 2. Session 2 does not transmit a layer 1 packet as $\min_{l' \in D(2,e_3)} B_{(2,1,l')}(t+1) = B_{(2,1,e_5)}(t+1) = 3 = W$, but transmits a layer 2 packet as $B_{(2,2,e_3)}(t+1) > 0$, and $B_{(2,2,e_5)}(t+1) = 2 < W$. The transmitted packet is added to the queue for transmission in link e_5. Now, $B_{(2,2,e_3)}(t+2) = 1$ and $B_{(2,2,e_5)}(t+2) = 3$. The rest of the buffer contents remain the same at time $t+2$ as at time $t+1$.

At time $t+2$, e_3 samples session 1. Session 1 does not transmit any packet as $\min_{l' \in D(1,e_3)} B_{(1,k,l')} = 3 = W$, for $k \in \{1,2\}$. So, e_3 samples session 2 next. Session 2 does not send any layer 1 packet as $\min_{l' \in D(2,e_3)} B_{(2,k,l')}(t+2) = 3 = W$, for $k \in \{1,2\}$. So, e_3 idles until e_4 and e_5 serve 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 evaluate the performance of our scheduling policy using analysis and simulation.

Let session i source transmit γ_i layers. Bandwidth of the kth layer of the ith session is $b_{i,k}$. Session i source is "well-behaved" if in any interval $[u,v)$ it generates 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, where $\xi_{i,k}$ are "transmission jitters." Many sources, e.g., outputs of leaky-bucket shapers, constant bit

‡Depending on the buffer management policy, the new packet may be added to the queue $B_{(1,1,e_5)}$, and an old packet in $B_{(1,1,e_5)}$ may be dropped.

rate (CBR) video sources, etc. are well-behaved. If the transmission jitters $\xi_{i,k}$ are appropriately selected, then variable bit rate (VBR) periodic sources can also be modeled by well-behaved sources. We assume that all sources are well behaved. Let the maxmin fair rate of receiver j be r_j.

Theorem 1: If $G \geq G^*$ and $W \geq W^*$, each 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 $[u,v)$. Here, $\delta_j, j = 1,\ldots,M, G^*, W^*$ are constants that do not depend on u, v.

The constants $\delta_j, j = 1,\ldots,M, G^*, W^*$ depend on $b_{i,k}, \xi_{i,k}$, path lengths, link capacities, G^* depends on W in addition and δ_js depend on both W and G in addition. Refer to the appendix for their formulation.

We now explain the significance of Theorem 1. Theorem 1 shows that the policy exhibits long term fairness as packets are delivered to the receivers at the maxmin fair rates. Also, the policy is fair in short intervals as the number of packets delivered to the receivers in any interval differ from the maxmin fair number by at most a constant that does not depend on the length of the interval; the constants are large though.

We now present the intuition behind the result in Theorem 1. For the bandwidth allocation to be maxmin fair, (a) the total link capacity must be divided equally among all sessions traversing the link provided the sessions are not congested elsewhere, and (b) if a session can not use its equal share due to congestion in other links, the residual bandwidth in a link must be used to serve other sessions. Round robin sampling of sessions in each link ensures (a). Also, since a session does not transmit a packet in a link when all its downstream links are congested (i.e., all downstream links of the session have W or more packets for each layer of the session) and other sessions receive the transmission opportunity, (b) is guaranteed. Thus, the resulting bandwidth allocation is maxmin fair.

The next theorem describes how packet losses are distributed across layers. Specifically, it shows that as required by the application, the packet loss is concentrated in the highest layer served. Here, $(u)^+ = \max(u,0)$.

Theorem 2: Let $G \geq G^*$ and $W \geq W^*$. Let receiver j belong to session i. The number of layer k packets lost in the path of receiver j in any interval $[u,v)$ is at most $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - r_j, b_{i,k}\right)\right)^+ (v-u) + \iota_{j,k}$. Here, $\iota_{j,k}$ is a constant that depends on $W, G, b_{i,k}, \xi_{i,k}$, path lengths, link capacities and not on u, v.

We now explain the significance of Theorem 2. Let receiver j belong to session i. As discussed before, under hierarchical encoding, lower layers contain more important information than higher layers. Thus, if the maxmin fair bandwidth r of receiver j is sufficient to serve all packets of the first k layers i.e., $\sum_{w=1}^{k} b_{i,w} \leq r_j$, then the application requires that j experience a loss rate of 0 for packets of the first k layers. In this case, $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - r_j, b_{i,k}\right)\right)^+ = 0$, and hence, by Theorem 2, the number of layer k packets lost in the path of receiver j in any interval $[u,v)$ is at most $\iota_{j,k}$. Thus, receiver j observes a long term loss rate of 0 for layer k packets. Next, if the maxmin fair rate r is sufficient to serve all packets of the first $k-1$ layers and only a portion of the packets of the kth layer, i.e., $\sum_{w=1}^{k-1} b_{i,w} < r_j < \sum_{w=1}^{k} b_{i,w}$, then the application requires that the residual bandwidth left after serving the first $k-1$ layer packets be used to serve the kth layer. Thus, receiver j must receive some of the layer k packets. In this case, $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - r_j, b_{i,k}\right)\right)^+ = \sum_{w=1}^{k} b_{i,w} - r_j < b_{i,k}$. Now, from Theorem 2, the long term loss rate for layer k packets is $\frac{\sum_{w=1}^{k} b_{i,w} - r_j}{b_{i,k}}$, which is less than 1. Thus, the application requirement is satisfied.

