Weighted Fairness and Correct Sizing of the Secondary Transit Queue in Resilient Packet Rings

Mete Yilmaz and Nirwan Ansari

***Abstract*—The IEEE 802.17 is a standardized ring topology network architecture, called the Resilient Packet Ring (RPR), to be used mainly in metropolitan and wide area networks. After a brief overview of the IEEE 802.17 RPR protocol, this article investigates the weighted fairness aspects as well as the requirements for sizing the secondary transit queue of IEEE 802.17 RPR stations (in the aggressive mode of operation). The analysis and suggested improvements presented in this article are then supported by performance evaluation results and theoretical calculations.**

***Index Terms*—MAN; WAN; Ring networks; Spatial reuse; Fairness**.

I. INTRODUCTION

The IEEE 802.17 Resilient Packet Ring (RPR) is a ring-based network protocol standardized by IEEE [1]. An RPR network is constructed by using point-to-point bidirectional connections between stations. The protocol is implemented on two counter-rotating rings called ringlets that provide protection and resilience.

A ring network is simpler to operate and administer than a complex mesh or an irregular network while virtually providing connection from each station to every other station. Synchronous optical network (SONET) rings are currently deployed by service providers in metropolitan area networks (MANs) or wide area networks (WANs), many of which consist of a dual-ring configuration where one of the rings is used as a backup ring and remains unused during normal operation to be utilized only in case of failure of the primary ring. The static bandwidth allocation and network monitoring requirements increase the total cost of a SONET network. While plain gigabit Ethernet does not require static allocation and provides cost advantages, it cannot provide desired features such as fairness and autorestoration.

The fairness aspects of RPR have been investigated in depth in light of interesting scenarios in [2–4]. Improvements for the current fairness algorithm of the IEEE 802.17 have been proposed in [2,3,5–7]. The weight aspect of the fairness algorithm has been investigated in [8]. While the requirements for the sizing of the secondary transit queue (STQ) in RPR stations have been studied in terms of overflows, the underflows were not investigated previously. In this article, an update to the weighted fairness will be provided with proof of convergence along with the underflow investigation of the STQ in RPR stations.

The rest of this article is organized as follows. In Section II an overview of RPR will be provided, with emphasis on weight allocation and fairness. Section III will present an example network topology. Weights are assigned to the stations to achieve the desired fairness, and the current fairness algorithm behavior will be illustrated through simulations. In Section IV, the observed behavior will be discussed and an improvement will be suggested with a proof of convergence. Section V will present how weights can be utilized to improve throughput when oscillations are present. Section VI will explain how the STQ should be adjusted to prevent underflow and avoid underutilization of the network. Finally, in Section VII, the conclusion will be drawn.

In order to demonstrate different operational modes, some performance figures of merit are included and discussed. The scenarios have been executed on the RPR simulator model developed at Simula Research Laboratory [9]. The simulation model is implemented in J-Sim [10] using Java. The suggested modification has been incorporated into the simulator model and its behavior is verified through simulations.

II. RPR OVERVIEW

The operation of the RPR protocol is discussed in detail in [11]. A brief overview of dual-queue operation will be provided in this article.

Figure 1 shows an example scenario. Stations 4 and 3 are transmitting to Station 2 on the inner ringlet, while Stations 2 and 1 are transmitting to Station 7.

Each time a frame is received at Station 2, the frame will be checked to see if that frame is destined to the station itself or not. The frame will be removed from the ring if it is destined to the station (in this case, any frame from Station 3 or 4). In parallel, Station 7 will be able to receive the frames from Stations 1 and 2 without being impacted by the traffic generated at Stations 3 and 4. This facilitates the

Manuscript received December 3, 2009; revised May 17, 2010; accepted September 9, 2010; published October 22, 2010 (Doc. ID 120454).

Mete Yilmaz (e-mail: myilmaz@cisco.com) is with Cisco, San Jose, California 95134, USA.

Nirwan Ansari (e-mail: nirwan.ansari@njit.edu) is with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, New Jersey 07102, USA.

Digital Object Identifier 10.1364/JOCN.2.000944

1943-0620/10/110944-8/$15.00 © 2010 Optical Society of America

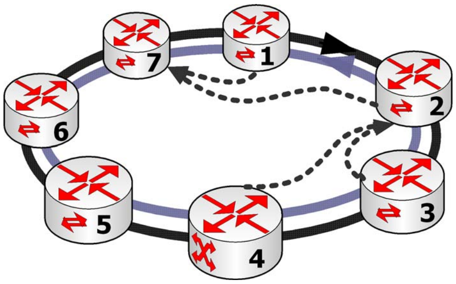

Fig. 1. (Color online) Destination stripping and spatial reuse illustrated on the inner ring (ringlet 1).

spatial reuse property of the RPR network except for multicast frames that are removed by the sender.

The unallocated bandwidth is distributed among the stations based on the RPR fairness algorithm. Assuming that all the stations have equal weights, Stations 3 and 4 will get an equal amount of the link bandwidth between Stations 2 and 3. Stations 1 and 2 will get an equal amount of the link bandwidth between Stations 1 and 7.

The RPR fairness algorithm utilizes a predefined scheduler operation and provides a fair sharing of the link bandwidth according to the weights of the stations when there is more traffic than can be transmitted through that link. Figure 2 shows an RPR media access control (MAC) of a dual-queue station. The primary transit queue (PTQ) is dedicated to high-priority ring traffic while the STQ is used for the remaining traffic. The client traffic is separated into three priorities, namely, ClassA, ClassB and ClassC.

