# Applying the DiffServ Model to a Resilient Packet Ring Network

February 2005
Revised August 2005

```
class point (x,y); real x,y;
   begin ref (point) procedure plus (P); ref (point) P;
                plus :- new point (x+P.x, y+P.y);
   end point;
```

Fredrik Davik Stein Gjessing

```
point class polar;
   begin real r,v;
         ref (polar) procedure plus (P); ref (point) P;
                plus :- new polar (x+P.x, y+P.y);
         r := sqrt (x↑2+y↑2);
         v := arctg (x,y)
  end polar;
```

[ simula . research laboratory ]

Research Report Simula 2005-01

Visiting address:
Martin Linges vei 17, Fornebu

P.O. Box 134
NO-1325 Lysaker, Norway

Telephone: +47 67 82 82 00
Telefax: +47 67 82 82 01

www.simula.no

# Applying the DiffServ Model to a Resilient Packet Ring Network

**February 2005**
**Revised August 2005**

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 new standard called Resilient Packet Ring (RPR), that is maintained in the 802 LAN/MAN Committee and designated standard number IEEE 802.17-2004. 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 simulation using some selected scenarios. 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 (RPR, IEEE 802.17), Differentiated Services (DiffServ), 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 the provisioning of 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 evaluation of our proposed mapping by use of three simulation scenarios. Then, in section 8, we asses the mapping based on both the analytic and simulation results. Finally, in sections 9, 10 and 11, we present related work, conclude and point out directions for future work.

## 2 The Resilient Packet Ring Architecture

RPR 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 there are no data packets in transit. This simple insertion buffer principle, however, must be extended considerably to constitute a full fledged Medium Access Control protocol providing fairness, service class differentiation and protection [9].

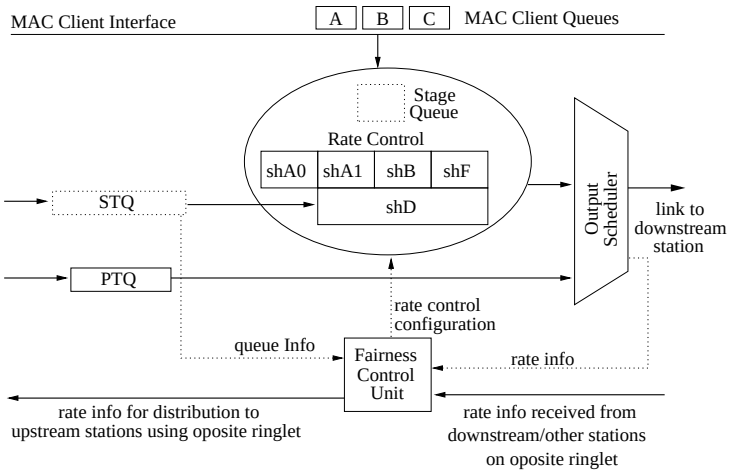


*Fig. 1: RPR node design, showing a node's attachment to the ring for transfer of data in the East direction and control information in the West direction. The solid lines indicates the flow of data through the node. The dotted lines indicates the exchange of control/configuration information between node internal function blocks.*

Fig. 1 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 control for the different service classes. Additionally, the *Stage Queue* can be considered a logical[5] queueing element, providing temporary storage for a packet received from the MAC client immediately before its transmission onto the ring.

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

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

[5] The RPR standard does not mandate the use of a stage queue, neither does it specify any requirements regarding its implementation.

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. Class $B$ traffic transmitted in excess of the allocated portion (the *B-CIR* (Committed Information Rate) portion), is sent on the expense of class $C$ traffic. The excess portion is denoted *B-EIR* (Excess Information Rate). 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[6]), 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

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

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.

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 in section 4.1 by providing 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 that section, we present our set of invariants that constitutes a specification of the differentiation between the RPR service classes.

Then, in section 4.2, we provide a brief introduction to RPR delay guarantees for high-priority traffic.

### 4.1 Rate Control

The MAC layer provides, via the rate control block (see Fig. 1), 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.

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 shaper *shF* is performed dynamically, based on rate measurements performed at the output of the *Output Scheduler* (see Fig. 1) 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.

Below to simplify the presentation form, we introduce some notation to be used for the reminder of this paper.

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.

In (3) and (4) below, we define reserved (preallocated) and unreserved (reclaimable) bandwidth on an RPR ring, denoted respectively $R_R$ and $R_U$. The bandwidth of the link is denoted $R_L$.

$$R_R \triangleq \sum_{j \in \{nodes\}} R_{A0_j} \tag{3}$$

$$R_U \triangleq R_L - R_R \tag{4}$$

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 specified 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 restrictions shown in invariants 3-8 effectively throttles the amount of $A0$ and $A1$ add traffic a node can add to the ring (to a configured value). The parts of the invariants relating to $A0$ traffic is enforced solely by the *shA0* shaper shown in Fig. 1. For the parts of the invariants related to $A1$ traffic, this is enforced by the union of the *shA1* and *shD* settings.

One interesting observation is that when both A0 and A1 traffic is supported (i.e. $R_{A0} > 0 \wedge R_{A1} > 0$), it is impossible to predict the portion of offered traffic that is accepted as respectively A0 or A1 traffic. This also applies on a per packet basis. Thus, when the *client* delivers a class A packet to the MAC layer, it does not know, whether the packet will be transferred as a A0 or A1 packet.

**invariant 3** $\forall nodes : \mathit{offered(A)} < R_{A0} \wedge R_{A1} = 0 \Rightarrow \mathit{accepted(A0)} = \mathit{offered(A)}$

**invariant 4** $\forall nodes : \mathit{offered(A)} < R_{A1} \wedge R_{A0} = 0 \Rightarrow \mathit{accepted(A1)} = \mathit{offered(A)}$

**invariant 5** $\forall nodes : \mathit{offered(A)} \geq R_{A0} \wedge R_{A1} = 0 \Rightarrow \mathit{accepted(A0)} = R_{A0}$

**invariant 6** $\forall nodes : \mathit{offered(A)} \geq R_{A1} \wedge R_{A0} = 0 \Rightarrow \mathit{accepted(A1)} = R_{A1}$

**invariant 7**
$$\forall nodes : R_{A0} > 0 \wedge R_{A1} > 0 \wedge \mathit{offered(A)} < R_{A0} + R_{A1} \Rightarrow \mathit{accepted(A0)} + \mathit{accepted(A1)} = \mathit{offered(A)}$$

**invariant 8**
$$\forall nodes : R_{A0} > 0 \wedge R_{A1} > 0 \wedge \mathit{offered(A)} \geq R_{A0} + R_{A1} \Rightarrow \mathit{accepted(A0)} = R_{A0} \wedge \mathit{accepted(A1)} = R_{A1}$$

In invariants 1-8, 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 9-12, it is more relevant to present per-link invariants, in order to express the relative priority of the RPR service classes.

The *B-CIR* portion of the accepted class $B$ traffic, is restricted by the union of *shB* and *shD* shapers. The *B-EIR* portion as well as the amount of accepted class $C$ traffic is controlled by union of *shF* and *shD* shapers. The rate setting of the *shF* shaper is controlled by the fairness algorithm.

In invariants 9 and 11, 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 9** $\forall links : \mathit{offered(B)} + \mathit{accepted(A1)} \leq R_U \Rightarrow \mathit{accepted(B)} = \mathit{offered(B)}$

In invariants 10 and 12, 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 10** $\forall links : \mathit{offered(B)} + \mathit{accepted(A1)} > R_U \Rightarrow \mathit{accepted(B)} = R_U - \mathit{accepted(A1)}$

**invariant 11** $\forall links : \mathit{offered(C)} + \mathit{accepted(A1)} + \mathit{accepted(B)} \leq R_U \Rightarrow \mathit{accepted(C)} = \mathit{offered(C)}$

**invariant 12** $\forall links : \mathit{offered(C)} + \mathit{accepted(A1)} + \mathit{accepted(B)} > R_U \Rightarrow \mathit{accepted(C)} = R_U - \mathit{accepted(A1)} - \mathit{accepted(B)}$

### 4.2 Delay Properties

The delay of class $A$ traffic consists 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 preemption of lower priority packets.

Access delay is measured from a packet is placed at head of line in the *client* queue 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, fairness and (high priority) control packets as well as transmission of packets from the $PTQ$. Thus when within its rate bounds, in the worst case, a class A packet can risk waiting several packet transmission times before being scheduled for transmission on the output link. Given a ring with $N$ nodes, utilizing shortest path routing of packets, in the worst case, we can receive a packet train consisting of N/2 or N back-to-back MTU-sized class $A$ packets. This observation builds on the assumption that upstream nodes are configured so that they are not able to transmit more than one (for a node supporting A0 traffic only) or two (for a node supporting both A0 and A1 traffic) back-to-back maximum sized class A packets and that they are not able to accumulate enough credits to insert more class $A$ packet into the packet train. For the simplicity of the analysis, we also disregard the sending of control traffic.

