A Rate Based RED Mechanism

Stefaan De Cnodder, Kenny Pauwels, Omar Elloumi

Alcatel – Network Strategy Group
Francis Wellesplein 1, B-2018 Antwerp, Belgium

{stefaan.de_cnodder, kenny.pauwels, omar.elloumi}@alcatel.be

ABSTRACT

The Random Early Detection (RED) algorithm [8] is becoming a de-facto standard for congestion avoidance in the Internet and other packet switched networks. Recently several research and working groups have been discussing parameter settings of RED. Although there were some recommendations dealing with the adequate settings of RED parameters, and some router vendors suggest some default values, the research community is still debating and still lacking the exact engineering rules network operators are looking for [13]. With the recent increasing interest in Differentiated Services, where packets with different priorities can share the same queue and where an enhanced variant of RED, i.e. weighted RED or n-RED, handling different priorities is used, the parameter settings' complexity increases accordingly. We propose Rate Based n-RED (RB n-RED): a new active queue management algorithm that preserves the design principles of RED and results in the same behavior as RED while reducing the number of parameters to only one. The basic idea is that the rate at which packets are dropped is a function of the long-term average arrival rate in addition to the average queue size used for RED. Our simulation results show that in all the cases RB n-RED gives the same performance results as a "well-configured" n-RED.

Keywords. Active queue management, RED, TCP.

I Introduction

It is becoming increasingly evident that active queue management schemes based on the Random Early Detection (RED) algorithm will be used in the next generation IP networks. For both Best-Effort, as well as IntServ and DiffServ based services, the Internet Engineering Task Force (IETF) is strongly recommending, if not stating it as a mandatory requirement, the use of an active queue management based on the RED algorithm [2,10]. Alternatively another, i.e. than RED, active queue management may be used if there are strong arguments behind its deployment.

RED aims at monitoring the queue occupancy and increasing the overall throughput while allowing transient bursts of data packets by relying on the average queue occupancy, instead of the actual queue occupancy, to drop packets probabilistically in a congestion avoidance phase and deterministically when the congestion is imminent. The basic design principle is that since reactive sources reduce their sending rate after packet losses, dropping probabilistically packets should avoid the oscillating behavior, and therefore performance degradation, by avoiding global synchronization of the sources. Without loss of generality, RED needs to be configured with four parameters: a minimum and a maximum threshold limiting the region in which packets are dropped probabilistically, a maximum drop probability determining an upper bound on the drop probability in the probabilistic dropping region and a weight used for estimating the average queue size.

For the specific case of the DiffServ model where the Type of Service (ToS) field of the IP header [15] is used to indicate the QoS requirements as well as a drop priority of the packet, extensions [5] to the RED algorithm have been proposed to provide different forwarding treatments based on the drop priority. The incoming data traffic, in particular the one using the Assured Forwarding Per Hop Behavior (AF PHB), is subject to marking aiming at assigning a low drop probability to the traffic that fits within the subscribed profile and a higher drop probability to the excess traffic. In case of congestion, packets marked with higher drop probability are preferentially dropped in order to make buffer space for packets marked with the lowest drop probability which may be dropped only in case of extreme congestion. While the AF PHB specifies three levels of drop precedence, several proposals [3,16] suggest the use of more than three drop precedences. For these schemes, a variant of RED handling different drop precedences can be used to achieve fair bandwidth allocation among competing flows and protection against unresponsive flows. While these are desired properties, there is a negative side effect resulting from using different colors consisting in increasing considerably the number of parameters and therefore the configuration complexity.

In [6,13], the authors have shown the impact of the parameter settings of the RED algorithm. Although in [6,7], setting guidelines are proposed, the engineering of RED is still an open area for research and testing, mainly because an optimal configuration is largely correlated with the number of flows, the round trip time, the buffer space, etc. Several proposals including [6] suggest the use of dynamic parameters, which are adjusted based on traffic patterns and the queue occupancy behavior. Although the effectiveness of these dynamic variants of RED has been demonstrated by means of simulations, their stability for very dynamic load conditions is still to be demonstrated. In this paper we propose a variant of RED, namely the Rate-Based n-RED (RB n-RED), that exhibits the same properties as RED with one or more colors. The strong advantage of RB n-RED is to reduce considerably the parameter setting complexity making it ideal for use for several load conditions and configurations.

This paper is organized as follows: Section II proposes an overview of RED based active queue management algorithms and describes RB n-RED. Section III explains how the estimation of the average arrival rate is made. In section IV we describe how to enhance RB n-RED to provide a uniform packet drop distribution. In Section V we give an overview of substantial benefits of deploying RB n-RED before showing some simulation results in Section VI. Finally we give some conclusions in Section VII.

II Rate Based n-RED

In this section we briefly discuss n-RED and define our new buffer acceptance algorithm, i.e. Rate Based n-RED.

