

Session-Aware Popularity Resource Allocation for Assured Differentiated Services*

Paulo Mendes^{1,2}, Henning Schulzrinne¹, and Edmundo Monteiro²

¹ Department of Computer Science, Columbia University
New York, NY 10027, USA

{mendes,schulzrinne}@cs.columbia.edu

² CISUC, Department of Informatics Engineering, University of Coimbra
3030 Coimbra, Portugal,
{pmendes,edmundo}@dei.uc.pt

Abstract. Differentiated Service networks (DS) are fair in the way that different types of traffic can be associated to different network services, and so to different quality levels. However, fairness among flows sharing the same service may not be provided. Our goal is to study fairness between multirate multimedia sessions for an *assured* DS service, in a multicast network environment. To achieve this goal, we present a fairness mechanism called *Session-Aware Popularity Resource Allocation* (SAPRA), which allocates resources to multirate sessions based upon their number of receivers. Simulation results in a multirate and multi-receiver scenario show that SAPRA maximizes the utilization of bandwidth and maximizes the number of receivers with high-quality reception.

Keywords: fairness, multimedia sessions, multicast, differentiated networks, multirate sources.

1 Introduction

Almost all multimedia applications in the Internet use unirate sources, generating flows with rates that don't change over time. For example, the SureStream technology from RealNetworks allows streams' broadcast with multiple rates by creating unirate stream copies. This approach leads to bandwidth waste in heterogeneous environments, such as the Internet, because sources broadcast copies of the same stream in order to satisfy receivers with different quality requirements. This can be solved by replacing unirate sources with multirate ones. Multirate sources [8,19] divide streams into cumulative layers. Each layer has a different rate and importance, and the stream rate is equal to the sum of all its layers' rates. This approach avoids waste of bandwidth, since sources broadcast only one stream to all receivers, sending each stream's layer to a different multicast group. Receivers join as many multicast groups as their connection speed

* This work is supported by POSI-Programa Operacional Sociedade de Informação of Portuguese Fundação para a Ciência e Tecnologia and European Union FEDER

allows them [14], starting by the most important layer. We use the designation of *session* to define the group of all layers belonging to the same stream.

Due to their real-time characteristics, multimedia sessions need quality guarantees from the network. These guarantees can be provided by the DS model [2], which allows network providers to aggregate traffic in different services at the boundaries of their network. Each service is based upon a per-hop behavior (PHB), which characterizes the allocation of resources needed to give an observable forwarding behavior (loss, delay, jitter) to the aggregate traffic. One important question about Assured Forwarding (AF) [5] services concerns their capability to be fair. AF services provide *intra-session* fairness, between receivers in the same session, since each session's layer can be mapped to a different drop precedence, considering its importance. However, how to achieve *inter-session fairness* in AF services, allowing receivers from all sessions to get their required quality level without wasting resources, is still a challenging research topic.

The goal of our work is to contribute to the study of *inter-session fairness* between sessions in AF services, keeping the *intra-session fairness* property. To achieve this goal, we propose the enhancement of AF services with a *Session-Aware Popularity Resource Allocation* fair mechanism (SAPRA), which provides *inter-session fairness* by assigning more service bandwidth to sessions with higher number of receivers. SAPRA is a session-based mechanism and not only multicast-based, since hiding session information from DS routers results in *intra-session* unfairness, higher quality oscillations and lower quality for all receivers. SAPRA also includes a resource utilization maximization function, because fairness policies based only upon the number of receivers could still lead to waste of resources. This can occur when the bandwidth assigned to a session is higher than the rate really used by that session, as might happen with mobile phone or personal digital assistant (PDA) sessions, since they have low rate requirements and normally a high number of receivers. SAPRA also detects and punishes high-rate sessions in times of congestion, as an incentive for sessions to adapt to the network capacity.

We present ns¹ simulations that evaluate SAPRA behavior in a multirate multi-receiver environment using a simple dropper, which we called SAPRAD, and using RIO, the dropper normally used in AF.

The remainder of the paper is organized as follows. In section 2, we present a brief description of some fairness definitions and some multirate source implementations. Section 3 describes SAPRA functionality and section 4 presents simulation results. Finally, section 5 presents some conclusions and future work.

