Modeling RED with Two Traffic Classes

P. Kuusela and J. T. Virtamo
Laboratory of Telecommunications Technology
Helsinki University of Technology
P. O. Box 3000, FIN-02015 HUT, Finland
Email: {Pirkko.Kuusela, Jorma.Virtamo}@hut.fi

July 5, 2000

Abstract

Random Early Detection (RED) [8] is a proposed mechanism to control congestion in network routers and gateways. We analyze the behavior of a queue in the case of two traffic classes with Poisson arrivals aggregated into a buffer managed by the RED algorithm. Each class has its own RED parameters, and the packet dropping probability in a lower priority class may or may not depend on the queue of the higher priority class, the former being the setting of a RIO algorithm [4]. An ODE approximation is presented to describe the time evolution of the expectations of the exponentially averaged queue lengths. The approximation, based on separation of the two different time scales in the system, is asymptotically accurate for a long averaging time. This paper focuses on the equilibrium values of the averaged queue lengths of the two classes, viewed as functions of their loads.

1 Introduction

Random Early Detection (RED) was proposed by Floyd and Jacobson [8] as an effective mechanism to control congestion in the network routers or gateways. Currently it is recommended as one of mechanisms for so-called active queue management by IETF (see Braden et al. [2].) and it has been implemented in vendor products. In addition, RED has been incorporated in various drafts for Differentiated Services for the Internet in such a way that RED operates on different flows with different parameters depending on the flows' priority, i.e., RED is used to provide different classes of services. Clark and Fang [4] propose RED with distinction of "in" and "out" packets as RIO. See also the Internet draft [1] for Differentiated Service proposal and [5] for a framework utilizing RED or RIO algorithms.

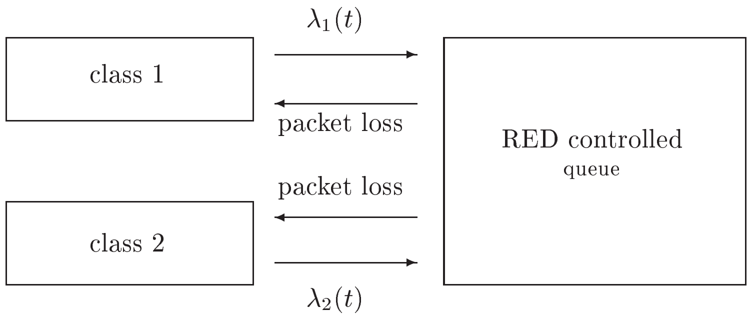

Figure 1: Illustration of a Differentiad Services framework.

RED has been analyzed using simulations or testbeds by Floyd and Jacobson [8], Lin and Morris [11] and by May et al. [12]. Simulation studies have been complemented by analytical studies; a detailed analysis of RED was presented by Peeters and Blondia [14]. Also, complete RED has been analyzed by Sharma et al. in [15]. May et al. use simple analytical models for RED in [13]. The exponentially averaged queue length, the key feature in RED, has not been fully addressed in [13], while it is the focus of our work.

The aim of this paper is to direct the analytical study of RED towards applications in Differentiated Services. A simple framework of Differentiated Services, see Figure 1, consists of two interacting parts: traffic sources devided into a collection of traffic classes and a RED controlled queue that differentiates the queue admission based on the traffic class. A traffic class may consist of a population of TCP sources or some non-TCP-friendly sources. In particular, a traffic class may or may not react to the congestion indications produced by the RED controlled queue. However, in this work we focus on the queue related analysis of the system, where the queue receives an aggregate flow of packets from each traffic class and reacts dynamically to the changes in the flow rates under the control of the RED algorithm. We model the dynamics of the RED controlled queue in this setting when arrivals from each class constitute a Poisson process with time varying intensity $\lambda_i(t)$. Such a model can be incorporated into the full system model describing the interaction of the sources and the RED controlled queue.

In this work we use an ordinary differential equation approximation for the mean queue length, an approach introduced by Sharma et al. in [15] for one traffic class, and study the mean queue lengths of two Poisson arrival streams that are aggregated into a buffer controlled by the RED algorithm. Each class has its own RED parameters, and the packet dropping probability in a lower priority class may or may not depend on the queue of the higher priority class, the former being the setting of the RIO algorithm.

The recent extension of the work in [15] by Lassila et al. [10] to capture the oscillatory transient behavior of the mean queue length is not exploited here, as our main interest is in the equilibrium of the system.

This paper is organized as follows: We begin by reviewing the RED algorithm in Section 2. The principle of the ODE approximation is given in Section 3 and after that in Section 4 we derive ODE approximations for the case of two traffic classes. We finish by numerical results in Section 5, where we first look at the equilibrium values of the exponentially averaged queue lengths in each class. We also consider the time evolution of the averaged queue lengths and we approximate the effect of the counter process omitted in our model.