ClassA service is utilized to satisfy guaranteed bandwidth and the low latency and bounded jitter requirements of high-priority traffic. ClassA service is given the highest priority in scheduling and utilizes the PTQ. ClassB traffic service has two parts. One part is the allocated portion called the ClassB committed information rate (CIR), which provides an allocated and guaranteed bandwidth. The second part is the ClassB excess information rate (EIR), which provides a best-effort service and is classified as fairness-eligible traffic and regulated by the fairness algorithm. The last traffic class is ClassC, which is a best-effort service with the lowest priority and is also classified as fairness eligible.

As shown in Fig. 2, the STQ and fairness-eligible traffic share the same priority from the scheduler point of view. The decision of which queue to select is based on the fairness algorithm. This decision is called "addRateOk" in the IEEE 802.17 standard. If the "addRateOk" parameter is evaluated as true, then the fairness-eligible station traffic will be selected; otherwise the STQ will be selected.

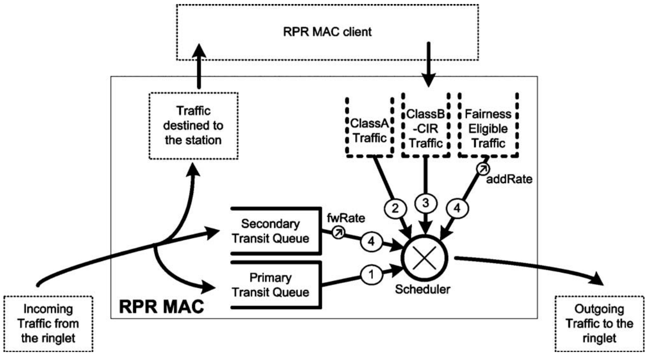

Fig. 2. RPR MAC of a dual-queue station. The numbers in circles provide the crude indication of scheduler priority.

RPR fairness is based on ingress aggregation, which is referred to as "Ring Ingress Aggregated with Spatial Reuse" (RIAS) fairness in [2] and follows the same methodology used in [12] for max–min flow control. The RIAS fairness definition does not include the station weights in the generalized formula, but this is included in the IEEE 802.17 standard. In order to provide a better representation of the fairness as in the IEEE 802.17 standard, a fairness definition with the inclusion of the administratively assigned station weights will be shown in this article as follows.

Denote N as the total number of stations on a ringlet. Let the capacity of each link on the ringlet be C. Each and every Station s on the ringlet is given a weight w_s for providing the weighted fairness. On this ringlet, a flow vector is defined by $\mathbf{F}=\{f_{st}\}$, in which each flow from Station s to Station t is denoted by f_{st}, which is also referred to as the path of the flow. A fair rate vector is defined by $\mathbf{R}=\{r_{st}\}$, in which the fair rate of flow f_{st} is denoted by r_{st}. By using the above definitions, the total allocated rate on link n of the ringlet is defined as

$$T_n = \sum_{\forall s,t:\text{link } n \in f_{st}} r_{st}. \tag{1}$$

On this ringlet, the vector $\mathbf{R}$ is said to be feasible if the following conditions are met:

$$r_{st} > 0 \quad \forall\, s,t{:}f_{st} \in \mathbf{F}, \tag{2}$$

$$T_n \leq C \quad \forall\, n \in \mathbf{N}{:}0 < n \leq N. \tag{3}$$

The sum of all flows originating from Station s and passing through link n is defined as

$$A_n(s) = \sum_{\forall t \in N:\text{link } n \in f_{st}} r_{st}. \tag{4}$$

For a feasible vector $\mathbf{R}$, the link n is a bottleneck link, $\mathbf{B_n}(s,t)$, with respect to $\mathbf{R}$ for f_{st} crossing link n if the following conditions are met with respect to all flows $f_{s't'}$ crossing link n:

$$T_n = C, \tag{5}$$

$$r_{s't'} \leq r_{st} \quad \forall\, s',t'{:}s' = s \text{ and } t' \neq t \text{ and link } n \in f_{s't'}, \tag{6}$$

$$A_n(s') \leq A_n(s) \quad \forall\, s',t'{:}s' \neq s \text{ and link } n \in f_{s't'}. \tag{7}$$

Note that if there are no other flows originating from any station other than Station s going through link n, $A_n(s')$ will be zero and Eq. (7) will be satisfied by default.

The vector $\mathbf{R}$ is said to be "weighted" RIAS fair if it is feasible as defined in Eqs. (2) and (3) and, if for each f_{st}, r_{st} cannot be increased while maintaining feasibility without decreasing the fair rate $r_{s't'}$ of some flow $f_{s't'}$ as defined in the following:

$$r_{s't'} \leq r_{st} \quad \forall\, s',t'{:}s' = s \text{ and } f_{s't'} \in \mathbf{F}, \tag{8}$$

$$\frac{A_n(s')}{w_{s'}} \leqslant \frac{A_n(s)}{w_s} \quad \forall s',t',n,:s' \neq s \text{ and } f_{s't'} \in \mathbf{F} \text{ and link } n \in f_{s't'} \text{ and link } n \in f_{st}. \tag{9}$$

Equation (8) ensures the fairness among the flows originating from the same station, while Eq. (9) ensures the fairness among ingress aggregated flows. The weights are used to normalize the comparison and thus achieve the weighted fairness.

For the scenario given in Fig. 1, if Station 4 has twice the weight of Station 3, it will get two times more bandwidth than Station 3. In this case, if Station 3 increases its share, Eq. (9) will not be satisfied.

III. WEIGHTED FAIRNESS IN RPR

In this section, an example of a weighted fairness scenario is provided to demonstrate how the weights on an RPR ring are utilized.