2 Related Work

There are several experimental multirate codecs, such as the Scalable Arithmetic Video Codec from the University of Berkeley² developed by D. Taubman [19],

¹ Network Simulator: <http://www.isi.edu/nsnam/ns/>

² Experimental software at: <http://www-video.eecs.berkeley.edu>

or the Scalable Video Conferencing project from the Framkom Research Corporation³ [8]. To fairly distribute AF resources between multirate traffic generated by these codecs, the *max-min* fairness definition [1] could be used since its formal definition is a well accepted criterion for fairness and its multicast definition [20] was extended to include multirate sessions [17]. However, Rubenstein et al. [17] show that *max-min* fairness can not be provided in the presence of discrete set of rates, as is the case of multirate sources.

The maximal fairness definition presented by Sankar et al. [18] exists in the presence of a discrete set of rates, but it doesn't consider the number of receivers in each session. Therefore, maximal fairness can't maximize resource utilization and at the same time maximize the number of receivers with good quality level.

Legout et al. present a proposal [11] to distribute bandwidth between sessions considering their number of receivers. However, this proposal assumes that every router in the path between the session's sender and its receivers keep information about the session's layers and the receivers receiving those layers. This proposal also doesn't maximize the utilization of resources and doesn't punish high rate flows.

Li et al. present [12] another proposal to improve *inter-session fairness* based upon the *max-min* fairness definition. Besides *max-min* limitation with discrete multirate sessions, this proposal only considers one shared link and doesn't consider the number of receivers and layers importance of a session.

3 SAPRA Fairness Mechanism

In this section, we introduce the *Session-Aware Popularity Resource Allocation* fairness mechanism (SAPRA), which is implemented only in DS-edge routers.

We assume that each possible multicast branch point is located only in DS-edge routers and that several multimedia applications can share the same host. We name each application *source* and each host *sender*. Since sources are multi-rate, they generate multimedia sessions with several layers, each layer identified by a Source-Specific Multicast (SSM) channel [6] - sender IP address and destination multicast group. Each receiver can join more than one session at the same time, even if those sessions belong to the same sender. To join a session, receivers start joining the SSM channel of the most important layer. They can try to increase their reception quality by joining more layers, always from the most important one. They can also get information about sessions using, for example, the Session Announcement Protocol (SAP) [4]. The number of receivers in each session corresponds to the number of receivers of the most important layer.

Implementing a fairness mechanism in DS-edge routers that only have information about multicast groups and not about sessions results in intra-session unfairness, higher quality oscillations and lower quality for all receivers. Fig. 1 shows the difference between a scenario where routers have information only

³ Project page: <http://mbc.framkom.se/projects/scale/>

about multicast groups and a scenario where routers have knowledge about session.

We assume that the session-based scenario has two sessions (S_1 and S_2) sharing a link with 1 Mb/s. Each session has 500 receivers, which mean that it has 0.5 Mb/s of bandwidth allocated. Session S_1 has three layers (l_0 , l_1 and l_2) joined by 500, 400 and 300 receivers respectively, and session S_2 has two layers (l_0 and l_1) joined by 500 and 400 receivers respectively. In the multicast-based scenario all layers are considered as independent multicast groups (flows f_1 to f_5 are layers from S_1 and flows f_4 and f_5 are layers from S_2), which means that the total number of receivers sharing the link is 2100. Therefore flow f_1 and f_4 have an allocated bandwidth of 0.24 Mb/s each, f_2 and f_5 of 0.19 Mb/s each and f_3 of 0.14 Mb/s. Considering for example session S_1 , Fig. 1 shows that the 100 receivers of S_1 that only join l_0 have the same reception rate (0.1Mb/s) and zero loss in both scenarios, since the rate is lower than the fair rate. However the 100 receivers that join l_0 and l_1 have a reception rate of 0.29 Mb/s and 5% loss in the multicast-based scenario and a rate of 0.3 Mb/s and zero loss in the session-based scenario. The situation becomes worst for the 300 receivers that join the three layers, since they have a reception rate of 0.43 Mb/s and 58% losses in the multicast-based scenario and a rate of 0.5 Mb/s and 16% losses in the session-based scenario. This shows that receivers have lower rate and higher loss percentage in a multicast scenario than in a session-based one. The multicast-based scenario isn't also *intra-session fair*, because AF drop precedences don't respect layers' importance. It also presents a higher quality oscillation, since receivers detect losses not only in the less important layer, but also in intermediary ones.

