I. INTRODUCTION

The Resilient Packet Ring (RPR) IEEE 802.17 standard [1] is under development as a new high-speed backbone technology for metropolitan area networks (MANs). RPR networks adopt a ring-based architecture that consists of two counter-rotating rings with each station connecting to two adjacent stations over a link pair. The primary merit of RPR networks is fault tolerance: all nodes remain connected with any single failure of a bi-directional link span. Moreover, rings have reduced deployment costs as compared to star or mesh topologies.

A key performance objective of RPR is to simultaneously achieve high utilization, spatial reuse, and fairness, an objective not achieved by current technologies such as SONET and Gigabit Ethernet nor by legacy ring technologies such as FDDI. The main technical challenge for RPR is the design of a bandwidth allocation algorithm that dynamically achieves these three performance properties. The difficulty therein lies in the distributed nature of the problem, requiring that upstream ring nodes inject traffic at rates according to congestion and fairness criteria downstream. There have been protocols proposed that aim to realize the above three designing objectives. These include Aladdin [7] and Gandalf [8]. These algorithms specify how upstream traffic should be throttled according to downstream measurements, namely, how a congested node will send fairness messages upstream so that upstream nodes can appropriately configure their rate limiters to throttle the rates of injected traffic to their fair rates. They also define the scheduling policy to arbitrate service among transit and station (ingress) traffic as well as among different priority classes. Unfortunately, these algorithms are inherited a certain limitation that they all suffer from severe and permanent oscillations even possibly spanning the entire range of the link capacity [3]. As a consequence, they are not efficient to fulfill the aforementioned performance objectives as oscillations hinder spatial reuse, decrease throughput, and increase delay jitter. In this paper, we overcome this drawback by designing a rate controller to stabilize the buffer occupancy of every node when allocating bandwidth.

In [5], the authors define Ring Ingress Aggregated with Spatial Reuse (RIAS) fairness; a reference model is now incorporated into the IEEE 802.17 standard’s targeted performance objective [9]. Approximately a so-called Distributed Virtual-time Scheduling in Rings (DVSR) algorithm further realizes this idealized fairness reference model in [3]. However, as revealed by [3] itself, there are still deviations between the DVSR service rates and the ideal RIAS fair rates for a realistic system. Therefore, how to fully realize the RIAS requirement in bandwidth allocation for RPR networks is still a challenging task.

In this paper, one fair bandwidth allocation algorithm, termed PID-RPR, is proposed that satisfies the performance requirements of RPR networks. To overcome the drawback inherited in the known algorithms, we propose to use the PID controller in rate allocation and study the requirements of stability in terms of queue length at the ring nodes. On the basis of PID control, we develop a dynamic bandwidth allocation...
algorithm for RPR network. We perform theoretic analyses as well as simulation studies. The results demonstrate that satisfactory performance of RPR networks can be achieved under the proposed bandwidth allocation scheme.

II. AN RPR FLOW MODEL FOR BANDWIDTH ALLOCATION

A simple RPR flow model is shown in Figure 1, where flow(i, j) denotes the data flow from the i-th source to its destination node j; and b(D, i) denotes the feedback message flow from Node D to the i-th source. For the sake of clarity, we classify the flows into two groups, namely the aggregated flows and the spare flows. The aggregated flows are those coming from the different sources but arriving at (or passing through) the same destination node. For example, in Figure 1, flow(n1, D), flow(n2, D), ..., flow(N, D) are the aggregated flows. Specially, among the aggregated flows, we call those inner flows, if they are oriented from the same source but terminated at different destination nodes. The two flows flow(4, D) and flow(4, N) in Figure 1 are such examples. Various inner flows should be treated as a single aggregated flow when allocating the bottleneck link bandwidth at the first stage. After this, for every aggregated flow who has inner flows, our next step is to further allocate the obtained bandwidth among the inner flows. Besides the aggregated flows, there may exist some kinds of flows in the ring, such as flow(1, 3) and flow(1, 2) in Figure 1. We call them spare flows. Finally, in order to achieve the maximal special reuse, we need to allocate the left bandwidth to the relevant spare flows that are competing for a common link.

