Abstract—Resilient Packet Ring (RPR) access method needs fairness algorithms to ensure all stations can transmit without being starved by the ones upstream. This paper tries to summarize the two standard fairness algorithms and the different proposals to improve their performance. The RIAS fairness definition is seen. The proposals DVSR, VQ, and Enhanced AM (which try to get the RIAS solution in a better way than the standard one) are commented. Also the non-RIAS proposals Equal Opportunity fairness and Weighted fairness are exposed.

Index Terms—RPR fairness, Virtual Queuing, DVSR, Equal Opportunity fairness, Enhanced AM, Weighted Fairness.

I. INTRODUCTION

Resilient Packet Ring [1], or RPR, is standardized as IEEE 802.17 (June 2004). This standard defines a packet access protocol addressed to fit in high rate metropolitan and wide area optical networks which are configured in a ring topology.

The optical ring consists of bidirectional point to point links, which results in a double-ringlet topology. This allows resilience, which is supposed to provide SONET/SDH protection features (50ms of maximum recovery time) in a packet based transmission while being physical layer independent.

In the normal case, only one of the rings is used to transmit user traffic. The other one is a backup ring to make the system capable of reaching any point even in case of one link failure.

RPR improves SONET/SDH bandwidth allocation, which is dynamic instead of being static. It also provides auto-restoration and fairness, which the dynamic bandwidth allocation competitor Gigabit Ethernet fails to.

In section II the basic concepts of RPR such as the traffic categories defined and their priorities are explained. Section III states the need of a fairness algorithm. In section IV the RIAS fairness definition is seen and in section V the two RPR standard fairness algorithms are exposed. In the following sections, DVSR, Virtual Queuing, Enhanced Aggressive Mode, Equal Opportunity and Weighted fairness algorithms are summarized. Finally, some conclusions and personal opinion are given.

II. RPR BASIC CONCEPTS

The two ringlets are called ringlet 0 and ringlet 1. Only one of them is used for user traffic when in a normal situation. When a station wants to send a packet, it inserts the data into the proper ringlet. When the frame reaches the next station, it’s removed from the ringlet if the destination direction is recognised to be part of the network of that node (outside of the ring), or it’s forwarded to the next node if it isn’t. This process can be done either in a store-and-forward mode or in a cut-through mode – start forwarding the packet when it hasn’t been completely received yet. To avoid a frame to infinitely go through the ring, each frame contains a finite TTL field.

It’s important to note that when a station routes a frame to outside of the ring, it isn’t forwarded inside the ring anymore to allow spatial reuse. That’s freeing the bandwidth of this packet in the rest of the ring where it hasn’t to be transmitted. This allows, as a simple example, simultaneous communication between 4 stations (two by two) with all the capacity available for each group of them if both paths don’t share any link (as shown in Figure 1, the two flows 1 → i, i + 1 → n).

A. Access method and traffic categories

As RPR is designed for ring networks, the traffic has to follow the only path, node over node, until it reaches its destination. So, two kinds of traffic can be distinguished in each node depending on its input.

On the one hand, the node receives traffic from the previous node in the ring, which can be addressed to this node and will be taken out of the ring, or can follow the ring. If the
second case occurs, it’s stored into the transit queue until it can be transmitted to the next node. This is called transit traffic. On the other hand, the traffic input can be the own node. In this case, it’s stored in the add queue and will compete with the transit traffic for being allocated into the link. The basic behaviour of RPR states that there’s only an add traffic packet inserted in the ring if the transit queue is empty. A schema of this structure is shown in Figure 2.

The design explained just above doesn’t contain any consideration about priorities and it should be improved. That’s why RPR, a part of his basic behaviour, also defines three classes of traffic. Here are the three classes with its own particularities and the subclasses defined for them:

- **Class A**: low-latency, low-jitter.
  - **A0**: reserved bandwidth, only used by the station who applied for it.
  - **A1**: reclaimable bandwidth, if it’s not used, can be temporally reallocated to other traffic.

- **Class B**: predictable latency, low-jitter.
  - **B-CIR**: committed information rate, also reclaimable bandwidth
  - **B-EIR**: excess information rate. <FE>.

- **Class C**: best effort traffic. <FE>.

The classes labelled with < FE > are called *Fairness Eligible traffic*, which means that they can be controlled by the fairness algorithms shown later in this paper. It’s also important to note that the A1 and B-CIR reservations are called reclaimable bandwidth, so when they are not used FE traffic can be allocated using these resources.