We propose two methods to implement SAPRA as a session-based mechanism in DS-edge routers. In the first method, each sender allocates consecutive multicast addresses to all layers inside a session and keeps one address gap between sessions. With SSM this method doesn't bring any address allocation problem, since each source is responsible for resolving address collisions between all the channels (232/8 addresses) they create. In this scenario each sender manages 2^{24} addresses in IPv4 and 2^{32} per scope in IPv6. With this method, DS-edge routers identify as belonging to the same session all layers that receivers join with consecutive SSM channels. The second proposed method is to change the way IGMPv3 [7] is used. The *auxiliary* data field of IGMPv3 reports can be used to include the multicast address of the most important layer - which identifies the session - in reports about other layers. So, DS-edge routers explicitly know what is the session of each layer. Routers that don't implement SAPRA ignore the

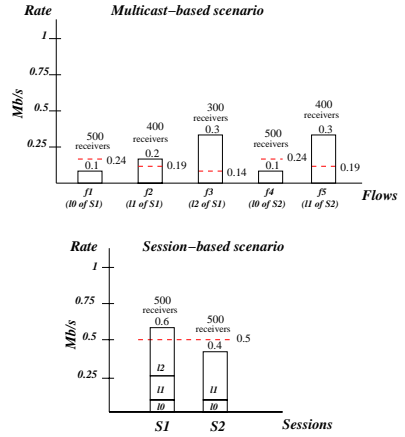


Fig. 1. SAPRA scenarios

auxiliary data field as is done in the current IGMPv3 implementations. In both proposed methods, routers know the relationship between layers in a session by the order receivers use to join sessions' layers. Receivers are motivated to join layers from the most important to the less important one, because less important layers are useless without the most important ones in the re-construction of the session's multimedia stream.

We assume that TCP and UDP traffic use different AF services. However unicast and multicast flows can share the same AF service. In this case SAPRA treats unicast flows as sessions with one layer and one receiver only. All layers of the same session use the same AF service, being however marked with different drop precedences. SAPRA only uses two drop precedences, IN and OUT, from the three allowed by AF services. We also assume that sources mark all their traffic as IN.

SAPRA has two components, one agent and one marker. Each DS-edge router has only one SAPRA agent and one marker for each downstream link. SAPRA agents exchange control information periodically with their neighbours. This information includes an *update* message sent to upstream neighbours with the number of receivers and fair rate of each session that presents changes in those values since the last time an *update* message was sent. This reduction of the *update* message size and the fact that agents don't need to have global network knowledge increases SAPRA scalability. The *update* messages information is used by agents to compute sessions' fair rates. Control information also includes a *sync* message sent to downstream neighbours. This message, which contains the lowest fair rate that each session has in the path from the source, can be used by quality adaptive mechanisms in the receivers. A brief description of the protocol used to exchange *update* and *sync* messages is presented in [16] and its performance study will be presented in a future paper. Next, we describe the SAPRA agent and marker.

3.1 SAPRA Agent

When a SAPRA agent receives an *update* message, it updates the local information about the sessions in the message and computes their new fair rates. In DS-edge routers that have local receivers, agents gather the number of receivers from IGMPv3 "State-Changes" reports.

Agents have to reserve local resources to store the received and computed information. For each upstream interface, agents reserve four bytes for each session and four bytes for each layer. For the local interface and for each downstream interface agents reserve twelve bytes for each session and eight bytes for each layer. As an example, consider 1000 sessions, each one with three layers, that are going through a DS-edge router with three downstream interfaces. Consider also that each session is present in each downstream interface and that the router doesn't have local receivers. In this situation the router reserves 124 Kb.

