Applying the DiffServ Model to a Resilient Packet Ring Network

Fredrik Davik[1,2,3] and Stein Gjessing[1]

[1] Simula Research Laboratory
[2] University of Oslo
[3] Ericsson Research Norway
{bjornfd,steing}@simula.no

Abstract. In June 2004, the IEEE approved a a new standard called Resilient Packet Ring (RPR), that is maintained in the 802 LAN/MAN Committee and designated standard number IEEE 802.17. Among the features provided by the RPR technology are built-in QoS capabilities for traffic class differentiation, bidirectional transfer of data with destination stripping and spatial reuse and fast protection against node and link failure(s). In this paper we introduce a framework used to specify the throughput of RPR, and propose a simple mapping between RPR's service classes and DiffServ Per Hop Behavior groups. We evaluate this mapping analytically, using a simple generic example, and by simulating a more complex example. All our findings support that our proposed mapping between RPR's traffic classes and the PHB groups is indeed a viable one.

Keywords: Resilient Packet Ring, Differentiated Services, Per Hop Behavior.

1 Introduction

Transfer of data using the Internet is commonly considered as being a best-effort service: there are no guarantees associated to the transfer of data along any of the traditional Quality of Service (QoS) dimensions: throughput, delay, jitter, data corruption or data loss. For the past two decades, a multitude of mechanisms have been proposed in order to enable Internet service providers to offer IP-based data transfer with QoS guarantees to their customers. These range from simple queueing and scheduling mechanisms, on a per router basis, to more advanced queueing and scheduling mechanisms in combination with resource reservation, packet classification, admission control, policing, shaping and different types of back-pressure. Despite the existence of technical frameworks and actual implementations that can provide IP QoS in some form, the availability of differentiated IP services is not commonplace [1]. In addition to the technical challenges related to providing IP QoS, there are also a number of non-technical issues that must be handled such as accounting, charging and billing [2].

Today, the DiffServ [3] framework, proposed by the Internet Engineering Task Force (IETF), appears to be the most promising and accepted technical solution for the provisioning of IP-based QoS [4].

A recent addition to the IEEE family of standards for LAN/MAN networks is the IEEE 802.17 Resilient Packet Ring (*RPR*) [5]. In this paper, we introduce a formal specification of parts of the service differentiation mechanisms of the RPR standard, and assess RPR's suitability for use in a DiffServ environment. We propose a simple mapping between RPR's service classes and three standardized DiffServ Per Hop Behavior groups. When using this mapping, conformance to the DiffServ Per Hop Behavior groups is discussed and evaluated based on analytical as well as simulation results. For the simulation, we have implemented the RPR standard in the OPNET Modeler discrete event simulator [6].

The rest of this paper is organized as follows: To provide the reader with sufficient background to understand our contribution, in the next sections we provide a short introduction to the RPR technology and the DiffServ Per Hop Behavior groups. Next, in section 4, we present a formal framework for the delay and per service class traffic differentiation mechanisms of RPR. Then, in section 5, our mapping between RPR and DiffServ is discussed and specified. By use of a simple generic example, the performance of this mapping is demonstrated analytically in section 6. In section 7, we proceed to the discussion of a simulation scenario used to demonstrate the mapping in a more complex setting with several competing traffic flows. Finally, in section 8, we asses the mapping based on both the analytic and simulated example, and in sections 9, 10 and 11, we present related work, conclude and point out directions for future work.

2 The Resilient Packet Ring Architecture

The Resilient Packet Ring architecture is a dual ring technology, i.e. data and control messages can be sent from a source node to a destination node on the

ring in either of the two directions (clockwise or counterclockwise). The buffer insertion principle [7,8] allows a node on the ring to transmit variable length locally-generated data packets on an output link as long as the transmitter is not currently transmitting a transiting packet received from an upstream node onto the link. Figure 1 shows a generic node design, utilizing the insertion buffer principle.

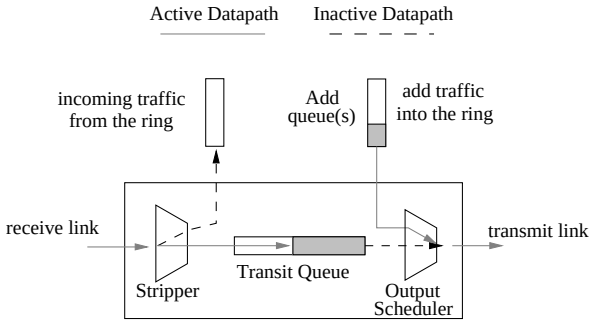

Fig. 1: Generic, simplified node design, showing a node's attachment to the ring for transferral of data in the clockwise direction.

In the figure, packets received from upstream nodes flow through the transit channel, which consists of a packet stripper, transit queue(s) (aka insertion buffer) and the *Output Scheduler*. If a packet is received from an upstream node as a node is adding a locally generated packet onto its outgoing link, the packet from the upstream node is queued in the transit queue until the output link becomes available. Note that a node has one attachment to each of the two ringlets. All links on the ring operate at equal data rate. Upon transmission of a packet, a node local decision is made on which ringlet, and consequently which attachment, to use for sending of the packet. This will typically be the ringlet that provides the minimum hop count on the path to the receiver. The functionality shown in figure 1 resembles the design used by Hafner et. al in [8]. As described in [9], the simple insertion buffer principle must be extended considerably to constitute a full fledged Medium Access Control protocol providing fairness, service class differentiation and protection.

Figure 2 shows the design of an RPR node. The standard allows for use of one or two transit queues in the transit path, denoted as respectively a *1TB*- or *2TB*[4]-design. In the case of a *1TB* design, the transit queue is referred to as the primary transit queue (PTQ), in the *2TB* design, the additional transit queue is referred to as the secondary transit queue (STQ). The *Rate Control Block* consist of a set of shapers, denoted *shA0*, *shA1*, *shB*, *shF* and *shD*, which enforces rate

[4] The purpose of the *2TB* design is to maximize bandwidth reuse by reducing the bandwidth allocation requirements for provisioning of high priority traffic.

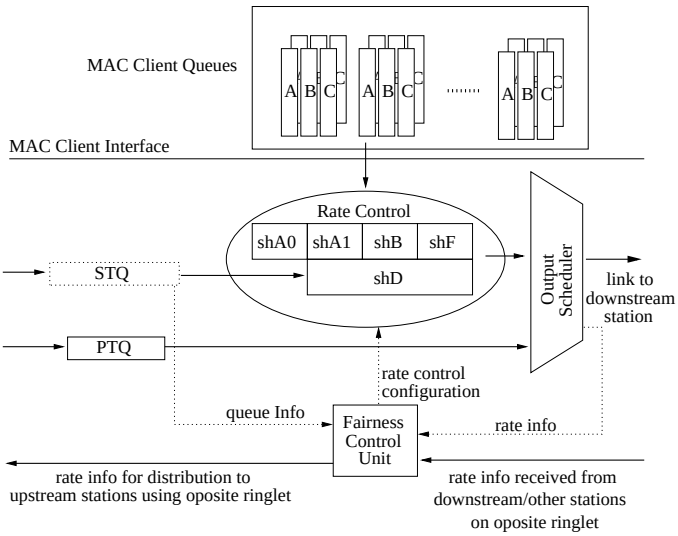

Fig. 2: RPR node design, showing a node's attachment to the ring for transfer of data in the East direction and control information in the West direction. The solid lines indicates the flow of data through the node. The dotted lines indicates the exchange of control/configuration information between node internal function blocks.

control for the different service classes. In the *1TB* design, all transit traffic, awaiting scheduling onto the output link, is stored in the PTQ. In the *2TB* design, the PTQ stores high priority transit traffic, while the STQ stores the remaining transit traffic. For the reminder of this article, the authors focus on the use of a *2TB* transit path design.The *Output Scheduler* is used for selecting between contending packet sources when the output link is ready to accept a new packet. Finally, the *Fairness Control Unit*, maintains the state for the distributed fairness algorithm, ensuring that for congested links, the contending nodes gets their fair share of the available link bandwidth. The operation of the RPR fairness algorithm is discussed in [9,10,11,12].

