Equation-Based Packet Marking for Assured Forwarding Services

Mohamed A. El-Gendy and Kang G. Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109-2122
{mgendy,kgshin}@eecs.umich.edu

Abstract— **This paper introduces a new packet marking algorithm that can be used in the context of *Assured Forwarding* (AF) in the *Differentiated Services* (DiffServ) framework [1], [2]. The new marking algorithm is called *Equation-Based Marking* (EBM) and is based on the TCP model in [3]. EBM is to handle the problems found in other marking schemes regarding fairness among heterogeneous TCP flows through a tight feedback-loop operation and adaptation of the packet marking probability to network conditions. We design a packet marker that uses EBM as the marking algorithm, and evaluate its performance using in-depth simulation. We also prove analytically the correctness of the marking algorithm and compare it with other marking schemes for different network scenarios. Our evaluation results demonstrate the effectiveness of EBM in providing the required fairness among heterogeneous flows and ensuring protection against non-assured traffic.**

I. Introduction

DIFFSERV is one of the recent proposals to provide Quality-of-Service (QoS) to IP networks, especially the Internet [1]. The DiffServ framework provides a small, yet effective, number of services which can be used to build end-to-end QoS for different applications on the Internet. Due to its simplicity and scalability, DiffServ has been receiving considerable attention.

Basically, the DiffServ framework introduces two[1] additional packet-handling schemes based on *Per-Hop Behaviors* (PHBs), besides the basic best-effort delivery mechanism used in the current Internet [2]. The two basic PHBs defined for DiffServ are the *Expedited Forwarding* (EF) and the *Assured Forwarding* (AF) [4] PHBs. The EF PHB is used to build services that require low delay, low jitter, low loss, and assured bandwidth like the *Virtual Leased Line* (VLL) services, while the AF PHB is used to build more "elastic" services that impose requirements only on throughput without any delay or jitter restrictions.

The idea behind the AF PHB is to differentiate packets by marking them, based on conformance to their target throughputs. Non-conformant packets are called *out-of-profile* (OUT), while conformant packets are called *in-profile* (IN). Then, by using a differentiated Random Drop Gateway like RIO [5] or a more general form thereof, at the time of congestion, OUT packets are more likely to be dropped than IN packets. This, in effect, protects IN packets from OUT ones, giving applications their required bandwidths. For three-color AF, IN packets are labeled *Green*, and OUT packets are divided into *Yellow* and *Red* in order to exert more control on the available bandwidth of the network. These three colors correspond to the AF's three drop precedences, AFx1, AFx2, and AFx3, respectively.

Recently, significant efforts have been invested into the performance evaluation of TCP congestion control for a random drop gateway like RED [6], or generalization thereof like RIO [5]. Some of these efforts placed an emphasis on AF services. The authors of [7], [8] found unfairness — in sharing the extra bandwidth in under-subscribed networks, or degrading performance in over-subscribed networks — among the TCP aggregates that have different round-trip times (RTTs), average packet sizes, target rates, or numbers of micro-flows in the aggregate. They also considered different models for the drop gateway configuration, such as RIO-C, WRED, overlapped, non-overlapped, single and multiple averages. Similar results have been reported in [9] and [10], the latter of which has also evaluated the difference in using token bucket and average rate estimator policing (marking).

On the other hand, packet marking algorithms have been proposed to work with AF, such as the *Two Rate Three-Color Marker* (TCM) in [11], which is based on token bucket metering, and the *Time Sliding Window Three Color Marker* (TSWTCM) in [12], [13] which is based on the average rate estimator algorithm in [5]. In these marking schemes, two target rates are defined: *Committed Information Rate* (CIR) and *Peak Information Rate* (PIR). The former is the minimum requirement to be achieved, and the latter is for a surplus of bandwidth when the network is lightly-loaded. An enhanced version of Time Sliding Window (TSW), called *Enhanced Time Sliding Window* (ETSW), was proposed in [14] to handle the TCP dynamics and over-marking in the original TSW marker.

An excellent analytical study of the achievable performance for TCP using token bucket marking is reported in [15] where the authors prove that token bucket markers cannot achieve all values of assured rates and the achieved rate is not proportional to the assured rate. They also introduced a method of choosing the correct profile to achieve a given service level.

The work reported in this paper was supported in part by Samsung Electronics, Inc. and by the Office of Naval Research under Grant No. N00014-99-1-0465.

[1] In fact, there are others, e.g., Class Selector Per-Hop Behavior group, but not as important as these two.

Unfortunately, these types of marking algorithms do not handle, nor have been evaluated with respect to the unfairness issues mentioned above. The authors of [7], [8], as well as our extensive simulation, find that all these marking algorithms suffer the following fact: TCP flows with different RTTs cannot achieve throughput proportional to their target rates. The same phenomenon occurs for TCP flows with different target rates, different mean packet sizes, or aggregates with different numbers of micro-flows. This is not acceptable in the operation of AF.

Several remedies have been proposed to overcome these fairness problems, such as the *TCP-Friendly* marker [16] and the *Fair Marker* [17], but they suffer from their inherent complexity, nor has their performance been evaluated. Two adaptive marking algorithms have been proposed in [18] and [19] as *Adaptive Packet Marking* (APM) and *Intelligent Traffic Conditioners* (rtt-aware and target-aware), respectively. APM is found to perform well in tracking the dynamics of TCP and preserving the target rate, but it is based on an inaccurate feedback model which causes performance fluctuations. Moreover, APM has to be implemented inside the TCP code itself, requiring modification of all TCP agents to be able to use this marking algorithm. The Intelligent Traffic Conditioner tries to use the simple TCP model in [20] to handle the unfairness associated with different RTTs and different target rates. This is somewhat similar to the approach proposed in this paper, except that these conditioners require external inputs and cooperation among markers for different traffic aggregates, which tends to be very complex in implementation and deployment.