To compute session S_u fair rate, F_{ui} , in a link i , agents use Eq. 1, which defines F_{ui} as the ratio between the session's number of receivers, n_{ui} , and the

total number of receivers in that link, considering the AF service capacity⁴, C_i . In Eq. 1 m_i is the number of sessions that share link i .

$$F_{ui} = \left(\frac{n_{ui}}{\sum_{x=1}^{m_i} n_{xi}} \right) * C_i \quad (1)$$

All computed fair rates are adjusted considering downstream fair rates. This adjustment is required to maximize the utilization of resources, because sessions whose fair rate is higher than their downstream fair rate waste resources, since packets are dropped downstream. Therefore, if a session has a computed fair rate, F_{ui} , higher than its downstream fair rate, F_{uj} (j is a link downstream of i), F_{ui} becomes equal to F_{uj} and the rate difference, $F_{ui} - F_{uj}$, is added to the available shared bandwidth in the link, w_i . The available shared bandwidth allows fair rate increase for sessions that have a fair rate lower than their downstream fair rate. If the difference $F_{uj} - F_{ui}$ is lower than w_i , then F_{ui} becomes equal to F_{uj} and w_i is reduced by that difference. However if $F_{uj} - F_{ui}$ is higher than w_i , then F_{ui} is only added by w_i , and w_i becomes zero.

The available shared bandwidth is used by all sessions, starting by those with the highest number of receivers. This maximizes the utilization of resources increasing the number of receivers with good quality.

Agents functionality can be described by a fairness definition, which can be stated as: Consider that F_{ui} and F_{uj} are the fair rates of a session S_u in a link i and in a link j downstream of i , respectively. A fair rate allocation vector $V_i^1(F_{1i}^1, \dots, F_{m_i}^1)$ in a link i is said to be *SAPRA-fairer* if for any alternative feasible⁵ fair rate allocation vector $V_i^2(F_{1i}^2, \dots, F_{m_i}^2)$:

$$\forall u \in [1, m_i], F_{ui}^2 > F_{ui}^1 \wedge F_{ui}^2 \leq F_{uj}^2 \Rightarrow \exists v \in [1, m_i], F_{vi}^2 < F_{vi}^1 \wedge F_{vi}^1 \leq F_{vj}^1 \quad (2)$$

After being adjusted, sessions' fair rates are passed by the agent to each SAPRA marker present in the downstream links.

3.2 SAPRA Marker

Fig. 2 shows the SAPRA marker - shadowed component - which replaces the usual marker in AF services.

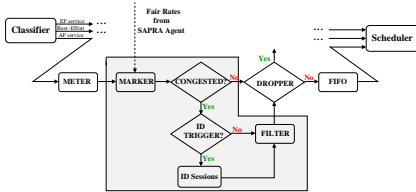


Fig. 2. SAPRA marker

This enhances the AF service with the capability to fairly distribute resources between sessions, based upon the fair rates computed by the SAPRA agent and the layers' average rate.

The marker needs to know the arrival rate of each layer. The easiest way to achieve this would be to obtain that information directly from the sources. How-

ever sources could indicate a lower rate than they actual have, trying to get a higher percentage of IN packets. Therefore a meter, included in the DS model

⁴ How the AF capacity in a DS-edge router is configured is a DS model implementation concern.

⁵ A feasible vector means that the sum of all fair rates is equal or lower than the AF capacity.

as shown in Fig. 2, is used to estimate average rates, maintaining the fairness mechanism independent of the sources.

With the information from the agent and the meter, the marker marks layer traffic as IN or OUT. Only packets that arrive already marked IN will be re-marked, since OUT packets are not compliant with upstream fair rates. All incoming IN packets are marked IN or OUT as follows: Considering that l_0 is the most importance layer and l_n the less important one of a session S_u , Eq. 3 and Eq. 4 give the probability that a layer l_k of that session has to be marked IN, P_{uki}^{in} , and OUT, P_{uki}^{out} , in a link i . With this marking strategy there is also a differentiation between sessions that have traffic marked OUT, since sessions with higher rates will have more packets marked OUT.

$$P_{uki}^{in} = \begin{cases} 1 & , L_{uki} \leq F_{ui} \\ \frac{M_{uki} - L_{u(k-1)i}}{r_{uki}^{in}} & , L_{uki} > F_{ui} \end{cases} \quad (3)$$

$$P_{uki}^{out} = \begin{cases} 0 & , L_{uki} \leq F_{ui} \\ \frac{L_{uki} - M_{uki}}{r_{uki}^{in}} & , L_{uki} > F_{ui} \end{cases} \quad (4)$$