It is the responsibility of the MAC client (denoted *client* onwards) to classify node ingress traffic in the three available service classes A, B and C. Service class A is the highest priority service class and is implemented to provide guaranteed throughput and low jitter values independent on the ring circumference. Service class B, which is the 2nd priority service class, is intended to provide guaranteed throughput and a bounded jitter range which is dependent on the circumference of the ring. Finally, service class C, which is the lowest priority service class, is a best-effort/scavenger service class and has no associated guarantees. We will discuss the properties of the RPR service classes more in section 4.

3 The DiffServ Architecture

In this chapter, we give a brief introduction to DiffServ and some of its most commonly used building blocks (PHBs). We also discuss the requirements that must be met by a DiffServ implementation, in order to claim conformance to these PHBs. In section 5, we will discuss these building blocks used in the context of an RPR network.

Differentiated Services (*DiffServ*) [3] was introduced as a "simple" and scalable framework for providing IP-based service class differentiation in an IP-network. In a DiffServ enabled network, IP packets are classified and marked at the network's ingress node(s). Based on some classification rule at the ingress node, the packet is assigned a DiffServ Code Point value (DSCP), carried within the packet's DS field [13]. This value maps the packet onto the network's available Per Hop Behaviors (PHB[5]), resulting in a specific packet forwarding at each DiffServ compliant node traversed by the packet.

DiffServ has a PHB named Expedited Forwarding (EF) [14]. The Expedited Forwarding PHB is specified with the intent of providing a DiffServ building block that can be used for the provisioning of services that provides low loss, low delay and low jitter. To be EF compliant, a DiffServ node has to comply to quantified delay and jitter values that are function of the rate R provided by the PHB. The EF basic conformance requirements for a DiffServ EF PHB implementation is specified in [14]. The most important conformance requirements are shown in (1) and (2) below.

$$\forall j > 0 : D_j <= F_j + E_a, \text{ where } F_j \text{ is defined iteratively by} \quad (1)$$

$$f_0 = 0, d_0 = 0, \forall j > 0 : F_j = max(A_j, min(D_j - 1, F_j - 1)) + \frac{L_j}{R} \quad (2)$$

In short, (1) (2), introduces a bound on the actual time (D_j), when a packet, j, should leave the node. In addition to the actual arrival time (A_j) of packet j, the ideal departure time (F_j) takes into account the actual- and ideal departure times of the previous packet, $j-1$, that was sent from the same node. The expression also includes an error term, E_a, that represents the worst case deviation between the actual and ideal departure time of any EF packet to traverse this node. Finally the fraction $\frac{L_j}{R}$ accounts for the ideal per packet transmission delay for an EF PHB with a committed service rate of R transmitting a packet j of length L_j.

Another DiffServ PHB, namely the Assured Forwarding (AF) PHB is a specification of PHB group, that may contain up to four independent AF classes [15]. Each class must be allocated a separate amount of forwarding resources. Within each AF class, an implementation must provide a minimum of two different drop probabilities and a maximum of four different drop probabilities. Conformance to the AF PHB by a DiffServ node is described in terms of the throughput obtained, relative service for the different drop probabilities within an AF class and no reordering of packets within an AF micro-flow.

[5] A single PHB is a special case of a PHB group.

A third PHB, is the so called Class-Selector PHB specified in [13]. This PHB can be implemented in a DiffServ node to provide a network that is compatible with the historical IP Precedence use.

The last PHB we want to introduce to the reader, is DiffServ's default PHB, which is specified in [13] as a best-effort (BE) forwarding behavior. For the reminder of the paper, we refer to traffic belonging to the default PHB, for BE traffic.

In this paper we limit the discussion the EF, AF and the default PHBs.

4 Delay Guarantees and Rate Control Functionality in RPR

In this chapter we start by providing a brief description of the RPR delay properties. Next, in section 4.2 we provide a detailed description of the workings of the RPR *Rate Control Block* and its associated interface towards the *client*. This is necessary in order to provide to the reader sufficient understanding of how the service differentiation between the three service classes A, B and C is performed in an RPR node. At the end of the chapter, we present our set of invariants that constitutes a specification of the differentiation between the RPR service classes.

4.1 Delay Properties

The delay of class A traffic consist of access delay at the ingress point, transmission delay (in ingress and transit nodes), link propagation delay and queuing delay in the transit path. The transmission delay per node transited is fixed and dependent on the packet size and link capacity. The transit queue delay for class A packets is in the worst case the aggregate of times we have to wait for nodes in the transit path to finish transmission of a locally added data packet (as well as locally added control packets), since RPR does not support pre-emption of lower priority packets.

Access delay is measured from a packet is delivered to the *Rate Control Block* until it is transmitted onto the link connected to the downstream node. When within its rate bounds, class A add traffic is prioritized over other add traffic, but has lower priority than transmission of idle and fairness packets as well as transmission of packets from the PTQ. Thus in the worst case, a class A packet can risk waiting several packet transmission times before being scheduled for transmission on the downstream link. Given a ring with N nodes, utilizing shortest path routing of packets, in the worst case[6], we can receive a packet train consisting of N/2 back-to-back MTU-sized class A packets. In this case, with L_{MTU} representing the bitsize of a MTU sized packet and R_L is the link rate, the the worst case access delay, E_a is given by (3) below. On average, the

[6] Assuming upstream stations are configured so that they are not able to accumulate enough credits to insert another class A packet into the packet train

access delay time for a class A packet will be less than the transmission time of an MTU sized packet.

$$E_a = \frac{N}{2} \cdot \frac{L_{MTU}}{R_L} \tag{3}$$

4.2 Rate Control

The MAC layer provides, via the rate control block (see figure 2), the client with per service class information on for which ringlet traffic can currently be accepted. In the case of class C traffic, the MAC layer also specifies an additional per ringlet constraint, namely the maximum distance (hop count) a packet is allowed to travel on the associated ringlet. If this value is less than 255 (the maximum number of nodes on an RPR ring), this indicates two things: i) there is a congestion point (a congested link) on the associated ringlet and ii) transmission of class C traffic beyond the congestion point is currently not allowed. This information can be utilized by *client*s implementing some form of Virtual Destination Queueing [16] to avoid Head-Of-Line (*HOL*) blocking [17]. In this context, *HOL* blocking would be that packets at the head of a client queue, if transmitted onto their associated ringlet, would traverse the congested link on their way to their destination and thus have to remain in the queue, while other packets in the same queue destined for nodes before the congested link are blocked. Figure 2 shows a possible construct where the *client* implement a set of queues per node on the ring. Each set contains two queues (one per ringlet) for respectively class A, B and C packets.

When the MAC layer, via the rate control block, indicates that it can accept class C traffic, the *client* can make a local decision on whether it wants to transfer class B traffic instead. This can be done, as long as the distance the class B packet will travel on the ring, is within the current maximum (see discussion above). The effect of this is that the MAC client may transmit class B traffic in excess of the (configured) shB shaper setting. If the *client* chooses to do so, when the demand is greater than the (allowed portion) of the link capacity, this is done on the expense of class C traffic.

Once a packet has been transferred from the *client* to the MAC layer, the rate control block assigns it into one of the following subclasses $A0$, $A1$ (*2TB* implementations only), *B-CIR*, *B-EIR* and C.

Subclass $A0$ traffic is rate controlled via the *shA0* shaper and is together with subclass $A1$ traffic the highest priority service class. Subclass $A1$ traffic is rate controlled via the the union of shapers *shA1* and *shD*.