Thus, based on the above assumptions and with $L_{MTU}$ representing the bit size of a MTU sized packet and $R_L$ representing the link rate, the worst case access delay, $D_a$, is given by (5) below. On average, with an offered load larger than the available capacity, the average access delay time for a class $A$ packet

will be close to the time required to accumulate a sufficient amount of credits for the transmission of the packet.

$$D_a = \begin{cases} \frac{L_{MTU}}{R_{A0}} + \frac{N}{2} \cdot \frac{L_{MTU}}{R_L}, R_{A1} = 0 \wedge R_{A0} > 0 \\ \frac{L_{MTU}}{R_{A1}} + \frac{N}{2} \cdot \frac{L_{MTU}}{R_L}, R_{A1} > 0 \wedge R_{A0} = 0 \\ \frac{L_{MTU}}{R_{A1}} + N \cdot \frac{L_{MTU}}{R_L}, R_{A1} \geq R_{A0} > 0 \\ \frac{L_{MTU}}{R_{A0}} + N \cdot \frac{L_{MTU}}{R_L}, R_{A0} > R_{A1} > 0 \end{cases} \quad (5)$$

In (5), the first and second cases represents a scenario where only A0 or A1 traffic is supported in the network. Using the above assumptions, the worst case access delay will be when a packet arrives at head of line in the *client* queue right after the previous packet has been accepted by the MAC layer. Thus in the worst case, following the transmission of the previous packet, the A0 or A1 shaper must accumulate an amount of credits equal to the packet size. Then, once the accumulation of credits has finished, upon arrival to the MAC layer, the transmission has to await the transmission of a transiting packet train of class A traffic consisting of $N/2$ packets. That is, all upstream nodes contributes one class A0 or A1 packet to the packet train.

In (5), in the third and fourth cases, both A0 and A1 traffic is supported in the network. In this case, the worst case time for the accumulation of credits for the sending of a packet equals the maximum packet size divided by the highest service rate In this case however, a station is able to send two maximum sized packets back-to-back. As a result of this, the worst case length of a transiting packet train is the double of the cases where only A0 or A1 traffic is supported.

## 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 onto RPR service classes $A$, $B$ and $C$. Given the the strict priority scheduling rules presented in section 4.2 and the various invariants used to maintain the per-service class rate configuration presented in section 4.1, 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 priorities in the *client* should be a relatively easy task, using some form of priority queueing scheduling algorithm. Thus achieving 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, in section 8, 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 Fig. 2, we have shown the analytical resulting throughput when mapping the EF-, AF and default PHBs to RPR's service classes $A$, $B$ and $C$.

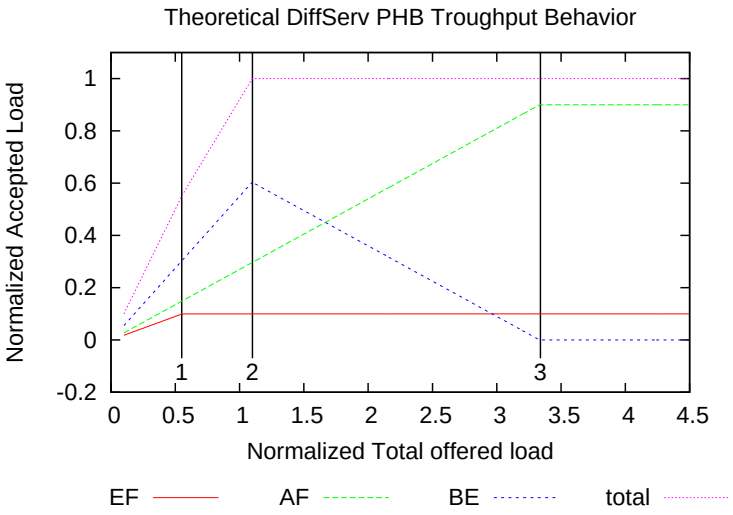


*Fig. 2: Theoretical Throughput for the EF-, AF and default PHBs when mapped to RPR service classes $A$, $B$ and $C$.*

The analytical example uses the invariants introduced in section 4.1 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.

Three points for offered load in this plot are of particular interest. As long as we are on the left side of point 1, the aggregate EF traffic can be increased further while maintaining invariant 3 or 7. At point 1, we have reached the load level where an additional load increase causes the premise of both invariants 3 and 7 to be false. Beyond this point, the maintenance of invariant 8 effectively prohibits the acceptance of additional EF traffic. On the left side of point 2, as long as invariant 11 is maintained, an increase in offered BE traffic leads to an equivalent increase in accepted BE traffic. At point 2, we have reached the load level where an additional load increase causes the invariant's premise to be false. Beyond this point, the maintenance of invariant 12 effectively prohibits the acceptance of more BE traffic. Finally, between points 2 and 3, as long as invariants 9 and 12 are maintained, an increase in offered AF traffic leads to an equivalent increase in accepted AF traffic, on the expense of an equivalent decrease in accepted BE traffic. At point 3, we have reached the load level where an additional load increase causes the premise of invariant 9 to be false. Beyond this point, the maintenance of invariants 10 and 12 effectively prohibits the acceptance of additional AF traffic as well as excluding BE traffic from the link.

## 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 section, we use our RPR discrete event simulator model implemented in the OPNET Modeler environment to perform the same evaluation by use of some simulation scenarios. It is impossible to verify the behavior of RPR by simulation for all conceivable combinations of traffic, node configurations and ring configurations. So in this section, we analyze three se-

lected scenarios that we believe are of importance for the evaluation of RPR's performance when used in a DiffServ context, as well as in a general context, where an RPR network is used to provide traffic differentiation.

In particular, we want to show that for the scenarios evaluated, delay guarantees for high-priority traffic (RPR class A/DiffServ EF) are kept, as well as throughput guarantees for medium priority traffic (RPR class B/DiffServ AF). And lastly, but not least, we want to show that traffic of the BE PHB is able to utilize the bandwidth not used by the EF and AF PHBs.

The discussion of conformance to the DiffServ PHB requirements, is postponed to section 8, where we will discuss the conformance, based on results obtained from the full set of simulation scenarios.

In all the simulated scenarios, we evaluate the delay (for EF traffic only) and throughput performance of an RPR ring as a function of increasing load. For all experiments where we calculate confidence intervals for the delay, we use the median as the estimator for the 90th and 99th percentiles. We have not presented delay results for the AF and BE PHBs, as there are no associated DiffServ delay compliance requirements for these PHBs. Further, as traffic on the two ringlets in an RPR network is handled independently, we evaluate traffic flowing in one direction only.

The load offered is DiffServ EF, AF and BE traffic, using the mapping between DiffServ's PHBs and RPR service classes as proposed in section 5.

In the cases where we use self-similar packet sources, these are implemented based on a model outlined by Mondragón et al., using chaotic maps [18]. The implementation is done by by Blomskøld and Nilsen and is described in [19].

For the reminder of the experimental part, we start in section 7.1, with the evaluation of delay and throughput performance for a so-called "Hot Receiver" or "Hub" scenario. This may be the case when an RPR ring is used as an access ring for the aggregation of traffic, with one node (the "Hot Receiver" or "Hub") on the ring providing access to the backbone.

Then, in section 7.2, we evaluate a scenario constructed to resemble an RPR ring used as a backbone ring.

Finally, in section 7.3, we perform an extensive series of tests for a set of randomly selected sender-receiver configurations.

### 7.1 Hot Receiver Scenario

In this experiment, we want to evaluate a configuration, where the majority of traffic on a ring(let) is destined for one particular node, a so-called "Hot-Receiver" or "Hub". This is typically the case in an access or metro-ring where we have one gateway, connecting the ring to a metro-ring or the backbone.

The scenario we use is shown in Fig. 3, where nodes 1-26 all send traffic to node 30. These nodes all send traffic of the DiffServ AF and BE PHBs, mapped onto RPR service classes $B$ and $C$ as proposed in section 5.

To make the scenario more realistic, we have added self-similar background traffic, entering the ring at node 0 and traversing the whole segment under observation. We model traffic flowing in one direction only. In a hot-receiver

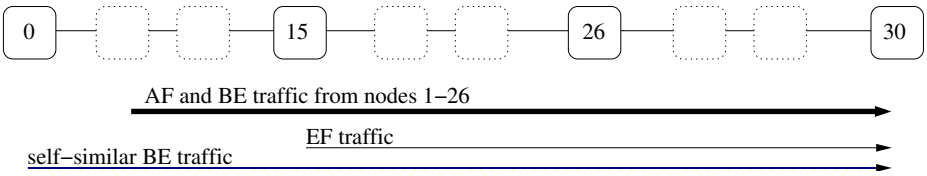


*Fig. 3: A hot receiver scenario for a 64 node ring, with nodes 0-26 sending data to node 30. Node 0 sends self-similar BE traffic only. Nodes 1-26 all send AF and BE traffic. Node 15 also sends EF traffic.*

scenario, in the reverse-direction, we will have a one-to-many configuration of flows, limited effectively by the hot-receiver's immediate downstream link as the load increases. Thus the likelihood of congestion in the reverse direction is much smaller than in the forward direction.