2 The RED algorithm

Consider a queue with a finite buffer that can hold up to K packets at a time. RED is a congestion control algorithm that may drop (or mark) packets randomly before buffer overflow. Random dropping takes place when the algorithm detects signs of permanent congestion. The congestion control variable in RED is the average queue size that is calculated from the instantaneous queue size by using exponential averaging. Persistent congestion results in an increase of the average queue size, whereas transient bursts in buffer have only little influence.

Next, we explain the RED algorithm in detail as in [8]. Let q_n denote the queue length, i.e., the number of packets in the system (including the one in service) at the time of nth packet arrival. Similarly, let s_n denote the average queue length (to be defined below) at the time of nth arrival. The nth packet will be discarded with probability p_n that depends on s_n and the past history of discarding packets, stored in the so-called counter variable c_n. RED is parametrized by choosing constants w for calculating the exponentially averaged mean queue length and $p_{\max}$, $T_{\min}$, and $T_{\max}$ specifying the dropping probability.

For each arriving packet we compute the average (smoothened) queue length s_n using

$$\begin{cases} s_n = (1-w)s_{n-1} + wq_n, & \text{if } q_n > 0, \\ s_n = (1-w)^m s_{n-1}, & \text{otherwise}, \end{cases} \tag{1}$$

where m = (idle time during the interarrival time of the nth packet)$/\tau$, τ = typical trasmission time for a small packet, and w is an appropriately fixed constant. If the buffer is full the packet is lost (i.e., $p_n = 1$ and we set $c_n = 0$). If $s_n < T_{\min}$, the arriving packet is accepted, i.e., $p_n = 0$ and we set $c_n = -1$. If $s_n > T_{\max}$, the packet is dropped, $p_n = 1$ and $c_n = 1$. However, if $T_{\min} \le s_n \le T_{\max}$, we set

$$c_n = c_{n-1} + 1, \quad C_n = \frac{1}{p_{\max}} \frac{T_{\max} - T_{\min}}{s_n - T_{\min}}, \qquad \text{and} \qquad p_n = \min\{1, 1/(C_n - c_n)\}.$$

Then, with probability p_n the packet is discarded.

Typically, in above, the value w is relatively small, i.e., Floyd and Jacobson [8] propose using $w \ge 0.001$. The role of the counter c_n in RED is to distribute the packet drops more evenly.

In this work we simplify the above RED algorithm in two ways: First, arrivals into an empty queue update s_n like arrivals into an occupied queue. Second, we omit the RED counter c_n in the analysis, i.e., we set $p_n = 1/C_n$. The first simplification was done as we are focusing on RED in presence of congestion. Later we see that the second simplification can be compensated by simple adjustment in the $p_{\max}$ parameter.

3 ODE approximation

Consider a process $\{q_n, s_n, c_n\}$ for a buffer controlled by the RED algorithm. The main observation is that two different time scales are involved in $\{q_n, s_n, c_n\}$. When w is small, as suggested for practical systems, $\{s_n\}$ changes very slowly in comparison to $\{q_n, c_n\}$ process, and $\{q_n, c_n\}$ can be viewed approximately stationary considering the queue admission control variable $s_k \equiv s$ for all k. We denote this hypothetical process by $\{q_n^{(s)}, c_n^{(s)}\}$.

Next we discuss a continuous time ordinary differential equation (ODE) approximation for the expected value of s. We assume that the packet arrival stream is a Poisson process with intensity $\lambda(t)$ and that the packet lengths are i.i.d. from exponential distribution with mean $1/\mu$.

We develop the differential equation model from the updating equation (1) for s_n, which can be written as

$$\Delta s_n = s_{n+1} - s_n = w(q_n - s_n). \tag{2}$$

In deriving a continuous time equation, we look at a short time interval of length Δt. Taking expectations and conditioning on the number of arrival events A during Δt gives us

$$\begin{aligned} E[\Delta s(t)] &= \sum_{i=0}^{\infty} E[\Delta s(t) \mid A = i] \, P\{A = i\} \\ &= 0 + wE[q(t) - s(t)]\lambda(t)\Delta t + O(\Delta t^2) \\ &= \lambda(t) w \left(E[q(t)] - E[s(t)] \right) \Delta t + O(\Delta t^2). \end{aligned}$$

Observing that $E[\Delta s(t)] = \Delta E[s(t)]$, letting Δt approach zero, and by denoting $\bar{s}(t) = E[s(t)]$, we obtain the differential equation

$$\frac{d}{dt}\bar{s}(t) = \lambda(t) w \left(E[q(t)] - \bar{s}(t) \right).$$

Due to slowness of $\{s(t)\}$ process in comparison to $\{q(t), c(t)\}$ one can assume that $E[q(t)]$ is sufficiently well approximated by $E_\pi[q^{(s)}]_{s=\bar{s}(t)}$, where $q^{(s)}$ is a random variable distributed according to the stationary distibution of $\{q^{(s)}(t)\}$[1]. Simulations in [15] indicate that the above approximation is not able to capture the damping oscillatory behavior of $E[q(t)]$ prior to reaching its equilibrium. To correct this deficiency Lassila and Virtamo [10] have extended the above model with an ODE describing the evolution of $E[q(t)]$.