We now present the intuition behind the result in Theorem 2. It follows from Theorem 1 that every receiver receives packets at its maxmin fair rate. Now, whenever a session is sampled in a link it first tries to transmit a lower layer packet, and transmits a higher layer packet only when it fails to do so. This happens if no lower layer packet is waiting for transmission or the downstream links have a large number of lower layer packets waiting for transmission. Due to this strict priority, the maxmin fair bandwidth allocated to a receiver is first used to deliver the lower layer packets, and the residual bandwidth is subsequently used to deliver higher layer packets. The result follows.

We now explain why the guarantees in Theorems 1 and 2 hold only when the credit value W and buffer size G exceed certain lower bounds. The packets of a layer j of a session i are not served in a link if the layer experiences congestion in downstream links. Now, the layer j of a session i is considered to experience congestion in a link only when W or more of its packets wait for transmission in the link. This may happen due to short term congestion which occurs due to burstiness of the packet generation and the service processes, or because the source of the link is receiving packets at a rate greater than the link's capacity. The service rate in preceding links should reduce only when the latter happens. But, if W is small, then the short term congestion may affect the service in preceding links which would in turn lead to oscillation of allocated rates. Similarly, if G is small, then the loss rate may increase beyond that guaranteed by the maxmin fair rates due to the burstiness of the packet generation and the service processes. Thus, the rate and loss guarantees in Theorems 1 and 2 hold only when W and G exceed certain lower bounds.

Now, the lower bounds can be quite large. For example, G^* and W^* may exceed $E^M H^{\max_i \gamma_i}$, where E is the maximum number of sessions traversing any link and H is the maximum path length in a session tree. Also, these lower bounds depend on the entire network topology which nodes may not know. Thus, nodes may not be able to select the credit and buffer sizes that exceed these lower bounds. The next theorem provides guarantees on the rate and loss when the credit and buffer sizes W, G are lower than the respective bounds W^*, G^* required in Theorems 1 and 2.

We introduce the notion of the *rank* of a receiver. If the maxmin fair rate of the receiver j is the mth smallest among the maxmin fair rates of all receivers, then the rank of receiver j is m. Let the number of ranks be F, where $F \leq M$; $F < M$ if the maxmin fair rates of some receivers are equal.

Theorem 3: Let $\hat{\lambda}_m$ be the mth smallest maxmin fair rate. Let receiver j belong to session i. There exists a sequence of constants, $W_1 < W_2 < \ldots < W_F = W^*$ and $G_1 < G_2 < \ldots < G_F = G^*$ such that, if $W \geq W_m$, and $G \geq G_m$, then (a) all receivers of rank m and above receive packets at rates greater than or equal to $\hat{\lambda}_m$, (b) all receivers of rank smaller than m, receive packets at their maxmin fair rates, and (c) the number of layer k packets lost in the path of receiver j in any interval $[u, v)$ is at most $\left(\min\left(\sum_{w=1}^{k} b_{i,w} - \min(r_j, \hat{\lambda}_m), b_{i,k}\right)\right)^+ (v-u) + \iota_{j,k}$.

We formulate the constants $W_1, W_2, \ldots, W_F, G_1, G_2, \ldots, G_F$ and $\iota_{j,k}$ in the appendix. We now explain the significance of Theorem 3. Theorem 3 shows that performance gradually improves with increase in buffer and credit sizes. Theorem 3 states that if $W \geq W_m$ and $G \geq G_m$ where $W_m < W^*, G_m < G^*$ then all receivers with lower values of maxmin fair rates (i.e., those with the maxmin fair rates lower than or equal to the mth lowest maxmin fair rate $\hat{\lambda}_m$) attain the maxmin fair rates and receive the same loss guarantees as in Theorem 2. The rest of the receivers are however not guaranteed to attain their maxmin fair rates, and may receive fewer layers than when $W \geq W^*$ and $G \geq G^*$. But, Theorem 3 guarantees that these receivers' rates are lower bounded by the mth lowest maxmin fair rate $\hat{\lambda}_m$ which is however less than their maxmin fair rates. Their loss rates are still concentrated in the highest layer served.

We now present the intuition behind the result in Theorem 3. We explain why the guarantees for receivers with lower ranks and hence lower maxmin fair rates require lower minimum values of W and G. First, the lower bounds on W and G increase with increase in the burstiness of the packet generation and the service processes. Now, due to round robin sampling, all sessions traversing a link are sampled at the same rate. A session i therefore receives higher rate than a session j in a link if due to congestion elsewhere in its path j does not transmit many times it is sampled, and instead i transmits packets at these epochs. Thus, i's service process depends on the burstiness in both j's and its own packet generation and service processes. Thus, a session which has a higher maxmin fair rate has a more bursty service process which increases the lower bounds on W and G required for allocating the maxmin fair rates to its receivers.

We have so far assumed that a link scheduler knows the queue lengths at the destination node of the link. More precisely, at all times t the scheduler for link l knows $\min_{l' \in D(i,l)} B_{(i,j,l')}(t)$ for all layers j of all sessions i traversing link l. The queue lengths at the next hop can be communicated in feedback packets, but feedback packets are never received instantaneously. Then, at time t the scheduler knows the queue lengths at the next hop at some previous time t'. Also, due to propagation delay in the link, packets do not reach the destination of a link immediately after the link's source complete transmission. Theorems 1 to 3 hold even when the scheduler decides whether to transmit a packet for a session on the basis of the queue lengths at previous times and even when packets reach the destination of a link some time after the source completes transmission, as long as these delays are bounded. We refer to these delays as propagation delays. The credit and buffer thresholds, W_i and $G_i, i = 1, \ldots, F$s depend on the propagation delays now. Refer to [14] for a formal proof. The intuition is as follows. At any node, the queue lengths at previous times differ from the current queue lengths by at most a constant that depends on the propagation delay and link capacities. This constant will increase the constants δ_is and $\iota_{i,j}$s but the long term throughputs do not depend on these.