For the interested reader, Appendix A shows a detailed overview of the parameter settings used for our experiment.

We run a set of simulations for a series of load-points, and present a subset of the obtained data (i.e. throughput and delay as a function of offered load). The relative division of offered traffic of the EF, AF and BE traffic classes is approximately 2/40/60 (EF/AF/BE percentages of aggregate offered load not considering the contribution from the self-similar traffic source). Given the base load settings as given in Appendix A, as the relative offered load increases, the network runs relatively rapidly into the congested area of operation, while it takes somewhat longer (e.g. a higher relative load value) before the amount of accepted EF traffic reaches its maximum value.

We model the traffic from nodes 1-26 so that it has as relatively fixed (CBR-like) communications demand on the ring that is a function of the relative load value. That is, for a simulation run, the expected average rate equals the base load times the relative load parameter. To introduce some variability however, the inter-arrival times of packets entering a node are drawn independently according to a Poisson distribution. To provoke worst-case jitter for the EF traffic, we let all nodes send traffic to node 30.

By experience, we know that for a reasonable mix of traffic of the various priorities, the access-delay part for high-priority traffic contributes heavily to its jitter values. Thus, we use a probe flow of high-priority traffic, which is inserted at a node, located in the middle of the congestion domain and that traverses all downstream nodes in the congestion domain. By this, we ensure that at the ingress point of the high-priority traffic, there will be a significant portion of transit traffic, competing for the capacity of the node's out-link. Furthermore, we have a significant share of AF traffic, not constrained by the RPR fairness algorithm. Finally, at the tail of the congestion domain (node 0), we have a self-similar source sending low-priority traffic, ensuring large variations in demand, especially at low values of aggregate load from the CBR-like sources. At higher loads, the significance of the self-similar traffic is less, as it is strongly throttled by the RPR fairness algorithm.

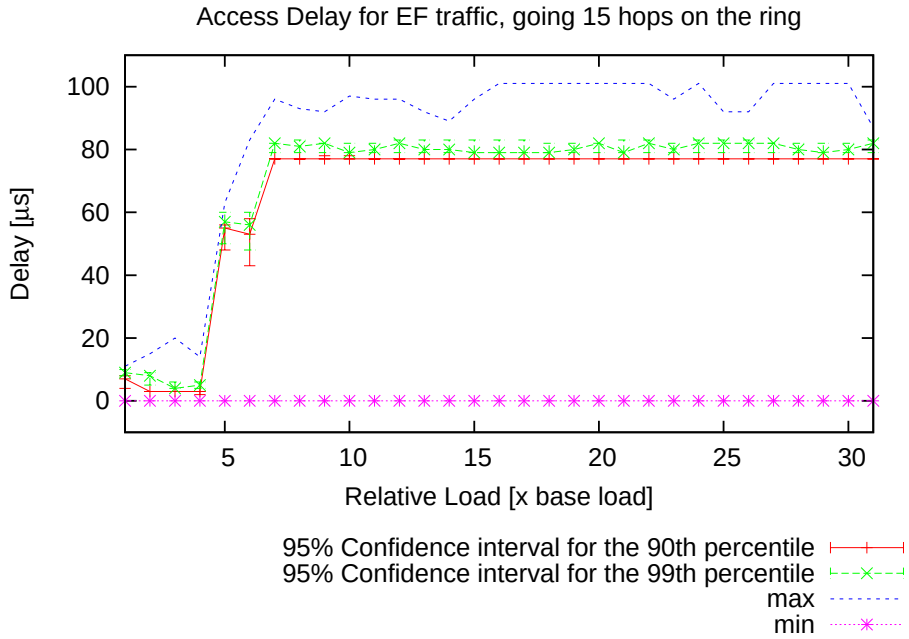


*Fig. 4: Access Delay for EF traffic going 15 hops on the ring for the "Hot Receiver" scenario. The worst-case jitter is 101[$\mu s$]. As the offered load increases, so does the delay. This is because packets must wait longer at the head of the client queue before a sufficient amount of A0 or A1 credits are available for the sending of the packet. Note that from the point where the relative load equals 10, the 95% confidence interval for the 90th percentile is a straight line. This is because the lower 90% of the delay measurement does not change significantly. For the upper 10% of the measurements, there are some minor differences, leading to some variations for the 95% confidence interval for the 99th percentile. Due to some bursts of high-priority control data related to conservative mode delay measurements from node 15, the maximum delay values are somewhat higher than the 99th percentiles.*

The delay measurements of EF traffic inserted at node 15, destined for node 30 are shown in Fig. 4. The figure shows maximum and minimum delays as well as 95% confidence intervals for the 90th and the 99th percentile.

In this scenario, we are serving EF traffic at a rate $R = 100$[Mbit/s], assigning $R/2$ to respectively A0 and A1 traffic, then, according to (5), the worst case analytical access delay is $\frac{L_{MTU}}{R_{A1}} + N \cdot \frac{L_{MTU}}{R_L}, R_{A1} \geq R_{A0}$. In this experiment however, except for high priority control traffic related to the RPR conservative fairness mode from upstream nodes, upstream nodes do not send neither class A nor EF traffic. Thus in this case, N is set to 2, and the worst case analytical access delay becomes $\frac{L_{MTU}}{R_{A1}} = \frac{500 \cdot 8}{50 \cdot 10^6} + 2 \cdot \frac{500 \cdot 8}{10^9} = 88[\mu s]$.

As seen in the figure, the maximum jitter is 101[$\mu$s], which is slightly above the analytical worst-case expression provided in (5). This is caused by some bursts of high-priority control data related to conservative mode delay measurements from node 15. The delay incurred on the the high-priority traffic however, can be reduced by setting the rate restriction for control traffic to a lower value (i.e. limit the burstiness of the control traffic).

The 95% confidence intervals for the 90th and 99th percentiles for the delay, for high values of offered load, are centered right below and above 80[$\mu$s] respectively.

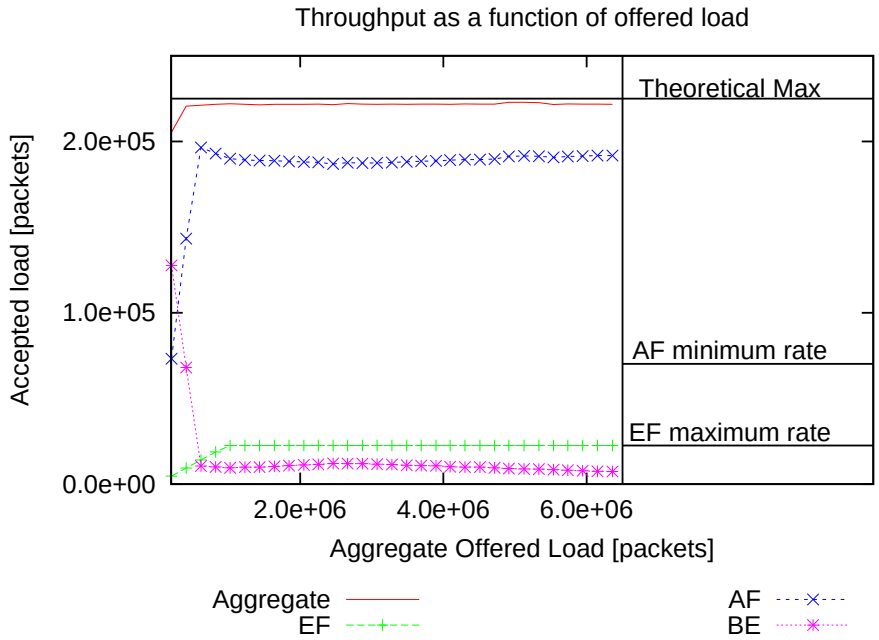


*Fig. 5: Accepted load as a function of offered load for the EF, AF and BE PHBs for the "Hot Receiver" scenario. On the right-hand side of the plot, the configured EF maximum and AF minimum rates are shown as well as the theoretical maximum for the given configuration and topology. The line labelled "AF Minimum rate" represents the minimum ordinate value that must be reached for accepted AF traffic as the amount of offered AF traffic reaches or exceeds the same value. As seen, the amount of AF traffic exceeds by far this level as the offered load increases, thus AF throughput guarantees are met.*

Throughput results for the EF, AF and BE PHBs are shown in Fig. 5. The figure presents the throughput as the aggregate number of packets accepted as well as the number of packet accepted per PHB as a function of the aggregate offered load. That is, for the range of relative load values presented in Appendix

A, for each value of the relative load parameter, the expected average offered load for each node equals the relative load parameter multiplied with the base load per PHB. The points plotted for accepted load are mean values, obtained from 12 independent simulation runs. The "noise" on the curves, is mainly caused by the self-similar traffic source, which for the given configuration, has an expected average throughput which equals the relative load times the base load. For different seed values however, the short term throughput of this packet generator will vary greatly. Thus introducing some "noise" on measured BE (and AF) traffic.