The purpose of the $A1$ traffic class for *2TB* implementations is for a given guaranteed amount of A traffic, to allow for lower priority traffic classes to reuse some of this capacity, the $A1$ capacity, when not in use. By use of the *STQ*, this is done without compromising the associated throughput and delay/jitter guarantees,

When a packet marked as class A is received by the *Rate Control Block* from the *client*, it is marked as belonging to subclass $A0$ or $A1$, depending on

the status of shaper *shA0* and the union of shapers *shA1* and *shD*. Packets of subclasses $A0$ and $A1$ experience the same delays both at the ingress node and in the transit path (both subclasses goes through the PTQ in transit nodes).

It it the user's (network operator) responsibility to configure the nodes on the ring. To facilitate (spatial) reuse of the available bandwidth resources, the amount of traffic of subclass $A0$ should be kept as low as possible.

Class B traffic is rate controlled by the union of shapers *shB* and *shD* and the union of shapers *shF* and *shD*. The class B traffic sent within the rate bounds of shaper *shB* is denoted *B-CIR* (Committed Information Rate). The class B traffic sent in excess of the rate bounds of shaper *shB*, maybe on the expense local class C traffic, is denoted *B-EIR* (Excess Information Rate).

Class C traffic is rate controlled by the union of shapers *shF* (shaper for fairness eligible traffic) and *shD*. The RPR standard defines different methods for implementing the *shF* shaper. The configuration of this shaper is performed dynamically, based on rate measurements performed at the output of the *Output Scheduler* (see figure 2) and calculations performed by the *Fairness Control Unit*.

The *Rate Control Block* imposes several per ringlet and per node rate constraints on local ingress (add)- and transit traffic. We use the notation R_X to specify the rate constraint in effect for a particular type of traffic. In the cases where $X \in \{A, B, C\}$, X specifies a rate constraint for a particular RPR service class. *offered(X)* represents the amount of traffic of a particular traffic class that a node or a set of nodes **want(s)** to transmit. *accepted(X)* refers to the corresponding amount of traffic that **can** be transmitted. First, in (4) and (5), we start with the definition of the reserved (preallocated) and unreserved (reclaimable) bandwidth, denoted respectively R_R and R_U. The bandwidth of the link is denoted R_L.

$$R_R \triangleq \sum_{j \in \{nodes\}} R_{A0_j} \tag{4}$$

$$R_U \triangleq R_L - R_R \tag{5}$$

The invariant below expresses that the sum of preallocated (subclass $A0$) bandwidth, cannot exceed the actual link capacity (R_L).

invariant 1 $R_R \leq R_L$

Invariant 1 is enforced only during ring initialization (start-up and any topology change (node addition or removal)) and triggers a node alarm if violated. In a running system, it is not possible to transmit preallocated traffic at a rate greater than the link-rate. However, if the configuration is done so that invariant 1 is violated, this effectively prohibits the sending of traffic of all other traffic classes. This follows from the invariants specfied below.

Invariant 2, restricts the sum of $A1$ and *B-CIR* traffic upwards to the rest of the available bandwidth. Currently, there is no functionality in the RPR standard that ensures that this invariant is met. If the configuration of corresponding shapers, *shA1* and *shB*, violates the invariant, this may break class A and B service guarantees. Thus, it is left to the operator of an RPR network to ensure that the configuration of *shA1* and *shB* shapers does not violate this invariant.

invariant 2 $\sum_{j \in \{nodes\}} \left(R_{A1_j} + R_{B_j} \right) \leq R_U$

The restriction shown in invariants 3, 4 and 5, effectively throttles the amount of $A0$ and $A1$ add traffic a node can add to the ring (to a configured value). For $A0$ traffic, this invariant is enforced solely by the *shA0* shaper shown in figure 2. For $A1$ traffic, this invariant is enforced by the union of the *shA1* and *shD* settings.

invariant 3 $\forall nodes : \mathit{offered(A)} \leq R_{A0} \Rightarrow \mathit{accepted(A0)} = \mathit{offered(A)}$

invariant 4 $\forall nodes : R_{A0} < \mathit{offered(A)} \leq R_{A0} + R_{A1} \Rightarrow$
$\mathit{accepted(A0)} = R_{A0} \wedge \mathit{accepted(A1)} = \mathit{offered(A)} - R_{A0}$

invariant 5 $\forall nodes : \mathit{offered(A)} > R_{A0} + R_{A1} \Rightarrow$
$\mathit{accepted(A0)} = R_{A0} \wedge \mathit{accepted(A1)} = R_{A1}$

In invariants 1-5, we have worked with per ringlet invariants. This was because for class A traffic, the sum of traffic is calculated, regardless of the destination of the traffic transmitted. For the amount of class C (and *B-EIR*) traffic accepted however, this may vary on a per link basis and depends on the number of stations sending traffic over the same link, and their individual sending pattern. Thus, for the remaining invariants 6-9, it is more relevant to present per-link invariants, in order to express the relative priority of the RPR service classes.

In invariants 6 and 8, no portion of the offered class B (and class C) traffic passes a congested link, or if it does, the amount of offered traffic is equal to or lesser than rate constraints in effect over the congested link. Thus all the offered class B (and class C) traffic traversing this link is accepted.

invariant 6 $\forall links : \mathit{offered(B)} + \mathit{accepted(A1)} \leq R_U \Rightarrow \mathit{accepted(B)} = \mathit{offered(B)}$

In invariants 7 and 9, the link under observation is congested (the demand is greater than the capacity). Assuming that the class B (and class C) traffic traversing the link does not traverse a downstream link that is more congested, the amount of accepted class B (and class C) traffic equals the portion of the R_U bandwidth not already in use by $A1$ traffic.

invariant 7 $\forall links : \mathit{offered(B)} + \mathit{accepted(A1)} > R_U \Rightarrow$
$\mathit{accepted(B)} = R_U - \mathit{accepted(A1)}$

invariant 8 $\forall links : \mathit{offered(C)} + \mathit{accepted(A1)} + \mathit{accepted(B)} \leq R_U \Rightarrow$
$\mathit{accepted(C)} = \mathit{offered(C)}$

invariant 9 $\forall links : \mathit{offered(C)} + \mathit{accepted(A1)} + \mathit{accepted(B)} > R_U \Rightarrow$
$\mathit{accepted(C)} = R_U - \mathit{accepted(A1)} - \mathit{accepted(B)}$

5 Proposal for DiffServ PHB Mappings in an RPR Network

When proposing a mapping between DiffServ PHBs and RPR service classes, we would like to remind the reader of the fundamental properties of the three RPR service classes discussed in section 2. In addition to a guaranteed and limited throughput, service class A class is characterized by low delay and jitter values. Class B traffic is characterized by guaranteed bandwidth and bounded delay and jitter values. Finally, class C is a best-effort/scavenger service class and has no associated guarantees.

What we propose, is to implement the three standard PHBs: Expedited Forwarding, Assured Forwarding and default PHB and to map these to RPR service classes A, B and C. Given the the strict priority scheduling rules presented in section 4.1 and the various invariants used to maintain the per-service class rate configuration presented in section 4.2, this results in a minimum access delay as well as per node transit delay for class A traffic. Also, the above invariants ensures a guaranteed bandwidth share for both class A and class B traffic. Based on this, the implementation of an EF PHB for a DiffServ enabled RPR node by use of RPR service class A seems feasible.

RPR's service class B does provide guaranteed throughput and in-order deliver of packets during normal operation of the ring. To provide an AF conformant PHB based on RPR's service class B however, the client has to implement the relative packet drop priorities. The implementation of relative drop priorites in the client should be a relatively easy task, using some form of priority queueing scheduling algorithm. Thus acheieving an AF compliant DiffServ implementation based on RPR seems feasible.