Now, we evaluate the performance of the scheduling policy using simulation. Simulations allow us to draw two important conclusions which we could not draw from the analytical results. First, the constants $\delta_j, \iota_{j,k}$ in Theorems 1, 2, 3 increase with increase in $W, G, b_{i,k}, \xi_{i,k}, M, N, E, \gamma_i$s. These constants can become quite large in actual networks. Thus, although the analysis guarantees that the rates attained by the receivers converge to the maxmin fair rates, it does not guarantee a fast convergence. The simulations demonstrate that the convergence rate guaranteed by the analysis is pessimistic and the rates attained by most receivers fast converge to the respective maxmin fair rates.

Second, intuition suggests that the guarantees on rate and loss will not hold when W and G are small. Consistent with this intuition, Theorems 1, 2 provide analytical guarantees on rate and loss only when W and G exceed thresholds W^*, G^* respectively. Theorem 3 shows that the guarantees progressively improve as W and G increase. But, the thresholds $W_1, \ldots, W_F$, $G_1, \ldots, G_F$ required for the progressive guarantees in Theorem 3 are still quite large and still depend on the global network topology. Using simulations, we seek to understand whether the analytical bounds are pessimistic and whether reasonable values of W and G usually suffice. The simulation results demonstrate that the convergence does not critically depend on the choice of W and G; thus the policy is robust. Specifically, even when the credit and buffer sizes are significantly less than W_1 and G_1 respectively (W_1, G_1 are the lowest thresholds required for any analytical guarantee), packets are delivered to the receivers at the maxmin fair rates, and the packet loss is concentrated in the highest layer served.

We seek to examine the time and the lower bounds required for convergence in networks where the analytical bounds W_i, G_i, $i = 1, \ldots F$, $\delta_j, \iota_{j,k}$ are large. We therefore consider a network with large number of nodes, links, sessions, receivers and layers. Specifically, we consider a network with 15 sessions, 96 receivers and 400 nodes. Nodes are points on a 20×20 grid. There exists an edge between any two nodes with a probability (p) that decreases with the increase in euclidean distance between the nodes (d), $p = \exp(\alpha(1-d))$, where α is the decay constant. We assumed $\alpha = 2$. The capacity of each link is uniformly distributed between 0 and 20. Source and receivers of the

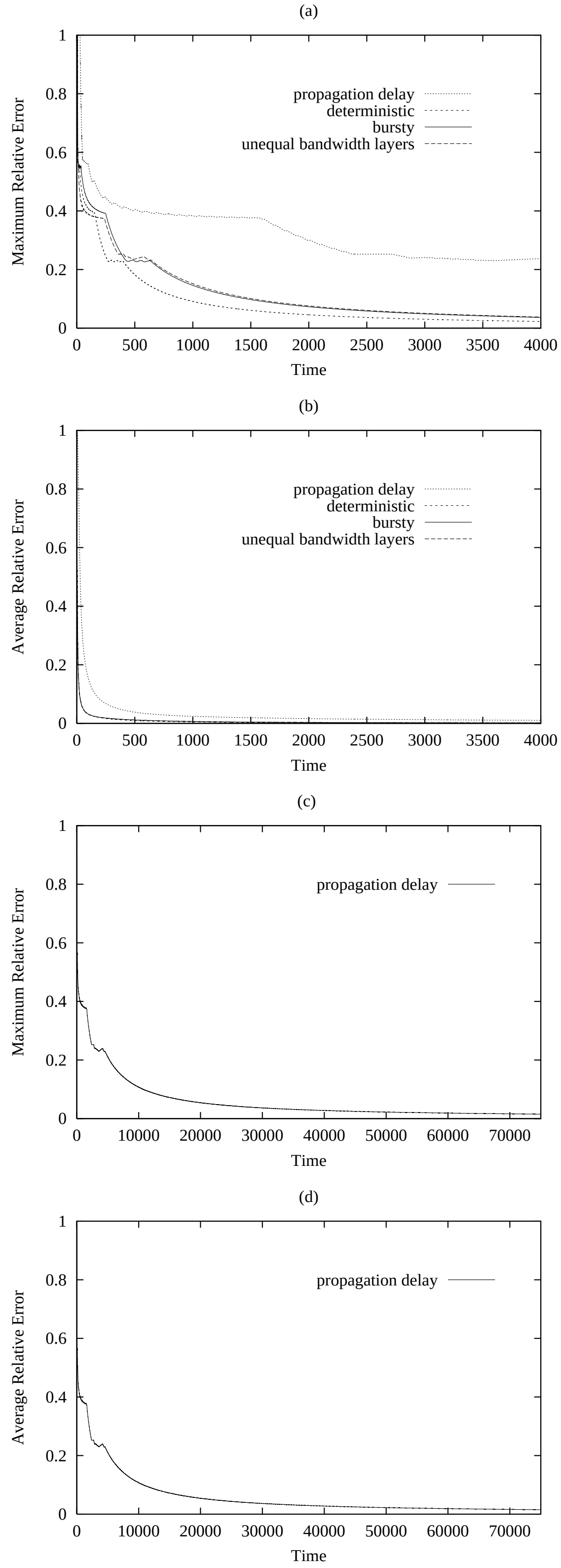

Fig. 7. These figures demonstrate the convergence of the packet delivery rates attained by the proposed scheduling policy to the maxmin fair rates. We have plotted the convergence errors as a function of time for different traffic models. Fig. (c) and (d) plot the errors for a larger range of time as compared to Fig. (a) and (b).