The reservation of the A0 traffic is done by broadcast over the full ring. After having received all the reservations from all the stations, each of them calculates the quantity of bandwidth to be permanently allocated for A0 traffic, and then they know the remaining capacity for reclaimable or Fair Eligible traffic.

Linked to the classes of traffic, there are several *traffic shapers*. Each node has one particular traffic shaper for each of the A0, A1, B-CIR and FE add traffics, which controls they don’t exceed their limitations. There’s also the downstream shaper, which manages all the traffic except for A0, and verifies it doesn’t go beyond the unreserved rate limit.

As a difference between ATM and other systems, no frame is going to be dropped to solve congestion problems. RPR doesn’t provide the option to mark exceeded traffic as discardable in case of congestion. So, all packets will arrive (sooner or later) to its destiny except for errors in their transmission.

To allow A traffic achieve its low-latency and low-jitter properties, transit queues should be almost always empty. That’s why transit traffic has a higher priority than add traffic, B transit traffic also has more priority than C add traffic, and both add an transit C traffics compete at the same level.

### B. Two transit queues

A more sophisticated structure to reach each traffic class features more accurately, consists of equipping each node with two transit queues: *Primary Transit Queue* (PTQ) and *Secondary Transit Queue* (STQ). Class A traffic will be stored in the PTQ while classes B and C will be redirected to the STQ. When choosing the packets to transmit over the link, PTQ frames are the first option. If this queue is empty, there is a chance for class A add traffic.

The next option is class B add traffic and finally, the last option is shared between class C add traffic and STQ frames. This hierarchy is shown in Figure 3. Figure 4 shows the latency in a simulation of a two queue nodes ring network, where the extremely constant latency of A traffic stands out, and B traffic’s latency is predictable compared to the C one’s. Details about the scenario of this simulation can be found in [3].
III. RPR Fairness

At the moment, FE traffic has been seen but it’s still to comment how this Fairness Eligible traffic is managed. Moreover, it hasn’t been clearly explained how the node selects between transmitting C class add traffic or STQ transit traffic. These points are kept on control by the fairness algorithm, which is now going to be commented and several alternatives are discussed in the following.

As explained before, each station transmits an add frame only if the transit queue is empty – in the simplified case. In this situation it’s usual that a station becomes starved by the upstream ones (the ones that fill it with transit traffic). So, if a station received a C rate transit traffic, it couldn’t have sent any add traffic. Hence, the fairness algorithm prevents the downstream station from this situation.

It’s important to clarify that the rates obtained from the fairness algorithm are only valid for the FE traffic. In the several studies to improve these algorithms, FE traffic is always supposed to be the only one transmitted through the ring. Other classes of traffic are omitted because of the bandwidth reservation they dispose of.

But first of all, to be able to implement different alternatives for a fairness algorithm, the concept of fairness has to be defined. Several definitions can be found but the RPR standard clarifies that the appropriate fairness definition is the RIAS one, which will be summarized at section IV.

As notation remarks, let the node origin of the congested link be called the head node, while the more upstream node transmitting through this link the tail node. The group of active nodes from the head to the tail one transmitting through the congested link is called congestion domain.

IV. RIAS Fairness Definition

The Ring Ingress Aggregated with Spatial Reuse (RIAS) is suitable for the RPR scenario. It defines two statements to clarify what can be understood as fair.

1) The granularity level is set to ingress-aggregated flows (IA). This is, all the flows originated at the same node that transit through the station running the fairness algorithm are treated as an only entity, without mattering its destination.

2) Extra bandwidth can be applied by IA flows when it’s not used. This is used to reach a maximum profit of the spatial reuse.

This algorithm can be generalized to obtain weighted fairness, which is later discussed.

Using the following notation,
- \( f(i, j) \) flows which go from \( i \) to \( j \).
- \( R_{i, j} \): candidate RIAS fair rate for \( f(i, j) \).
- \( F_n \): allocated rate on link \( n \).

\( R \) matrix is said to be feasible if
\[
R_{i,j} > 0 \quad \forall f(i,j) F_n \leq C \quad \forall \text{link } n \quad (1)
\]

It’s easy to understand that the allocated rate on a link can’t be greater than this link’s capacity, and that any existing flow should have an assigned fair rate greater than 0.