A service provider offers Internet and video service over its RPR network using an OC12 ring as shown in Fig. 3. The provider wants to make sure that there is always enough bandwidth to accommodate the video requests of the subscribers. The video server is connected to Station 5 and the Internet connection is through Station 4 on the ring. Assume that the service provider is utilizing MPEG4 compression for a high-definition video service where each connection is taking approximately 8 Mbps of bandwidth [13].

Also assume that in this scenario, a total of 50 different channels are being requested by the customers of the video service. This requires a total of 400 Mbps of traffic to be originated from Station 5. These video service customers are connected to Stations 6, 7, and 8 on the ring. At the same time, some 200 other customers with 1.5 Mbps Internet connection services at Stations 6, 7, and 8 are downloading files through the Internet, generating a total traffic of 300 Mbps. For simplicity, other stations will not be included in the discussion and only the outer ringlet will be used in this example. In the case of RIAS fairness, which does not account for weights, stations on the ring will share the ring bandwidth equally. This means that Station 4 and Station 5 will add an equal amount of traffic to the ring when there is congestion. This will be the case when there is a total of 400 Mbps video and 300 Mbps of Internet traffic being requested on an OC12 (~600 Mbps net data throughput) ring. In this case, Station 4 will become the congestion tail, and Station 5 will become the congestion head. Both Stations 4 and 5 will add approximately 300 Mbps of traffic each to the ring. Therefore, the service provider will not be able to accommodate the requests for 50 different channels. In this scenario, only 37 different channels can be distributed.

By definition, the issue can be resolved by assigning weights to the stations on the ring. When there is a contention for resources, the weights will control the RPR network operation. The service provider can estimate the maximum bandwidth that will be expected from Station 5. For the scenario being discussed, this is 400 Mbps. Under normal conditions, Station 4 will be the next biggest contender for the ring bandwidth. In the worst case, Station 4 should get the rest of the bandwidth, which is approximately 600−400 =200 Mbps of bandwidth. Since the ratio between these estimates is 2, a weight of 2 can be assigned to Station 5, while the weight of Station 4 will remain 1. This setting will ensure that customers will be able to enjoy watching 50 different programs simultaneously with the other 200 customers sharing the remaining 200 Mbps of bandwidth on the outer ringlet.

The scenario is simulated using the modified Simula RPR simulator to allow per-station weight adjustment. An OC12 ring that is composed of nine stations is created with 20 km of distance between every two adjacent stations. Each station is configured as a dual-queue station with the aggressive fairness mode enabled. The size of the STQ at each station is 512 KB and the “lp_coef” [1] parameter of the RPR MAC is set to 16. Bandwidth is *not* allocated for ClassA and ClassB CIR around the ring and only fairness-eligible traffic is generated, i.e., ClassB EIR and ClassC.

Figure 4 shows the total traffic sourced by Stations 4 and 5 to the outer ringlet starting at time 0.1 s. The available bandwidth is being shared by Stations 4 and 5 equally, which is around 300 Mbps, and the total amount of traffic sourced by all active stations (only 4 and 5 in this scenario)

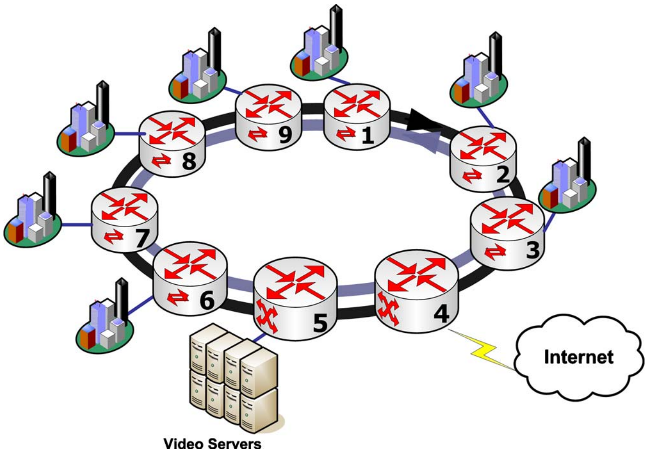

Fig. 3. (Color online) Weighted fairness scenario.

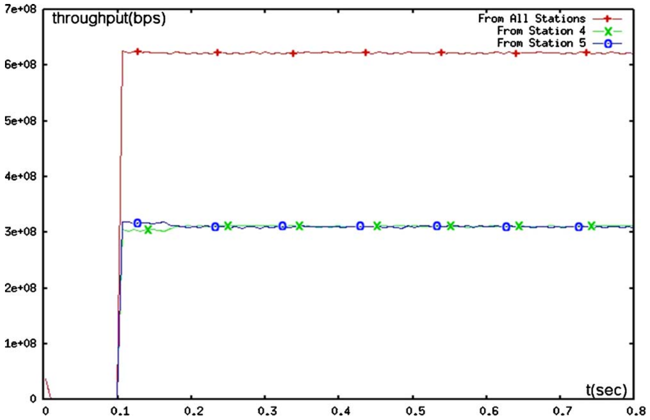

Fig. 4. (Color online) Throughput versus time graph for the scenario where Stations 4 and 5 have equal station weights.

is approximately 600 Mbps. This behavior is expected as Stations 4 and 5 have equal station weights.

The scenario is then updated where Stations 4 and 5 are assigned to weights of 1 and 2, respectively. However, the stations obtain almost the same amount of ring bandwidth with the observation that there is some additional oscillation as shown in Fig. 5.