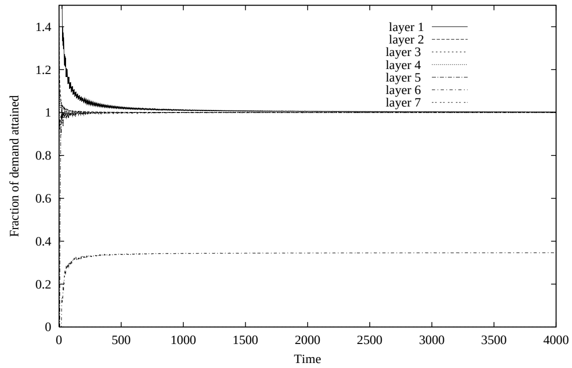

Fig. 8. The figure shows the fraction of packets of different layers delivered to one particular receiver in the random network.

sessions are selected randomly. The session tree consists of shortest paths between the source and the receivers. Every source transmits 20 layers. We implement the scheduling policy in C.

Fig. 7 demonstrates the convergence of the receiver rates to the respective maxmin fair rates for different traffic patterns. We study the difference between the rate of delivery of packets for each receiver i and the receiver's maxmin fair rate, r_i. The rate of delivery of packets at time t for receiver s, $r_s^a(t)$, is the number of packets delivered to s in the interval $[0, t)$ divided by t. The error for receiver s is $|1 - \frac{r_s^a(t)}{r_s}|$ at time t. Fig. $7(a)$ and $7(c)$ plot the maximum relative error, and Fig. $7(b)$ and $7(d)$ plot the average relative error, the maximum and average are taken over all receivers. We consider the following different traffic patterns. For the curves labeled "deterministic," every source generates packets of every layers periodically at rate 1 per unit time. For the curves labeled "bursty," in any interval of length t, each source generates at most $t+3$ and at least $t-3$ packets of every layer. For the curves labeled "unequal bandwidth layers," in any interval of length t, a source i generates at most $b_{i,j}t + \xi_{i,j}$ and at least $b_{i,j}t - \xi_{i,j}$ layer j packets. We randomly selected the layer bandwidths $b_{i,j}$ and transmission jitters $\xi_{i,j}$. For each i, $b_{i,1}$ is uniformly distributed between 0 and 5, and for each j, $b_{i,j}$ is uniformly distributed between 0 and $5 - \sum_{k=1}^{j-1} b_{i,k}/4$, and $\xi_{i,j}$s are uniformly distributed between 0 and 20. For example, $b_{1,1} = 0.21$, $b_{1,2} = 2.78$, $b_{1,3} = 1.81$, $b_{2,1} = 3.5$, $b_{2,2} = 3.21$, $b_{2,3} = 3.3$, $\xi_{1,1} = 2$, $\xi_{1,2} = 3$, $\xi_{1,3} = 0$, $\xi_{2,1} = 2$, $\xi_{2,2} = 0$, $\xi_{2,3} = 1$ respectively. For all these curves, we assume that a packet transmitted at time t reaches the next hop at time $t+1$ and the scheduler at a node knows the exact queue lengths at the next hop nodes. The average error converges to 0 much faster than the maximum error, indicating that for most of the receivers the reception rates converge to the maxmin fair rates fast, whereas convergence is relatively slow for a few others. Convergence is fastest for the deterministic traffic model. We used $W = 5$ and $G = 10$ respectively for the deterministic traffic model, and $W = 8$ and $G = 16$ respectively for the bursty and unequal bandwidth layer traffic models. Note that in this network W_1

and G_1 are much larger than 8 and 16 respectively. Thus, the receivers receive packets at the maxmin fair rates even when the network is large and W and G are significantly smaller than W_1 and G_1 respectively.

We also considered the effect of delayed feedback. For the curves labeled "propagation delay" we assume that the propagation delay for the data and the feedback packet in each link equals the euclidean distance between the end points of the link. We plot the errors for different ranges of time in Fig. 7(a) to 7(d). Here, $W = 100$ and $G = 200$. As expected, rates of delivery of packets still converge to the maxmin fair rates, but the convergence is slower than when the propagation delay is ignored. Propagation delay increases buffer and credit size requirements, but these requirements are still reasonable.

We show in Fig. 8 that packet losses are confined to the highest layer served and packets from different layers suffer different loss rates. We plot the fraction of packets delivered to a receiver that has maxmin fair rate equal to 9 packets per unit time; this fraction for layer i at time t 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 t. We consider the unequal bandwidth traffic model, and ignore the propagation delays. Here, $W = 8$ and $G = 16$. 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. Analytical results guarantee that the receiver should receive all packets of the first 5 layers, 35% packets of layer 6 and possibly no packet of any higher layer. As the figure shows, for the first 5 layers, the fraction of packets delivered to the receiver fast converge to 1. Initially, the fraction is greater than 1 for some lower layers, as the source sends an initial burst of packets for every layer due to transmission jitters and the network delivers some of these bursts for the lower layers. The fraction of packets of layer 6 delivered to the receiver converges to 0.35 as well. Very few packets of other layers reach the receiver. This shows that the packet loss is confined to the highest layer served, i.e., layer 6 in this case.