As seen in the figure, the amount of accepted EF traffic equals the amount of offered EF traffic as long as the amount of EF traffic is below the configured maximum. When the amount of offered EF traffic exceeds the configured maximum, the amount of accepted EF traffic is limited to the configured maximum (as given by the configuration in Appendix A).

For AF traffic, it is clear that the amount of AF traffic is not restricted to the value given by the configured AF rate given by the configuration in Appendix A. This is however not the intention. As specified by the RPR standard, AF traffic (when mapped to RPR service class B) has precedence[7] over BE traffic (when mapped to RPR service class C). Thus, as specified by invariant 12, when the sum of AF and BE traffic over a link is larger than the link can support, the BE traffic will only receive the remaining portion when the AF demand is satisfied. This is clearly shown in Fig. 5. As the aggregate offered load increases, the aggregate demand of AF traffic is satisfied on the expense of BE traffic. For high values of offered load, the small fraction of accepted BE traffic is most likely caused by BE traffic transmitted during the initial transient period, where the fairness algorithm has not yet converged.

### 7.2 Backbone/Metro Ring

In this experiment, we will consider a scenario where an RPR ring is operating as a large metro or backbone network. In this scenario, each RPR node on the ring serves as a ring ingress/egress point, handling traffic of the three PHBs EF, AF and BE.

In this scenario, illustrated in Fig. 6, let us assume that the EF and AF PHBs are used as a service for providing low-latency (EF only) and guaranteed throughput (both EF and AF) for communication between fixed node-pairs on the ring. This may be the case where a company subscribes to a gold and silver service from a service provider, to enable real-time (e.g. voice and video conferencing) as well as high-quality multimedia-traffic between two (or more) geographically distributed locations.

We model the EF traffic as traffic with a relatively fixed (CBR-like) communications demand on the ring, but to introduce some variability, the inter-arrival times of EF packets entering a node is drawn independently according to a Pois-

[7] On the ring ingress point.

son distribution. To provoke worst-case jitter for the EF traffic, we let all nodes send EF traffic transiting node 24..

The AF traffic is expected to be more bursty (e.g. variable rate multimedia traffic) and is modelled as self-similar traffic. The destination for AF traffic is chosen randomly according to a uniform distribution at simulation start-up, so that that the hop-count of the per node AF traffic will be in the range $\langle min, max \rangle$ (these are configuration settings specified in Appendix B). Once the destination is chosen, it remains fixed for the reminder of the simulation run. Note that for all destinations drawn, the traffic is sent on ringlet 0.

For BE traffic, we do not expect a fixed communication pattern on the ring, rather it is reasonable to assume a long-term behavior, where for traffic entering a ring node, the egress point of this traffic will be distributed in a uniformly manner between the various egress points on the ring. In the short term, we will typically have a behavior termed "source locality" [20], where, if we observe an arbitrary packet on the ring, there is a very high probability that the next packet on the ring will have the same source and destination address. Thus for a burst of packets entering the ring from an ingress point, a number of the packets in the burst is expected to have the same ring egress point.

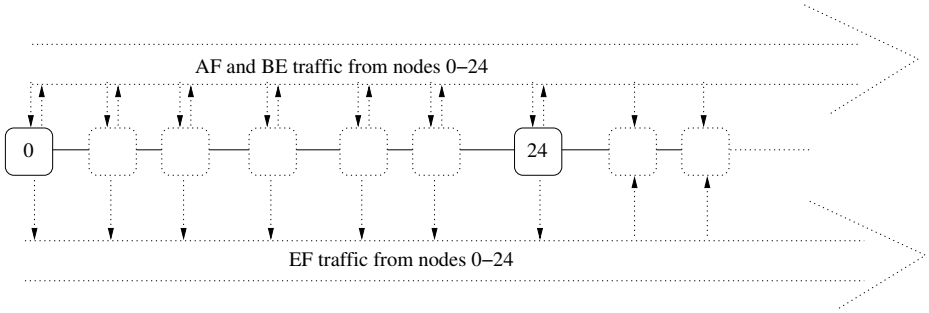


*Fig. 6: A "Backbone/Metro Ring" scenario for a 64 node ring, with nodes 0-24 sending data to various nodes on ringlet 0 (i.e. the ringlet with data flowing to the right in the figure). For each simulation run, All EF traffic is sent to random destinations located beyond node 24. All AF traffic is sent a minimum number of hops (according to the configuration parameters), to a destination selected according to a uniform distribution at simulation startup. For BE traffic, the destination of traffic is selected so that the traffic is sent a minimum number of hops. The destination is selected according to a uniform distribution at the start of each on-period of the self-similar generator.*

Just as for AF traffic, we also expect the BE traffic to be self-similar, with great variations in throughput, both on a short time-scale and a long time-scale. To obtain the "source locality" behavior of the BE traffic, the destination address is kept fixed for the duration of an on-period of the self-similar packet source, before we, at the end of the on-period, draw the destination address to use for

the next packet burst uniformly (in the same manner as is done at simulation startup for the AF traffic).

To provoke worst case access delays for EF traffic, we use a traffic mix with a relatively high maximum share of EF traffic (10% of the line rate). For the base load used, we have a 10/20/70 division for the offered traffic of the EF/AF/BE PHBs.

For the interested reader, Appendix B shows a detailed overview of the parameter settings used for our experiment.

If we start by evaluating the delay performance of the EF traffic, we find the worst case jitter for EF traffic entering the ring at node 10. As is shown in Fig. 7 (and with an excerpt showing the details for high values of offered load in Fig. 8), as the offered load increases, so does the access delay of the EF traffic. But as expected, as the offered load increases, the maximum delay stabilizes at an upper bound, close to time it takes to accumulate an amount of credits, sufficient for the transmission of a maximum sized package.

The results obtained fall within the upper bound provided by the worst-case analytical expression given in (5). In this expression, for a network supporting both A0 and A1 traffic, the worst case access delay is specified as $\frac{L_{MTU}}{R_{A1}} + N \cdot \frac{L_{MTU}}{R_L}, R_{A1} \geq R_{A0}$. In this experiment, with a 64 node ring, $N = 64$, however, as there are only 25 active nodes, the value to use for $N$ is 50. Thus we get the worst case analytical access delay: $D_a = \frac{500 \cdot 8}{2 \cdot 10^6} + 50 \cdot \frac{500 \cdot 8}{10^9} = 2.2ms$

In the results obtained, the worst case access delay is measured to 2.121ms, a value well within the predicted worst case.

If we consider the throughput results shown in Fig. 9, the behavior is as expected. As the offered EF traffic is increased, as given by invariant 7, the accepted EF traffic increases in an equivalent fashion as long as the offered load stays below the (class A0 and class A1) rate constraints in effect. This is shown in the region in the plot to the left of the vertical line labelled 1. Then, to the right of this line, the amount of offered EF traffic exceed the rate constraints specified for A0 and A1 traffic, thus as specified in invariant 8, no additional amounts of EF traffic is accepted.

Similarly, for BE traffic, as specified in invariant 11, as long as there are no congested links on the ring, the amount of accepted BE traffic increases linearly with the offered amount of BE traffic. However, as links starts to become congested, as specified in invariant 12, the BE traffic gets the remaining bandwidth not used by EF and AF traffic. The vertical line labelled 2 marks the point where, to accommodate an increase in demand for AF capacity, the amount of BE traffic accepted must be correspondingly reduced.

Finally, the vertical line labelled 3 marks the point where the ring starts to become saturated. To maintain invariant 10 as the offered AF traffic increases, the ring is no longer able to support a linear growth rate of accepted AF traffic.