A n-RED

The n-RED mechanism consists of n separate RED mechanisms, i.e. one for each color, making use of the same queue but using different thresholds and estimates for dropping an arriving packet. Routers using RED are able to keep the overall throughput high while maintaining a small average queue length, and tolerate transient congestion without resulting in global synchronization of TCP connections. Since n-RED is defined as an extension to RED it still retains these attractive features in addition to the ability to discriminate some packets with respect to others on the basis of a mark, i.e. drop-precedence or color, in times of congestion. When two drop-precedences[1] are used, n-RED is also known as RIO [5].

For every drop precedence, four parameters need to be configured: a weight $w_{drop\text{-}prec}$ (can be the same for all drop-precedences), a minimum threshold $min_{drop\text{-}prec}$, a maximum threshold $max_{drop\text{-}prec}$ and a maximum drop probability $maxp_{drop\text{-}prec}$. The operation of a single RED, with respect to the last three parameters, can be represented as shown in Figure 1**.** Figure 2 shows n-RED for two drop precedences: in and out, where we see that the number of parameters is doubled compared to Figure 1. When the average queue occupancy is below the minimum threshold, no packets are dropped. When the average queue size exceeds the minimum threshold, packets are dropped with an increasing probability up to $maxp_{drop\text{-}prec}$. When the average queue size exceeds the maximum threshold all arriving packets are dropped. In [9] an improvement is discussed: a further linear continuation from $maxp_{drop\text{-}prec}$ up to 1 of probabilistic dropping, until the average queue occupancy reaches 2 x max_{th}.

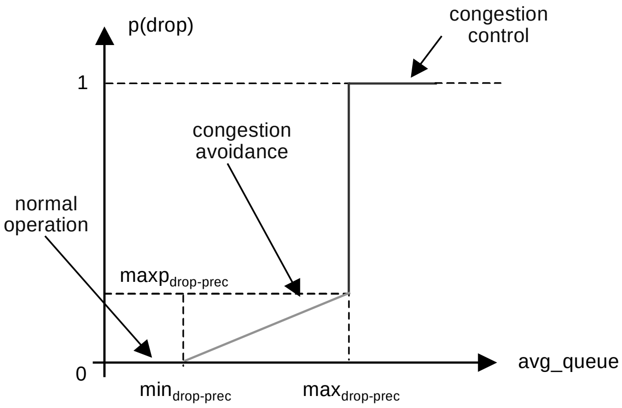


Figure 1: drop probability in function of the average queue length in RED

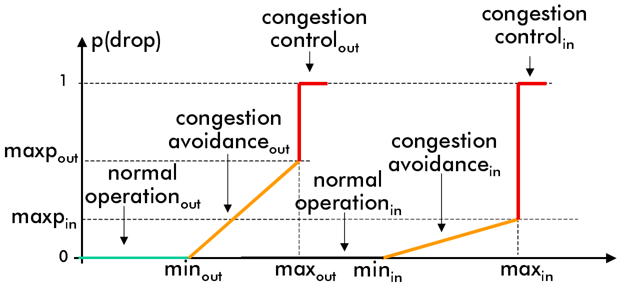


Figure 2: n-RED for two drop precedences, in and out

---

[1] Starting from this point we will use drop precedence and color interchangeably.

B Rate Based n-RED

With Rate Based n-RED, as with n-RED we assume that all packets are colored. The key idea behind RB n-RED is that the long term average arrival rate is estimated for each color. The total average arrival rate should never be larger than the rate R at which the queue is served. If the total average arrival rate is smaller, then all packets are accepted. When it is larger, packets have to be dropped such that the total arrival rate of the accepted packets is equal to the service rate R.

Let us assume that we have three colors: green, yellow and red. We denote the estimated arrival rate in bits per second of color c at time t by EAR(c,t). For example EAR(green,t) is the estimated arrival rate in bits of green packets at time t. Let TEAR(t) be the total arrival rate of all packets, i.e.

$$TEAR(t) = EAR(green,t) + EAR(yellow,t) + EAR(red,t) \qquad \text{Eq. 1}$$

then no packets should be dropped when TEAR(t) is smaller than the service rate R; when TEAR(t) is larger than R, packets should be dropped such that the total arrival rate of accepted packets is equal to the service rate R. This means that the drop probability should be equal to (see Figure 3):

$$P_{drop} = max\left(0, \frac{TEAR(t) - R}{TEAR(t)}\right) \qquad \text{Eq. 2}$$