Before we explain why that is so in the next section, let us try swapping the locations of the video server and the Internet connection so that Station 4 becomes the video server and Station 5 provides the Internet connection. The results are shown in Fig. 6.

Interestingly, this scenario behaves as expected and the new video server (Station 4) is able to acquire two times more bandwidth out of the ring than what Station 5 gets.

IV. Analysis of Weighted Fairness

In this section, we will first investigate the unexpected behavior of the fairness algorithm upon which an improvement will be suggested.

Since only fairness-eligible traffic is being generated by the clients, the scheduling decision is among packets from the STQ of the station and the station traffic. As mentioned earlier, this decision is controlled by the "addRateOk" of RPR:

$$\begin{aligned} addRateOk = {} & (addRate < allowedRate)\&\&(nrXmitRate \\ & < unreservedRate)\&\&(STQ.empty() \,\|\, (fwRate \\ & > addRate)\&\&\ (STQ.depth() < stqHighTh))). \end{aligned} \tag{10}$$

The parameters "fwRate" and "addRate" are the measured rates of fairness-eligible traffic from the STQ and the station, respectively. The "allowedRate" is the calculated fair rate at which the station is allowed to add fairness-eligible traffic. The "nrXmitRate" is the measured rate of traffic other than reserved high-priority traffic on the ringlet. The "unreservedRate" is the difference between the link rate and the total reserved bandwidth (for high-priority traffic) on the ringlet based on administratively assigned parameters. The scheduler also monitors the STQ state ("STQ.empty" and "STQ.depth") and compares the occupancy of the STQ with a administratively defined threshold called "stqHighTh" for selecting which packet to transmit.

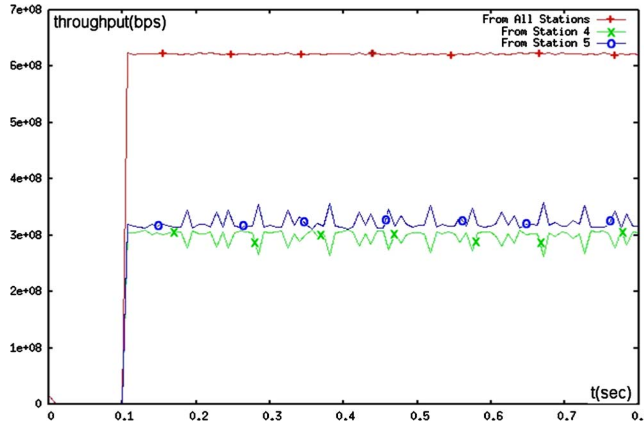

Fig. 5. (Color online) Throughput versus time graph of the scenario where Station 5 weight is set to 2.

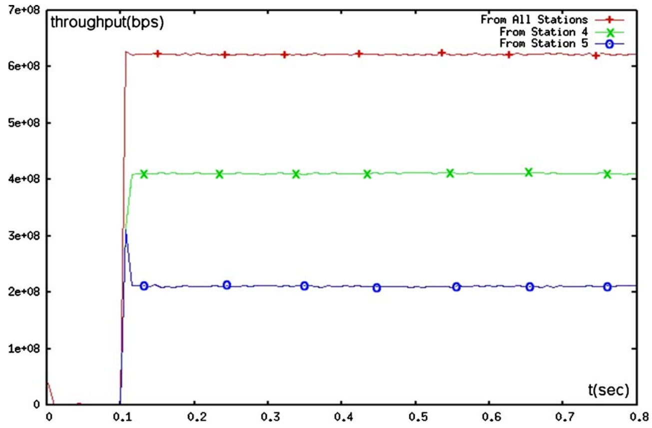

Fig. 6. (Color online) Throughput versus time graph of the scenario in which the stations of the video server and the Internet connection are swapped.

The first two parameters are not relevant in our example since Station 5 is the head of the congestion and there is no reserved traffic on the ring. So these expressions will always be evaluated as true in this scenario. The third expression checks for the availability of a packet in the STQ. If there is a packet, then it ensures the fair distribution of bandwidth unless the STQ occupancy has reached the high threshold level. The fair distribution in this case is equal bandwidth for both the transit and station traffic, and this equal distribution is the culprit. When the station is assigned to a higher weight, it is supposed to get a weighted share out of the ring.

The current calculation in the IEEE 802.17 standard shown in Eq. (10) is flawed because it will not allow the current node to transmit enough bytes when the "fwRate" and "addRate" parameters are compared even if the node is given a higher weight. This will cause the node to slow down as in Fig. 5 when the node with the higher weight is the congested node. This behavior is not observed in Fig. 6 when the node with the higher weight is an upstream node. The reason is that the "allowedRate" (estimation of the fair rate) in IEEE 802.17 already includes the node weights and in this case the "fwRate" and "addRate" comparison will not be evaluated to be true if the node is not congested.

Equation (11) shows the improved equation to resolve the unexpected behavior that includes the administratively assigned "**localWeight**" factor:

$$\begin{aligned} addRateOk = {} & (addRate < allowedRate)\&\&(nrXmitRate \\ & < unreservedRate)\&\&(STQ.empty \\ & \|\, (fwRate * \boldsymbol{localWeight} \\ & > addRate)\&\&\ (STQ.depth < stqHighTh))). \end{aligned} \tag{11}$$

This modification allows the station to add "**localWeight**" number of bytes to the ringlet for each byte forwarded from the STQ. After updating the "addRateOk" decision in the model as in Eq. (11), the simulation is run again. The results are shown in Fig. 7.