Finally, we have an obvious match between DiffServ's default PHB, specified in [13] as a best-effort forwarding behavior, and RPR's service class C. In an RPR network, when mapping DiffServ's default PHB to RPR's service class C, this will work in conformance with the requirements of DiffServ's default PHB. Following the presentation of simulation scenarios and results in section 7 we will conclude on the conformance to the DiffServ PHB requirements for the proposed mapping between the DiffServ PHBs and RPR service classes. But first, in section 6, we show an analytical example using the proposed mapping.

6 Evaluation of RPR Service Class Differentiation by Analytical Example

In figure 3, we have shown the analytical resulting throughput when mapping the EF-, AF and default PHBs to RPR's service classes A, B and C. The analytical example uses the invariants introduced in section 4.2 to assign bandwidth to the DiffServ PHBs and plots the accepted amount of traffic for the three PHBs for a linearly increasing load level. In the example, we calculate the per PHB- as well as the total throughput for a single node aggregate, using all bandwidth resources on the ring-segment from the source- to the destination node. For scenarios with

more than one node sending DiffServ traffic over a common ring segment, use of analytical models becomes too difficult, thus we use our simulator model to analyze the PHB behavior. For the analytical example, the initial load consist of 55% best-effort traffic, 27% AF traffic and 18% EF traffic. This represent a reasonable setting, where the majority of the traffic consist of low-priority (BE) traffic, and the remaining fraction of this is divided using a ratio of 3:2 between the amount of medium (AF)- and high-priority (EF) traffic offered. The aggregate of the initial load is lesser than the link-bandwidth. From this starting point, we increase the amount of traffic offered linearly for the three PHBs while maintaining the 55/27/18 ratio. The R_R fraction is set to 10% of the link bandwidth. The offered load is increased until the point where the division of link bandwidth remains at a constant level, regardless of any additional increases in the amount of offered traffic.

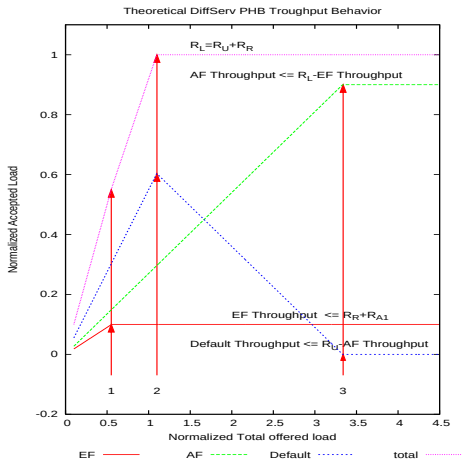

Fig. 3: Theoretical Throughput for the EF-, AF and default PHBs when mapped to RPR service classes A, B and C.

In the figure, as the offered load increases, the first threshold we approach is the point labelled 1. At this point, to maintain invariant 5, no more EF traffic can be accepted onto the ring. Thus, above this threshold, the curve showing the aggregate of accepted EF traffic remains at a constant level. As the offered load increases more, the next threshold we approach is the one labelled 2 in the figure. At this threshold, we have reached the load-point where the sum of traffic on the link equals the available capacity, and to maintain invariants 6 and 9, the amount of best-effort traffic has to be reduced to accommodate an equivalent increase in AF traffic. The last threshold is the load-point labelled 3 in the figure. This is the point where, to maintain invariant 7, no more AF traffic can be accepted onto the link. Thus above this threshold, the AF throughput stays constant.

7 Performance Evaluation by Simulations

In section 6 the suitability of the Resilient Packet Ring's built-in mechanisms for service differentiation in a DiffServ environment, was demonstrated analytically by a simple example. In this chapter we use our RPR discrete event simulator model implemented in the OPNET Modeler to perform the same evaluation based on a more complex example.

In section 6 we analyzed one link only, or, in the case of more links, assumed that all resources on all links are at any time maximally utilized.

This will not be the case when RPR is deployed. Then some links will be more utilized than others, and some links will be congested while other links could accept more traffic.

The simulated example uses a topology corresponding to a large (800km) metro- or small WAN-ring, interconnecting a number of lower-capacity access-rings. For such rings, link speeds in the range 1-10Gb is commonly used. In this paper, we present results obtained for a 16 node ring, where the link delay and capacity is respectively 250 μs and 1Gbit/s for all links.

In a backbone-ring, aggregating traffic from several access network, the traffic pattern is typically that all nodes on the ring send to all other nodes on the ring. However using an all to all traffic matrix will utilize all links of the ring almost equally, and we are back to our analytical example. To keep this all-to-all traffic characteristics, and at the same time ensure a more unbalanced load, we instead let each node on the ring send all its traffic to one other randomly selected node. Also, each node receives data from one other node. The random source-destination pairs stay fixed throughout the duration of each simulation.

For each simulation an offered load parameter is given. 101 load values are used, linearly distributed on a scale from 1 to 10 times the base load. For each load value 16 simulations are executed with different sets of source destination pairs, in order to give some statistical significance to the final result.

The base load is also the same as in section 6. All nodes transmit traffic of each PHB group using a Poisson distribution.

The presentation of the simulation results is split in two parts. In section 7.1, we present the throughput performance of the individual PHB groups as well as the aggregate throughput performance of the ring. In section 7.2, we present the delay/jitter performance of the EF, AF and default PHB groups.

7.1 Per Hop Behavior Group Throughput

Figure 4 plots the throughput for the three PHBs EF, AF and default (Best Effort, BE), as well as the aggregate of all PHBs. Each point on the curves is the mean of the throughput values obtained from the 16 different sender-receiver simulations for that particular offered load.

When looking at the throughput-performance of the ring, the behavior is not as clear-cut as that of a single link, as illustrated in figure 3. The general trend however follows the theoretical pattern, given by the invariants specified in section 4.2. Thus, as the offered load increases, the increase in accepted EF

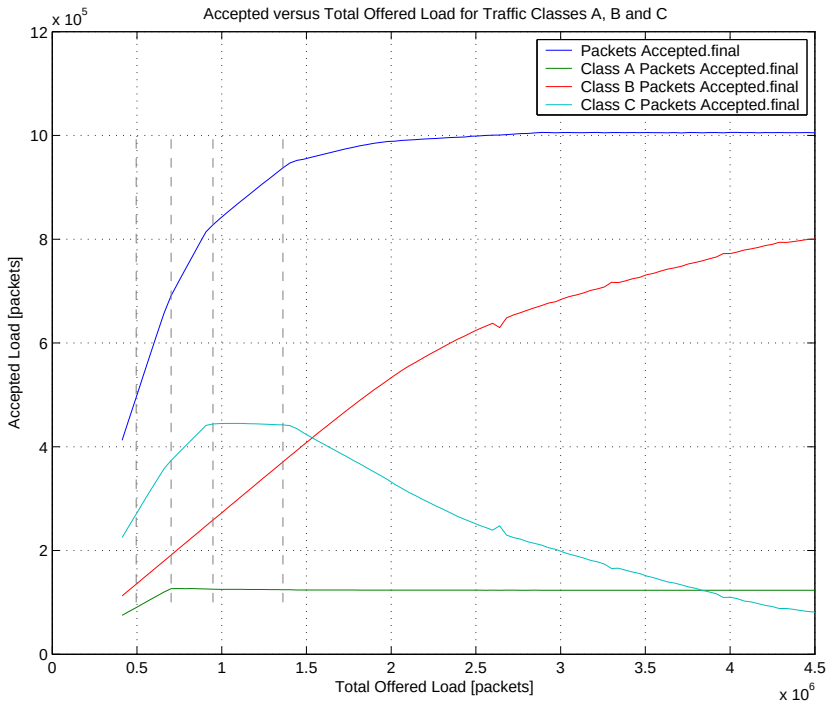

Fig. 4: Accepted vs. total offered load for EF=A, AF=B and default=C PHB Groups

traffic is linear up to some point and then remain at a constant level. For both AF- and BE traffic, the accepted load increase linearly initially, and as some links get congested, the amount of BE traffic accepted has to be decreased to allow for an increase in accepted AF traffic. Up to an total offered load of $\sim 0.7 \cdot 10^6$, the accepted traffic for all PHBs increases linearly as a function of offered load. At this point however, the amount of offered EF traffic has reached the point where a further increase in accepted EF traffic would violate invariant 5, thus no further increases in accepted EF traffic is allowed. For BE traffic, at an offered-load of $\sim 0.95 \cdot 10^6$, some links in the simulated sender-receiver pair scenarios have become congested, resulting in a stop in the increase of accepted BE traffic (to give room for AF traffic and hence maintain invariant 9), while others link still manage to absorb more BE traffic. At a total offered load of $\sim 1.4 \cdot 10^6$. sufficiently many links have become congested, resulting in a clear decrease in the acceptance rate for BE traffic while maintaining a linear increase in accepted AF traffic. Shortly thereafter, the increase in accepted AF traffic starts to decrease slowly towards 0, as more links utilize all their capacity, and hence, to maintain invariant 7, all further increases in accepted AF traffic is stopped.