[1] By conditioning $E[q(t)] = E[\,E[q(t) \mid s]]$, where we approximate $E[q(t) \mid s]$ by $E[q^{(s)}]$. Further, the distribution of s is assumed to be narrow so that $s(t) \approx E[s(t)] = \bar{s}(t)$, resulting in the approximation $E[q(t)] \approx E[q^{(s)}]_{s=\bar{s}(t)}$.

4 Two traffic classes

In this section we derive ODE approximations for two Poisson streams with intensities $\lambda_1(t)$ and $\lambda_2(t)$ that are aggregated into a buffer controlled by the RED algorithm. In both streams the packet lengths are assumed to be i.i.d. from exponential distribution with mean $1/\mu$, and for simplicity we also take $\mu = 1$ in the analysis.

We refer to packets arriving from Poisson stream with intensity $\lambda_1(t)$ as belonging to class 1, and similarly, class 2 refers to packets arriving with intensity $\lambda_2(t)$. We assume that w is given together with RED parameters $p_{i,\max}$, $T_{i,\min}$ and $T_{i,\max}$ for each class $i = 1, 2$. Denote the number of class i packets in queue at the time of nth arrival by $q_{i,n}$ and the exponentially averaged (the averaging parameter w is the same for both classes) queue sizes by $s_{i,n}$ for $i = 1, 2$.

4.1 Two uncoupled classes

We derive an ODE approximation for the case of two uncoupled classes that are aggregated into a single buffer. By uncoupled classes we mean that each class operates with its own RED parameters and the probability of discarding a packet from the class i depends only on $s_{i,n}$.

The updating equation (1) written for both classes gives us $\Delta s_{i,n} = \hat{q}_{i,n} - \hat{q}_{i,n-1} = w\,(q_{i,n} - s_{i,n})$, for $i = 1, 2$. Again, in continuous time we look at a small time interval of length Δt. Taking expectations and conditioning on the number of arrival events A_i in class i during Δt results in

$$\begin{aligned} E[\Delta s_i] &= \sum_{j=0}^{\infty} E[\Delta s_i \mid A_i = j] P\{A_i = j\} \\ &= w\,(E[q_i] - E[s_i])\,\lambda_i \Delta t + O(\Delta t^2), \text{ for } i = 1, 2. \end{aligned}$$

As in Section 3, we get

$$\frac{d}{dt}\bar{s}_i(t) = \lambda_i(t) w\,(E[q_i(t)] - \bar{s}_i(t))\,, \text{ for } i = 1, 2,$$

where $\bar{s}_i(t)$ denotes $E[s_i(t)]$.

We approximate $E[q_i]$ (where explicit time dependency has been omitted from the notation) by $E_\pi[q_i^{(s_1,s_2)}]$, i.e., the stationary mean queue length of q_i if $s_{1,n}$ and $s_{2,n}$ were identically equal to $\bar{s}_1$ and $\bar{s}_2$, respectively. Also, we use the approximation $s_{i,n} \approx \bar{s}_i$ for $i = 1, 2$ in the RED drop probabilities.

Let $p_1(\bar{s}_1)$ and $p_2(\bar{s}_2)$ denote the probabilities of the RED algorithm discarding packets from classes 1 and 2, respectively. Observe that the arrival processes thinned by the RED drop are again Poisson arrivals with intensities $\lambda_1(1 - p_1(\bar{s}_1))$ and $\lambda_2(1 - p_2(\bar{s}_2))$, respectively. Moreover, as the packet lengths in both classes are identical, we may consider a queue with the aggregated arrival process and independently label each packet into class 1 with probability

$$\frac{\lambda_1(1 - p_1(\bar{s}_1))}{\lambda_1(1 - p_1(\bar{s}_1)) + \lambda_2(1 - p_2(\bar{s}_2))}$$

and similarly for class 2.