Firstly, we consider the bandwidth allocation among the aggregated flows in a certain bottleneck link. From Figure 1, we can see in the ring there are N aggregated flows at the most congested link. Let us consider such a specific bottleneck link. At the destination node D, the dynamic of the queue of the link can be described by

$$\dot{q}(t) = \max(0, \sum_{i=1}^{N} R_i(t - \tau_f^i) - \nu),$$  \hspace{1cm} (1)

where $q(t)$ is the queue length at time $t$; $\tau_f^i$ is the forward delay from the i-th source to the destination Node D, specifically, we have $\tau_f^i = \sum_{j=1}^{N} \tau_f^j$. The component $R_i(t)$ is the sending rate of the i-th source at time $t$; $\nu$ is the transmitting rate of the destination node; $N$ denotes the number of aggregated flows in a certain bottleneck link.

For analyzing the effectiveness of the proposed algorithm, we assume the sources are persistent until the closed-loop system reaches steady state. By “persistent”, we mean that the source always has enough data to transmit at the allocated rate. In the next section, we will propose a control-theoretic iterative algorithm on the basis of the deterministic fluid model (1).

III. A HIGH PERFORMANCE FAIR RATE BANDWIDTH ALLOCATION ALGORITHM

In this section we discuss the operations of rate controller in RPR networks, and then propose the PID-RPR control approach.

A. The basic operations of rate controller for RPR networks

The basic operations of rate controller for RPR networks are described in detail as follows.

First, the rate controller throttles all station traffic entering the ring. Next, RPR nodes have measurement modules (byte counters) to measure demanded and/or serviced stations and transit traffic. These measurements are used in our algorithm to compute a feedback control signal to throttle upstream nodes to attain the desired rates. Once a node receives any feedback message from its downstream nodes, it uses the carried control information therein together with the local information, and then to set the bandwidth for its rate controller. In our algorithm, we use a single-queue buffering mode. In this mode, the transit path consists of a single first-in-first-out (FIFO) queue, which is called the Primary Transit Queue (PTQ). In this case, we employ a strict priority of transit traffic over station traffic. The objective is to ensure hardware simplicity and to ensure that the transit path is lossless, i.e., once a packet is injected into the ring, it will not be dropped at any downstream node.

For each link of RPR network illustrated in Figure 1, we implement a PID rate controller in a distributive manner. The working mechanism for each PID rate controller is displayed in Figure 2, where the broken line denotes the virtual connections of the forward data flows and the backward control flows.

B. The PID controller

For a given bottleneck link shown in Figure 1, we propose the following PID controller

$$f(t) = a - \frac{1}{Q_w} \left[ K_pe(t) + K_i \int_0^t e(t) dt + K_D e(t) \right],$$  \hspace{1cm} (2)

where $a$ is the available bandwidth in the bottleneck link; the function $f(t)$ is the control rate in the bottleneck link; the component $e(t)$ is the error between the target of queue and the instantaneous queue, i.e., $e(t) = q(t) - q_T$, $q_T$ is the target of queue length, $w_i$ is the weight of the i-th flow, $\rho_i = \frac{w_i}{Q_w}$, so we have

![Flow model](image-url)
\[ \sum_{i=1}^{N} \rho_i = 1. \] The coefficients \( K_P, K_I \) and \( K_D \) are control gains, which are to be determined on the basis of system stability.

![Fig. 2. The PID closed-loop control model for RPR networks](image)

If we let \( a = \mu - \sum_{i=1}^{N} MR(i) - B \), where \( B \) is the bandwidth that has already been allocated in terms of the capacity of bottleneck links elsewhere, \( \mu \) is the capacity of the bottleneck link, the equation (2) can be rewritten as

\[
\begin{align*}
    f(t) &= \mu - \sum_{i=1}^{N} MR(i) - B - \frac{1}{Q_w} [K_P e(t)] \\
    &\quad + K_I \int_{0}^{t} e(t) dt + K_D e(t).
\end{align*}
\] (3)

The congested node then feedbacks the above obtained rate to its corresponding sources, from which the aggregated flows are issued. Taking into account the backward delay from the destination node to Source \( i \), i.e., \( \tau_i^b \), the weighted fair rate of Source \( i \) is computed according to

\[
r_i(t) = w_i f(t - \tau_i^b). \tag{4}
\]