From the simulation results, it is clear that when making an implementation according to the standard, the invariants established in section 4.2 will be

maintained. Thus seemingly, we get a clear service differentiation between traffic of the EF, AF and default PHBs, regardless of the offered load and location of active nodes on the ring. Additionally the EF traffic gets its intended guaranteed (and rate-limited) throughput. Similarly the AF traffic also gets its guaranteed throughput and absolute priority over the BE traffic (on ingress). Finally, it is clear that the best effort traffic is able to utilize the bandwidth not already in use by EF and AF traffic.

7.2 Per Hop Behavior Grop Delay

In this section, we present the delay measurements for the EF-, AF- and default PHBs for the scenarios presented in section 7 above.

In the delay measurements, we have removed the transmission- and propagation delay from the samples in the set. By doing this, it is easier to compare the delay components caused by access delay at the ingress as well as queueing delay in the transit path.

We start by presenting the delay measurements for traffic of the EF PHB. Figure 5 shows the delay distribution for EF traffic, while figure 1 shows the associated statistics broken down on a per-hop level. The mean values are in the range $[0.6, 5.3]\mu s$, which is well within the transmission time of two 500B packets. The standard deviation is very low (less than transmission time of a 500B packet). For packets traversing less than 3 hops on the ring, the median of the sample distribution is located at 0 μs, meaning that at least 50% of the packets experience no access delay in the ingress node and no queueing delay in the transit path. As expected, the median, mean, standard deviation and max values all increase as a the number of hops increase for the samples observed.

hops	median $[\mu s]$	$\overline{x}$ $[\mu s]$	σ $[\mu s]$	max $[\mu s]$	min $[\mu s]$	n [sample count]
1	0.0	0.6	1.3	24.1	0.0	12633535
2	0.0	1.2	1.7	24.4	0.0	13060494
3	1.2	1.8	2.0	27.0	0.0	13581686
4	2.0	2.4	2.3	24.0	0.0	14681577
5	2.3	2.7	2.6	32.9	0.0	15525550
6	2.9	3.3	2.8	28.6	0.0	15241110
7	3.9	4.3	3.1	28.0	0.0	10212709
8	4.9	5.3	3.4	30.0	0.0	4283579

Table 1: EF traffic delay statistics broken down on a per hop level.

When comparing the sample distribution for EF traffic that traverses 1 hop vs EF traffic that traverses 8 hops on the ring, we clearly see in figures 6 and 7 that the added transit queue delay introduces some jitter. But as shown, the jitter is very low and in the worst case within 32.9 μs.

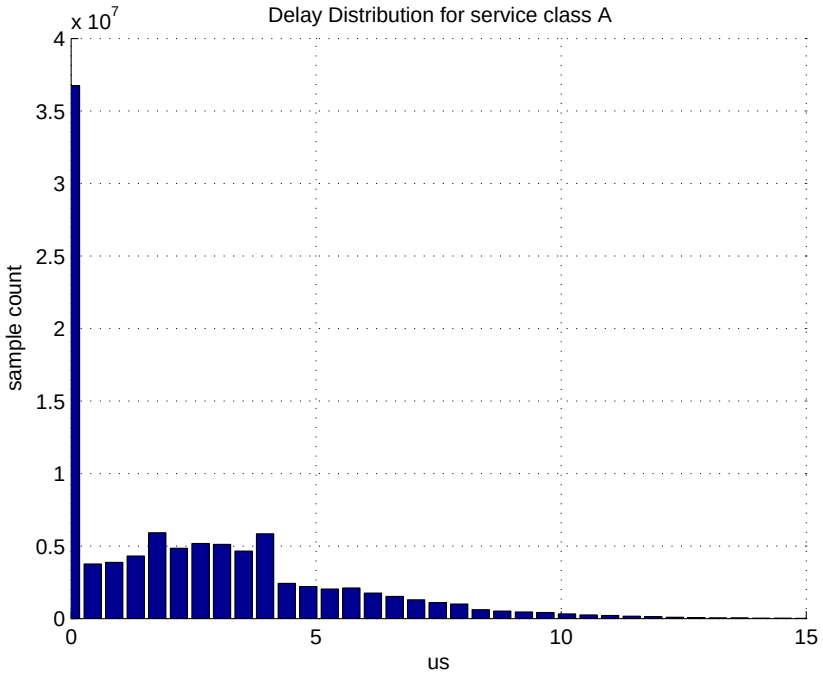

Fig. 5: EF traffic delay distribution

The measurement of AF- and BE delay includes an extra delay, namely the time waited from an AF- (or BE) packet becomes *HOL* in the corresponding *client* queue until it is delivered to the *Rate Control Block*. The reason this delay is not measured for EF traffic, is that EF traffic is rate limited at a hard limit, namely the aggregate of rates as configured for the class A traffic shapers *shA0* and *shA1*. From the time a class A (EF) traffic packet is delivered from the *client* to the *Rate Control Block* and it is transmitted onto the ring, an aggregate class A traffic transmission rate of 3.7% means that on average, it will take $\frac{packet\ size}{rate} = \frac{500 \cdot 8}{0.037 \cdot 1E9} = 108\mu s$ before the *Rate Control Block* will accept the next EF packet from the *client*. When packets are delivered to the *client* faster than they are accepted by the *Rate Control Block*, which is the case as the load increases, this will add significant to the delay values measured for EF traffic. Actually, if the queue time while waiting at *HOL* position in the *client* queue is included in the delay measurement, the delay values for EF traffic will exceed those of AF- and BE at low hop-count values. For AF- and BE traffic, the added delay value is much smaller, because the rate value in the denominator is much larger. Of course, this is not true for BE traffic at high loads, as the BE rate goes towards 0. But as BE traffic has no associated rate or delay guarantees, measurements of BE delays at high loads are expected to be high, regardless of measurement method.

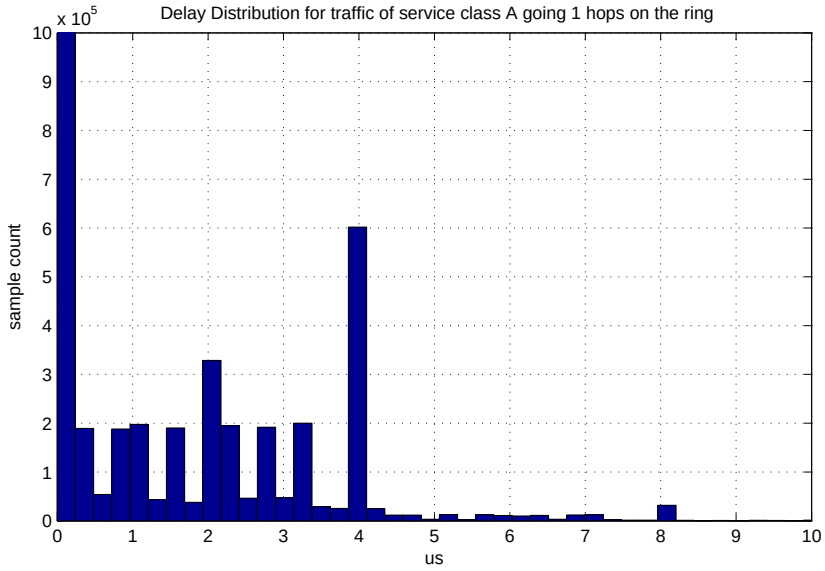

Fig. 6: Delay Distribution for EF traffic going 1 hop on the ring. The maximum sample count value (approximately 1E7) in the first bar is not shown.