Using this, we are sure that the accepted arrival rate is never higher than the service rate R. It is preferable that packets should be dropped less aggressively when the buffer occupancy is low and that packets should be dropped somewhat more aggressively when the buffer occupancy is high to be sure that the queue remains stable. For this purpose we multiply the drop probability P_{drop} with a correction factor, CF. CF is calculated based on the average buffer occupancy in the same way as it is calculated with RED. If the average buffer occupancy is low, CF should be smaller than 1. However when the average buffer occupancy is high, then CF should be larger than 1. Let QS be the queue size and avgQ the average queue occupancy, then CF is calculated in the following way (Figure 4):

$$CF = \frac{avgQ}{QS} \cdot AP \qquad \text{Eq. 3}$$

where AP is the aggressiveness parameter. A typical value for AP is 2. This parameter determines how aggressive the drop mechanism will be. A small value means that it is not aggressive and allows large queue occupancies and, therefore, large delays. A large value for AP makes the drop mechanism more aggressive resulting in smaller queue occupancies and smaller delays. Using the correction factor, the corrected drop probability, CP_{drop}, becomes

$$CP_{drop} = P_{drop} \cdot CF \qquad \text{Eq. 4}$$

This means that the drop rate at time t, i.e. the amount of bits discarded per second at time t, is equal to:

$$drop\ rate(t) = CP_{drop} \cdot TEAR(t) \qquad \text{Eq. 5}$$

The total drop probability must be P_{drop} or CP_{drop} (when the correction factor is used).

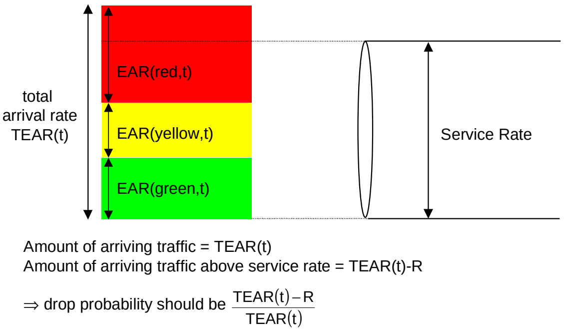


Figure 3: Basic idea behind Rate Based n-RED

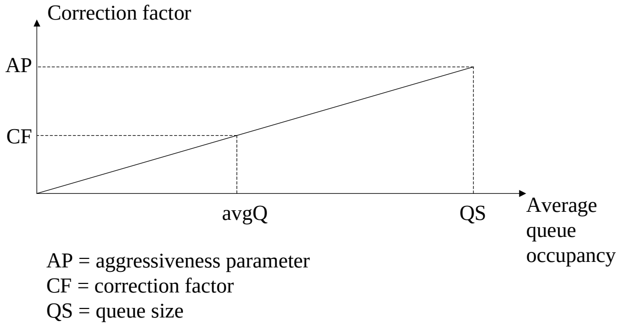


Figure 4: Calculation of the correction factor

C Configuration

The parameters, which need to be configured by the network operator, are:

- ✓ the service rate, R,
- ✓ the queue size, QS, and
- ✓ the aggressiveness parameter, AP.

The latter parameter, AP, only has to be configured when the correction factor is used. The service rate, R, and the queue size, QS, are typical parameters of a queue.

The distribution of the individual drop probabilities among the different colors can be chosen in different ways. In the following three subsections, we explain how RB n-RED can provide loss differentiation and a minimum guaranteed rate service.

III Estimation of the Arrival Rates

In order to calculate the drop probability we have to estimate the arrival rate of each color, i.e. EAR(green,t), EAR(yellow,t), and EAR(red,t). This can be done using the formula in [17] where the estimated arrival rate is updated at every packet arrival as follows:

$$EAR_{new}^{arr}(c,t) = \left(1 - e^{-T/K}\right) \cdot \frac{L}{T} + e^{-T/K} \cdot EAR_{old}^{arr}(c,t) \qquad \text{Eq. 6}$$

c is the color, K a constant (typical values are 0.1, 0.5 or 1 second), L the packet size, and T the time between the current time and the last update of $EAR^{arr}(c,t)$. Taking K small means that the estimated average arrival rate will follow the instantaneous arrival rate closely. On the other hand a large K means that EAR(t) varies more smoothly in time. This means also that it reacts slow with respect to bursts. $EAR^{arr}(c,t)$ is updated each time a packet of color c arrives.

The problem why we cannot use $EAR^{arr}(c,t)$ as EAR(c,t) is because $EAR^{arr}(c,t)$ is updated only on packet arrival. When there are no packet arrivals for that color, the estimate remains constant. In order to solve this, we calculate an upper bound, $EAR^{upper}(c,t)$, of the mean arrival rate, which is updated on every packet arrival, irrespective of the color of that arriving packet, in the following way:

$$EAR^{upper}(c,t) = \left(1 - e^{-T/K}\right) \cdot \frac{L_{max}}{T} + e^{-T/K} \cdot EAR^{arr}(c,t) \qquad \text{Eq. 7}$$

where T is again the time between the current time and the previous update of $EAR^{arr}(c,t)$ (and not the previous update of $EAR^{upper}(c,t)$!). Each time $EAR^{arr}(c,t)$ is updated, $EAR^{upper}(c,t)$ is set equal to $EAR^{arr}(c,t)$. $EAR^{upper}(c,t)$ is an upper bound on the estimated arrival rate, which decreases when no packets of that color arrive. The final estimated arrival rate, EAR(c,t), is updated on every packet arrival (irrespective of the color) according to the following formula:

$$EAR(c,t) = min\left(EAR^{arr}(c,t), EAR^{upper}(c,t)\right) \qquad \text{Eq. 8}$$