These results are in line with the weighted RIAS definition and the bandwidth is shared according to the weights of the stations as desired. Station 5 is getting two times more bandwidth than Station 4 on the ringlet. Specifically, Station 5 is adding 400 Mbps and Station 4 is adding 200 Mbps of traffic to the outer ringlet.

The addition of a new factor in the calculation of the "addRateOk" parameter as in Eq. (11) requires an additional multiplication operation in the scheduler. To simplify the calculation, one can require the weight to be a power of 2 so that simple shift operations can replace the more complicated multiplication logic. Another approach is to add a new parameter called "weightedFwRate," and per each byte transmitted from the STQ, increment the "weightedFwRate" with the weight of the station.

There are also other practices a network operator can follow to avoid encountering the scenario discussed above. One such workaround, as shown in Fig. 6, is to make sure that a station with a larger weight does not become the head of the congestion domain. Also, another desirable approach is to distribute the high throughput servers evenly around the ring when possible, since this will allow efficient use of both ringlets and will decrease the contention on the ring.

One last thing to note is the oscillations observed in Fig. 5. This is mainly due to the feedback control mechanism of RPR in the aggressive mode of operation. Once the STQ reaches a certain threshold (in the aggressive mode of operation), a station is considered to be congested. At this point, the station starts transmitting a message with its own normalized addRate to the upstream stations. When an upstream station receives this message, it will adjust its transmit rate to the fair rate (addRate) of the station that transmitted the congested message. In this case, the video server transmits half of its own addRate to the upstream stations due to its assigned weight. The upstream station that receives the notification slows down to this rate. However, there are already packets waiting in the STQ of the congested station, and the scheduler is transmitting those packets. Once the station starts letting go of some of those packets in the STQ, the station is no longer congested and stops transmitting its own normalized rate, which in turn lets Station 4 increase its share on the ring. This mechanism creates an oscillatory behavior in this specific case that can be smoothed out by increasing the available STQ size, and this will result in equal sharing of the ring bandwidth (which is not desired in this scenario). On the other hand, if the STQ size is decreased, there will be more oscillations while the ratio of traffic added by each station will approach the ratio of station weights.

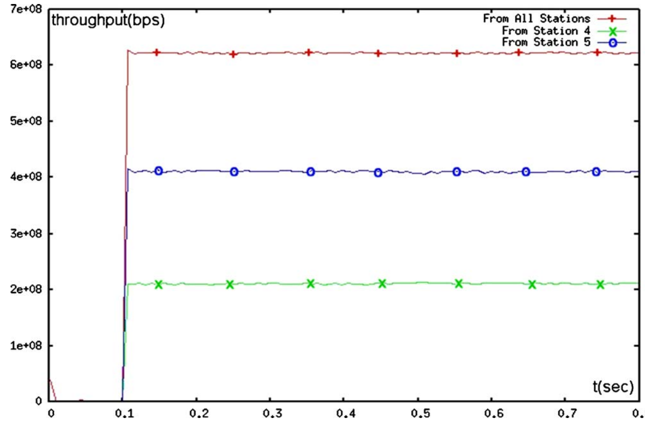

Fig. 7. (Color online) Throughput versus time graph of the scenario where Station 5 weight is set to 2 with the updated *addRateOk* calculation.

For a stable network configuration as defined in [4], the convergence can be shown analytically for the updated addRateOk calculation of Eq. (11) as follows.

Assume *unreservedRate* is set to the capacity of the line and the calculation takes place at the node that is the congestion head. By definition, the STQ of the congested node will not be empty. Then, Eq. (11) can be reduced to

$$addRateOk = (fwRate * \boldsymbol{localWeight} > addRate) \&\& \quad (STD.depth < stqHighTh))). \tag{12}$$

An RPR station will schedule transit or transmit packets based on Eq. (12). If the transit packets are not scheduled at the rate they are coming in, then the STQ will start to build up. When the STQ hits the *stqLowTh* threshold, an RPR station will start sending out congested messages as defined in the standard.

As long as the occupancy of the STQ is less than *stqHighTh*, the scheduling of the packets will be directly controlled by Eq. (12) and will satisfy

$$addRate = \boldsymbol{localWeight} * fwRate. \tag{13}$$

Let C be the capacity of the link. Then, *addRate* and *fwRate* also need to satisfy

$$C = addRate + fwRate. \tag{14}$$

Based on Eqs. (13) and (14), one can derive the *addRate* as

$$addRate = C * \boldsymbol{localWeight}/(\boldsymbol{localWeight} + 1). \tag{15}$$

If a node is congested, it will transmit congested messages that will regulate the upstream traffic. The congested message from the congested node will set the *allowedRate* of the upstream stations based on the RPR standard:

$$allowedRate = addRate/\boldsymbol{localWeight}. \tag{16}$$

Define $\tilde{x}$ as the effective number of upstream nodes in the congested domain; then, the fwdRate can be derived from Eq. (16) as

$$fwRate = \tilde{x} * addRate/\boldsymbol{localWeight}. \tag{17}$$

Only when $\tilde{x}$ is equal to 1 will both Eqs. (13) and (17) be satisfied. Therefore, the initial assumption that the STQ is less than *stqHighTh* but higher than *stqLowTh* will only be satisfied when there is a single effective upstream flow for a stable network configuration as defined in [4].