Therefore, the expected numbers of queued packets in classes 1 and 2 are given by

$$E_\pi[q_1^{(s_1,s_2)}] = \frac{\lambda_1(1-p_1(\bar{s}_1))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))} E_\pi[q^{(s_1,s_2)}],$$
$$E_\pi[q_2^{(s_1,s_2)}] = \frac{\lambda_2(1-p_2(\bar{s}_2))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))} E_\pi[q^{(s_1,s_2)}],$$

where $E_\pi[q^{(s_1,s_2)}]$ is the stationary mean queue length of a M/M/1/K queue with load $\rho = \lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))$, i.e.,

$$E_\pi[q^{(s_1,s_2)}] = \frac{\rho}{1-\rho} - (K+1)\frac{\rho^{K+1}}{1-\rho^{K+1}}.$$

From now on, in this paper, we simplify the notation so that $E_\pi[q^{(s_1,s_2)}]$ is replaced by $E_\pi[q]$. Hence our ODE approximation for two uncoupled classes is given by

$$\begin{aligned} \frac{d}{dt}\bar{s}_1 &= \lambda_1 w \left(\frac{\lambda_1(1-p_1(\bar{s}_1))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))} E_\pi[q] - \bar{s}_1 \right), \\ \frac{d}{dt}\bar{s}_2 &= \lambda_2 w \left(\frac{\lambda_2(1-p_2(\bar{s}_2))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))} E_\pi[q] - \bar{s}_2 \right). \end{aligned} \tag{3}$$

4.2 RIO, two coupled classes

Next we derive the ODE approximation for two coupled classes. In this paradigm class 1 has a higher priority than class 2. In RED algorithm this in implemented by taking the RED drop in class 2 to depend on congestion levels in both classes. That is, congestion in class 1 may result in discarding packets from class 2, whereas in terms of RED the class 1 operates independently from class 2.

ODE approximation can be derived as in the previous section with only minor modifications. We take RED drop functions p_1 and p_2 for classes 1 and 2, respectively, such that $p_1 = p_1(\bar{s}_1)$ but $p_2 = p_2(\bar{s}_{12})$, where we indicate the averaged total queue length by $\bar{s}_{12} = \bar{s}_1 + \bar{s}_2$. With this choice of RED drop functions we get a RIO system, an extension of RED algorithm introduced in [4].

The updating equations for the RIO system are given by $\Delta s_{1,n} = w\,(q_{1,n} - s_{1,n})$ and $\Delta s_{12,n} = w\,(q_{12,n} - s_{12,n})$, where $q_{12,n} = q_{1,n} + q_{2,n}$. In taking expectations of the updating equations we condition on the number of arrival events A_1 and A_2 in classes 1 and 2 in the following manner:

$$\begin{aligned} E[\Delta s_1(t)] &= \sum\nolimits_{j=0}^{\infty} E[\Delta s_1(t) \mid A_1 = j] P\{A_1 = j\} \\ &= wE[q_1(t) - s_1(t)]\lambda_1(t)\Delta t + O(\Delta t^2) \end{aligned}$$

and

$$\begin{aligned} E[\Delta s_{12}(t)] &= \sum\nolimits_{j=0}^{\infty} E[\Delta s_{12}(t) \mid A_1 + A_2 = j] P\{A_1 + A_2 = j\} \\ &= wE[q_{12}(t) - s_{12}(t)](\lambda_1(t) + \lambda_2(t))\Delta t + O(\Delta t^2). \end{aligned}$$

The approximation for $E[q_1]$ and $E[q_{12}] = E[q]$ does not change with priorities of the classes, and hence letting Δt approach zero we arrive at the ODE approximation for the RIO system:

$$\begin{aligned} \frac{d}{dt}\bar{s}_1 &= \lambda_1 w \left(\frac{\lambda_1(1-p_1(\bar{s}_1))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_{12}))} E_\pi[q] - \bar{s}_1 \right), \\ \frac{d}{dt}\bar{s}_{12} &= (\lambda_1+\lambda_2) w \left(E_\pi[q] - \bar{s}_{12} \right), \end{aligned} \tag{4}$$

where we denote $\bar{s}_1 = E[s_1]$, $\bar{s}_{12} = E[s_{12}]$ and $E_\pi[q]$ stands for the stationary mean queue length of a M/M/1/K queue with load $\rho = \lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_{12}))$.

5 Numerical results

In this section our main focus is on the equilibrium of the expected averaged queue sizes. Due to space limitations we illustrate only results in the RIO setting. We also consider the time evolution of the expected values, comment on the impact of the counter omission, and show how to adjust our model in order to approximate RED with the counter.

5.1 Equilibrium

We have studied the equilibria of ODE approximations (3) and (4) with different RED parameters; in a symmetric case we have the same RED parameters $p_{i,\max} = 0.1$, $T_{i,\min} = 10$ and $T_{i,\max} = 30$ for both classes 1 and 2, whereas in the nonsymmetric case class 2 has worse RED parameters $p_{2,\max} = 0.5$, $T_{2,\min} = 5$ and $T_{2,\max} = 20$ while the first class has the previous parameter values. Parameter choice for class 1 is according to guidance given in [9], namely $p_{1,\max} \leq 0.1$, $T_{1,\min} \geq 5$ and $T_{1,\max} \geq 3 \times T_{1,\min}$.