VI. Discussion and Conclusion

We discuss some salient features of our scheduling policy.

If the maximum rates of sessions form a feasible rate allocation, then the maximum rates are maxmin fair. Thus, our scheduling policy attains the maximum rates, if the maximum rates are feasible; therefore, it satisfies all users subject to bandwidth limitations.

The scheduling policy is adaptive as its execution does not require any knowledge of the maximum rates of users, layer bandwidths, or the statistics of the packet arrival process. Note that we could prove the analytical performance guarantees for the case that the credit and buffer values are greater than certain lower bounds, $W_1, \ldots, W_F$ and $G_1, \ldots, G_F$ that depend on the topology; but the simulations reveal that these lower bounds are pessimistic and fair rates are obtained even when the credit and buffer values are below these thresholds. Specifically, in the simulations even in large networks with propagation delays, fair rates were obtained for all moderately large values of W and G (e.g., $W = 100, G = 200$). The analytical guarantees do not depend on the hierarchical structure of signals and unequal bandwidth layers are permitted.

A malicious session can not increase the throughput of its receivers by selecting layer bandwidths suitably, as the maxmin fairness of receiver throughputs are guaranteed irrespective of the layer bandwidths.

Our scheduling policy is computationally simple.

A link scheduler takes scheduling decisions whenever the link is free to transmit a packet, and need not synchronize with schedulers for other links.

Our 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 sessions with different priorities instead of different layers. This would allow differentiation of service within the framework of fairness.

Sometimes receivers need some minimum quality of service guarantees, which can be attained only when their rates exceed certain minimum acceptable values. A rate allocation is now feasible if it satisfies the capacity constraint and the rate of each receiver i is greater than or equal to its minimum required rate of μ_i. As before, a feasible rate vector is maxmin fair if it is not possible to maintain feasibility and increase the rate of a receiver without decreasing the rate of any other receiver that has equal or lower rate. (Note that the definition for feasibility has changed.) We now describe how the scheduling policy in Figure 6 can be generalized to attain the maxmin fair rates in presence of these minimum rate constraints. Only the sampling strategy in Figure 6 need to be changed. Let μ_{il} be the maximum of the minimum rates of receivers of session i downstream of link l. For simplicity of exposition, assume that time is slotted, and the transmission duration of each packet is a slot. Now, link l must sample only session i in μ_{il} fraction of slots. For this purpose, link l may divide the slots in frames and reserve a certain number of slots for each session. In each of the remaining $1 - \sum_{i \in n(l)} \mu_{il}$ fraction of slots, l samples the session that has so far been sampled the least number of times. The rest remains the same. The analytical guarantees presented in Theorems 1, 2 and 3 still hold. The proofs are similar as well.

Our policy does not assume any particular drop strategy. When a packet arrives and finds the buffer full, the new packet need not be dropped. Dropping an old packet may be a better option for real time transmission, as packets delivered after a certain delay become useless. The routers may use a drop tail (drop the new packet) or random drop (drop a random 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.

A link scheduler needs 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 W. Whenever queue length of a session at a node becomes lower (higher) than W after being higher (lower) than W, a message can be sent to the node at the previous hop. This message can be piggy backed in data packets. Thus overhead is low. This hop by hop congestion feedback has certain advantages over end to end congestion control, e.g., it can control short-lived congestion better than TCP, and is used in local area networks [16].

Since the propagation delays do not alter the throughput of the policy as long as the delays are bounded, this policy can be used in networks where propagation delay is significant, e.g., networks with satellite links.

Our scheduling policy requires per session states in the routers, but the resulting increase in complexity is not drastic. Arguing in the lines of Grossglausser and Bolot [6], implementing a multicast/multilayer service requires per-session state in the routers anyway. So, the incremental cost of maintaining some more information for each session and using this additional information in the scheduling is much smaller than that in the unicast case. If however these additional session states become an issue, then this policy can be used in the VPNs and intranets, and state aggregation may be used in the backbones.

We conclude that this scheduling policy is suitable for use in large, dynamic, high speed decentralized networks in which nodes have access to only local and delayed information.

Appendix

Theorems 1 to 3 follow from a general result, Theorem 4, which we state in this section. We outline the proof of this general result here, and prove it in technical report [14]. We first introduce some additional notations which we require in stating Theorem 4.