The algorithm is summarized in Figure 5**.**

```
When a packet of color c arrives:

1. Calculate $EAR^{arr}(c,t)$ according to
   formula (Eq. 6)
2. Set $EAR^{upper}(c,t)$  equal to $EAR^{arr}(c,t)$
3. For each color cc different from c,
   calculate $EAR^{upper}(cc,t)$  according to
   formula (Eq. 7)
4. For each color cc, calculate EAR(cc,
   t), according to formula (Eq. 8)
```

Figure 5: Estimating the averages

IV Variable Service Rate

In the previous sections, it was always assumed that the service rate, R, of the queue is fixed. In practice, there are also other QoS queues, such as a queue for the expedited forwarding (EF) PHB [12] and a queue for best effort traffic. This means that the service rate, R, is time-dependent and in order to calculate a drop probability at time t, the service time at time t, R(t), has to be calculated. R(t) can be estimated in the same way as the arrival rates are estimated: .

$$R(t) = \left(1 - e^{-T/K}\right) \cdot \frac{L}{T} + e^{-T/K} \cdot R(t\_previous) \qquad \text{Eq. 9}$$

where t is the last service time of a packet and t_previous, the previous service time. T is the difference between t and t_previous. In this case, the calculation of the drop probability, for example in Eq. 2, is based on R(t) rather than on R.

V Supporting Different Services

Rate-Based n-RED can be used to support different services. This section explains how it can be used for two specific services: minimum rate guarantees and loss differentiation.

A Minimum Rate Guarantees

For minimum rate guarantees, red packets are discarded first. If this is not sufficient we also start to discard yellow packets, and if there is still congestion all red and yellow packets are dropped. Finally, green packets are dropped only in case of extreme congestion. CP_{drop} is computed as described in section 2 and this value is used to calculate the acceptance rate. The average queue occupancy is thus taken into account when calculating the (transformed) acceptance rate. The drop rate, as defined in section 2, is:

$$drop\ rate(t) = CP_{drop} \cdot TEAR(t) \qquad \text{Eq. 10}$$

This means that the amount of accepted traffic, AT(t), is equal to:

$$AT(t) = (1 - CP_{drop}) \cdot TEAR(t) \qquad \text{Eq. 11}$$

All traffic above AT(t) has to be discarded. Four different scenarios can be identified. The first scenario is where the total arrival rate TEAR(t) is smaller than or equal to AT(t). In this case, no packets are dropped. The second scenario is where EAR(green,t) + EAR(yellow,t) is below AT(t), but TEAR(t) is above AT(t), then all green and yellow packets are accepted, but the red packets are probabilistically dropped (Figure 6):

$$P_{red} = \frac{TEAR(t) - AT(t)}{EAR(red,t)} \qquad \text{Eq. 12}$$

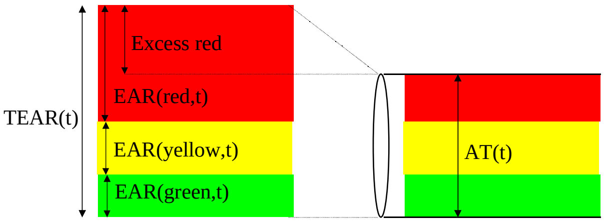
The third scenario is similar: EAR(green,t) is below AT(t), but EAR(green,t) + EAR(yellow,t) is above AT(t). In this case we accept all green packets, discard all red packets, and discard yellow packets with probability (Figure 7):

$$P_{yellow} = \frac{EAR(green,t) + EAR(yellow,t) - AT(t)}{EAR(yellow,t)} \qquad \text{Eq. 13}$$

In the fourth scenario EAR(green,t) is above AT(t). This means that all yellow and red packets must be dropped and that the green packets are dropped with probability (Figure 8):

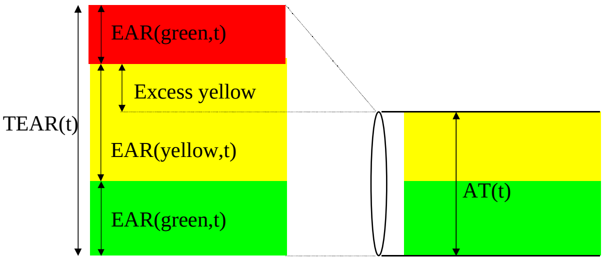
$$P_{green} = \frac{EAR(green,t) - AT(t)}{EAR(green,t)} \qquad \text{Eq. 14}$$

The four scenarios are summarized in Figure 9.