Then, \( n \) is called a bottleneck link with respect to \( R \) for \( f(i, j) \) crossing \( n \), denoted by \( B_n(i, j) \) if these two conditions are satisfied:

1) \( F_n = C \)
2) If \( IA(i) \) is the only IA at \( n \),
   - \( R_{i,j} \geq R_{i,j'} \).
   - If isn’t,
     \[
     IA(i) \geq IA(i') \quad \forall i = i'
     \]
     and inside this \( IA(i), R_{i,j} \geq R_{i,j'} \).

for all flows crossing link \( n \) in each case. Then, the RIAS fair proposition is stated as:

**Proposition 1 (RIAS fair):** \( R \) is RIAS fair if it is feasible and for each \( f(i, j) \), \( R_{i,j} \) can’t be incremented while keeping feasibility without having to reduce \( R_{i,j'} \) for any other flow which satisfies \( R_{i,j} \leq R_{i,j'} \) or \( IA(i') \leq IA(i) \) at the common link.

The former condition is to introduce intra-station fairness, while the latter to inter-station. No flow can improve its rate by gaining it against other flows with a lower rate. Intra-station does it by considering all flows coming from the same source station while inter-station just considers the ingress-aggregates. This behaviour is equivalent to the max-min but here it’s applied to two different granularities.

As a result, each IA flow will have at it’s disposal no less than \( \frac{C}{N} \), but only if has enough add traffic to fill it – \( N \) is the number of active nodes at the selected link \( n \).

To summarize, using this idea of fairness, only one fair rate \( F_n \) has to be imposed by the congested node \( n \) to the other nodes contributing to the congestion. The ones that have lower add traffic rates will only use part of their rate, while the ones exceeding will have to limit their output rate. As node \( n \) isn’t clear about the real requested rate of each node, \( F_n \) could be difficult to calculate and could lead to oscillating estimations.

V. RPR Standard Fairness Algorithms

The IEEE 802.17 standard defines two fairness algorithms. Both aim to converge to the RIAS fair solution, but they do it in different ways.

A link is considered as congested when its transit queue occupation reaches a previously configured threshold. The main idea, then, is to share the bandwidth between the stations interested in it when demand can’t be afforded. The congested station will determine a fair rate for the stations transmitting through it, which will be communicated by a fairness message using the backup ringlet. The reason why this message goes upstream is to reach as soon as possible the stations causing the congestion, which usually are the ones immediately before the congested one.

The fair rate is calculated (in a different way depending on RPR-AM or RPR-CM) without having to measure each flow...
rate. This fair rate value will be only estimated because there’s even no way to know the exact required rate for each IA in the actual RPR standardization. As not all of the active stations might fully use the fair rate, link $n$ capacity might not be filled even with node $n$’s add traffic. Then, the fair rate would be increased to reach the RIAS result. Note that this would produce some oscillation before reaching the theoretical value. The way of calculating and updating this rate for new more accurate ones is the difference between the two standardized fairness modes.

A. Conservative mode: RPR-CM

The fair rate is initially established as

$$\frac{BW}{N}$$

where $N$ is the number of stations transmitting frames through the congested link. This rate is applied only to the FE traffic.

Then, it’s sent to the upstream stations contributing to the congestion and it’s not updated until all the stations have adopted their new rate. This adoption is detected by measuring the congestion state. We can see that in a simulation using the scenario of Figure 5 it results in the oscillation shown in Figure 6 where 27% of the capacity is lost at the head node. This behaviour is due to the slow convergence of this algorithm. The simulation considers a dynamic traffic produced by two state sources.

So, some kind of improvement should be done to avoid this quite high link usage loss.

B. Aggressive mode: RPR-AM

The fair rate estimation is calculated by low-pass filtering the add rate of the congestion station — an exponential averaging filter. This algorithm continuously transmits the new fresh estimated fair rates, by default every 100μs. When the congestion has already been solved, a fair message indicating no congestion is sent upwards. Then the previously fair rate advised nodes will start to rise gradually their add rates and the head node will become congested again. At this time, the average will be more accurate to the RIAS solution and slightly greater than the previous one. Figure 7 summarises this process and illustrates where bandwidth loss happens.

In case of static traffic — where changes are due only to the fairness algorithm not to variable rates of the source traffic — when the STQ gets empty, there’s a lack of a message to tell the upstream nodes that there are not enough packets in the STQ to completely fill up the outgoing frames. At this moment, they keep gradually increasing their rates according to the expression in the same schema, while a faster increase process would reach a better bandwidth usage. This possibility is studied at section VIII.

The simulation in the Figure 5 scenario with the same dynamic traffic as before is shown in Figure 8. In this case, a 16% of usage loss is given due to the same oscillation also present here.

VI. DVSR

The Distributed Virtual time Scheduling in Rings (DVSR) [5] fairness algorithm tries to reach the theoretical RIAS result not by oscillating around it by using estimations based on the unused capacity of the congested link, but measuring each IA rate and defining the fair rate as a function of these. Of course, it requires much more computing resources because
of the separate measurement of each IA. But, in contrast, it converges faster than the previously commented algorithms.