When observing the delay distribution for AF traffic shown in figure 8 with associated statics values broken down to per hop traffic shown in figure 2, we see from the figure that the distribution is contained within the range $[0, 39053.8]\mu s$. As expected, when the hop-count increases, so does the max, median, mean and standard deviation. The distribution for a hop-count of 1, shown in figure 9 is very narrow (contained within the range $[0, 347.5]\mu s$). One thing to note here, is that for traffic going only one hop on the ring, there is no transit path delay, as the traffic does not pass through any transit queues. The reason that the delay values are larger than those of EF traffic going one hop on the ring, is partly due to lower priority at the ingress.

For AF traffic going one hop on the ring, with its delay distribution shown in figure 9, if we were to exclude the delay component measuring the time waited at *HOL* in the associated *client* queue, the delay distribution would be similar to that of EF traffic.

In figure 10, we can observe the effect of the transit queue on delay for AF traffic traversing 2 hops on the ring. As shown in the figure and in figure 2, the center of the distribution is shifted right with a factor 3 and the distribution becomes much wider (increases with a factor 40). When comparing EF traffic traversing respectively one and two hops on the ring, the distribution center remained constant and the distribution width increased with a factor 1.01.

When comparing the delay statistics for AF- and BE traffic listed in figures 2 and 3, we note that the median values for hop count values > 3 and mean

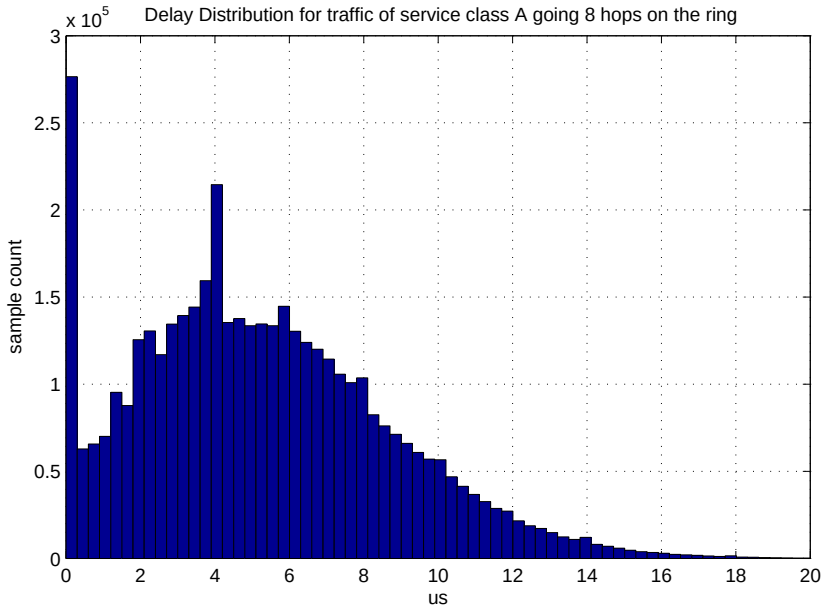

Fig. 7: Delay Distribution for EF traffic going 8 hops on the ring.

values for hop count > 1 are smaller for BE traffic than for AF traffic. The main reason for this is that as the load increases, the amount of AF traffic increases and BE traffic decreases, as seen in figure 3. Hence most of the BE packets used in the statistics are sent under light load, while most of the AF traffic packets are sent over a highly loaded ring.

Some of BE's traffic associated delay properties has already been discussed above. As already said, BE traffic has no associated rate or throughput guarantees. Figure 11 shows the delay distribution for BE traffic with associated statistics values broken down to per hop traffic shown in figure 3. The width of the distribution is 20 times that of the AF traffic distribution.

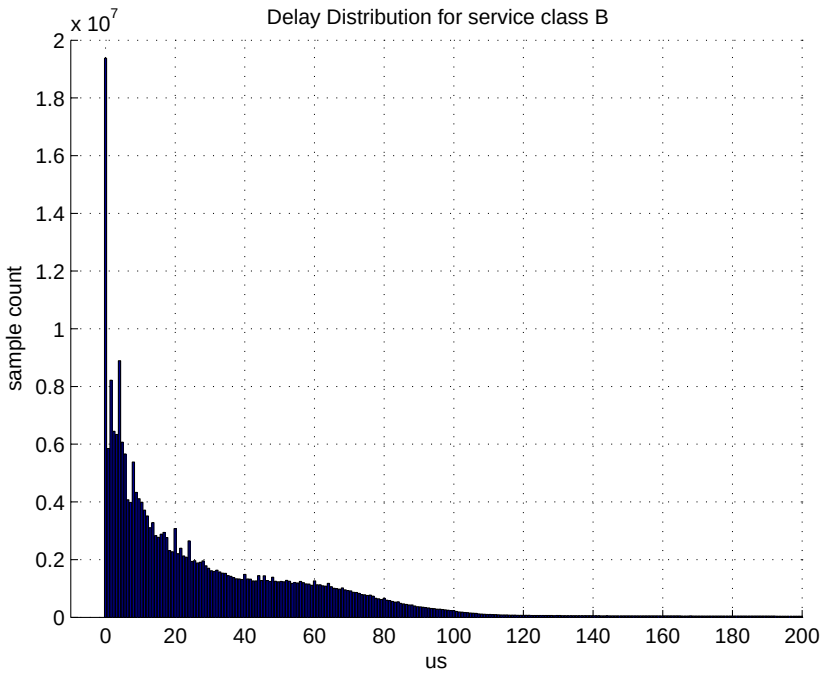

Fig. 8: Delay Distribution for AF traffic.

hops	median $[\mu s]$	$\overline{x}$ $[\mu s]$	σ $[\mu s]$	max $[\mu s]$	min $[\mu s]$	n [sample count]
1	7.7	17.7	23.4	347.5	0.0	64456636
2	27.9	1089.6	2426.2	15065.5	0.0	62972181
3	71.0	1890.3	3198.4	24688.0	0.0	64384109
4	396.5	2488.5	3484.9	21913.9	0.0	68195750
5	481.1	2844.6	4190.4	33481.8	0.0	67673544
6	2769.3	4134.0	4742.0	33172.9	0.0	63227469
7	4134.8	5541.9	5524.2	30980.4	0.0	42576855
8	5589.2	7075.1	7004.2	39053.8	0.0	16360568

Table 2: AF traffic delay statistics broken down on a per hop level.