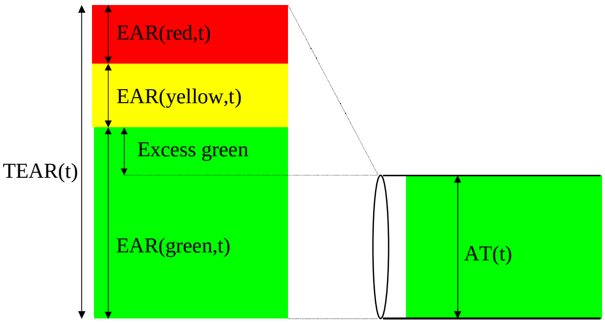
Accept all green and yellow packets
Discard red packets with probability: Excess red / EAR(red,t)

Figure 6: Discard red packets probabilistically



Accept all green packets
Discard all red packets
Discard yellow packets with probability: Excess yellow / EAR(yellow,t)

Figure 7: Discard yellow packets probabilistically



Discard green packets with probability: Excess green / EAR(green,t)
Discard all yellow and red packets

Figure 8: Discard green packets probabilistically

B Loss Differentiation

For loss differentiation the loss probability for the red packets should be q_{red} times the loss probability of the green packets and the loss probability of the yellow packets should be q_{yellow} times the loss probability of the green packets. The drop rate (t), i.e. the amount of bits discarded per second at time t, is:

$$\begin{aligned} drop\ rate(t) &= P_{green} \cdot EAR(green,t) + P_{yellow} \cdot EAR(yellow,t) + P_{red} \cdot EAR(red,t) \\ &= P_{green} \cdot EAR(green,t) + q_{yellow} \cdot P_{green} \cdot EAR(yellow,t) + q_{red} \cdot P_{green} \cdot EAR(red,t) \\ &= P_{green} \cdot \left(EAR(green,t) + q_{yellow} \cdot EAR(yellow,t) + q_{red} \cdot EAR(red,t) \right) \end{aligned} \qquad \text{Eq. 15}$$

Because $drop\ rate(t) = CP_{drop} \cdot TEAR(t)$ (assume we use the corrected drop probability), we have the following:

$$\begin{aligned} & P_{green} \cdot \left(EAR(green,t) + q_{yellow} \cdot EAR(yellow,t) + q_{red} \cdot EAR(red,t) \right) = CP_{drop} \cdot TEAR(t) \\ & \Rightarrow P_{green} = \frac{CP_{drop} \cdot TEAR(t)}{EAR(green,t) + q_{yellow} \cdot EAR(yellow,t) + q_{red} \cdot EAR(red,t)} \end{aligned} \qquad \text{Eq. 16}$$

Using this formula for calculating the drop probability of the green packets, we can derive the drop probability for the yellow and red packets:

$$P_{yellow} = q_{yellow} \cdot P_{green}, \text{ and}$$
$$P_{red} = q_{red} \cdot P_{green}. \qquad \text{Eq. 17}$$

These formulas are sufficient to compute the different drop probabilities in order to support loss differentiation.

```
if
(EAR(green,t)+EAR(yellow,t)+EAR(red,t))
 ≤ AT(t))

then accept all packets;

else
if ((EAR(green,t)+EAR(yellow,t) ≤ AT(t))

then accept all green packets;
     accept yellow packets;
     discard red packets with probability

     $P_{red} = \frac{TEAR(t) - AT(t)}{EAR(red,t)}$ ;

else
if (EAR(green,t) ≤ AT(t))

then accept all green packets;
     discard all red packets;
     discard red packets with probability

     $P_{yellow} = \frac{EAR(green,t) + EAR(yellow,t) - AT(t)}{EAR(yellow,t)}$ ;

else discard all yellow packets;
     discard all red packets;
     discard green packets
     probabilistically;

     $P_{green} = \frac{EAR(green,t) - AT(t)}{EAR(green,t)}$ ;
```

Figure 9: algorithm for bandwidth differentiation

VI A Uniform Drop Probability

Unpublished simulation results performed by Sally Floyd show that light-tailed drop distributions give better performance than heavy-tailed ones (see the note in http://www.aciri.org/floyd/REDdistributions.txt for more details). In particular a uniform drop distribution is a desired property of RED. In this section we explain how RB n-RED can be modified to provide a uniform drop distribution.