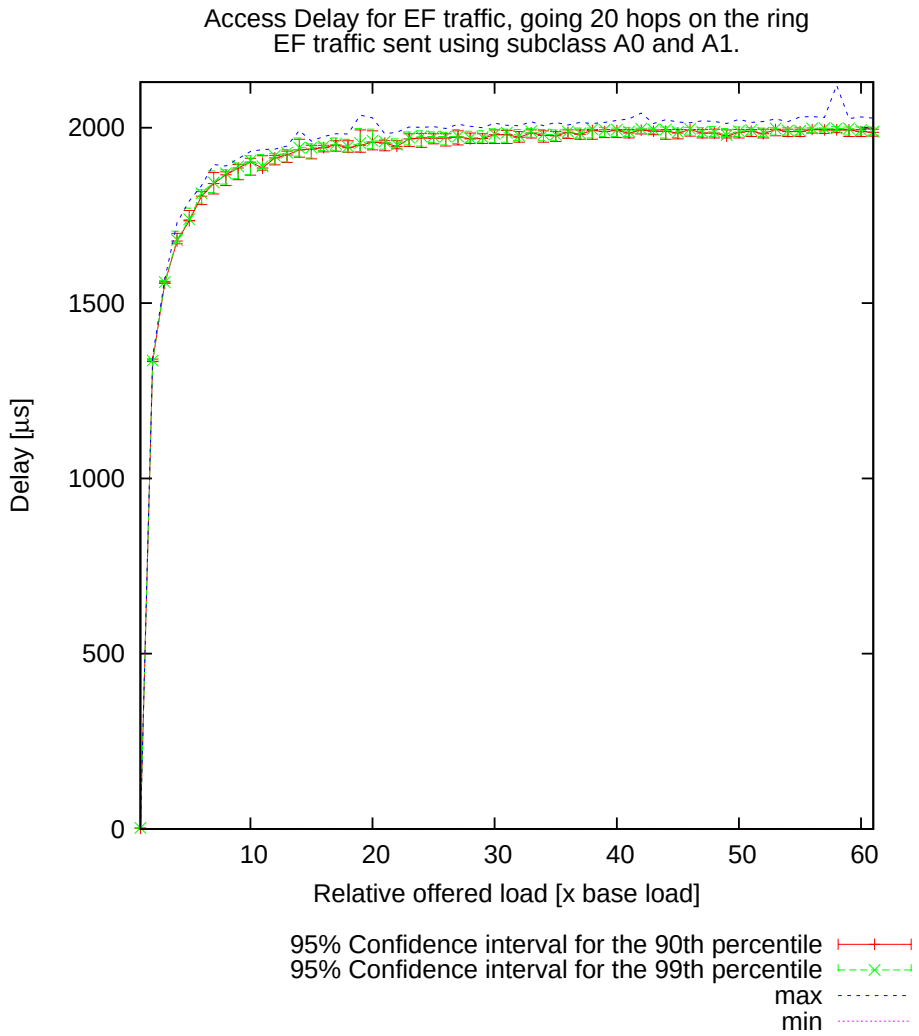


*Fig. 7: Worst case Access delay for the worst case "Backbone/Metro Ring" scenario is is found for EF traffic entering the ring at node 10. The worst-case jitter is found for the value of 57 for the relative offered load, and is* $2121[\mu s]$. *As is seen from the plot, when the offered load increases, the 90 and 99th percentiles for the access delay converges towards the time it takes to accumulate credits for the transmission of an EF packet, using subclass A0 or A1. The maximum delay values are caused by delayed access to the output link, caused mainly by a train of EF packets from upstream nodes transiting the node. However, the sending of control traffic also contributes to the deviation. Fig. 8 contains an excerpt of these results, showing more details for the delay results for high values of offered load.*

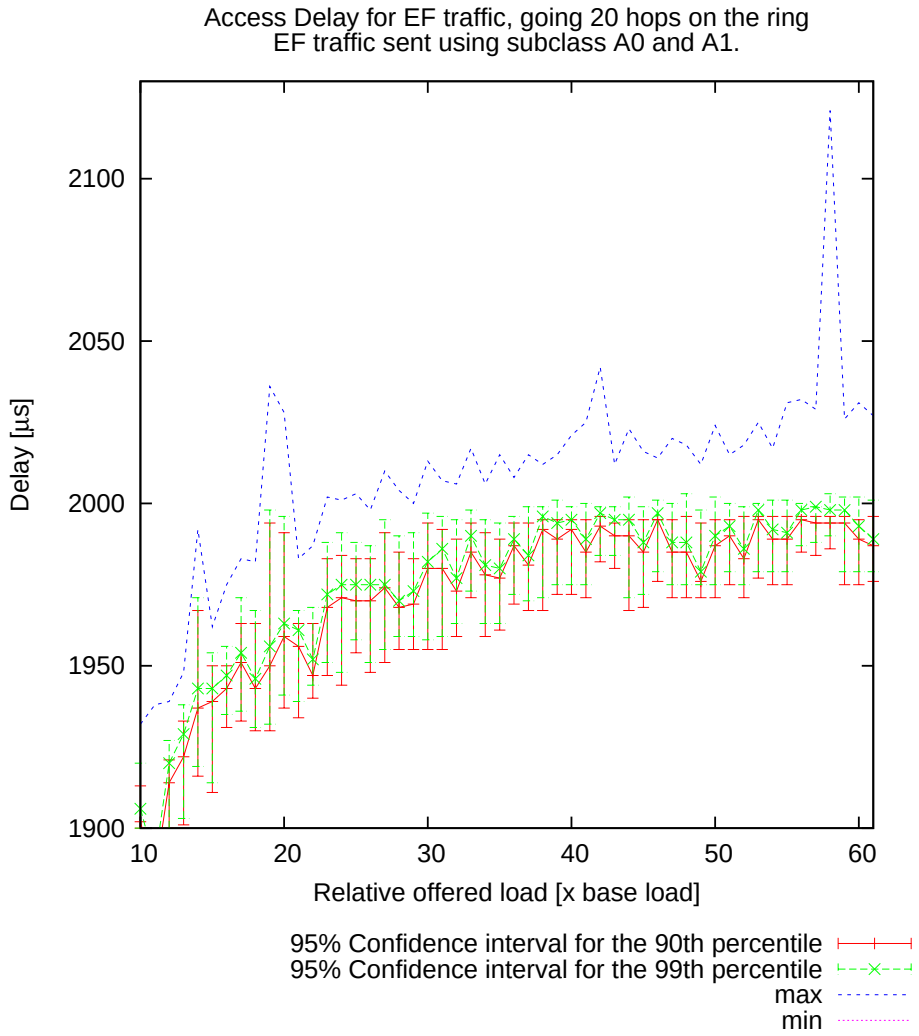


*Fig. 8: Excerpt of results shown in Fig. 7 for the "Backbone/Metro Ring" scenario. As seen, the 95% confidence intervals for the 90th and 99th percentiles are rather close. The maximum delay value at a value of 57 for relative offered load, is caused by a number of back-to-back EF packets transmitted by upstream nodes at simulation startup in addition to a burst of control packets.*

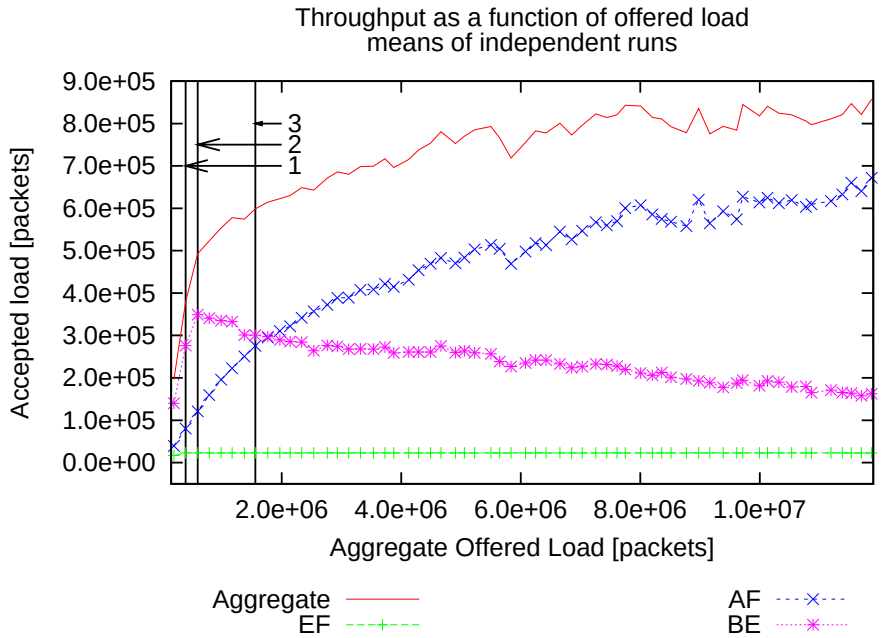


*Fig. 9: Average throughput as a function of offered load for traffic of the three PHBs EF, AF and BE for the "Backbone/Metro Ring" scenario. The figure also shows the aggregate throughput. At point 1, the aggregate of accepted EF traffic has reached its maximum value. At point 2, some links start to become congested, thus to accept more AF traffic, the amount of BE traffic accepted must be reduced. Finally, at point 3, the ring starts to become saturated as more links become congested, thus the increase of accepted AF traffic is no longer a linear function with respect to the offered load. The "noise" on the curves is caused by the self-similar traffic generators which, depending on the seed value of the simulation run (and otherwise equal parameter settings), will produce an average amount of offered traffic which varies.*

### 7.3 Random Fixed Pairs Communication

In the "Hot Receiver" and "Backbone/Metro Ring" scenarios, described in sections 7.1 and 7.2, we measured the time taken from a packet was available at the head of the *client* queue, until it was scheduled for transmission on the output link. This performance metric is essential in order to claim conformance to the DiffServ EF requirements.

However, given that an EF packet is transmitted onto the ring, it is also important that it is not delayed significantly on its transmit path from its ring ingress to its ring egress point.