Let the congested node be called node $n$, and $r_i^n$ the rate of traffic arriving at node $n$ originated at node $i$, which has to be forwarded to the next node. The fair rate for node $n$, $F_n$ is calculated as:

$$F_n = \begin{cases} \max_i \{r_i^n\} + (C - \sum_i r_i^n) & \text{if } \sum_i r_i^n \leq C \\ C - \sum_{i \neq n, r_i^n < F_n} r_i^n & \text{otherwise} \end{cases}$$ (2)

where $I = \{i : r_i^n \geq F_n\}$. 

So, in the first case, $\sum_i r_i^n \leq C$ includes the case $i = n$, which is the $n$ add traffic. Then, this sum takes into account all the traffic to be transmitted through the congested link, and the inequality means that the link capacity is not exceeded. In this case, the fair rate is set as the maximum IA rate plus the left capacity.

In the second case, the rate to be transmitted by the link (including node $n$’s add traffic) exceeds the link capacity. In the formula there, the sum takes into account only the already fair IA rates. The capacity not used by these fair IA rates is shared between all the unfair rates. This will reduce the fair rate to fit in $C$.

As seen in eq. (2), the previous $F_n$ is used to calculate the fair rate $F_n$ (in the sum of the fair IA rates). This makes that the algorithm being iterative what requires a loop. Moreover, a sort operation has to be performed. That’s why the computing requirements of this algorithm are important. Its complexity sort operation has to be performed. That’s why the computing requirements of this algorithm are important. Its complexity rises at $O(N \log_2 N)$ where $N$ is the number of IA flows active at $n$, while RPR-CM and RPR-AM complexity doesn’t depend on $N$.

DVSR algorithm is compared to other proposals in section VII-C.

VII. VIRTUAL QUEUING ALGORITHM (VQ)

The Virtual Queuing (VQ) algorithm [4] tries to reach the RIAS fair solution converging faster than RPR-AM and RPR-CM modes while being less costly than DVSR.

A. The main idea

Let $r_i^n(k)$ be the rate of traffic arriving at node $n$ originated at node $i$, which has to be forwarded to the next node during the control time interval $k$. Other notation like previous sections might be used by adding $(k)$ to indicate the $k$ control time interval instance. Then, the rate of traffic arriving to node $n$ is

$$r^n(k) = \sum_i r_i^n(k) = \sum_{i \neq n} r_i^n(k) + r_i^n(k) \leq C + F_n(k)$$

The sum

$$\sum_{i \neq n} r_i^n(k)$$

has to be lower than $C$ because, in fact, it comes from the $n-1$ link which has, in principle, the same capacity as link $n$. The remaining term $r_i^n(k)$ represents the congested node $n$ add traffic. If the previous sum is equal but not less than $C$, then node $n$ can’t fairly send its own traffic. Of course, $n$ node add traffic should be lower or equal than its own fair rate $F_n$.

As the traffic that exceeds the $C$ rate is stored in a queue before being transmitted, [4] compares each node to a virtual queue which service rate is $C$ and its arrivals rate is $r^n(k)$. Then, the congestion domain results as shown in Figure 9.

B. The VQ algorithm

In this section the algorithm used by VQ to reach the RIAS fair solution is exposed and briefly commented. Let’s comment it’s code, which is in Algorithm 1. Let $n$ indicate the congested node and $F_n(k)$ the fair rate just previously advertised, which will be used to calculate $F_n(k+1)$. Some of the values in the algorithm are:

- $N_i(k)$: Traffic received at $n$ from node $i$.
- $E_i$: Total quantity of traffic originated at $i$ going through $n$, according to the last $F_n(k)$.
- $E_S$: Total traffic quantity of rate-limited nodes.
- $E_U$: Total traffic quantity of input-limited nodes — this is, nodes that can’t fill up their fair rate because their add traffic rate is lower.
- $f$: coefficient to multiply to $F_n(k)$ to get the updated value.

```
1: $N_i(k) \leftarrow r_i^n(k) \cdot T$
2: $N_n(k) \leftarrow \max \{r_i^n \cdot T, \text{local_queue_size}\}$
3: $E_i \leftarrow \min \{N_i(k), T \cdot F_n(k)\}, \forall i$
4: $E_S \leftarrow \sum_{i \in S} E_i$
5: $E_U \leftarrow \sum_{i \in U} E_i$
6: if $E_S \neq 0$ && $E_U < C \cdot T$ then
7:   $f \leftarrow C \cdot T - E_U$
8: end if
9: if $E_U \geq C \cdot T$ then
10:   $f \leftarrow C \cdot T - E_S$
11: end if
12: if $E_S = 0$ && $E_U < C \cdot T$ then
13:   $f \leftarrow 1$
14: end if
15: $F_n(k+1) \leftarrow \min \{f \cdot F_n(k), C\}$
```