In Eq. 3 and Eq. 4 session S_u has the following values in link i : fair rate F_{ui} ; rate of IN packets of its layer l_k , r_{uki}^{in} ; sum of all rates from layer l_0 to layer l_k - $L_{uki} = \sum_{x=0}^k r_{u,x,i}^{in}$; maximum value between the session's fair rate and the sum of all layers' rate from l_0 to l_{k-1} - $M_{uki} = \max(F_{ui}, L_{u(k-1)i})$.

When the meter detects that the link is congested, the marker filters all layers from sessions with rate higher than their fair rate plus their share of the available bandwidth, before sending the marked packets to the DS dropper. We define these sessions as *high-rate sessions*. The strategy to identify and punish high-rate sessions is based upon the Random Early Detection with Preferential Dropping mechanism (RED-PD) [13]. However, contrary to RED-PD, SAPRA uses fixed length intervals in congested periods to identify high-rate sessions and doesn't need to maintain a list of all layers that suffer drops in each interval. This simplifies the mechanism avoiding the estimation of the recent average packet drop rate used by RED-PD to compute their variable interval length.

In each identification interval, SAPRA starts by verifying which sessions have total (IN and OUT packets) rate, r_{ui} , higher than their fair rate. Session S_u total rate in a link i , is given by $r_{ui} = \sum_{x=0}^{n-1} r_{u,x,i}$, considering that each layer l_k from l_0 to l_{n-1} has rate r_{uki} .

A session with rate lower than its fair rate isn't using all its share of the link bandwidth, so the unused bandwidth becomes available for other sessions' OUT packets. Fig. 3 shows that SAPRA distributes this available bandwidth in equal shares between all sessions with rate higher than their fair rate and identifies which of these sessions are high-rate sessions.

To punish each high-rate session S_u in a link i , SAPRA computes its dropping probability in each identification interval t , $D_{ui}(t)$, using Eq. 5, where z_i is the available bandwidth in link i .

$$D_{ui}(t) = (D_{ui}(t-1) + \sigma_d + \frac{1}{100} * (1 - \frac{F_{ui} + \frac{z_i}{m_i}}{r_{ui}})) \quad (5)$$

This equation shows that in each interval the dropping probability of high-rate sessions is increased by two values: a drop factor, σ_d , and a value proportional to the excess rate the session is using. This excess rate corresponds to the difference between the session rate and the sum of the session fair rate and its share of the available bandwidth, as shown in Fig. 3.

The dropping probability of each session is used to compute dropping probabilities for their layers, being the less important layer the first to suffer an increase of its dropping, because losses induce a higher quality degradation if they happen in more important layers [9]. Since hierarchical codecs are tolerant to loss in less important layer, SAPRA computes layers' dropping probability with a linear quality degradation. However SAPRA can be configured to be more aggressive, dropping all layers that have a dropping probability higher than a predefined limit Θ_d .

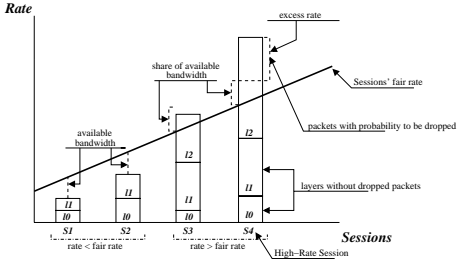


Fig. 3. Punishment mechanism

If in an identification interval a session is no longer identified, its dropping probability is halved until it reaches a minimum value, after which the session will stop to be filtered.

Packets that aren't dropped by the filter are sent to the DS dropper as happens with all packets from non-identified sessions. In the DS model the dropper is managed with RIO (RED with in/out) [3]. However, RIO introduces some complexity, since it needs to compute the total average queue size, the average queue size of IN packets, has different dropping scheme (random, front and tail) and its four thresholds can introduce oscillations. Therefore, we show that SAPRA has similar behavior with RIO and with a simpler dropper, which we named SAPRAD (SAPRA Dropper). SAPRAD manages a FIFO queue preferentially dropping OUT packets. When the queue is full an OUT packet is randomly discarded. Only if the queue doesn't have OUT packets, an IN packet is randomly discarded. This guarantees that layers with higher rates are more severely punished.