In this experiment, we want to measure the transit path delay, starting the measurement as a packet is accepted by the MAC layer at the ingress point (i.e. put into the stage queue), and stopping the measurement as the packet is stripped from the ring at its egress point.

To ensure that there will be a competition for network resources, we use a scenario where for a ring of 16 nodes, we let each node transmit traffic to one other node on the ring. Additionally, we let each node receive traffic from one other node on the ring. Further, a node is not allowed to send traffic to itself. Thus we get a configuration with 16 pairs of communicating nodes. The communicating pairs are are randomly selected on beforehand, and 16 different configurations are generated.

For each simulation an offered load parameter is given. 20 load values are used, linearly distributed on a scale from 1 to 10 times the base load. In order to calculate confidence intervals for the delay, 12 independent simulation runs are performed per load-point for each configuration.

The base load uses the same 55/27/18 ratio for offered load of the EF/AF/BE PHBs as is used in section 6. The traffic is modelled in the same CBR-like fashion as is used in the two previous scenarios, where the inter-arrival time of packets are drawn independently according to a Poisson distribution. Thus resulting in an expected average throughput per relative load value, with some variation in the arrival time of the packets at the ingress nodes.

For the interested reader, detailed configuration information for this scenario can be found in Appendix C.

The delay measurements for this scenario are shown in Fig. 10. In the plot, we observe that the variations in delay values are relatively insignificant compared to their actual values. The worst case jitter is 33[$\mu$s], which constitutes 1.85% of the minimum delay value for a distance of 7 hops on the ring. One interesting observation is that with a lower offered load, the delay values for the 90th and 99th percentiles are slightly higher than for higher offered loads. We believe this is because at low values of offered loads, there is a larger chance that a congestion domain has not been established on the ring. Without one (or several) assigned node(s) (i.e. congestion domain head) working to constrain the amount of BE traffic accepted onto the ring, there is a greater chance for an EF packet at the ingress point or in the transit path having to await the completion of a BE packet being transmitted. However, the majority of variations are relatively small, as can be observed by the size and locations of the confidence intervals.

This confirms the expected behavior for EF delay, which is expected to be limited upwards, as the load increases. Both in terms of the access delay at the ingress point (with the analytical worst case expression provided in (5)), as well as the delay incurred on the packets as they propagates on their path between their ingress and egress points on the ring.

Fig. 11 plots the throughput for the three PHBs EF, AF and BE, as well as the aggregate of all PHBs. For each value of offered load, each point on the curves is the mean taken across all the 16 sender-receiver configurations and their independent simulation runs.

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 Fig. 2. The general trend however follows the theoretical pattern, given by the invariants specified in section 4.1. Thus, as the offered load increases, the increase in accepted EF traffic is linear up to some point and then remains 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 $\sim 0.8 \cdot 10^6$ (marked with the vertical line labelled 1), 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 8, thus no further increases in accepted EF traffic is allowed. For BE traffic, at an offered-load of $\sim 10^6$ (marked with the vertical line labelled 2), some links in the simulated sender-receiver pair configurations have become congested, resulting in a stop in the increase of accepted BE traffic (to give room for the increased amount of AF traffic and hence maintain invariant 12). From this point onwards, sufficiently many links have become congested, resulting in a clear decrease in the accepted amount of BE traffic while maintaining a linear increase in the accepted amount AF traffic.

As the offered load exceeds a level of $\sim 2.5 \cdot 10^6$ (marked with the vertical line labelled 3), to maintain invariant 10 as more links utilize all their capacity for the sending of EF and AF traffic, the growth rate of accepted AF traffic starts to decrease slowly towards 0.

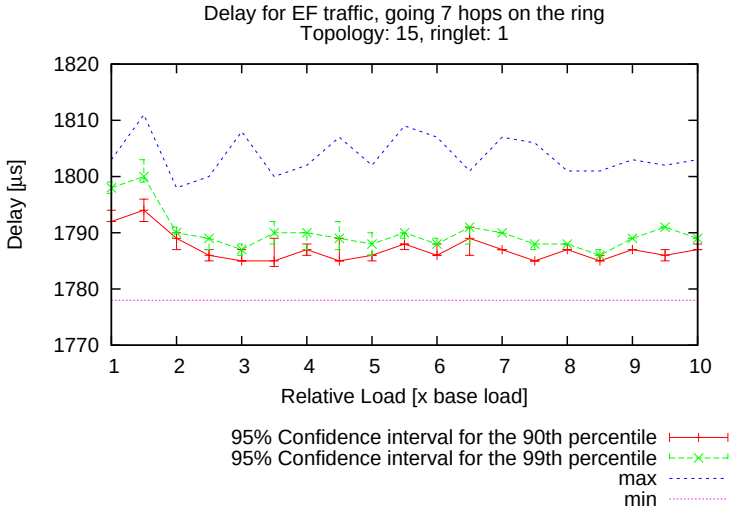


*Fig. 10: Delay measurements for the "Random Pairs" scenario. The plot shows max and min delays as well as the 95% confidence intervals for the 90th and 99th percentiles. Worst case jitter (max-min) for Random Fixed Pair Scenario is measured to 33[µs] at relative load 1.5 for "topology" 15, for traffic going 7 hops on ringlet 1. Note that in this plot, the delay consists of the waiting time in the stage queue at the ingress node, as well as propagation and queueing delay in the transit path. With a lower offered load, there is a larger chance that a congestion domain has not been established on the ring. Thus without one (or several) assigned node(s) working to constrain the amount of BE traffic accepted onto the ring, there may be a greater chance for an EF packet at the ingress point or in the transit path having to await the completion of a BE packet being transmitted. Note however that the difference is in the order of 10-20[µs] and is insignificant compared to total delay experienced by a packet. Another interesting observation is that the delay does not follow a monotonically increasing behavior as the offered load increases. Rather, it varies around a "steady-state" value. The majority of variations are very small however, as can be observed by the size of the confidence intervals. This confirms the expected behavior for EF delay, which is expected to be limited upwards, as the load increases. Both in terms of the access delay at the ingress point (with the analytical worst case expression provided in (5)), as well as the delay incurred on the packets as they propagate on their path between their ingress and egress points on the ring.*

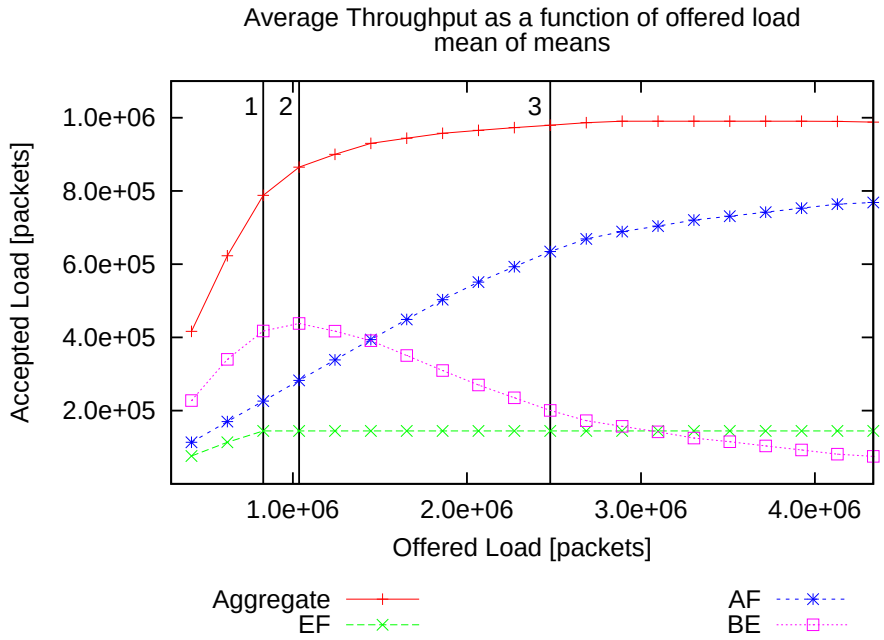


*Fig. 11: Accepted vs. total offered load for traffic of the DiffServ PHBs EF, AF and BE for the "Random Pairs" scenario. The plot shows the mean of mean-values for all the independent sender-receiver configurations. That is, for each sender-receiver configuration, a number of independent simulations are run for each value of offered load. For each value of offered load, the mean is taken across all the 16 sender-receiver configurations and their independent simulation runs. Note the smoothness of this curve when compared to the throughput curves shown in figures 5 and 9. This is due to the type of traffic generators used. In this experiment, we use CBR-like traffic generators, producing a relatively similar aggregate offered load for different seed values. As a result of this, the accepted load varies more smoothly as the offered load is varied.*

## 8 Conformance to DiffServ PHB Requirements

In section 3, the conformance requirements for the DiffServ PHBs EF, AF and BE (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 simulation results meet the DiffServ conformance requirements. We start by evaluating conformance to the DiffServ EF requirements in section 8.1, before evaluating conformance to the DiffServ AF requirements in section 8.2. Then in section 8.3, we end by evaluating conformance to the DiffServ BE requirements.