Let N_c be the number of packets of color c after a dropped packet of the same color, i.e. the number of accepted packets of color c after a dropped packet of color c plus 1 (the dropped packet at the end). Then with the dropping method of the previous subsections, N_c has a geometric distribution. This is because every packet has the same drop probability independent of any previous drops:

$$P[N_c = n] = (1 - p_c)^{n-1} \cdot p_c \text{ , } n \geq 1 \qquad \text{Eq. 18}$$

where p_c is the drop probability of color c. The mean of N_c, $E(N_c)$, is in this case equal to:

$$E(N_c) = \frac{1}{p_c} \qquad \text{Eq. 19}$$

If we want to drop a packet according to a uniform distribution as in [8], then we have to change p_c such that the mean remains the same, i.e. such that we are still dropping at the same rate. Assume that P_b, is this transformed uniform drop distribution and that the final drop probability, P_a, is calculated as follows:

$$P_a = \frac{P_b}{1 - count_c \cdot P_b} \qquad \text{Eq. 20}$$

where count is incremented with 1 if the packet was not dropped and set to 0 if it was dropped. In this case N_c will have a uniform distribution:

$$\begin{aligned} P[N_c = n] &= \frac{P_b}{1-(n-1)\cdot P_b} \cdot \prod_{i=0}^{n-2} 1 - \frac{P_b}{1 - i \cdot P_b} \\ &= \frac{P_b}{1-(n-1)\cdot P_b} \cdot \prod_{i=0}^{n-2} \frac{1-(i+1)\cdot P_b}{1 - i \cdot P_b} \\ &= P_b \end{aligned} \qquad \text{Eq. 21}$$

when n is smaller or equal to $1/P_b$ (assume for simplicity that $1/P_b$ is an integer). Otherwise the probability is zero. This means that the mean of N_c, $E(N_c)$, is equal to:

$$E(N_c) = \frac{1/P_b + 1}{2} = \frac{1 + P_b}{2 \cdot P_b} = \frac{1}{2 \cdot P_b} + \frac{1}{2} \qquad \text{Eq. 22}$$

In order to have the same drop probability for the uniform case as with the geometric case, we should choose P_b such that the two means in Eq. 19 and in Eq. 22 are equal to each other. This means that we have to calculate P_b as follows:

$$P_b = \frac{p_c}{2 - p_c} \qquad \text{Eq. 23}$$

In order to obtain a uniform drop probability, the algorithm depicted in Figure 10 should be executed. Notice that each color has its own count-variable.

```
When a packet of color c arrives:

1. Calculate p_c using one of the methods
   described in the previous subsections
2. Calculate P_b:

        $P_b = \frac{p_c}{2 - p_c}$

3. Calculate $p_a = \frac{p_b}{1 - count_c . p_b}$

4. Drop packet with probability P_a. If
   packet is not dropped then increment
   count_c with 1, otherwise reset count_c
   to zero.
```

Figure 10: A uniform drop probability

VII Simulation Model and Results

In order to show that RB n-RED gives a similar behavior as an optimal configured n-RED, some simulations with bursty TCP sources are performed. The intention of these simulations is not to show that RB n-RED outperforms n-RED. We will consider a scenario, called the generic fairness configuration (GFC), which is a rather complex scenario and it is depicted in Figure 11. We have 10 users connected to an ISP, called A-short, A-long, B-short, B-long, C-short, C-long, X-short, X-long, Y-short, and Y-long. Users A-long, B-long, C-long, X-long, and Y-long have a larger RTT than the other users connected to the same bottleneck router. The shaded region in Figure 11 represents the ISP to which all users are connected. All access links, i.e. the links from the routers of the users towards the ISP, are 45 Mbps with a fixed propagation delay of 2.5 ms. The bandwidths in the ISP backbone are shown in Figure 11 in Mbps and the links have a fixed propagation delay of 10 ms. The arrows in the figure represents the direction in which the data packets flow, the acknowledgements flow in the opposite direction. Each user has 20 TCP connections attached to the routers via a link of 10 Mbps and a fixed delay of 1 ms, but these individual TCP connections are not shown in the figure.

The aggregated flow is colored according to a two rate three color marker (tr-TCM) [11]. These have a CIR of 6 Mbps, a PIR of 12 Mbps, a CBS of 10 Kbytes and a PBS of 20 Kbytes. We take the link rates such that green packets can consume at most 50% of the second link of the ISP. Because traffic from users A-short, A-long, B-short, B-long, C-short, and C-long go through this link, the maximum green packets rate is at most 36 Mbps (6 times 6 Mbps). If we want that the green packets to consume at most 50% of this link, we have to set the link rate equal to 72 Mbps. Users B-short and B-long use link 1 and 2 and they should have their bottleneck at the first router of the ISP such that they can consume 10% of the excess bandwidth of link 2, i.e. 3.6 Mbps. This means that the users B-short and B-long send their traffic at an average rate of 7.8 Mbps, each (6 Mbps CIR and 1.8 Mbps excess). By setting the link rate of the first link equal to 31.2 Mbps, users X-short, X-long, B-short and B-long should have an equal amount bandwidth. We make a similar reasoning for users A-short and A-long: they have their bottleneck at the third router and we want that they consume 30% of the excess bandwidth of link 2. This leads us to a bandwidth of 45.6 Mbps for the third link. Users C-short and C-long have their bottleneck at the second router.