If the effective number of active upstream stations is greater than 1, the STQ of the congested station will be filled up beyond *stqHighTh*. For this case also, *addRate* and *fwdRate* are bounded by the capacity of the link and need to satisfy Eq. (14). In addition, by definition, the *allowedRate* will be calculated according to Eq. (16). Then, the following relation can be derived:

$$C = \text{addRate} + \tilde{x} * \text{addRate}/\boldsymbol{localWeight}. \tag{18}$$

Therefore, for the stable network configuration as defined in [4], the addRate will converge to

$$\text{addRate} = \text{C} * \boldsymbol{localWeight}/(\tilde{x} + \boldsymbol{localWeight}). \tag{19}$$

Note that if the upstream traffic increases, the addRate will decrease, which in turn will limit the upstream traffic. Once the upstream traffic is limited, the additional bandwidth will be utilized by the station itself, which will increase the addRate to converge to the value given in Eq. (19).

V. WEIGHTED FAIRNESS UNDER INSTABILITY

It was shown in [14] that the RPR algorithm can suffer from oscillations under some special scenarios where the congested node has very minimal traffic. It is quite clear that as a result of these oscillations, the network utilization will go down. In this section, we show how station weights can solve the underutilization of the network.

The scenario that was introduced in Section IV as shown in Fig. 3 is modified to create the oscillatory behavior. We will use the corrected weighted fairness algorithm in the simulation. Similar behavior was also observed in the original fairness algorithm [14].

In this scenario, Station 5 has 400 Mbps, Station 4 has 300 Mbps, and Station 6 has 20 Mbps of traffic, all of which are destined to Station 7. The service provider still wants to make sure that Station 5 will get 400 Mbps when needed to support 50 different channels.

The difference from the previous scenario of Section III is that the only destination is Station 7 and a new traffic source, Station 6, is added. Note that all the stations have a weight of 1. The oscillations are observed as a result of having Station 6 adding a very small amount of traffic while being the congested station at the same time. As Station 6 gets congested, it advertises its current add rate, which slows down Stations 4 and 5 more than it should periodically and hence results in the oscillatory behavior as shown in Fig. 8.

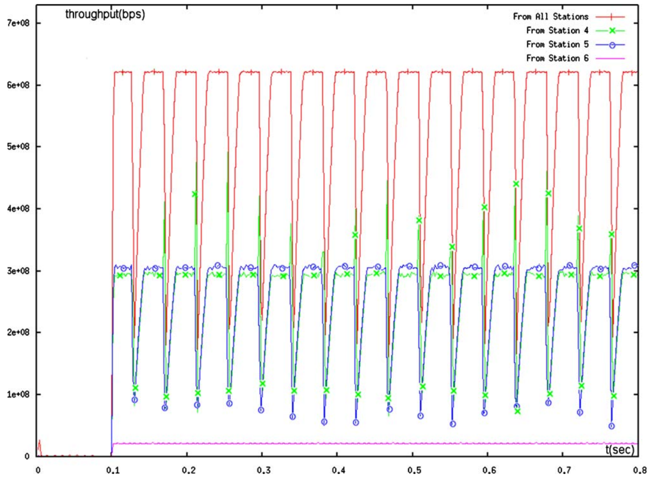

Fig. 8. (Color online) Throughput versus time graph of the scenario where the weight of stations are set to 1.

The next scenario has the same traffic pattern but with different weights for the stations. The weight of Station 4 is 10, the weight of Station 5 is 20, and the weight of Station 6 is 1. In this case, the stations share the bandwidth as desired, and the oscillations are gone as shown in Fig. 9.

Note that the period of the congestion interval depends on the amount of traffic added to the ring by Station 6 when other parameters such as ring size and buffer thresholds remain the same [4]. When that traffic decreases, the congestion interval will increase. Under this condition, even though some oscillations might still be observed, the impact on the total network utilization will be minimal.

If traffic added by Station 6 increases, the fairness algorithm will function better because the congested node (Station 6) will have more traffic to advertise.

The adjustment of weights should not be confused with static bandwidth assignments. The reason is that the weights will only be active if there is traffic from the node with higher weight and there is some congestion down the path. Otherwise, the stations that are assigned with smaller weights will still utilize the unused bandwidth. In addition, the adjustment of weights is well suited to the current network architectures where the upload limit for the nodes in the network is generally much less than the download limit.

Also note that the behavior of the fairness algorithm is tightly coupled with the STQ thresholds, round-trip time of the network, and the amount of smoothing of the instantaneous measurements [4]. By adjusting these parameters carefully, the network behavior can be optimized further, some of which we will explore in the next section.

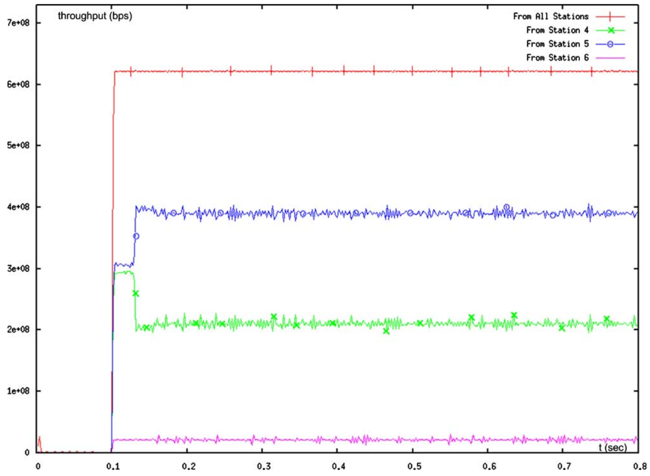

Fig. 9. (Color online) Throughput versus time graph of the scenario where the weight of Station 5 is set to 20 while the weight of Station 4 is set to 10.

VI. SIZING OF SECONDARY TRANSIT QUEUE

The example provided in the previous section shows how imbalanced traffic can cause oscillations in an RPR network. The control mechanism of RPR has been very well studied, and different algorithms have been suggested. However, these algorithms require deviation from the current standard.