Algorithm 1: VQ fair rate calculation

The aim of line 3 is to avoid having used old data if the node $i$ wouldn’t have applied $F_n(k)$ yet. So, the minimum of both possible rates is stored in $E_i$. From line 6, this is equivalent
to a max_min* algorithm but written quite efficiently. If line 6 condition is satisfied, \( f \) should be greater than 1 to alleviate the fair rate because all the capacity isn’t been used. Note that what’s done in this case is to share the total amount of traffic that can be transmitted without the one transmitted by the input-limited nodes, between the rate-limited ndoes — in fact it’s a way to grant input-limited node rates and distribute the sparse bandwidth between the rate-limited ones (which will have a greater rate).

On the other hand, line 9 results in \( f \leq 1 \), as max_min would also do because input-limited rates already fill up all the available bandwidth (including \( n \)'s add rate). Finally, in the last case at line 12, the same fair rate is kept because there are only input-limited flows and they don’t need the full bandwidth, this means that the congestion has been controlled.

C. Simulation and comparison

In the paper, it’s stated that the head node of the congestion domain is the worst hit one. Therefore, the Accumulated throttled traffic (ATT) of this head node is considered a right figure of merit to characterize how good is the fair behaviour of each algorithm. It’s defined as follows, where \( \langle \cdot \rangle \) represents the mean value over a time interval and \(ATT(L)\) means to take into account the \( L \) control intervals \( K = 1 \ldots L \).

\[
ATT(L) = \sum_{k=1}^{L} (F(k) - r(k)) \cdot T
\]

\[
\langle ATT \rangle (L) = \sum_{k=1}^{L} \frac{F(k) - r(k)}{L}
\]

\[
\langle ATT \rangle \% = 100 \cdot \frac{\langle ATT \rangle }{\sum_{k=1}^{L} r(k)} \cdot L
\]

In regard to the simulation and comparison, despite also having to measure each IA rate, VQ presentation paper boasts about having lower computational complexity while better fairness properties and the same convergence speed as DVSR. Figure 10 shows the analytical comparison of ATT at the head node upper bound between DVSR and VQ, depending on the on-off ratio for the dynamic traffic two-state sources. It shows that VQ analytically gives better fairness performance than DVSR. Figure 11 compares this bound to a simulation and the upper bound is correctly greater than the simulation results. It’s also clear that VQ behaviour is better at the ATT simulation.

Figure 12 compares the convergence time between RPR-AM, RPR-CM and VQ depending on the number of nodes in the congestion domain. As RPR-CM is better for few nodes than RPR-AM, VQ shows an incredibly better performance at all scales.

Finally, Figure 13 shows the throughput loss depending on the off-time to on-time ratio of the two-state traffic sources. Here, VQ is also shown as a better alternative.

DVSR isn’t shown at these last pictures because it’s supposed to behave similarly to VQ compared to RPR-X algorithms. But what the paper doesn’t speak about is how much computational requirements does VQ use compared to DVSR and RPR-X. It’s said to cost less than DVSR but no idea is shown about how significantly less to be implemented to the nodes. How far is it from DVSR and near to RPR-X? It doesn’t seem to provide such an improvement to this point because it’s also based in measuring IA rates and an iterative process.
although it’s said to avoid a sort operation which DVSR uses.

VIII. ENHANCED AGGRESSIVE MODE

Enhanced Aggressive Mode (Enhanced AM, E-AM) [2] is a proposal to improve the standard RPR-AM algorithm.

Regarding Figure 7, which shows the standard RPR-AM behaviour, it can be observed that after having solved the congestion (because a fair rate has been imposed), the head node sends a null fair rate packet to indicate other nodes to gradually increase their rates — and then it proceeds to calculate the new updated fair rate. Then, it’s exposed in the paper that when the head node STQ is empty, other nodes rate is still rising quite slowly while it could immediately be set to their rate requirements. From the moment when the STQ becomes empty to when the load becomes greater than $C$, some bandwidth waste has occurred: this period should be as short as possible. RPR-AM has no way to shorten it because each source rate rising scale has no methods to be changed from the head node.