The total sending rate of Source \( i \) is thus determined by

\[
R_i(t) = MR(i) + r_i(t) = MR(i) + w_i f(t - \tau_i^b). \tag{5}
\]

From (3), (4) and (5), we have the following closed-loop system description

\[
R_i(t) = MR(i) + w_i (\mu - \sum_{i=1}^{N} MR(i) - B - \frac{1}{Q_w} [K_P e(t - \tau_i^b)]) \\
+ K_I \int_{0}^{t} e(t - \tau_i^b) dt + K_D e(t - \tau_i^b)]. \tag{6}
\]

Suppose that the above system has an equilibrium point at which the derivatives of the variables are zero, i.e., \( \lim_{t \to \infty} \dot{q}(t) = 0, \lim_{t \to \infty} q(t) = q_r \) and \( \lim_{t \to \infty} f(t) = f_r \). Then the equation (1) leads to

\[
\sum_{i=1}^{N} R_{is} = v. \tag{7}
\]

Furthermore, the equation (5) at the equilibrium point can be written as

\[
R_{is} = MR(i) + r_s = MR(i) + w_i f_r. \tag{8}
\]

Using (7) and (8), we obtain

\[
\sum_{i=1}^{N} [MR(i) + r_s] = \sum_{i=1}^{N} [MR(i) + w_i f_r] = v, \tag{9}
\]

which establishes that the PID controller achieves the following weighted max-min fairness property among the aggregated flows

\[
r_{is} = w_i f_r = \frac{w_i}{Q_w} [v - \sum_{i=1}^{N} MR(i)]. \tag{10}
\]

C. The system stability analysis

In this section, we propose the stability conditions for the proposed controller. The stability conditions provide a guideline in setting the control gain parameters.

Using equation (1) and (6), we have

\[
\ddot{q}(t) = \ddot{e}(t) = \sum_{i=1}^{N} \ddot{R}_i(t - \tau_i^f), \tag{11}
\]

\[
\ddot{R}_i(t) = -\rho_i [K_P \dot{e}(t - \tau_i^b) + K_I e(t - \tau_i^b) + K_D \dot{e}(t - \tau_i^b)]. \tag{12}
\]

Since \( \tau_i \) is the round trip delay between the \( i^{th} \) source and its destination node, i.e., \( \tau_i = \tau_i^f + \tau_i^b \), submitting (2) into (11), we obtain

\[
\ddot{e}(t) = -\sum_{i=1}^{N} \rho_i [K_P \dot{e}(t - \tau_i^b) + K_I e(t - \tau_i^b) + K_D \dot{e}(t - \tau_i^b)]. \tag{13}
\]

By taking the Laplace transform of equation (13), we derive

\[
s^2 \tilde{E}(s) = \sum_{i=1}^{N} \rho_i e^{-\tau_i^b} [K_P \tilde{S}(s) + K_I \tilde{E}(s) + K_D s^2 \tilde{E}(s)] \tag{14}
\]

i.e.,

\[
\{s^2 + \sum_{i=1}^{N} \rho_i e^{-\tau_i^b} [K_P s + K_I + K_D s^2] \} \tilde{E}(s) = 0 \tag{15}
\]

Where \( \tilde{E}(s) \) denotes the Laplace transform of \( e(s) \). Then we obtain the characteristic equation:

\[
\Delta(s) = s^2 + \sum_{i=1}^{N} e^{-\tau_i^b} \rho_i (K_P s + K_I + K_D s^2) = 0 \tag{16}
\]

For simplicity, we use a two-order Taylor series to approximate the exponential function, i.e., by supposing \( e^{-\tau_i^b} \approx 1 - \tau_i^b s + \frac{\tau_i^b s^2}{2} \), we can have the following polynomial equation:

\[
\Delta(s) = s^2 + \sum_{i=1}^{N} \rho_i (1 - \tau_i^b s + \frac{\tau_i^b s^2}{2}) (K_P s + K_I + K_D s^2) \nonumber\]

\[
= \left(\sum_{i=1}^{N} \rho_i \frac{\tau_i^b K_P}{2}\right) s^2 + \sum_{i=1}^{N} \rho_i \left(\frac{\tau_i^b K_P}{2} - K_P \tau_i\right) s \nonumber\]

\[
+ (1 + K_D) \sum_{i=1}^{N} \rho_i (\frac{\tau_i^b K_I}{2} - K_D \tau_i) s^2 \nonumber\]

\[
+ \sum_{i=1}^{N} \rho_i (K_P - K_I) s + K_I. \tag{17}
\]