The previous section provides a means to improve the deficiency by use of weighted fairness. This section will provide another option for the RPR MAC designer in terms of queue sizing. Appendix G of the RPR standard [1] provides implementation guidelines. However, the section does not provide all the requirements for STQ sizing. Specifically, the guidelines provided in the standard are not satisfactory for the underflow case.

While the transit queue sizing has been investigated for the overflow case [15], it has not been investigated for the underflow case. It is well known that queue underflow will result in low utilization of the available bandwidth in a network. In order to prevent the underflow, the STQ needs to be sized accordingly.

In the standard, the maximum fairness round-trip time (*maxFRTT*) is defined as the round-trip time for propagation of a fairness value around an entire ring and for the first effected traffic to return to the congested node.

Denote N as the total number of stations on a ringlet. Let *advertisingInterval* be the interval on which each station advertises its own *addRate* and *ringKM* be the circumference of the ringlet, then *maxFRTT* can be calculated as

$$maxFRTT = N * advertisingInterval + 2 * (5\mu s * ringKm). \tag{20}$$

Note that the constant 5 μs is used as the propagation delay of a signal per km of the medium.

However, *maxFRTT* does not provide the total delay for the sizing of the STQ to prevent underflow. There is another major component that comes from the fairness algorithm of RPR. When a congested node is no longer congested, it will start advertising *FULL_RATE* to indicate the absence of congestion. The source station will then start incrementing its *allowedRate* up to a maximum rate defined as *LINK_RATE*. The *allowedRate* is incremented according to

$$allowedRate = allowedRate + (LINK\_RATE - allowedRate)/rampUpCoef. \tag{21}$$

Define *agingInterval* as the interval a source station increments its own *allowedRate*, and define *rampUpCoef* as an arbitrary constant. Define the additional delay before a station reaches its maximum rate of *LINK_RATE* as *rampUpDelay*. The *rampUpDelay* can then be calculated as

$$rampUpDelay = \frac{agingInterval * LINK\_RATE}{rampUpCoef}. \tag{22}$$

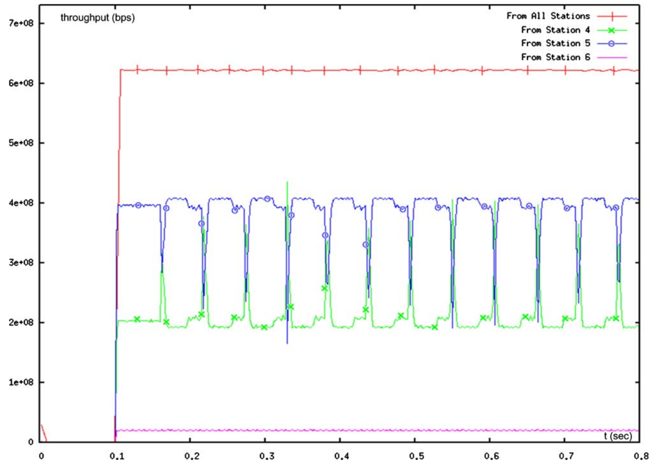

Fig. 10. (Color online) Throughput versus time graph of the scenario with buffer threshold at Station 6 adjusted for underflow.

The impact of oscillations on link utilization can be resolved by correct sizing of the STQ for the underflow case. To prevent underflow of the STQ, the queue needs to be sized so that it cannot be emptied before the feedback control loop takes effect. Therefore, after the congested station declares that it is not congested anymore (which is defined as the *queueSize* being less than *lowThreshold*), it should have enough buffer buildup in order to transmit for the sum of *maxFRTT* and *rampUpDelay*:

$$lowThreshold > (maxFRTT + rampUpDelay) * lineRate. \tag{23}$$

Rerunning the scenario for instability from Section V by using the guideline according to Eq. (23) for correct sizing of the STQ generates the results shown in Fig. 10.

As shown in Fig. 10, the oscillatory behavior for the total traffic received from all stations is not there anymore, and the link utilization is at 100%. There are periodic interruptions to the traffic sourced from Stations 4 and 5. The oscillations occur as a result of the already buffered traffic at the upstream stations while these buffers are being depleted during traffic adjustment periods.

As long as a system has enough buffering and the traffic can tolerate jitter, one can utilize the additional buffer for the fairness-eligible packets and prevent oscillations at the destination to provide maximum utilization of the network. However, if the buffers are not available at the MAC client, then one can always utilize the mechanism described in the previous section via weighted fairness parameters and completely eliminate oscillations.

VII. CONCLUSION

This article has discussed and explained the use of weighted fairness in an RPR network. It has extended the definition of ring ingress aggregated fairness by incorporating weights into the formulation. Performance evaluations by using the latest version of the IEEE 802.17 RPR standard have demonstrated how the bandwidth is shared by using different weights. In particular, we have identified a flaw and suggested improvements to circumvent that flaw as substantiated by the simulation results and proof of convergence. In addition, we have shown that by adjusting various parameters already available in the fairness algorithm of IEEE 802.17, one can eradicate the oscillatory behavior under certain scenarios. Furthermore, we have shown that by providing the right queue size for the network, the utilization can be improved because the feedback loop works more efficiently.

REFERENCES

[1] IEEE Standard 802.17-2004, "Resilient Packet Ring."

[2] V. Gambiroza, P. Yuan, L. Balzano, Y. Liu, S. Sheafor, and E. Knightly, "Design, analysis, and implementation of DVSR: a fair high-performance protocol for packet rings," *IEEE/ACM Trans. Networking*, vol. 12, no. 1, pp. 85–102, 2004.