What Enhanced-AM proposes is to set a threshold $\text{alv\_threshold}$ over the STQ, and when its occupation is lower than it, an alleviation advertisement is sent upstream. The nodes receiving this warning should increase their rate to fit all their requirements. Then, the behaviour of this improvement would result as shown in Figure 14. Node’s rate would gradually increase from the null rate advertise to the alleviation advertise following the RPR-AM progression $a_k = \frac{C-a_{k-1}}{\text{Coeff}}$. $\text{Coeff}$ is set to 64 by default. When this last advertisement is received, the node sends at its maximum rate.

The $A_T$ threshold can be either over or under zero. The over zero position is clear because it’s compared to the STQ length. The under zero value is compared to the number of free slots transmitted in the link which couldn’t be filled with data because STQ was empty, during one full packet transmission time.

A. Simulation and comparison

The simulation in the paper uses the scenario in Figure 15 where $C = 622\text{Mbps}$, flow(1,3) requires 622Mbps and flow(2,3) requirements vary between 5, 10, 50, 100, 200 and 250 Mbps in the different simulations. Figure 16 shows the results for different $A_T$ alleviation thresholds. The case $A_T = -\infty$ is equivalent to the standard RPR-AM. From my point of view, this figure must be mistaken because according to its legend the RPR-AM would have achieved nearly 100% link utilization, while it should have been Enhanced-AM. The paper also shows a table with example values for $A_T$ where can clearly be seen that the minimum utilization threshold are values like $-1760$ or $-751$ (depending on f(2,3) rate) while ideal $A_T$ has values like 11 or 9, much closer to 0.

So, despite this probably switch mistake in the chart legend, the proposal would succeed to get better utilization results. Otherwise, it wouldn’t have any sense because if it’s even not useful in their simulations, I can’t imagine how it would be in adverse situations.

IX. EQUAL OPPORTUNITY FAIRNESS

The Equal Opportunity Fairness (EO) [6] is a proposal to improve the RIAS fair solution. In contrast to the previous proposals, it’s not a different way to reach the RIAS solution but it obtains a different one.

According to [6], RIAS gives unbalanced conditions between long and short flows — length is understood as number of nodes they have to cross. For a long flow, it’s more difficult to get a certain rate because it has to be allowed for the fair rate of each of the nodes it crosses. On the contrary, a short flow depends on fewer nodes. The situation is illustrated in Figure 17, where the short flow only depends on 2 links, while the long one depends on more than $N$ of them. In other words, it’s more likely to find a congested node if you have to traverse $N \gg 2$ than if you only go through a couple of them.
Still referring to Figure 17, the nodes inside the congestion are modeled as two-state nodes switching between $C = 100\text{Mbps}$ and $0.2C$ in periods of $T$ much greater than the convergence time (to avoid discussing about convergence and keep the interest in final solutions). The short and long flows, which are the case of study, have both the same demand $0.5C$ so that they can completely use the first link equally.

Figure 18 (a) shows the simulation results (for $N=7$ nodes in the congestion domain) comparing the RIAS model (a and b) and de EO fairness (c and d). As we can see in (b), the short flow is assigned 50% of the full capacity (let $R$ be the short flow rate requirements, $R = 0.5C$), while the long flow oscillates depending on the head node congestion. Its maximum rate (when there’s no congestion at node $n$) is $0.5C$.

What EO fairness tries to improve is this punctual issue in RIAS inter-station fairness. It tries to keep some kind of history of the penalization a flow gets and compensate it when the downstream congestion allows it.

In the example, when nodes $\{2 \ldots N-1\}$ are in high state, there’s no choice for the long flow to reduce its speed under the fair rate it’s been assigned, while the short flow is free to transmit its required $0.5C$. But, when nodes $\{2 \ldots N-1\}$ are in low state, RIAS would allow the long flow to share the first link bandwidth with the short flow in equal parts: $0.5C$ for each.

What EO fairness would do in that case is to compensate the previous penalty for the long one and let it transmit as much as the $N$ node fair rate permits. Moreover, when the long flow was limited by the $N$ node fair rate, the short flow would be capable to send the previously throttled traffic, now faster than $0.5C$.

The normalized mean rates of the example and $R = 0.8$ (short flow rate requirements), for the RIAS case are $\bar{r}_{1,2} = 0.650$ and $\bar{r}_{1,7} = 0.333$. In the case of EO fairness, they would improve to $\bar{r}_{1,2} = 0.517$ and $\bar{r}_{1,7} = 0.483$ as seen in Figure 19.