- The rank of a receiver s is $\vartheta(s)$.
- Recall that F is the total number of ranks.
- Let session i traverse link l. The rank of i in l, $\Delta(i,l)$, is the maximum rank of i's receivers downstream of l.
- Recall that $\hat{\lambda}_p$ is the pth smallest maxmin fair bandwidth, $p \geq 1$. Let, $\alpha(i,p)$ be the maximum number of layers of session i that can be fully served if it is allocated $\hat{\lambda}_p$ amount of bandwidth, i.e., $\alpha(i,p) = \max\{k : \sum_{m=1}^{k} b_{i,m} \leq \hat{\lambda}_p\}$, if $b_{i,1} \leq \hat{\lambda}_p$. If $b_{i,1} > \hat{\lambda}_p$, then $\alpha(i,p) = 0$. By convention, $\alpha(i,0) = 0$.
- Let E_i be the maximum number of sessions traversing a link in session i's path. Recall that E is the maximum number of sessions traversing any link. Thus, $E = \max_i E_i$.
- All links that originate from the same node and are in session i's paths are said to be session i siblings of each other.
- Let ν_i be the maximum number of links that originate from a node and are in session i's path, and $\nu = \max_i \nu_i$.
- The set of links in receiver s's path is L_s.
- The session of receiver s is $\chi(s)$.
- The number of packets of session i waiting to be served in link l at time t is $B(i,l,t)$.
- The number of layer j packets of session i waiting to be served in link l at time t is $B(i,j,l,t)$.
- The number of times session i is sampled in link l in time interval $[s,t)$ is $C(i,l,s,t)$.
- The number of times layer j of session i is sampled in link l in time interval $[s,t)$ is $C(i,j,l,s,t)$.
- The number of session i packets that finish service in link l in time interval $[s,t)$ is $P(i,l,s,t)$.
- The number of session i layer j packets that finish service in link l in time interval $[s,t)$ is $P(i,j,l,s,t)$.
- The routing tree of each session has different paths. The length of a path is the number of links in the path. Let H_i be the maximum length of a session i path. Since every session has an access link, every session has at least two links in each of its paths, and $H_i > 1$, for all sessions i. Recall that H is the maximum length of a path in routing tree of any session. Thus, $H = \max_i H_i$.

Now, we define some recursive constants that depend on rank p, session i and layer j. We use these recursive constants in stating Theorem 4.

$$\varepsilon_1(p,i,j) = \begin{cases} \max\left(\xi_{i,j}, (E-1)\gamma(p-1) + 2 \right. \\ \left. + \sum_{m=1}^{j-1} \varepsilon_3(p,i,m)\right), & \text{if } j \leq \alpha(i,p)+1, \\ 0, & \text{if } j > \alpha(i,p)+1. \end{cases}$$

$$\begin{aligned}
\varepsilon_2(p,i,j) &= \begin{cases} H_i\varepsilon_1(p,i,j) + H_i - 1 + \xi_{i,j}, \\ \qquad\qquad \text{if } j \leq \alpha(i,p)+1, \\ 0, \qquad\quad \text{if } j > \alpha(i,p)+1. \end{cases} \\
\varepsilon_3(p,i,j) &= \varepsilon_2(p,i,j). \\
\varsigma_1(p,i,j) &= \begin{cases} \sum_{m=1}^{j-1} \varsigma_3(p,i,m) + (E-1)\kappa(p-1), \\ \qquad\qquad \text{if } j \leq \alpha(i,p)+1, \\ 0, \qquad\quad \text{if } j > \alpha(i,p)+1. \end{cases} \\
\varsigma_2(p,i,j) &= \left(H_i - 1 + \nu_i \frac{2\nu_i^{H_i-1} - \nu_i^{H_i-2} - 1}{\nu_i - 1}\right) \varsigma_1(p,i,j). \\
\varsigma_3(p,i,j) &= \begin{cases} \varsigma_2(p,i,j) + (2^{H_i+1})\left(\varsigma_2(p,i,j) + \varepsilon_2(p,i,j)\right) \\ +2^{H_i} \max_{l:i\in n(l)} B(i,j,l,T_{p,j}), \\ \qquad\qquad \text{if } j \leq \alpha(i,p+1), \\ \varsigma_2(p,i,j) + W(H_i - 1) \\ +2^{H_i-1}\left(\left(\varrho(p) + \tau(p)\right) E_i + 1\right) \\ +(H_i - 1) \max_{l:i\in n(l)} B(i,j,l,T_{p,j}), \\ \qquad\qquad \text{if } j > \alpha(i,p+1). \end{cases} \\
\varepsilon_1(p) &= \max_{s:\vartheta(s)=p} \varepsilon_1\left(p, \chi(s), \alpha(\chi(s),p)+1\right). \\
\varsigma_1(p) &= \max_{s:\vartheta(s)=p} \varsigma_1\left(p, \chi(s), \alpha(\chi(s),p)+1\right). \\
\theta_i(x,y) &= \begin{cases} y, & \text{if } x = 1, \\ y\nu_i \frac{2\nu_i^{x-1} - \nu_i^{x-2} - 1}{\nu_i - 1}, & \text{if } x > 1. \end{cases}
\end{aligned}$$

$$\Phi_i(x+1,T,p,j)$$

$$= \begin{cases} T, \qquad \text{if } x = 0, \\ \Phi_i(x,T,p,j) + \frac{1}{\min_{r_m \neq r_n} |r_m - r_n|}\left(W + 1\right. \\ +(H_i - x)\varepsilon_1(p,i,j) + (H_i - x)2^{H_i - x}\varepsilon_1(p,i,j) \\ +x\varepsilon_1(p,i,j) + x + \varepsilon_3(p,i,j) + (H_i - x)\varsigma_1(p,i,j) \\ +\theta_i\left(x, \varsigma_1(p,i,j)\right) + \varsigma_3(p,i,j) \\ \left. + \max_l \left(B(i,m,l,0) + C_l \Phi_i(x,T,p,j)\right)\right), \qquad \text{if } x > 0. \end{cases}$$

$$T_{p,j} = \begin{cases} 0, & \text{if } p = 1, \\ T_{p-1,\max_i \alpha(i,p)+1}, & \text{if } p > 1, j = 0, \\ \max_i \Phi_i \left(H_i, T_{p,j-1}, p, j\right), & \text{if } p > 1, j \geq 1. \end{cases}$$