4 SAPRA Simulations

In this section we present simulations that aim to analyse the ability of SAPRA to distribute AF bandwidth between multirate sessions with different number of receivers, considering the number of receivers and the relationship between layers inside each session. We use a scenario - Fig. 4 - with three DS-edge routers and two congested links. The upstream link is configured with 10 Mb/s and the downstream with 5 Mb/s of bandwidth. The queue in each link has a size of 64 packets - default value in Cisco IOS 12.2 -, and each packet has a size of 1000

bytes. We analyse SAPRA’s behavior in the presence of two types of droppers, RIO, the dropper normally used in AF, and SAPRAD.

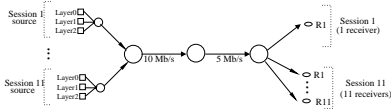


Fig. 4. Simulation scenario

S_{11} , increasing by one receiver per session. Although SAPRA can deal with any number of layers, we consider sessions with three layers in the present simulations, since this partitioning provides a good quality/bandwidth trade-off and additional layers only provide marginal improvements [9]. We performed sixty seconds simulations with sources that have increasing rates, from S_1 to S_{11} , in multiples of 25 Kb/s from session to session, starting with 25 Kb/s for S_1 . The session rate is the rate of its most important layer, l_0 , and each layer l_k has a rate equal to twice the rate of l_{k-1} . The dropping probability of all sessions is computed using Eq. 5, with σ_d equal to 0.5% and the dropper used is SAPRAD.

Fig. 5 shows that sessions’ fair rates are proportional to sessions’ number of receivers, since SAPRA distributes resources considering the number of receivers in each session. They also show that, in the upstream link, sessions use fair rates lower than the computed ones. This happens because SAPRA adjusts the upstream link computed fair rates, since they are higher than the downstream link ones. Another conclusion is that SAPRA respects layers’ relationship, since packet dropping starts always by l_2 , which can be clearly seen in Fig. 5 (right) where the rate of l_2 is reduced to a minimum value.

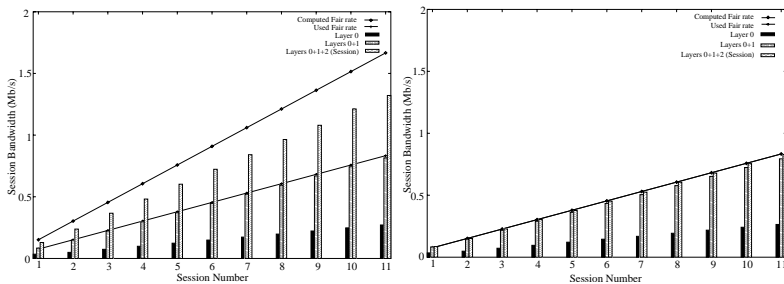


Fig. 5. Sessions’ fair rates in upstream (left) and downstream (right) links

In Fig. 5 sessions with higher rate suffer higher drop rates. For example, in the upstream link, S_{11} incoming rate is 1925 Kb/s and it has a loss rate of 31.35%, while S_{10} has an incoming rate of 1213 Kb/s and 30.72% loss. Fig. 5 also shows that in the upstream link all sessions are identified as high-rate sessions, since their incoming rates are higher than their fair rates and therefore the available bandwidth is zero.

The filter aggressiveness can be configured by changing σ_d and the dropping probability limit Θ_d . To better show the punishment effect, we used an equal number of receivers per session, i.e., sessions have the same fair rate. Fig. 6 shows results for the upstream link using a σ_d value of 5%: In the left figure, Θ_d isn't defined and so layers suffer a linear dropping increase and in the latter; in the right one, Θ_d is equal to 50%, after which layers are completely dropped.