We denote \( a_0 = K_I \), \( a_1 = \sum_{i=1}^{N} \rho_i (K_P - K_I \tau_i) \), \( a_2 = 1 + K_D + \sum_{i=1}^{N} \rho_i (\frac{\tau_i^b K_I}{2} - K_P \tau_i) \).
The equation (17) can be rewritten as
\[ \Delta(s) = a_0 s^4 + a_2 s^2 + a_4 s + a_0. \] (18)

Based on the Routh-Hurwitz stability test [8] in the control theory, we know the system is stable if and only if
\[ a_0 > 0, \quad a_3 > 0, \quad a_2 - \frac{a_4 a_0}{a_3} > 0, \quad a_1 - \frac{a_3^2 a_0}{a_2 a_3 - a_4 a_1} > 0, \quad a_0 > 0. \]

Finally, we obtain the following conditions under which the closed-loop system is stabilized:
\[ K_D > 0, \quad K_I > 0, \quad K_P > 0, \quad K_I \geq \frac{K_D}{\sum_{i=1}^{N} \rho_i \tau_i K_D}, \]
\[ K_I \geq \frac{(K_P - \sum_{i=1}^{N} \rho_i \tau_i K_D)K_D}{\sum_{i=1}^{N} \rho_i \left(\frac{\tau_i^2 K_P}{2} - K_D \tau_i\right)} \]
\[ + \frac{1 + K_D \sum_{i=1}^{N} \rho_i \left(\frac{\tau_i^2 K_P}{2} - K_D \tau_i\right)}{\sum_{i=1}^{N} \rho_i \left(\frac{\tau_i^2 K_P}{2} - K_D \tau_i\right)} > 0. \]

Once the control gains \( K_D, K_I, \) and \( K_P \) in (2) are chosen to satisfy the above stability conditions, the queue length \( q(t) \) is stabilized at the target. Therefore, our PID-RPR scheme has the capability of avoiding queue oscillations [8-9] and the network performance is enhanced.

D. The algorithm of PID-RPR

Based on the RPR flow model discussed in Section 2 and the results of system stability of PID-RPR, we are able to choose the proper control gains to ensure the system stability. Such designing can also achieve the max-min fairness among the aggregated flows, and subsequently the RIAS fairness among all flows is satisfied by maximal spatial reuse.

We now present the specific PID-RPR algorithm. This algorithm constitutes three steps. Firstly, we use PID controllers to allocate the bandwidth of the aggregated flows that come from different upstream nodes, terminate at a common destination node and share a common bottleneck link. This allocation is determined dynamically by the queue length at the destination node, it satisfies the stability condition of queue length and further in steady state it meets the requirement of weighted fairness between these flows exactly. Secondly, we allocate the bandwidth of the inner flows, which originate from the same node but terminate different destination nodes, based on weighted fairness. Thirdly, in order to maximize spatial reuse in the relevant links, we allocate the remaining bandwidth to the spare flows. Hence, the algorithm realizes the distributed function that upstream ring nodes inject traffic at rates

\[ \text{Fig. 3. Pseudocode for the PID-RPR} \]
according to congestion and weighted fairness criteria downstream. Furthermore, it is established that RIAS fairness has been exactly realized by this rate allocation scheme.

The pseudocode of PID-RPR algorithm is shown in Figure 3, where the component $k$ denotes the number of links in the ring; $F[i]$, $FA[i]$ and $FI[i]$ denote the number of flows, the number of aggregated flows and the number of inner flows in link $i$ respectively; $B_i$ denotes the allocated bandwidth in link $i$; $a$ is the spare bandwidth and $MR(i)$ represents the minimal rate in link $i$.

### IV. PERFORMANCE EVALUATION

To verify the performance of the proposed algorithm, we carry out simulations on the basis of the model shown in Figure 4.