$$\begin{aligned} \varrho(p) &= \max_i \sum_{m=1}^{\alpha(i,p)+1} \varsigma_2(p,i,m), \qquad p > 0. \\ \tau(p) &= \max_i \sum_{m=1}^{\alpha(i,p)+1} \varepsilon_2(p,i,m), \qquad p > 0. \\ \kappa(p) &= \begin{cases} 0, & \text{if } p = 0, \\ \max_i \sum_{m=1}^{\alpha(i,p)+1} \varsigma_3(p,i,m), & \text{if } p > 0. \end{cases} \\ \gamma(p) &= \begin{cases} 0, & \text{if } p = 0, \\ \max_i \sum_{m=1}^{\alpha(i,p)+1} \varepsilon_3(p,i,m), & \text{if } p > 0. \end{cases} \end{aligned}$$

$$\begin{aligned} W_p &= \left(1 + \max_{s:\vartheta(s)=p} H_{\chi(s)} \varepsilon_1\left(p, \chi(s), \alpha(\chi(s),p)+1\right)\right) \quad (1) \\ G_p &= \max_{j:\vartheta(j)=p} \left(H_{\chi(j)} + \frac{\max_{l:\chi(j)\in n(l)} C_l}{\min_{r_m \neq r_n} |r_m - r_n|}\right) W + 3 \\ &\quad + \frac{\max_{l:\chi(j)\in n(l)} C_l}{\min_{r_m \neq r_n} |r_m - r_n|} \left((H_{\chi(j)} - 1)\varsigma_1(p) + \right. \\ &\quad \theta_{\chi(j)}\left(H_{\chi(j)} - 1, \varsigma_1(p)\right) + (H_{\chi(j)} + 1)\varepsilon_1(p) + 1 \\ &\quad \left. + \kappa(p-1) + \gamma(p-1)\right) + E\left(\rho(p) + \tau(p)\right). \qquad (2) \end{aligned}$$

Theorem 4: Let the number of layer j packets a session i generates in any interval $[s,t)$ differ from $b_{i,j}(t-s)$ by at most $\xi_{i,j}$. Then, the following hold. For all $p = 1, \ldots, F$, if $W \geq W_p$ and $G \geq G_p$, s, t, $t \geq s \geq T_{p+1,0}$, sessions i and links l such that $\Delta(i,l) \geq p$,

$$P(i,l,s,t) \geq \hat{\lambda}_p(t-s) - \varrho(p) - \tau(p), \qquad (3)$$

$$\begin{aligned} &\forall \text{ layer } j, P(i,j,l,s,t) \\ \geq\ & \min\left(\left(\hat{\lambda}_p - \sum_{m=1}^{j-1} b_{i,m}\right), b_{i,j}\right)(t-s) \\ & -\varsigma_2(p,i,j) - \varepsilon_2(p,i,j). \qquad (4) \end{aligned}$$

If $\Delta(i,l) = p$, then

$$P(i,l,s,t) \leq \hat{\lambda}_p(t-s) + \kappa(p) + \gamma(p), \qquad (5)$$

$$\begin{aligned} &\forall \text{ layer } j, P(i,j,l,s,t) \\ \leq\ & \min\left(\left(\hat{\lambda}_p - \sum_{m=1}^{j-1} b_{i,m}\right)^+, b_{i,j}\right)(t-s) \\ & +\varsigma_3(p,i,j) + \varepsilon_3(p,i,j). \qquad (6) \end{aligned}$$

Let $l(j)$ be the link serving packets to receiver j. Theorem 1 follows from (3) and (5) of Theorem 4 with $l = l(j)$, $W^* = W_F$, $G^* = G_F$ and $\delta_j = C_{l(j)} T_{\vartheta(j)+1,0} + \kappa\left(\vartheta(j)\right) + \gamma\left(\vartheta(j)\right)$, since $\varrho(p) \leq \kappa(p)$ and $\tau(p) \leq \gamma(p)$. Theorem 2 follows from (4) and (6) of Theorem 4 with $l = l(j)$, $W^* = W_F$, $G^* = G_F$ and $\iota_{j,k} = b_{i,k} T_{\vartheta(j)+1,0} + \xi_{i,k} + \varsigma_3(\vartheta(j),i,k) + \varepsilon_3(\vartheta(j),i,k)$, since $\varsigma_3(p,i,k) \geq \varsigma_2(p,i,k)$ and $\varepsilon_3(p,i,k) \geq \varepsilon_2(p,i,k)$. Theorem 3 follows from Theorem 4, with $l = l(j)$, W_is and G_is given by (1) and (2).