[3] F. Davik, A. Kvalbein, and S. Gjessing, "Performance evaluation and improvement of non-stable resilent packet ring behavior," in *Proc. Part II of the 4th Int. Conf. on Networking (ICN'05)*, LNCS 3421, Reunion Island, 2005, pp. 551–563.

[4] F. Davik, A. Kvalbein, and S. Gjessing, "An analytical bound for convergence of the resilient packet ring aggressive mode fairness algorithm," in *Proc. 40th Annu. IEEE Int. Conf. on Communications (ICC'05)*, Seoul, South Korea, 2005, pp. 281–287.

[5] F. Alharbi and N. Ansari, "Distributed bandwidth allocation for resilient packet ring networks," *Comput. Netw.*, vol. 49, no. 2, pp. 161–171, Oct. 2005.

[6] F. Alharbi and N. Ansari, "SSA: simple scheduling algorithm for resilient packet ring networks," *IEE Proc.-Commun.*, vol. 153, no. 2, pp. 183–188, Apr. 2006.

[7] M. Yilmaz, N. Ansari, J. H. Kao, and P. Yilmaz, "Active queue management for MAC client implementation of resilient packet rings," in *Proc. of the Int. Conf. on Communication (ICC'09)*, Dresden, Germany, 2009, pp. 1–5.

[8] M. Yilmaz and N. Ansari, "Weighted fairness in resilient packet rings," in *Proc. of the 2007 IEEE Int. Conf. on Communications (ICC'07)*, Glasgow, 2007, pp. 2192–2197.

[9] "Simula RPR Simulator" [Online]. Available: http://software.simula.no/nd/rpr/.

[10] H. Tyan, "Design, realization and evaluation of a component-based compositional software architecture for network simulation," Ph.D. dissertation, Ohio State University, 2002.

[11] F. Davik, M. Yilmaz, S. Gjessing, and N. Uzun, "IEEE 802.17 resilient packet ring tutorial," *IEEE Commun. Mag.*, vol. 42, no. 3, pp. 112–118, Mar. 2004.

[12] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1987, pp. 524–530.

[13] M. Alvarez, E. Salami, A. Ramirez, and M. Valero, "A performance characterization of high definition digital video decoding using H.264/AVC," in *Proc. IEEE Int. Symp. on Workload Characterization*, 2005, pp. 24–33.

[14] V. Gambiroza, P. Yuan, and E. Knightly, "The IEEE 802.17 media access protocol for high-speed metropolitan-area resilient packet rings," *IEEE Network*, vol. 18, no. 3, pp. 8–15, May–June 2004.

[15] P. Setthawong and S. Tanterdtid, "Inter-ring traffic management in bridged resilient packet rings: global fairness and buffer overflow prevention," *Int. J. Comput. Sci. Network Secur.*, vol. 6, no. 11, pp. 190–201, Nov. 2006.

Mete Yilmaz received his B.S. (1995) and M.S. (1999) degrees from Bogazici University, Istanbul, Turkey, in electrical engineering and computer engineering, respectively. He is currently pursuing his Ph.D. degree at New Jersey Institute of Technology, Newark, in computer engineering. He works at Cisco, San Jose, California, as a Technical Leader with responsibilities ranging from architecting new line cards to developing packet processing FPGAs for Cisco's high-end routers. He has been working on communication networks for 14 years. His research interests include quality of service, bandwidth reservation, and fairness algorithms. He holds three patents on fairness algorithms.

Nirwan Ansari received the B.S.E.E. (summa cum laude, with a perfect GPA) from the New Jersey Institute of Technology (NJIT), Newark, in 1982; the M.S.E.E. degree from the University of Michigan, Ann Arbor, in 1983; and the Ph.D. degree from Purdue University, West Lafayette, Indiana, in 1988. He joined NJIT's Department of Electrical and Computer Engineering as an Assistant Professor in 1988 and became a tenured Associate Professor in 1993 and a Full Professor in 1997. He has also assumed various administrative positions at NJIT. He authored *Computational Intelligence for Optimization* (Springer, 1997, translated into Chinese in 2000) with E. S. H. Hou and edited *Neural Networks in Telecommunications* (Springer, 1994) with B. Yuhas. His current research focuses on various aspects of broadband networks and multimedia communications. He has also contributed over 350 technical papers, over one third of which are in widely cited refereed journals/magazines. For example, one of his seminal works was the sixth most cited article published in the *IEEE Transactions on Parallel and Distributed Systems*. He was/is serving on the Advisory Board and Editorial Board of eight journals, including as a Senior Technical Editor of *IEEE Communications Magazine* (2006–2009). He had/has been serving the IEEE in various capacities such as Chair of IEEE North Jersey COMSOC Chapter, Chair of IEEE North Jersey Section, Member of IEEE Region 1 Board of Governors, Chair of IEEE COMSOC Networking TC Cluster, Chair of IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and Chair/TPC Chair of several conferences/symposia. He has been frequently invited to deliver keynote addresses, distinguished lectures, tutorials, and talks. Some of his recent awards and recognitions include an IEEE Fellow (Communications Society), IEEE Leadership Award (2007, from Central Jersey/Princeton Section), the NJIT Excellence in Teaching in Outstanding Professional Development (2008), IEEE MGA Leadership Award (2008), the NCE Excellence in Teaching Award (2009), Thomas Alva Edison Patent Award (2010), and designation as an IEEE Communications Society Distinguished Lecturer (2006–2009, two terms).