In this model, we assume that there are 10 nodes in the resilient packet ring. There exist data flows $flow(n1, n5)$, $flow(n2, n5)$, $flow(n3, n5)$, $flow(n4, n5)$, $flow(n1, n2)$, $flow(n2, n3)$, $flow(n2, n4)$ and $flow(n4, n5)$. They all achieve the weighted max-min fairness property, which is consistent with our theoretical analyses. The rate allocations among inner flows, namely, $flow(n2, n3)$ and $flow(n2, n4)$, are shown in Figure 7. Likewise, the rate allocations among them also achieve the weighted fairness. Figure 8 and Figure 9 respectively demonstrate the spatial reuse of link L1 and link L2. Obviously, all flows in two links can reclaim unused bandwidth dynamically in a weighted fair manner. Furthermore, from Figure 10 we observe that the link utilizations of link L1, L2, L3 and L4 are driven to as high as possible due to stability and maximal spatial reuse merits of the algorithm.

$K_D = 0.02$, $K_I = 0.1$, $K_P = 0.001$ which satisfy the stability conditions.

The simulation results are shown in Figure 5 – 10. Figure 5 displays the queue length dynamic of the congested node n5. Convinced by Figure 5, the designed PID control scheme stabilizes the queue length of the congested node to the target exactly. Figure 6 shows the rate allocation among aggregated flows, i.e., $flow(n1, n5)$, $flow(n2, n5)$, $flow(n3, n5)$ and $flow(n4, n5)$. They all achieve the weighted max-min fairness property, which is consistent with our theoretical analyses. The rate allocations among inner flows, namely, $flow(n2, n3)$ and $flow(n2, n4)$, are shown in Figure 7. Likewise, the rate allocations among them also achieve the weighted fairness. Figure 8 and Figure 9 respectively demonstrate the spatial reuse of link L1 and link L2. Obviously, all flows in two links can reclaim unused bandwidth dynamically in a weighted fair manner. Furthermore, from Figure 10 we observe that the link utilizations of link L1, L2, L3 and L4 are driven to as high as possible due to stability and maximal spatial reuse merits of the algorithm.
V. CONCLUSIONS

Using the PID principle of classical feedback control theory, this paper developed a dynamic bandwidth allocation algorithm termed PID-RPR for RPR networks. We analyzed the traffic model, specified the basic operations of PID-RPR and provided a controller parameter setting guideline. The algorithm can be implemented in a real-time and distributed manner in RPR networks. We showed through both theoretical analysis and simulations that PID-RPR overcomes the limitations of the existed RPR bandwidth allocation algorithms and achieves stability, maximal spatial reuse and fairness simultaneously. In Gandalf proposal for IEEE standard 802.17 [7], two kinds of access control, namely Global access control and Local access control have been specified to ensure fairness, this is exactly what we need to realize the fairness of PID-RPR scheme. Therefore, the algorithm of PID-RPR is believed to be applicable to real RPR networks by utilizing the explicit congestion detection method as specified by IEEE 802.17.

RPR networks have high reliable capability due to their merit that, when a failure occurs in RPR network, the new data transfer paths are reconfigured very quickly by its high-speed protection capability. As PID control has very quick response, i.e., the controller is able to reach its equilibrium very quickly, the proposed PID-RPR can be expected to maintain very good performance as well after reconfiguring new data transfer path when a failure occurs.

REFERENCES


J. Kao et al., “Gandalf proposal for IEEE standard 802.17”, Draft 0.4, Nov. 2001

Liansheng Tan, A. C. Pugh and Min Yin, “Rate-Based Congestion Control in ATM Switching Networks Using a Recursive Digital Filter”, Control Engineering Practice (Special Issue on Control Methods for Telecommunication Networks), vol. 11, no. 10, 2003, pp. 1171-1181.


Liansheng Tan is now a Full Professor and Head of Department in Department of Computer Science, Central China Normal University, PR China. Professor Tan received his Ph.D. degree from Loughborough University in the UK in 1999. He has done research in computer communication network in School of Information Technology and Engineering at University of Ottawa, Ontario, Canada as a postdoctoral research fellow and a visiting research scientist in 2001. He has published over forty referred papers. His research interests are in modeling, congestion control analysis and performance evaluation of computer communication networks.