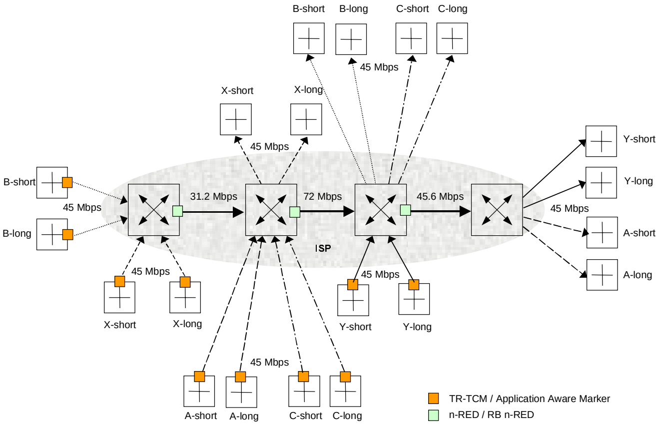


Figure 11: Simulation model - GFC scenario

We perform simulations where all routers implement n-RED, RB n-RED and 2 mixes. Mix 1 where router 1 and router 3 implement n-RED and where router 2 implements RB n-RED. A second mix (Mix 2) consists of a RB n-RED implementation in router 1 and router 3, and an n-RED implementation in router 2.

Figure 12 shows the simulation results for bandwidth differentiation. We see that our RB n-RED mechanism achieves the same results as an optimal configured n-RED mechanism. All users get their CIR in all cases, therefore achieving the minimum rate guarantee promised by the mechanisms. Issues related to bandwidth guarantees using this method can be found in [14]. From the simulation results of the two mixes we can notice that RB n-RED performs well in a heterogeneous network where different RED mechanisms are deployed. The mechanism doesn't degrade the performance of a well-configured n-RED. We remind the reader that correctly configuring n-

RED isn't obvious, while with RB n-RED this problem is resolved.

In a second simulation scenario, the RB n-RED mechanism was set such that it provides loss differentiation between green and yellow packets (no red packets here to keep the simulation scenario relatively simple). The chosen loss ratio between green and yellow was set to 2. For loss differentiation no tr-TCMs are used, the marking is done at the hosts. Each user has 40 TCP connections where 20 connections tag their packets as green whereas the others tag their packets as yellow. Figure 13 shows the packet loss ratio in each queue and it is clear that the targeted loss ratio between green and yellow are met.

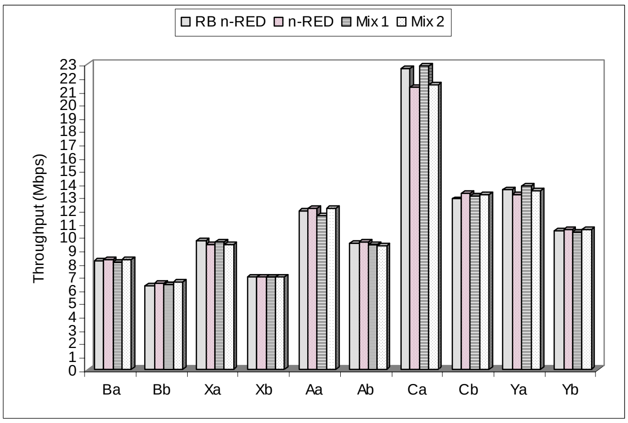


Figure 12: simulation results for bandwidth differentiation

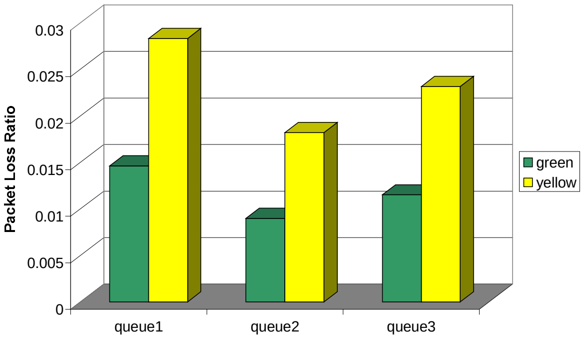


Figure 13: Simulation results for loss differentiation

VIII Relationship with Other Network Mechanisms

The Rate Based n-RED mechanism proposed in this document, is somewhat more complex to implement compared to n-RED. The estimation of the rates for each color is the most complex task from an implementation point of view. However, in addition to substantial benefits in terms of configuration, there are other positive side effects that compensate for this additional complexity: the different rates can also be used for other purposes such as traffic engineering, measurement based flow admission control, feedback mechanisms, rate adaptive shapers and performance monitoring. How the rate estimates can be used for these purposes is briefly discussed in the following subsections but a specification of such a mechanism is outside the scope of this paper.