We consider a buffer of size $K = 40$. Our aim is to study $\bar{s}_1 = E[s_1]$ and $\bar{s}_2 = E[s_2]$ at the equilibrium (together with other performance characteristics derived from those) as a function of constant loads (intensities) from sources 1 and 2.

Equilibrium equations for two uncoupled classes can be obtained from (3) yielding:

$$\begin{aligned} E[q_1] - \bar{s}_1 &= \frac{\lambda_1(1-p_1(\bar{s}_1))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))} E_\pi[q] - \bar{s}_1 = 0, \\ E[q_2] - \bar{s}_2 &= \frac{\lambda_2(1-p_2(\bar{s}_2))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_2))} E_\pi[q] - \bar{s}_2 = 0. \end{aligned} \tag{5}$$

Similarly, the equilibrium equations for a RIO system can be written in terms of $\bar{s}_1$ and $\bar{s}_2$. Observe that $E[q] = E[q_1] + E[q_2]$ and similarly $\bar{s}_{12} = \bar{s}_1 + \bar{s}_2$, so the equilibrium equations from (4) give us

$$\begin{aligned} E[q_1] - \bar{s}_1 &= \frac{\lambda_1(1-p_1(\bar{s}_1))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_1+\bar{s}_2))} E_\pi[q] - \bar{s}_1 = 0, \\ E[q_2] - \bar{s}_2 &= \frac{\lambda_2(1-p_2(\bar{s}_1+\bar{s}_2))}{\lambda_1(1-p_1(\bar{s}_1)) + \lambda_2(1-p_2(\bar{s}_1+\bar{s}_2))} E_\pi[q] - \bar{s}_2 = 0. \end{aligned} \tag{6}$$

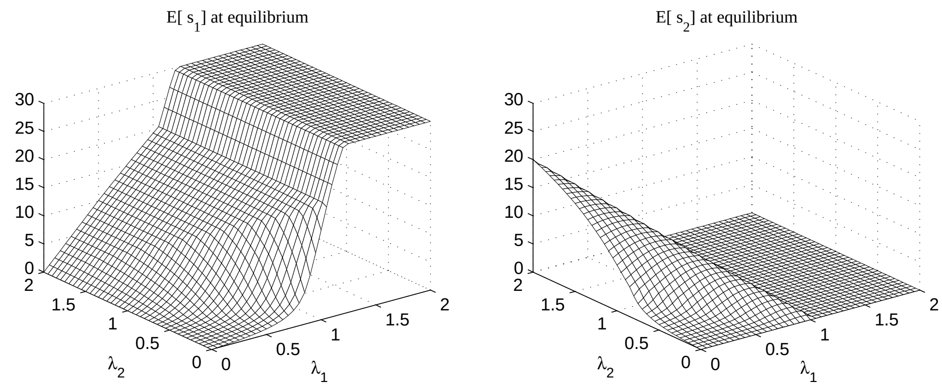

Figure 2: Equilibrium of $\bar{s}_1$ and $\bar{s}_2$ for a RIO system.

It turns out that the approximations for $E[q_1]$ and $E[q_2]$ do not work well in cases with small $p_{\max}$-values but large intensities. For simplicity, we explain the approximation in the context of one Poisson class. The approximation assumes that arrival process is a Poisson process that is thinned by the factor of $(1-p)$, where p is the RED drop probability. However, with large load and small $p_{\max}$, s_n increases fast to some value above $T_{\max}$, and RED algorithm discards all arriving packets. Gradually s_n decreases until values below $T_{\max}$ result in a burst of accepted packets, increase in s_n, and a period of discarging again all packets. This oscillatory behavior of s_n together with almost on-off behavior of accepting packets is not approximated well by a thinned arrival process. (See also [13] for observation on consecutive packet drops under high loads as w tends to zero.)

In our model the above problem arises with uncoupled classes when λ_2 is small but λ_1 is large, or vice versa. In the first case we define the equilibrium of $\bar{s}_1$ at $T_{1,\max}$, and correspondingly in the other case. For coupled classes, when $\lambda_1 + \lambda_2$ is large, we define an equilibrium state $(\bar{s}_1, \bar{s}_2)$ such that $\bar{s}_1 + \bar{s}_2 = T_{2,\max}$ (assuming that $T_{2,\max} \leq T_{1,\max}$), unless $\bar{s}_2 = 0$. We stress out that our analysis is for the expected values of $s_{1,n}$ and $s_{2,n}$ at equilibrium. In the above cases single realizations of $s_{1,n}$ and $s_{2,n}$ oscillate around the expected values.