We now outline the proof of Theorem 4. Note that $\Delta(i,l) \geq 1$, for each session i and link l in i's tree. Due to round robin sampling, every session is sampled at a rate that is more than $\hat{\lambda}_1$ in every link in its tree. A session first tries to transmit a layer 1 packet whenever it is sampled. Now, we assume that $\hat{\lambda}_1$ is greater than or equal to the bandwidth of the first layer of each session i, $b_{i,1}$, since the first layer must be transmitted without any packet loss for an acceptable quality of reception. Thus, the first layer of a session i is sampled at a rate of at least $b_{i,1}$ in every link in i's tree. We next prove that when $W \geq W_1$, and $G \geq G_1$, this lower bound on the sampling rate guarantees that every link in i's tree transmits at least $b_{i,1}t$ layer 1 packets of i in any interval of length t. When a session is sampled, it tries to send layer j packets if it can not send layer $1, \ldots j-1$ packets. Note that layer j packets of session i are generated at a rate of $b_{i,j}$, and $\sum_{k=1}^{\alpha(i,1)} b_{i,k} \leq \hat{\lambda}_1$. Thus, layers $j, j \leq \alpha(i,1)$ of session i are sampled at the rate of $b_{i,j}$ in every link in i's tree, and layer $\alpha(i,1)+1$ is sampled at a rate that is greater than or equal to $\hat{\lambda}_1 - \sum_{j=1}^{\alpha(i,1)} b_{i,j}$. We prove that when $W \geq W_1$, and $G \geq G_1$, this lower bound on the sampling rate guarantees that every link in i's tree transmits at least $b_{i,j}t$ layer j packets of i in any interval of length t, if $j \leq \alpha(i,1)$, and at least $\left(\hat{\lambda}_1 - \sum_{j=1}^{\alpha(i,1)} b_{i,j}\right) t$ packets in any interval of length t, if $j = \alpha(i,1)+1$. Thus, (4) of Theorem 4 follows for rank $p = 1$. Thus, if $W \geq W_1$, and $G \geq G_1$, every link in session i's tree transmits at least $\hat{\lambda}_1 t$ packets of session i in any interval of length t. Thus, (3) of Theorem 4 follows for rank $p = 1$. Also, clearly every link in i's tree transmits at most $b_{i,j}t$ layer j packets of i in any interval of length t, if $j \leq \alpha(i,1)$. Thus, (6) of Theorem 4 follows for rank $p = 1$, and layers $1, \ldots, \alpha(i,1)$.

Now, consider a session i and link l such that $\Delta(i,l) = 1$, i.e., all receivers of session i downstream of l have rank 1. Then, either session 1 generates packets at rate $\hat{\lambda}_1$ or there exists a link in the path of each receiver of session i downstream of l that offers a bandwidth of $\hat{\lambda}_1$ to session i. Such links are referred to as "bottleneck" links. In the first case, from the definition of $\alpha(i,1)$, session 1 generates only $\alpha(i,1)$ layers. Therefore, in any interval of length t, every link in i's tree transmits at most $\sum_{j=1}^{\alpha(i,1)} b_{i,j}t$ packets of session i, and $\hat{\lambda}_1 = \sum_{j=1}^{\alpha(i,1)} b_{i,j}$. In the second case, layer $\alpha(i,1)+1$ of session i is served at a rate of at most $\left(\hat{\lambda}_1 - \sum_{j=1}^{\alpha(i,1)} b_{i,j}\right)$ in the bottleneck links, and layers higher than $\alpha(i,1)+1$ of session i are rarely served in the bottleneck links. Thus, layer $\alpha(i,1)+1$ of session i is served at a rate of at most $\left(\hat{\lambda}_1 - \sum_{j=1}^{\alpha(i,1)} b_{i,j}\right)$ in l, and layers higher than $\alpha(i,1)+1$ are rarely served in l. Thus, (6) of Theorem 4 follows for rank $p = 1$, and layer $\alpha(i,1)+1$. Thus, session i packets are served at a rate of at most $\hat{\lambda}_1$ in l. Thus, (5) of Theorem 4 follows for rank $p = 1$.

Now, consider a session i and link l such that $\Delta(i,l) \geq p$. Using the upper bound (5) on the transmission rates of sessions with ranks lower than p, and round robin sampling, we show that i is sampled at a rate of at least $\hat{\lambda}_p$ in l. The rest of the argument is similar to the case for $p = 1$.

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. Hierarchical packet fair queueing algorithms *Proceedings of ACM SIGCOMM'* 96, *pp* $143-156$, Palo Alto, CA, August $'96$

[4] 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

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

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

[7] 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 $'96$

[8] 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 $1024-1039$

[9] 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

[10] 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

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

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

[13] 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

[14] S. Sarkar and L. Tassiulas, "Back pressure based multicast scheduling for fair bandwidth allocation," *Technical report*, Electrical and Systems Engineering Department, University of Pennsylvania, Available at http://www.seas.upenn.edu/ ∼swati

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

[16] 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

[17] 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

Saswati Sarkar (S'98, M'00, Member number 40260893) received Master of Engineering in Electrical Communication Engineering from the Indian Institute of Science in 1996 and Phd in Electrical and Computer Engineering from University of Maryland, College Park in 2000. She is currently an Assistant Professor in the department of Electrical and Systems Engineering in University of Pennsylvania. Her research interests are in resource allocation and performance analysis in communication networks. She received the motorola gold medal for the best masters student in the division of electrical sciences at the Indian Institute of Science and a National Science Foundation (NSF) Faculty Early Career Development Award in 2003. She has been associate editors of IEEE Transaction on Wireless Communications from 2001 and Journal of Computer Networks from 2005.

Leandros Tassiulas (S'89, M'91, Member number 00618256) was born in 1965, in Katerini, Greece. He obtained the Diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Maryland, College Park in 1989 and 1991 respectively.

He is Professor in the Dept. of Computer and Telecommunications Engineering, University of Thessaly, Greece and Research Professor in the Dept. of Electrical and Computer Eng and the Institute for Systems Research, University of Maryland College Park since 2001. He has held positions as Assistant Professor at Polytechnic University New York (1991-95), Assistant and Associate Professor University of Maryland College Park (1995-2001) and Professor University of Ioannina Greece (1999-2001).

His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols of wireless systems, sensor networks, high-speed internet and satellite communications.

Dr. Tassiulas received a National Science Foundation (NSF) Research Initiation Award in 1992, an NSF Faculty Early Career Development Award in 1995 an Office of Naval Research, Young Investigator Award in 1997 and a Bodosaki Foundation award in 1999 and the INFOCOM '94 best paper award.