### 8.1 EF Conformance

For DiffServ EF traffic, we start by analyzing the results obtained for access delays, presented for the "Hot Receiver" scenario, presented in section 7.1 and the "Backbone/Metro Ring" scenario, presented in 7.2.

When considering the EF delay conformance requirements, one term in the expression provided in (2) is of particular interest:

In (2), 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 $L_j$. In our scenario, with a maximum packet size of $L_{max} = 500$[bytes], this accounts for a delay component of $\frac{500 \cdot 8}{R}$. The question then becomes, for an RPR network, what is the maximum committed DiffServ EF rate $R$ that can be supported?

With a configuration where EF traffic is transported using RPR subclasses A0 and A1, at high loads when traffic is backlogged, the worst case access delay component is due to the accumulation of credits for the sending of a $L_{max}$ sized packet. This component, which we term $D_{ac}$, is shown in (6) below.

$$D_{ac} = min\left(\frac{L_{max}}{R_{A0}}, \frac{L_{max}}{R_{A1}}\right) = L_{max} \cdot min\left(\frac{1}{R_{A0}}, \frac{1}{R_{A1}}\right) \tag{6}$$

Knowing that $D_{ac}$ is not the only component of the access delay, $D_a$, clearly, we cannot commit to a service $R$, that requires a maximum value of the access delay, lower than $D_{ac}$. Thus we get a relation between the maximum committed EF service rate, $R$, and the credit accumulation component of the access delay as shown in (7) below.

$$R < \frac{L_{max}}{D_{ac}} = max(R_{A0}, R_{A1}) \tag{7}$$

Thus in our "Hot Receiver" scenario, with $R_{A0} = R_{A1} = 50$ [Mbit/s], we have $R < R_{A0} = 50$[Mbit/s].

In the same scenario, if we were to transport EF traffic using A0 capacity only (i.e. $R_{A0} = 100$[Mbit/s] and $R_{A1} = 0$) then clearly, $R < R_{A0} = 100$[Mbit/s].

Thus in the two configuration alternatives, we have the same aggregate bandwidth (100 [Mbit/s]) available for high-priority traffic. However, in the first configuration, the committed bandwidth $R$ of a DiffServ EF conformant service is limited upwards to half that of the second configuration.

For the "Hot Receiver" scenario, serving EF traffic at a service rate $R = 50 + 50 = 100$[Mbit/s], the worst case access delay was found to be 101[$\mu$s]. Although slightly higher than the worst case $D_{ac}$ of 80[$\mu$s], the results obtained indicates that by enforcing a somewhat stricter rate limitations on control traffic, it is possible to provide a DiffServ compliant EF PHB at a committed rate $R$, close to the A0 and A1 rate, that is 50Mbit/s.

Similarly, for the access delay results obtained for the "Backbone/Metro Ring" scenario, the worst-case value of 2121[$\mu$s] is clearly within the analytical worst-case of 2200[$\mu$s]. And as such, these results strengthen the above claim that it possible to provide a DiffServ compliant EF PHB offering a rate close to the A0 and A1 rate, which in the case of the "Backbone/Metro Ring" scenario was 2[Mbit/s].

However, the results for access delay alone, is not sufficient to conclude on the suitability of RPR for the provisioning of a DiffServ compliant EF PHB.

Thus we proceed to the study of throughput results obtained, starting with the "Hot Receiver" scenario.

In this scenario, the per-node aggregate rate of class $A$ ($A0$ and $A1$) traffic is set to 10% of the line rate. The duration of the simulation runs constituting the throughput results in Fig. 5 is 0.9[s]. Thus, with $N = 1$ node, $R_L = 10^9$[bits/sec], $t_{sim} = 0.9$[s] and $L_p = 500$[bytes] , this corresponds to: $\sum_{A\ traffic} = \frac{N \cdot (R_{A0}+R_{A1}) \cdot R_L \cdot t_{sim}}{L_p} = \frac{1 \cdot 0.1 \cdot 10^9 \cdot 0.9}{4000} = 22500$[packets]. For all points of relative offered load, where the aggregate offered load exceeds 22500[packets], the aggregate accepted EF load is restricted to a value $\sim 22530$[packets].

Similarly, for the "Backbone/Metro Ring" scenario, the per-node aggregate rate of class $A$ ($A0$ and $A1$) traffic is set to 0.4% of the line rate. The duration of the simulation runs constituting the throughput results in Fig. 9 is 0.9[s]. Thus, with $N = 25$ nodes, $R_L = 10^9$[bits/sec], $t_{sim} = 0.9$[s] and $L_p = 500$[bytes], this corresponds to: $\sum_{A\ traffic} = \frac{N \cdot (R_{A0}+R_{A1}) \cdot R_L \cdot t_{sim}}{L_p} = \frac{25 \cdot 0.004 \cdot 10^9 \cdot 0.9}{4000} = 22500$[packets]. For all points of relative offered load, where the aggregate offered load exceeds 22500[packets], the aggregate accepted EF load is restricted to a value $\sim 22780$[packets].

Lastly, for the "Random Fixed Pairs" scenario, the per-node aggregate rate of class $A$ ($A0$ and $A1$) traffic is set to 4% of the line rate. The duration of the simulation runs constituting the throughput results in Fig. 11 is 0.9[s]. Thus, with $N = 16$ nodes, $R_L = 10^9$[bits/sec], $t_{sim} = 0.9$[s] and $L_p = 500$[bytes], this corresponds to: $\sum_{A\ traffic} = \frac{N \cdot (R_{A0}+R_{A1}) \cdot R_L \cdot t_{sim}}{L_p} = \frac{16 \cdot 0.04 \cdot 10^9 \cdot 0.9}{4000} = 144000$[packets]. For all points of relative offered load, where the aggregate offered load exceeds 144000[packets], the aggregate accepted EF load is restricted to a value $\sim 144500$[packets].

Thus in sum, all the throughput results support the claim, that the use of RPR's A0 and A1 subclasses meets the throughput conformance requirements of DiffServ's EF PHB.

### 8.2 AF Conformance

When studying the conformance requirements of the AF PHB, we see clearly from both the analytical results in Fig. 2 and the simulation results in Fig. 11, that as long as the ring is not saturated, the AF traffic is provided the requested throughput on the expense of BE traffic. Clearly, this is in conformance with the AF and default PHB requirements.

Also, although not supported by simulation results as this is not an issue in our simulation model, in-order delivery of AF packets is guaranteed by the RPR standard under normal operation of the ring.

However, as noted in section 5, the relative packet drop priorities have to be implemented in the *client*.

### 8.3 BE Conformance

Conformance to the default PHB is clearly obtained, as most of the bandwidth not utilized by the EF and AF traffic, is utilized by the BE traffic. Given the operation of the RPR fairness algorithm, 100% link utilization on all links in an RPR network is difficult to obtain (as is the case for any congestion control algorithm working concurrently to maximize throughput and minimize delay).

## 9 Related Work

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"*. Currently, there is a preliminary Internet draft available (first version published July 10, 2005) from the IETF IPORPR Working Group[8]. The main focus of the draft however, is the detailed mapping proposal between various DiffServ PHBs (as well as mapping for MPLS traffic) and RPR service classes. The Internet draft has no discussions on the fulfillment of DiffServ PHB conformance requirements, neither does it address the implementation of various drop priorities for the AF PHB group.

Apart from the ongoing work in the IETF IPORPR working group, we have only found one other paper addressing the problem of using RPR in a DiffServ context. One of the problems with the work by Wang et al. [21] is that a proposal is made to map the DiffServ AF PHB to RPR's A1 service class and to map the DiffServ EF PHB to RPR's A0 service class. By this, the use of the 1TB node architecture is effectively prohibited. Further, for a node using the 2TB-architecture, the design of the RPR MAC is done so that the *client* never knows

[8] http://www.ietf.org/html.charters/iporpr-charter.html

whether a high-priority packet is sent as an A0 or an A1 packet. Thus with the mapping proposed, one will never know, whether an EF or AF packet will be transported using RPR service classes A0 and A1. In effect this makes it impossible to provide a clear service differentiation between DiffServ traffic of the EF and AF PHBs, traversing an RPR network.

## 10 Conclusion

In this paper we have evaluated the suitability for 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 also introduced a set of invariants which specify important parts of the RPR traffic class priorities and rate controls. Further, we proposed and discussed a simple mapping proposal between the RPR traffic classes and some standardized DiffServ PHB groups. Based on the formal invariants and the proposed mapping, an analytical model of a single RPR flow between two nodes was developed and analyzed.

We also performed a comprehensive test of the proposed mapping by use of three different simulation scenarios utilizing the RPR conservative fairness mode. In the first two, we focused on the evaluation of access delay for DiffServ EF traffic as well as throughput performance of DiffServ EF, AF and BE (default) traffic.