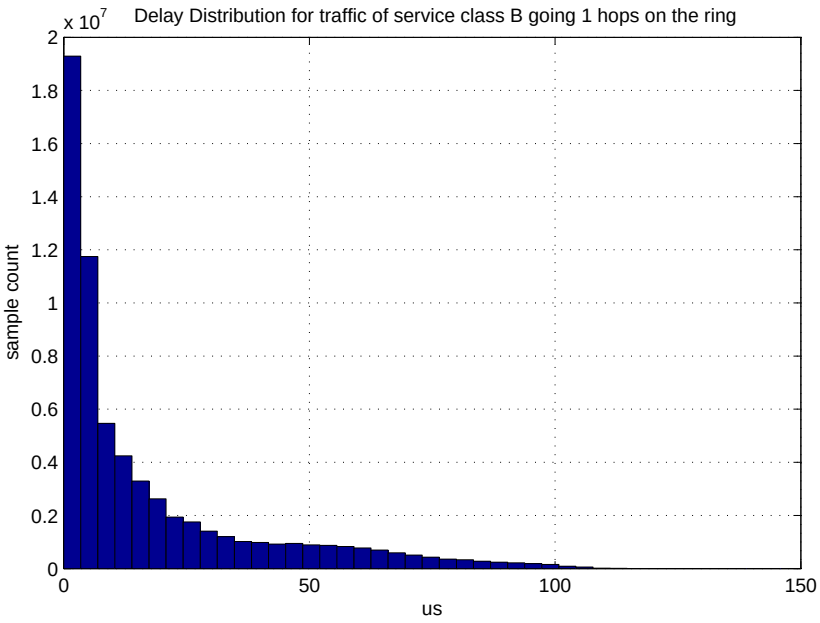

Fig. 9: Delay Distribution for AF traffic going 1 hop on the ring.

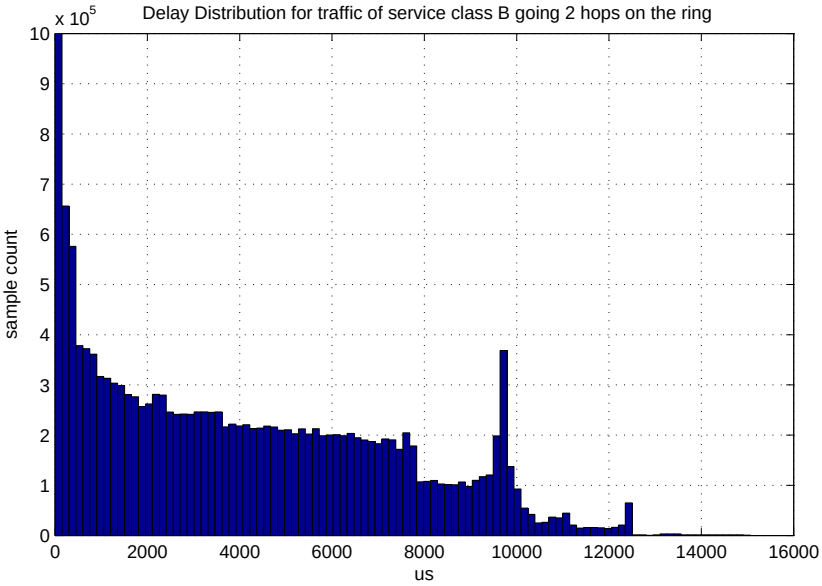

Fig. 10: Delay Distribution for AF traffic going 2 hops on the ring. The maximum sample count value (not shown on figure) for delays in the range 0-150 microseconds is approximately 4.7E7.

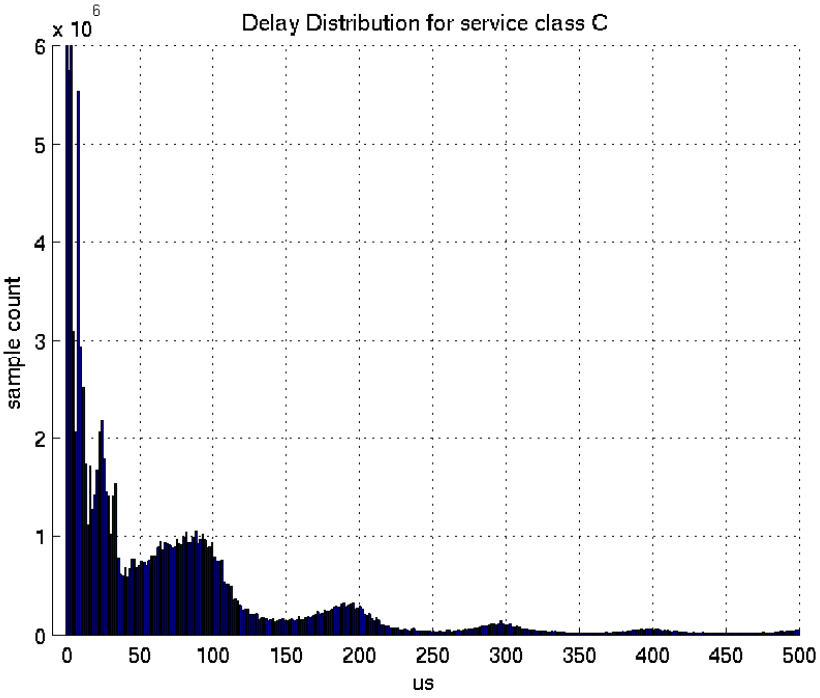

Fig. 11: Delay Distribution for BE traffic. The maximum sample count value (not shown on figure) for the bar at 0, is approximately 2.8E7

hops	median $[\mu s]$	$\overline{x}$ $[\mu s]$	σ $[\mu s]$	max $[\mu s]$	min $[\mu s]$	n [sample count]
1	23.6	57.4	907.0	877588.0	0.0	51523947
2	71.5	1041.5	2687.1	879586.0	0.0	32931628
3	114.3	1672.6	3174.6	627512.0	0.0	33659285
4	142.8	1925.7	3397.7	851505.0	0.0	29163589
5	170.1	2357.4	4554.2	866889.0	0.0	25276927
6	542.8	3163.1	5057.4	856188.0	0.0	20603736
7	1824.8	4129.7	5584.2	881981.0	0.0	14428445
8	1946.5	4687.8	7199.0	788670.0	0.0	3870403

Table 3: BE traffic delay statistics

8 Conformance to DiffServ PHB Requirements

In section 3, the conformance requirements for the DiffServ PHBs EF, AF and default were presented. Having presented both analytical and simulation results evaluating throughput and delay performance of the proposed PHB mappings, we will now discuss how well our results meets the conformance requirements. For DiffServ EF traffic, we start by analyzing the throughput results. Given a service rate of $R = 10\%$ of the link bandwidth (for the analytical example), this is clearly in compliance with the EF throughput shown in figure 3. For the scenarios simulated, the per-node aggregate rate of class A ($A0$ and $A1$) traffic is set to 3.7% of the line rate. The simulation time for the plot shown in figure 4 is 0.9s. Thus, with $N = 16$ nodes, $R_L = 10^9 bits/sec$, $t_{sim} = 0.9s$ and $L_p = 500B$, this corresponds to: $\sum_{A\ traffic} = \frac{N \cdot (R_{A0}+R_{A1}) \cdot R_L \cdot t_{sim}}{L_p} = \frac{16 \cdot 0.037 \cdot 10^9 \cdot 0.9}{4000} = 133200[packets]$. If we study the throughput plot, we see that this matches the EF throughput in figure 4 well. We do however loose some of the EF capacity to the sending of RPR fairness and control packets. Thus, for an EF PHB requiring a service rate R, we should reserve some additional RPR class A bandwidth to allow for the overhead required by the RPR fairness algorithm. When studying conformance to EF packet departure time requirements, using a packet size of 500B and $R = 0.1 \cdot R_L$, this results in the ratio $\frac{L_j}{R} = \frac{500 \cdot 8}{1e9 \cdot 10\%} = 40\mu s$. The E_a element, corresponds to the worst case access delay, specified in (3), which equals $E_a = \frac{16 * 1500 * 8}{2 \cdot 1e9} = 96\mu s$. Thus, assuming that the previous packet, $j-1$, was sent at its ideal time ($D_{j-1} = F_{j-1} \geq A_j$), the ideal departure time of packet j equals $F_j = D_{j-1} + \frac{L_j}{R} + E_a = D_{j-1} + 40 + 96[\mu s]$. On average, the access delay of an EF packet, is less than the transmission time of 1 MTU sized packet, and the probability that we get a packet train of N/2 MTU sized EF packets is diminishingly small. Even if this happens, we have a safety margin of $40\mu s$. Thus, clearly, the actual departure time, D_j will be smaller than the ideal departure time. Hence, compliance to the EF PHB packet departure time requirement is clearly achieved. The results obtained from simulations supports this. From figure 1, we see that the worst case delay (including transit path queueing delays), is $32.9\mu s$. This is clearly within the margins calculated above.