For numerical reasons (due to the fact that equilibria may lie close to discontinuities of the RED drop functions), we approximate the RED drop function with a continuous piecewise linear function that differs from the original RED drop function by the linear increase from $p_{\max}$ to 1 in $[T_{\max}, T_{\max} + 1/2]$. In general, continuous RED drop functions have been proposed to reduce the algorithm's sensitivity to the $T_{\max}$ and $p_{\max}$ parameters. However, our linear increase of the drop probability from $p_{\max}$ to 1 is less "gentle" than the recent "gentle_"-mode extension of the RED algorithm implemented in the NS simulator, see [6] and references therein.

Figure 2 illustrates equilibria of $\bar{s}_1$ and $\bar{s}_2$ for a RIO system with RED parameters $p_{1,\max} = 0.1$, $T_{1,\min} = 10$, $T_{1,\max} = 30$, $p_{2,\max} = 0.5$, $T_{2,\min} = 5$, and $T_{2,\max} = 20$. We see that for $\lambda_1 \geq 1.05$, RED control completely discards packets from class 2 and $\bar{s}_1$

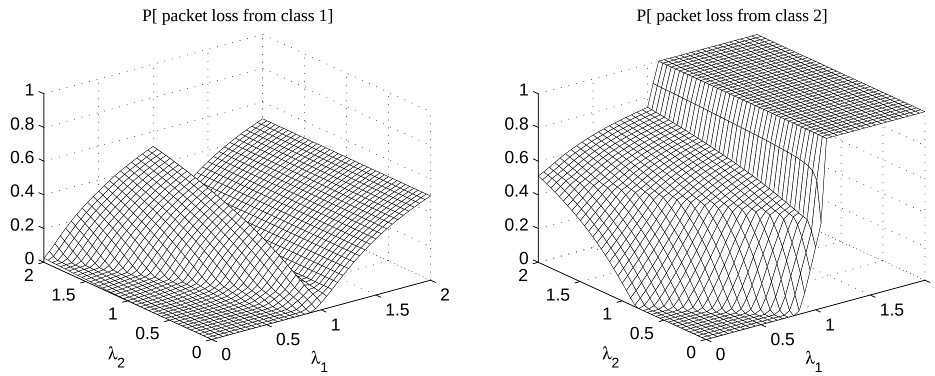

Figure 3: Probabilities of losing packets for a RIO system at the equilibrium.

increases fast to $T_{1,\max}$. For intensities λ_1 and λ_2 so that $\lambda_1+\lambda_2 \geq 1$ but $\lambda_1 \leq 1.05$ the equilibrium satisfies $\bar{s}_1+\bar{s}_2 = T_{2,\max}$. If the priority of class 2 is increased by using the same RED parameters for both classes, the surface of $\bar{s}_1$ as a function of the intensities is similar without the rapid incease observable in Figure 2 for λ_1 increasing from 1.05 to 1.2.

Next we illustrate the proportion of lost traffic at the equilibrium as a function of intensities λ_1 and λ_2. For each arriving packet the RED algorithm described in Section 2 checks if the buffer is full and applies the random drop. Thus, using the PASTA property, in RIO setting (and for two uncoupled classes with the modification $p_2 = p_2(\bar{s}_2)$) the probabilities of losing a packet in classes 1 and 2 are given by

$$P_{l,1}(\bar{s}_1,\bar{s}_2) = 1-(1-\mathrm{Ovf}(\rho,K))\,(1-p_1(\bar{s}_1))\,,$$
$$P_{l,2}(\bar{s}_1,\bar{s}_2) = 1-(1-\mathrm{Ovf}(\rho,K))\,(1-p_2(\bar{s}_1+\bar{s}_2))\,,$$

where $\mathrm{Ovf}(\rho,K) = \rho^K / \sum_{i=0}^{K} \rho^i$ is the overflow probability of a M/M/1/K queue with load $\rho = \lambda_1\,(1-p_1(\bar{s}_1)) + \lambda_2\,(1-p_2(\bar{s}_1+\bar{s}_2))$.

Figure 3 illustrates the probability of losing packets from classes 1 and 2 with the RED parameter setting of Figure 2. The main observation is that for fixed intensity λ_2, $P_{l,1}$ is not an increasing function of the intensity λ_1. We point out that the probability of losing packets from class 1 sharply decreases when $\lambda_1 \approx 1.05$, and then starts to increase again. Decrease in the packet loss probability takes place when the RED algorithm discards packets from class 2, and the increase in the packet loss thereafter is due to load from class 1 packets solely.

Similarly, for class 2 we observe around $\lambda_1 \approx 1.05$ a sharp increase in packet losses when RED algorithm starts to discard class 2 completely. However, now for any fixed intensity λ_1, the probability of packet losses in class 2 is a nondecreasing function of λ_2.