This behavior would increase the long flow mean rate while slightly decreasing the short flow one, but the first link ($1 \rightarrow 2$) total throughput would be kept near 100%. In this scenario, RIAS results in a 17% throughput loss in link $1 \rightarrow 2$.

**A. The proposed algorithm**

The concrete proposed algorithm is shown in Figure 20. Let’s see what some of the variables are used for.

After each control interval, the fair rates $F_i$ of the downstream nodes are received. Then, the EO fair service...
rates are calculated for each VDQ, $r_{i,j}$. The variable $d_{n,j}$ represents the current service demand of the VDQ and $e_j$ keeps the history of each VDQ's penalty in respect to other VDQs. $k$ represents the further node through which information now transiting node $i$ will transit.

The compensation is done by assigning part of a $j$ node available bandwidth to other VDQs previously worst affected. This compensation bandwidth, $E_j$, is lower or equal than $z \cdot C$, where $z$ is used to equilibrate the percentage of this bandwidth to be used for compensation. This is, $z$ represents the trade off between fairness and bandwidth availability for every flow. For $z < 1$ every flow will be assigned some part of the bandwidth. Of course, the bandwidth not used for compensation ($F_j - E_j$) is shared between the local flows by max-min (like in RIAS was).

The variable $v$ is used to determine how long is the history kept for. For $v < 1$ the deficit records are reduced exponentially each cycle, while for $v = 0$ there’s only 1 control interval memory. Finally, $v = 1$ represents infinite memory.

### B. Evaluations and conclusions

What’s proposed in this paper is to create a history, which is a memory to mix inter-station and intra-station fairness the way they can provide a maximum throughput in each node. With this, it’s achieved to get — in some conditions — a quite better mean rate for long flows while slightly lowering the short flows mean rate. This would succeed only in some traffic conditions like the one shown. But in the cases it wasn’t useful it wouldn’t deteriorate the system performance.

Some of the simulations in the paper have been shown in section IX where the improvements of this proposal were commented. The proposal’s drawback is the introduction of some delay jitter to the short flow. This is because it has to slow down when the long flow is being compensated. The paper justifies that it’s acceptable because this jitter is bound to the order of μs. But, from my point of view, each particular network and even each particular traffic should be considered if it can suffer the delay jitter increase. As it only affects FE traffic, in most of the cases, probably will.

### X. WEIGHTED FAIRNESS IN RPR

The weighted fairness version of the RIAS definition isn’t by default included in the RPR fairness algorithm. Weighted Fairness in RPR [7] proposes an extension to make this feature available.

### A. The proposal

First of all, let’s see some particular notation used in the paper:

- $N$: total number of stations in the ringlet.
- $w_n$: station $n$ weight.
- $f_{st}$: s to t stations flow and the path followed.

A VDQ is defined as the aggregation of different flows crossing a node with the same destination.

- $E = \{ f_{st} \}$: vector with all active flows.
- $R = \{ r_{st} \}$: fair rate vector with the rates for each of the flows.

Then, the total allocated rate on link $n$ is

$$ T_n = \sum_{\forall s,t: \text{ link } n \in f_{st}} r_{st} $$

Using this notation, the RIAS condition for $R$ feasible (eq. 1) becomes

$$ r_{st} > 0 \quad \forall s,t: f_{st} \in E $$

$$ T_n \leq C \quad \forall n \in N: 0 < n \leq N $$

All the flows originated at $s$ which go through $n$ are

$$ A_n(s) = \sum_{\forall t \in N: \text{ link } n \in f_{st}} r_{st} $$

For a feasible vector $R$, the $n$ link is a bottleneck link $B_n(s,t)$ to $R$ because of $f_{st}$ if the next conditions are satisfied:

$$ T_n = C $$

$$ r_{s't'} \leq r_{st} $$

$$ \forall s', t': s' \neq t' \& (t' \neq t) \& (\text{link } n \in f_{s't'}) $$

$$ A_n(s') \leq A_n(s) $$

$$ \forall s', t': s' \neq t' \& (s' \neq s) \& (\text{link } n \in f_{s't'}) $$

Condition 3 means all the capacity of the congested link is used. Condition 4 shows that the flow $f_{st}$ is the one that makes the congestion because its fair rate has to be at least equal or greater than the rest. In fact, it is the one asking for more resources so in RIAS fairness it should be assigned at least the same resources as the other flows. Finally, condition 5 forces that the aggregate flows from other stations (not the congesting one) have a lower or equal rate.