A Traffic Engineering

The purpose of traffic engineering is to reroute a part of the traffic when some links along the current path become congested. A problem that traffic engineering faces is how to determine which links are congested and which links are not. One method is to estimate the packet loss ratio. Alternatively, this can also be done using the rate estimates of the RB n-RED. With these rate estimates it is easy to determine which links are congested and the available bandwidth on each link.

Henceforth by using the rate estimated by RB n-RED, a powerful traffic engineering mechanism can be designed without introducing too much complexity in the routers because the rate estimates are already made by the buffer acceptance algorithm.

B Measurement Based Flow Admission Control

For measurement based flow admission control, the available bandwidth has to be estimated in order to accept a new flow or to reject it. An example is the "simple marking" scheme proposed in [18]. It is obvious that the rate estimates provided by the RB n-RED can also be used for measurement based flow admission control.

C Feedback Mechanisms and Rate Adaptive Shapers

Core routers can give feedback to the edge routers about the level of congestion they experience. When a core router detects a congestion some proposals suggest to introduce a feedback mechanism where a message is sent back to the edge routers in order to reduce their sending rate [4]. With the rate estimates provided by the RB n-RED mechanism, the edge routers can see how the level of congestion in the network increases and decreases in time. Based on the increase or decrease of the level of congestion, the edge routers can adapt their sending rate appropriately.

The same can be done for rate adaptive shapers such as the one proposed in [1]. The rate estimates provided by Rate Based n-RED in the core routers can also be used to determine a more appropriate shaping rate.

IX Conclusions

In this paper we proposed a buffer acceptance algorithm that exhibits the same characteristics as RED while avoiding the parameter setting complexity. The mechanism adapts itself to the changing traffic patterns but without changing parameter values, and drops the correct amount of packets. It is based on rate estimates, which can be re-used by various other QoS-related mechanisms.

X Acknowledgement

This work was carried out within the framework of the project TRENDS, sponsored by the Flemish institute for the promotion of scientific and technological research in the industry (IWT).

XI References

[1] O. Bonaventure and S. De Cnodder, "A Rate Adaptive Shaper for Differentiated Services", IETF draft, draft-bonaventure-diffserv-rashaper-01.txt, October, 1999, work in progress.
[2] B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.
[3] Z. Cao, Z. Wang and E. Zegura, "Rainbow Fair Queueing: Fair Bandwidth Sharing Without Per-Flow State", in proceedings INFOCOM '00, March 2000.
[4] H. Chow and A. Leon-Garcia, "A feedback control extensions to Differentiated Services", draft-chow-diffserv-fbctrl-00.txt, March 1999, work in progress.
[5] D. Clark and W. Fang, "Explicit Allocation of Best-Effort Service", IEEE/ACM Transactions on Networking, Vol6, Num 4, August 1998.
[6] W. Feng, D. D. Kandlur, D. Saha, K. G. Shin, "A self-configuring RED gateway", in Proceedings INFOCOM '99, March 1999.
[7] W. Feng, D. D. Kandlur, D. Saha, K. Shin, "Blue: A New Class of Active Queue Management Algorithms", Technical report U. Michigan CSE-TR-387-99, April 1999.
[8] S. Floyd and V. Jacobson, "Random Early Detection Gateways for congestion Avoidance", IEEE/ACM Transactions on Networking, Vol 1, Num 4, August 1993.
[9] S. Floyd; Mail – "Optimum functions for computing the drop probability", available from http://www.aciri.org/floyd/REDfunc.txt.
[10] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, February 1999.
[11] J. Heinanen and R. Guerin, "A Two-Rate Three Color Marker", RFC 2698, September 1999.
[12] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999.
[13] M. May, J. Bolot, C. Diot, B. Lyles, "Reasons not to deploy RED",
[14] B. Nandy, N. Seddigh, P. Pieda, J. Ethridge, "Intelligent Traffic Conditioners for Assured Forwarding Based Differentiated Services Networks", Proc. of Networking 2000, May 2000, Paris, France.
[15] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
[16] K. Pauwels, S. De Cnodder, O. Elloumi, "A Multi-Color Marking Scheme to Achieve Fair Bandwidth Allocation", accepted for QofIS 2000, September 2000, Berlin, Germany.
[17] I. Stoica, H. Zhang, S. Shenker, R. Yavatkar, D. Stephens, A. Malis, Z. Wang, F. Baker, J. Wroclawski, C. Song and R. Wilder, "Per Hop Behaviors Based on Dynamic Packet State", draft-stoica-diffserv-dps-00.txt, February, 1999, work in progress.
[18] L. Westberg and Z. R. Turanyi, "Load control of real-time traffic", draft-westberg-loadcntr-02.txt, October 1999, work in progress.