Again, we comment on packet losses with other RED parameter settings. In a RIO paradigm such that both classes have the same RED parameters $p_{\max} = 0.1$, $T_{\min} = 10$ and $T_{\max} = 30$, for fixed λ_1 the packet loss probability for class 1 is an

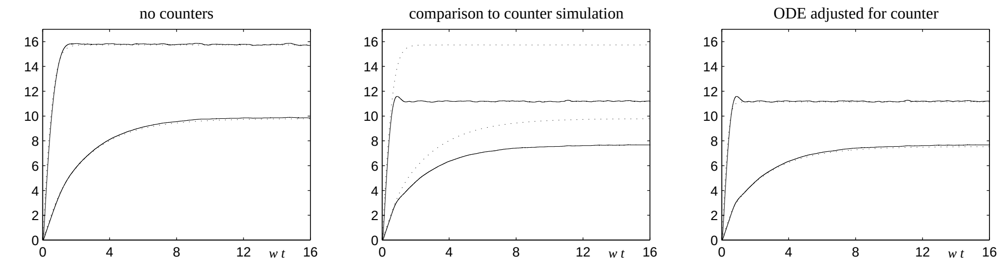

Figure 4: Comparing the ODE approximation with simulated queue, two uncoupled classes.

increasing function of λ_2. Moreover, for fixed λ_2 in this case the packet losses in class 1 increase monotonically as λ_1 increases. Contrary to Figure 3 the packet losses appear to be almost symmetric in λ_1 and λ_2. The surface describing the packet losses for class 2 is similar to the one for class 1, with the distinction that packet losses are higher (by 0.1 or less).

5.2 Comparison with simulated systems

In addition to equilibrium studies the ODE approximations (3) and (4) allow us to study the time evolution of $\bar{s}_1$ and $\bar{s}_2$, and compare those with simulated systems with and without the counter process. Recall that our model was simplified by omitting the counter process in the original RED algorithm. The purpose of the counter in to distribute the consecutive packet drops more evenly (and in that way to help to reduce the probability of the global TCP synchronization effect). Let X denote the number of accepted packets between two drop events and assume that the arriving packets see the same value of $s_n = s$, which is a good approximation for small values of w. Floyd and Jacobson [8] have shown that with the counter process X is uniformly distributed in $\{0, 1, \dots, \lfloor 1/p(s) \rfloor\}$, with mean $E[X] = \frac{1}{2p(s)} + \frac{1}{2}$, where $p(s) = p_{\max}(s - T_{\min})/(T_{\max} - T_{\min})$. Without the counter process X would be geometrically distributed with mean $E[X] = 1/p(s)$.

We look for the probability $\tilde{p}(s)$ for the geometric distribution so that the means of the above distibutions match, resulting to

$$\frac{1}{2p(s)} + \frac{1}{2} = \frac{1}{\tilde{p}(s)} \Rightarrow \tilde{p}(s) = \frac{2p(s)}{1 + p(s)} \approx 2p(s). \tag{7}$$

Therefore, to approximate the effect of the counter process, we adjust in our model the values of $p_{\max}$, i.e., for both classes we redefine $p_{i,\max} := 2p_{i,\max}$ (see also [7], where a similar approximation has been used.)

The effect of the correction is with two uncoupled classes approximated by (3). The RED parameters are taken to be $w = 0.001$, $p_{1,\max} = 0.1$, $T_{1,\min} = 10$, $T_{1,\max} = 30$, $p_{2,\max} = 0.5$, $T_{2,\min} = 5$, $T_{2,\max} = 20$, together with constant intensities $\lambda_1 = 0.4$, $\lambda_2 = 1.0$ and the buffer size $K = 40$.

Figure 4 presents the comparison of the ODE approximation (3) with simulated systems in which arrivals into an empty queue are handled as in the analytical model. The numerical solution of the ODE (3) is graphed using dashed lines whereas solid lines correspond to averages of 1000 simulated realizations from the queue evolution. The undermost pairs of solid and dashed lines correspond to $\bar{s}_1$, while the topmost pairs illustrate $\bar{s}_2$. The leftmost graph in Figure 4 shows the solution of (3) and the simulated system without the counter process. The equilibrium values are $\bar{s}_1 = 9.8$ and $\bar{s}_2 = 15.7$. Here we see that the approximation agrees well with the RED algorithm without the counter. However, in the center graph of Figure 4 we compare the ODE approximation with the evolution of the RED algorithm in which the counter process has been implemented, and we note that the ODE approximations are above the simulated curves. The problem is corrected in the rightmost graph, where we have adjusted $p_{i,\max} := 2p_{i,\max}$ for $i = 1, 2$. The last graph shows that the adjusted ODE approximation can model well the behavior of the RED algorithm with the counter process.