Then, the RIAS fair proposition 1 is adapted to its weighted version as

**Proposition 2 (Weighted RIAS fair):** $R$ is “weighted” RIAS fair if it’s feasible and if for each $f_{st}$, $r_{st}$ can’t be increased while maintaining feasibility without decreasing $r_{s't'}$ of some flow which

$$ r_{s't'} \leq r_{st} \forall s', t': s' = s \& f_{s't'} \in E $$

$$ \frac{A_n(s') + A_m(s')}{w_{s'}} \leq \frac{A_n(s) + A_m(s)}{w_s} $$

$$ \forall s', t', m, n: s' \neq s $$

$$ f_{s't'} \in E $$

$$ (\text{link } n, \text{link } m) \in f_{s't'} $$

$$ (\text{link } n, \text{link } m) \in f_{st} $$

Condition 6 ensures fairness for the flows originated at the same station while condition 7 takes care of the fairness between the different $IA$ flows. There is where the weights are introduced. If $m = n$ the flows $f_{st}$ and $f_{s't'}$ have the
same bottleneck — they are both to blame for the congestion there. If \( m \neq n \) both flows don’t have any common bottleneck but one goes though the other’s bottleneck.

Note that this proposal requires measuring the rate of each particular ingress aggregated flow, operation not performed by the two standard algorithms and which significantly increases the computational complexity of the process.

B. Example scenario

The paper [7] proposes an example scenario where to use weighted fairness. This same scenario is also used for simulations in section X-C. Figure 21 shows the way the nodes are connected. Node 5 is a video server with a 400Mbps upload rate (50 TV channels), which is sent to the nodes 6, 7 and 8 (video customers). Moreover, node 4 is the gateway to the internet while nodes 6, 7 and 8 download 300Mbps from there (an aggregated of 200 connections of 1.5Mbps per final user). The link capacity is \( C = 600 \text{Mbps} \).

If the RIAS fair solution is applied to this scenario, node 4 and node 5 get the same rate (the link \( 5 \rightarrow 6 \) is supposed to be congested \((400\text{Mbps} + 300\text{Mbps} > C)\)). In this case, only 37 of the 50 TV channels could be transmitted.

The weighted RIAS solution would allow assigning a higher weight to the video traffic so that all the TV channels could be transmitted. To determine the weight to assign, the 400Mbps of video have to be taken into account. Then, for node 4 should remain \( 600 - 400 = 200 \). From this result,

\[
w_5 = \frac{400\text{Mbps}}{200\text{Mbps}} = 2
\]
\[
w_4 = 1
\]

From my point of view, this scenario is not really typical because what I would do to combine both internet and video traffic is establish a different class for the video one. If it had been class A or class B-CIR it wouldn’t have needed such weights to ensure its required rate. So, concept of the scenario may be a little bit unrealistic but, of course, the need of weights could be applied to other situations more credible.

C. Simulation results

The results show that the previous scenario doesn’t work as expected: they get the bandwidth shared 50%-50% between the nodes 4 and 5, despite their different weights. Moreover, there’s a quite big amplitude undesired oscillation. The problem is that the highest weighted node selects to transmit a packet from the STQ than the add traffic queue too often. That’s why if the nodes 4 and 5 positions are switched, the results are the ones expected.

To definitively solve this issue, the selection algorithm of the RPR has to be modified by inserting the local weight when choosing between STQ or add traffic. It’s necessary to divide the add rate or to multiply the forward rate with the local weight. This increments the computational requirements of the algorithm but the paper suggests to assign the weights as power of 2 because then the new multiplication is just a shift right operation.

This solution works properly and the results are the ones expected without such a big oscillation.

XI. Conclusions

There are several proposals to improve the RPR-AM algorithm which mostly try to address the oscillation and the speed of convergence. As there are such proposals trying to solve the same two problems, they can be understood as the main standard fairness algorithms weaknesses.

There’s the trade off between speed of convergence and computing requirements. From my point of view, the best equilibrated one for this seems to be the VQ proposal, but it’s just an opinion because there aren’t many details about the complexity reducement of this proposal against DVSR. Also the Enhanced-AM is a nice way to improve the throughput without too many changes.

There are also a few algorithms that discuss the possibility of changing the RIAS solution. The first of them is the Equal Opportunity fairness. It could be a good idea in some cases but the delay jitter introduced could be a problem. The other one is the Weighted extension of the RIAS solution, which in my opinion could be introduced in the standard because it hasn’t any important drawback (a part from a probable increase of computing requirements which isn’t detailed).

Finally, just saying that these are some examples of lots of improvements published which were selected to get an idea about what are the weak points in RPR fairness algorithms and different aproaches for solving them.

REFERENCES