In the last simulation scenario, we evaluated delay incurred on DiffServ EF traffic, starting the measurement at the time an EF packet is delivered to the MAC layer at its ring ingress node, and stopping the measurement when the packet was stripped at its ring egress node. In this scenario, we also evaluated throughput performance of DiffServ EF, AF and BE traffic.

The results obtained were compared to and discussed by use of our established analytical expressions.

Finally, in section 8, we discussed how well the obtained results adhered to the conformance requirements for the DiffServ EF, AF and BE PHBs.

In concluding, we found that most of the achieved access delay measurements support our analytical worst case expressions. However, care must be taken in the handling of control traffic, to avoid large bursts of high priority control data. Further, we found that when utilizing both RPR subclasses A0 and A1 (with corresponding rate constraints $R_{A0}$ and $R_{A1}$) for the transport of DiffServ EF traffic, the committed EF rate $R$ is limited upwards so that $R < max(R_{A0}, R_{A1})$. In the case were only one of the two subclasses is used, $R < R_{AX},\ X \in \{A0, A1\}$.

In the last scenario, we also found that EF traffic was not significantly delayed on its path between its ingress and egress points on the ring.

Further, all our throughput results indicated that our proposed mapping conforms to the DiffServ conformance requirements for the DiffServ EF, AF and BE PHBs.

However, as noted in section 5, for the provisioning of an AF conformant DiffServ PHB in an RPR network, the relative packet drop priorities have to be implemented in the *client*.

As a side-node, we will mention that the most interesting lessons learned are related to the last scenario, were communications pairs were established in a randomized fashion. In this scenario, our simulator model was extensively tested, and many subtleties related to RPR conservative mode rate control were revealed. Indeed, we believe that this type of test scenario represents a scenario that may be used as a type of sanity check for the RPR conservative mode.

## 11 Further Work

A natural extension of this paper, would be to use the same simulation scenarios for the testing of RPR aggressive fairness mode.

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 *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.

## 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
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
3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An architecture for differentiated services (1998) IETF, RFC 2475.
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)
5. IEEE Computer Society: IEEE Std 802.17-2004 Resilient packet ring (RPR) access method and physical layer specifications (2004)
6. OPNET Technologies, Inc: (OPNET Modeler)

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)
8. Hafner, E., Nendal, Z., Tschanz, M.: A digital loop communication system. IEEE Transactions on Communications **22** (1974) 877–881
9. Davik, F., Yilmaz, M., Gjessing, S., Uzun, N.: IEEE 802.17 Resilient Packet Ring tutorial. IEEE Communications Magazine **42** (2004) 112–118
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
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
12. Davik, F., Kvalbein, A., Gjessing, S.: An analytical bound for convergence of the Resilient Packet Ring Aggressive mode fairness algorithm. In: Proceedings of the 40th annual IEEE International Conference on Communications, Seoul, Korea (2005)
13. Nichols, K., Blake, S., Baker, F., Black, D.: (Definition of the Differentiated Services field (DS field) in the IPv4 and IPv6 headers)
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.
15. Heinanen, J., Baker, F., Weiss, W., Wroclawski, J.: Assured Forwarding PHB group (1999) IETF, RFC 2597.
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
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
18. Mondragón, R.J., Arrowsmith, D.K., Pitts, J.M.: Chaotic maps for traffic modelling and queueing performance analysis. Performance Evaluation **43** (2001) 223–240
19. Horn, G., Kvalbein, A., Blomskøld, J., Nilsen, E.: An empirical comparison of generators for self similar simulated traffic. Submitted to Performance Evaluation (2004)
20. Jain, R., Routhier, S.: Packet trains–measurements and a new model for computer network traffic. IEEE Journal on Selected Areas in Communications **4** (1986) 986–995
21. Wang, X., Huang, B., Yu, X., Zhang, F.: Edge QoS study of RPR equipment. Proceedings of the SPIE The International Society for Optical Engineering **5281** (2004) 396–403

# A Hot-Receiver Configuration

| **Parameter Name** | **Value** |
|---|---|
| Fairness Mode | Conservative |
| Line Rate | 1 [Gbit/s] |
| Packet size | 500 [B] (fixed) |
| Packet Generator Settings: | Relative load: Offered load $\in [1, .., 30] \cdot (Baseload)$<br>Nodes 1-26 sending behavior: Poisson<br>Node 15 Base load (average rate), EF: 20[Mbit/s]<br>Nodes 1-26 Base load (average rate), AF: 12[Mbit/s]<br>Nodes 1-26 Base load(average rate), BE: 20[Mbit/s]<br>Node 0 sending behavior: self-similar<br>Hurst parameter: 0.85<br>Base load, BE: $E(\overline{Rate}) = 20$[Mbit/s] |
| Rate Constraints (shaper settings) | Per Station AF Rate (nodes 1-26) : 12[Mbit/s]<br>Node 15 EF Rate (subclass A0): 50[Mbit/s]<br>Node 15 EF Rate (subclass A1): 50[Mbit/s] |
| STQ Thresholds | low: $125 \cdot 10^3$[bytes]<br>medium: $185 \cdot 10^3$ [bytes]<br>high: $250 \cdot 10^3$ [bytes] |
| rampUpCoef | 64 |
| rampDnCoef | 64 |
| ageCoef | 4 |
| lpCoef | 32 |
| link-delay | 250 $[\mu s]$ |
| Start of traffic | 1.1s |
| Simulation Duration (simulated time) | 2s |
| Number of independent simulations | 12 |

*Table 1: Detailed configuration information for "Hot-Receiver" simulation experiment.*

## B Backbone/Metro Ring Configuration

| **Parameter Name** | **Value** |
|---|---|
| Fairness Mode | Conservative |
| Line Rate | 1 [Gbit/s] |
| Packet size | 500 [B] (fixed) |
| Packet Generator Settings<br>for nodes 0-24: | Note that we have a 10/20/70 ratio of<br>offered traffic for the EF/AF/BE PHBs |
| Relative load: | Offered load $\in [1, .., 60] \cdot (Baseload)$ |
| EF Traffic | sending behavior: Poisson<br>Base load (average rate) = 2.86[Mbit/s] |
| AF Traffic | sending behavior: self-similar<br>Hurst parameter: 0.85<br>Baseload $(E(\overline{Rate}))$ = 5.71[Mbit/s] |
| BE Traffic | sending behavior: self-similar<br>Hurst parameter: 0.85<br>Baseload $(E(\overline{Rate})$ = 8[Mbit/s]<br>$min = 1, max = 30$<br>(limited upwards so that traffic is not sent beyond node 30) |
| Rate Constraints<br>(shaper settings for nodes 0-24) | <br>Per Station AF Rate: 8[Mbit/s]<br>Per Station EF Rate (subclass A0): 2[Mbit/s]<br>Per Station EF Rate (subclass A1): 2[Mbit/s] |
| STQ Thresholds | low: $125 \cdot 10^3$[bytes]<br>medium: $185 \cdot 10^3$ [bytes]<br>high: $250 \cdot 10^3$ [bytes] |
| rampUpCoef | 64 |
| rampDnCoef | 64 |
| ageCoef | 4 |
| lpCoef | 32 |
| link-delay | 100 $[\mu s]$ |
| Start of traffic | 1.1s |
| Simulation Duration<br>(simulated time) | 2s |
| Number of independent simulations | 12 |

*Table 2: Detailed configuration information for "Backbone/Metro Ring" scenario.*

## C Random Fixed Pairs Configuration

| **Parameter Name** | **Value** |
|---|---|
| Fairness Mode | Conservative |
| Line Rate | 1 [Gbit/s] |
| Packet size | 500 [B] (fixed) |
| Packet Generator Settings<br>for all nodes: | Note that we have a 55/27/18 ratio of<br>offered traffic for the BE/AF/EF PHBs |
| Relative load: | Offered load $\in [1, .., 10] \cdot (Baseload)$ |
| AF Traffic | sending behavior: Poisson<br>$Baseload$ (average rate) = 30[Mbit/s] |
| BE Traffic | sending behavior: Poisson<br>$Baseload$ (average rate) = 60[Mbit/s] |
| EF Traffic | sending behavior: Poisson<br>$Baseload$ (average rate) = 20[Mbit/s] |
| Rate Constraints<br>(shaper settings for all nodes) | <br>Per Station AF Rate: 40[Mbit/s]<br>Per Station EF Rate (subclass A0): 20[Mbit/s]<br>Per Station EF Rate (subclass A1): 20[Mbit/s] |
| STQ Thresholds | low: $100 \cdot 10^3$[bytes]<br>medium: $400 \cdot 10^3$ [bytes]<br>high: $700 \cdot 10^3$ [bytes] |
| rampUpCoef | 64 |
| rampDnCoef | 64 |
| ageCoef | 4 |
| lpCoef | 32 |
| link-delay | 250 $[\mu s]$ |
| Start of traffic | 1.1s |
| Simulation Duration<br>(simulated time) | 2s |
| Number of independent simulations | 12 |

*Table 3: Detailed configuration information for "Random Fixed Pairs" simulation experiment.*