We finish this section with some remarks about the weight parameter w. Our model is based on approximating $E[q]$ by $E_\pi[q^{(s)}]$, i.e., assuming that $\{s_n\}$ process is slow in comparison to $\{q_n\}$. This approximation can be done if w is small. However, we have observed that for a RIO system with high load ($\lambda_1 = 0.9$ and $\lambda_2 = 1.5$) a w parameter value as low as 10^{-5} was needed so that $\bar{s}_1 + \bar{s}_2$ from (4) approximates well (difference less than 0.4) $\bar{s}_1 + \bar{s}_2$ from the simulated system without counter process. On the other hand, uncoupled systems in general and RIO with similar RED parameters are easier to approximate than the case we have focused on. For example, the time evolution for two uncoupled classes with RED parameters $w = 0.001$, $p_{i,\max} = 0.1$, $T_{i,\min} = 10$ and $T_{i,\max} = 30$ for $i = 1, 2$, and intensities $\lambda_1 = \lambda_2 = 1.5$ together with the counter process can be approximated well even without the adjustment in $p_{i,\max}$-values.

6 Summary

We provided first steps towards analytical models for the behavior of the RED algorithm with two traffic classes, which hopefully helps in gaining a better understanding of Differentiated Service implementations utilizing the RED algorithm. The analysis of the RED algorithm was simplified by omitting the counter process and later we adjusted parameters to model RED with the counter. Our ODE model was based on the stationarity approximation motivated by the different time scales of the queue length averaging process and the instantaneous queue size process.

The ODE approximation was used to study the time evolution of the expected averaged queue sizes $\bar{s}_1$ and $\bar{s}_2$ in each class. Moreover, the approximation allows us to study the equilibrium values of $\bar{s}_1$ and $\bar{s}_2$ as functions of loads from the two traffic sources. Such analysis via simulations of the queue behavior would require excessive computations. From the equilibrium results we point out that the qualitative behavior of packet loss probability in class 1 depends on the RED parameters for class 2 in the RIO setting. Lower $T_{\min}$ and $T_{\max}$ parameter values for class 2 result in very rapid discarding of class 2 in some cases, which may not be desirable.

Future extensions of the work include incorporating the ongoing modeling work on

TCP sources into the setting of Differentiated Services. This extension also relaxes the assumptions on the packet size distribution.

Acknowledgments

Authors are thankful to Pasi Lassila for a Matlab code simulating the RIO system and for helpful discussions on parameter adjustment.

References

[1] S. Blake et al., *An architecture for differentiated services*, Internet RFC-2475, December 1998.

[2] B. Braden et al., *Recommendations on queue management and congestion avoidance in the internet*, Internet RFC-2309, 1998.

[3] Cisco, *Cisco ios configuration*, `http://www.cisco.com`, 1999.

[4] D. Clark and W. Fang, *Explicit allocation of best-effort packet delivery service*, IEEE/ACM Transactions of Networking **6** (1998), no. 4, 362–373.

[5] Bernet et al., *A framework for differentiated services*, Internet draft, February 1999.

[6] S. Floyd, *Recommendations on using the "gentle_" variant of RED*, available at `http://www.aciri.org/floyd/red/gentle.html`, March 2000.

[7] S. Floyd and K. Fall, *Ns simulator tests for Random Early Detection (RED) queue management*, technical report dated April 29, 1997, available at `http:/www.aciri.org/floyd/papers.html`.

[8] S. Floyd and V. Jacobson, *Random Early Detection gateways in congestion avoidance*, IEEE/ACM Transactions on Networking **1** (1993), no. 3, 397–413.

[9] S. Floyd et al., *Discussions on setting parameters*, `http://www.aciri.org/floyd/REDparameters.txt`, November 1997.

[10] P. Lassila and J. Virtamo, *Modeling the dynamics of the RED algorithm*, to be presented in QofIS'00, available at `http://www.tct.hut.fi/tutkimus/com2/publ/redoscil.pdf`, February 2000.

[11] D. Lin and R. Morris, *Dynamics of random early detection*, Proceedings of ACM SIGCOM, 1997.

[12] M. May, J. Bolot, C. Diot, and A. Jean-Marie, *Simple performance models for differentiated services schemes for the internet*, Proceedings of IEEE INFOCOM, March 1999.

[13] M. May, T. Bonald, and J. Bolot, *Analytic evaluation of RED performance*, Proceedings of IEEE INFOCOM, March 2000.

[14] S. Peeters and C. Blondia, *A discrete time analysis of random early detection with responsive best-effort traffic*, Proceedings of IFIP ATM & IP Workshop, 1999.

[15] V. Sharma, J. Virtamo, and P. Lassila, *Performance analysis of the Random Early Detection algorithm*, submitted for publication, available at `http://www.tct.hut.fi/tutkimus/com2/publ/redanalysis.pdf`, 2000.