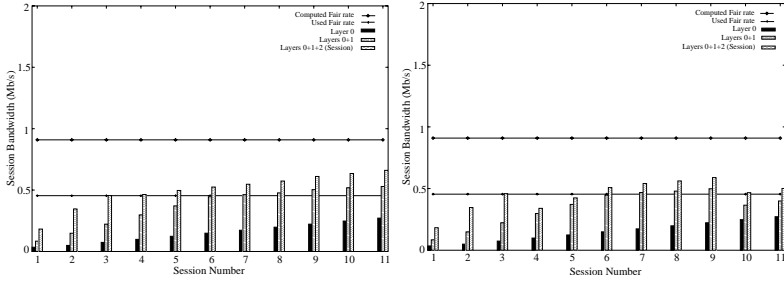


Fig. 6. Punishment mechanism with σ_d value of 5%

Fig. 6 shows that all layers in S_1 and S_2 have null dropping probability. The same happens for l_0 in any session. As for l_2 - Fig. 6 (right) - in S_3 presents a dropping probability of 22%, in S_4 of 61%, growing up until 100% from S_7 to S_{11} , which means that it's completely dropped from S_4 to S_{11} . Nevertheless, l_2 doesn't have a null rate in these sessions. This is due to the time gap between the beginning of the simulation and the moment agents receive the first *update* message, during which agents don't have any information about sessions, being unable to differentiate them. Consequently all layers have the same dropping probability, making possible for receivers to get a certain number of packets from all layers. SAPRA has similar results for the downstream link. These results can be found in [15].

To compare SAPRAD behavior against RIO, we used again a sixty seconds simulation but sessions with equal rate and one receiver only. SAPRA uses a value of 0,5% for σ_d and Θ_d isn't defined. RIO's minimum and maximum thresholds have the following values: IN_min of 60 packets, IN_max of 64 packets, IN_drop of 0.5%, OUT_min of 32 packets, OUT_max of 48 packets and OUT_drop of 50%. With these values the dropping of OUT packets is higher than the one of IN packets for the 64 packets queue, which approximates the behavior of the SAPRAD dropper.

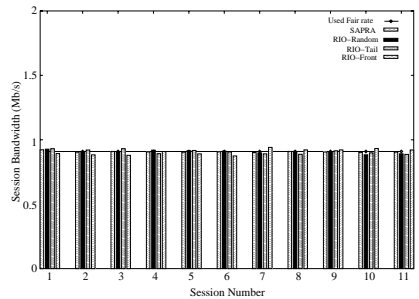


Fig. 7. SAPRAD and RIO

approximates the behavior of the

Fig. 7 shows, for the upstream link, that SAPRAD and RIO behavior is similar, being the session rate closer to its fair rate in the link when SAPRAD is used. These results show that an Internet Service Provider (ISP) that uses RIO can still use SAPRA to distribute AF resources between multimedia sessions. However implementing SAPRA with SAPRAD instead of RIO reduces the mechanism complexity. To understand what is the best RIO's configuration we made several simulations by changing the value of IN and OUT thresholds. These simulations show that SAPRA behavior with RIO has a high variation between different RIO configurations. Detailed results can be found in [15].

5 Conclusion and Future Work

This paper describes and evaluates SAPRA, whose components are only installed in DS-edge routers, computing sessions' fair rate based upon SAPRA fairness definition. SAPRA enhances DS functionality by fairly distributing bandwidth and by punishing high-rate sessions.

SAPRA distributes bandwidth between sessions considering their number of receivers, which increases receivers motivation to use multicast, since they will experience higher quality than unicast ones. SAPRA also increases providers' motivation to use multicast: ISPs can have more clients using fewer resources and multimedia providers can deploy new services that scale with large number of receivers. However, SAPRA doesn't attempt to be the optimal fairness mechanism, because social and economic issues can influence fairness as much as technical ones. But being based upon sessions' number of receivers and a maximal resource utilization function, SAPRA can be the base of a hierarchical fairness mechanism for multirate multicast sessions.

To evaluate SAPRA behavior, we presented simulations with two congested links that showed its performance with a simple dropper, SAPRAD, and also with RIO. Simulations showed that SAPRA maximizes the utilization of bandwidth and the number of receivers with high quality reception.