When studying the conformance requirements of the AF PHB, we see clearly from both the analytical results in figure 3 and the simulation results in figure 4, that the AF traffic acquires the bandwidth share not used by the EF traffic. We also see that when the sum of offered AF- and BE traffic is greater than the available capacity, the AF traffic demand is satisfied on the expense of the BE traffic. Thus, this is in conformance with the AF and default PHB requirements. However, as noted in section 5, the relative packet drop priorities have to be implemented in the client. Finally, in-order delivery of AF packets is guaranteed by the RPR standard under normal operation of the ring.

Conformance to the default PHB is clearly obtained, as the bandwidth not utilized by the EF and AF traffic, is utilized by the BE traffic.

9 Related Work

Several papers have been published studying different RPR performance aspects, both for hardware implementations [10,18] and simulator models [9,10,11,19,20]. Huang et al. presents a thorough analysis of ring access delays for nodes using only one transit queue [11]. Robichaud et al presents ring access delays for class B traffic for both one- and two transit queue designs [19]. Gambiroza et al. focus on the operation of the RPR fairness algorithm and their alternative proposal, DVSR, and their ability, for some given load scenarios to converge to the fair division of rates according to their RIAS fairness reference model [10]. We are not aware of any results presented by others analyzing the suitability of the RPR technology used in a DiffServ setting.

IETF has created a working group named IP over Resilient Packet Rings (iporpr) chartered to investigate the problem area of *"developing the necessary standards for efficient interaction between L2 and L3"*. From their web site, it appears that the working group is currently inactive and there are no listed published Internet drafts or request for Comments.

10 Conclusion

In this paper we have evaluated the suitability of use of RPR in a DiffServ environment. We have discussed the fundamental mechanisms used in RPR to perform rate control according to given constraints, of which some are given by the network topology (link rates, propagation delays, number of nodes), some are statically configured (rate settings for traffic classes A and B) and some are configured dynamically (rate constraints for classes *B-EIR* and C). We have introduced a set of invariants which specify important parts of the RPR traffic class priorities and rate controls. A simple mapping between the RPR traffic classes and the standardized DiffServ PHB groups is also discussed and proposed. Based on the formal invariants and the mapping, an analytical model of a single RPR flow between two nodes was developed and analyzed. A set of more complex scenarios with 16 concurrent flows was evaluated using simulations of a 16-node ring with nodes using the *2TB* transit queue design and running the conservative fairness algorithm. The established invariants were also used to discuss the validity of the simulation results obtained and how the relative priority of the three traffic classes affects the per class throughput and delay under increasing load conditions. Finally the conformance between our implementation of the DiffServ PHBs and the DiffServ PHB requirements was discussed and found to be in partial conformance. The point where there was no conformance is outside the scope of the RPR standard however, and is left to the implementation of the MAC client.

11 Further Work

As discussed above, the relative drop priorities within an Assured Forwarding PHB class is not supported by the RPR MAC, and thus has to be implemented

in the MAC client. A study on the implementation of this mechanism, to allow for full conformance to the AF PHB requirements would be interesting. Also, a study of admission control methods applicable for RPR networks used in a DiffServ environment seems reasonable. Such a study can maybe be performed in cooperation with the IETF iporpr working group, once they resume their activities.

12 Acknowledgements

We would like to thank Sven-Arne Reinemo, Tor Skeie and Olav Lysne for providing helpful insights into the DiffServ problem area.

References

1. Davie, B.: Deployment Experience with Differentiated Services. In: Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS, ACM Press (2003) 131–136 1
2. Burgstahler, L., Dolzer, K., Hauser, C., Jahnert, J., Junghans, S., Macian, C., Payer, W.: Beyond technology: the missing pieces for qos success. In: Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS, ACM Press (2003) 121–130 1
3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services (1998) IETF, RFC 2475. 1, 3
4. El-Gendy, M.A., Bose, A., Shin, K.G.: Evolution of the Internet QoS and Support for Soft Real-Time Applications. Proceedings of the IEEE **91** (2003) 1
5. IEEE Computer Society: IEEE Std 802.17-2004 (2004) 1
6. : (OPNET Modeler. http://www.opnet.com) 1
7. Reames, C.C., Liu, M.T.: A loop network for simultaneous transmission of variable-length messages. In: Proceedings of the 2nd Annual Symposium on Computer Architecture. Volume 3. (1974) 2
8. Hafner, E., Nendal, Z., Tschanz, M.: A digital loop communication system. IEEE Transactions on Communications **22** (1974) 877 – 881 2, 2
9. Davik, F., Yilmaz, M., Gjessing, S., Uzun, N.: IEEE 802.17 Resilient Packet Ring Tutorial. IEEE Communications Magazine **42** (2004) 112–118 2, 2, 9
10. Gambiroza, V., Yuan, P., Balzano, L., Liu, Y., Sheafor, S., Knightly, E.: Design, analysis, and implementation of DVSR: a fair high-performance protocol for packet rings. IEEE/ACM Transactions on Networking **12** (2004) 85–102 2, 9
11. Huang, C., Peng, H., Yuan, F., Hawkins, J.: A steady state bound for resilient packet rings. In: Global Telecommunications Conference, (GLOBECOM '03). Volume 7., IEEE (2003) 4054–4058 2, 9
12. Davik, F., Gjessing, S.: The Stability of the Resilient Packet Ring Aggressive Fairness Algorithm. In: Proceedings of The 13th IEEE Workshop on Local and Metropolitan Area Networks. (2004) 17–22 2
13. Nichols, K., Blake, S., Baker, F., Black, D.: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (1998) IETF, RFC 2474. 3, 3, 5
14. Davie, B., Charny, A., Bennett, J., Benson, K., Boudec, J.L., Courtney, W., Davari, S., Firoiua, V., Stiliadis, D.: An Expedited Forwarding PHB (Per-Hop Behavior) (2002) IETF, RFC 3246. 3

15. Heinanen, J., Baker, F., Weiss, W., Wroclawski, J.: Assured Forwarding PHB Group (1999) IETF, RFC 2597. 3
16. Tamir, Y., Frazier, G.L.: High-performance multi-queue buffers for vlsi communications switches. In: Proceedings of the 15th Annual International Symposium on Computer architecture, IEEE Computer Society Press (1988) 343–354 4.2
17. Karol, M.J., Hluchyj, M.G., Morgan, S.P.: Input vs. output queueing on a space-division packet switch. IEEE Transactions on Communications **35** (1987) 1347 – 1356 4.2
18. Kirstadter, A., Hof, A., Meyer, W., Wolf, E.: Bandwidth-efficient resilience in metro networks - a fast network-processor-based rpr implementation. In: Proceedings of the 2004 Workshop on High Performance Switching and Routing, 2004. HPSR. (2004) 355 – 359 9
19. Robichaud, Y., Huang, C., Yang, J., Peng, H.: Access delay performance of resilient packet ring under bursty periodic class b traffic load. In: Proceedings of the 2004 IEEE International Conference on Communications. Volume 2. (2004) 1217 – 1221 9
20. Schuringa, J., Remsak, G., van As, H.R.: Cyclic queuing multiple access (CQMA) for RPR networks. In: Proceedings of the 7th European Conference on Networks & Optical Communications (NOC2002), Darmstadt, Germany (2002) 285 – 292 9