As future work we'll simulate SAPRA in more complex scenarios in order to analyze the SAPRA protocol oscillations with the variation of the number of receivers. We'll also create a receiver-driven adaptive mechanism that will use SAPRA network support, mainly fair rates collected in *sync* messages, trying to solve some of the problems presented by other adaptive mechanisms such as RLM [14] and RLC [21]. Legout et al. [10] show that RLM presents *inter-session* unfairness and has low convergence time and low link utilization, while RLC is unfair to TCP for large packets and its bandwidth inference mechanism is very sensitive to queue size. Also, both mechanisms can induce losses in all layers when a join experience occurs. This can be avoided if the adaptive mechanism is based upon SAPRA, since it guarantees *intra-session* fairness.

References

1. D. Bertsekas and R. Gallager. "*Data Networks*". Prentice-Hall, 1987.

2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. "An architecture for differentiated service". Request for Comments 2475, Internet Engineering Task Force, December 1998.
3. D. Clark and W. Fang. "Explicit allocation of best-effort packet delivery service". *Journal of IEEE/ACM Transactions on Networking*, 6(4):362–373, August 1998.
4. M. Handley, C. Perkins, and E. Whelan. "Session announcement protocol". Request for Comments 2974, Internet Engineering Task Force, October 2000.
5. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. "Assured forwarding PHB group". Request for Comments 2597, Internet Engineering Task Force, June 1999.
6. H. Holbrook and B. Cain. "Source-specific multicast for IP". Internet draft, Internet Engineering Task Force, March 2001.
7. H. Holbrook and B. Cain. "Using IGMPv3 for source-specific multicast". Internet draft, Internet Engineering Task Force, March 2001.
8. M. Johanson. "Scalable video conferencing using subband transform coding". In *In Proc. of ICSPAT'99*, Orlando, FL, USA, November 1999.
9. J. Kimura, F. Tobagi, J. Pulido, and P. Emstad. "Perceived quality and bandwidth characterization of layered MPEG-2 video encoding". In *In Proc. of SPIE International Symposium on Voice, Video and Data Communications*, Boston, MA, USA, September 1999.
10. A. Legout and E. W. Biersack. "Pathological behaviors for RLM and RLC". In *In Proc. of NOSSDAV'00*, Chapel Hill, NC, USA, June 2000.
11. A. Legout, J. Nonnenmacher, and E. W. Biersack. "Bandwidth allocation policies for unicast and multicast flows". In *In Proc. of IEEE INFOCOM'99*, New York, NY, USA, March 1999.
12. X. Li, S. Paul, and M. Ammar. "Multi-session rate control for layered video multicast". In *In Proc. of Multimedia Computing and Networking*, San Jose, CA, USA, January 1999.
13. R. Mahajan and S. Floyd. "Controlling high-bandwidth flows at the congested router". Tr-01-001, ICSI, April 2001.
14. S. McCanne, V. Jacobson, and M. Vetterli. "Receiver-driven layered multicast". In *In Proc. of ACM SIGCOMM'96*, pages 117–130, Palo Alto, CA, USA, August 1996.
15. P. Mendes. "Session-aware popularity resource allocation". <http://www.cs.columbia.edu/~mendes/sapra.html>.
16. P. Mendes, H. Schulzrinne, and E. Monteiro. "Multi-layer utilization maximal fairness for multi-rate multimedia sessions". Cucs-007-01, Columbia University, July 2001.
17. D. Rubenstein, J. Kurose, and D. Towsley. "The impact of multicast layering on network fairness". In *Proc. of ACM SIGCOMM'99*, Cambridge, MA, USA, September 1999.
18. S. Sankar and L. Tassiulas. "Fair allocation of discrete bandwidth layers in multicast networks". In *Proc. of IEEE INFOCOM'00*, Tel Aviv, Israel, March 2000.
19. D. Taubman and A. Zakhor. "Multirate 3-D subband coding of video". *Journal of IEEE Transactions on Image Processing*, 3(5):572–588, September 1994.
20. H. Tzeng and K. Siu. "On max-min fair congestion control for multicast ABR service in ATM". *IEEE Journal on Selected Areas in Communications*, 15:542–556, April 1997.
21. L. Vicisano, L. Rizzo, and J. Crowcroft. "TCP-Like Congestion control for layered multicast data transfer". In *Proc. of IEEE INFOCOM'98*, San Francisco, CA, USA, March/April 1998.