We propose in this paper a new AF packet marking algorithm that solves most of the unfairness and protection problems associated with the other marking schemes. First, we define the fairness term as follows: *"In an under-subscribed network, all flows should get a share of the excess bandwidth proportional to their target rates (an equal share for equal target rates). In an over-subscribed network, all flows should experience throughput degradation proportional to their target rates (equal degradation for equal target rates)"*.

Our marking algorithm is called *Equation-Based Marking* (EBM), and works similarly to TCP, but on the packet-marking level. It senses the current network conditions and adapts the packet marking probabilities (IN and OUT for two-color marking; Green, Yellow, and Red for three-color marking) accordingly. This adaptation adjusts the loss probabilities of heterogeneous TCP flows, thus providing an equal or proportional share of extra bandwidth in under-subscribed networks, and an equal or proportional degree of throughput degradation in over-subscribed networks. The new marking algorithm predicts the behavior of the TCP sender and adjusts the marking probabilities accordingly, distinguishing itself from the other approaches to AF marking. This behavior gives a close interaction between the marker and the TCP sender.

The paper is organized as follows. In Section II, we discuss the main idea behind EBM and give a brief overview of the operation of EBM. Design of the EBM is detailed in Section III with a brief reasoning of how it works against unfairness. We present an analysis for the operation of EBM in Section IV and outline the proof of its correctness. Section V presents the results of EBM performance evaluation under different parameters and comparing EBM with the other marking algorithms. We end the paper with concluding remarks in Section VI.

II. Equation-Based Marking

EBM solves the unfairness problems associated with the other AF marking schemes by using a compact feedback control model based on the TCP model which was introduced in [3] and is summarized in AppendixA for completeness. The model encodes all the previously-mentioned factors affecting performance, in a single equation, Eq. (5).

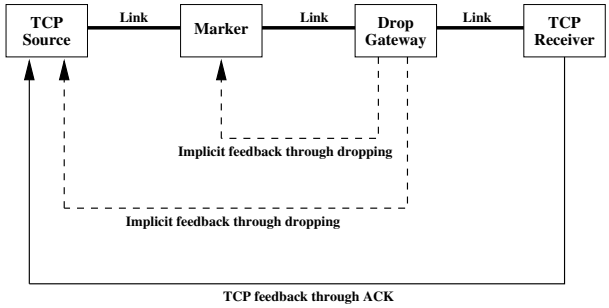

Fig. 1. Feedback loop operation of EBM

The marker works in the feedback loop shown in Figure 1 by sensing the losses that the TCP connection experiences, and tries to estimate the current network conditions and adjust the marking probabilities accordingly.

EBM works just like TCP which adjusts the sending rate by sensing the level of congestion in the network via observation of packet losses. So, EBM, through the innermost feedback loop in Figure 1, controls the TCP congestion control feedback loops and provides appropriate marking to achieve the required target rate. This, in effect, provides fairness among different TCP flows as each flow has its packets marked, depending on whether it received its share of bandwidth or not.

EBM uses estimated loss probabilities, instead of estimated average throughput, in calculating the required marking probabilities, and uses the duality between throughput and loss probability as shown in Figure 2.

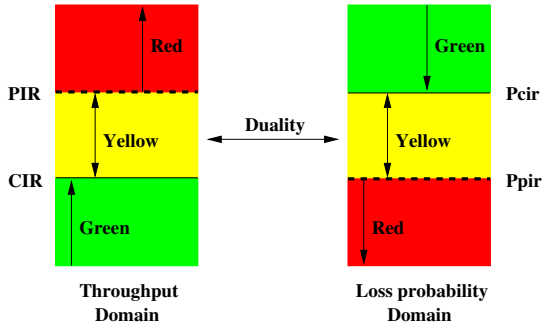

Fig. 2. Duality between throughput and loss probability

In this figure, we identify the target loss probabilities, p_{cir} and p_{pir}, corresponding to target throughput rates, CIR and PIR, respectively. EBM uses these two parameters in its operation, and calculates them from the TCP model in Eq. (5) using

the current network conditions like RTT, mean packet size, and maximum TCP window size. Then, as described later, it uses the current loss probability seen by this TCP flow as well as these target loss probabilities to calculate the packet-marking probabilities.

Several researchers attempted to model TCP congestion control [3], [20], [21] using model-based rate control [22], and equation-based congestion control [23]. However, EBM is different from these, as it works in the context of AF Services which employ packet marking to provide QoS differentiation based on TCP's reaction to packet losses.

III. The Design of EBM

The EBM is structured as shown in Figure 3. The marking engine makes use of two other modules, one for current RTT estimation (Section III-B), and the other for current loss probability estimation (Section III-C). It takes the updated values of RTT and current loss rate and puts them into the inverse of the TCP equation, $T_p^{-1}(r_t, RTT, T_o, W_{max})$[2], to get the target loss probabilities from the target throughput rates. This calculation is the key, as it adapts to the current values of loss probability, RTT, and the other derived parameters, reflecting different values for different TCP flows/aggregates. The target loss probabilities are used through an appropriate marking function to get the packet marking probabilities for the three colors: Green, Yellow, and Red.

Figure 4 illustrates the main steps in the EBM operation. These steps are executed periodically, and the execution period is tunable to achieve the best performance.

In the following sections we detail each of the building blocks of the EBM and the steps of the algorithm.

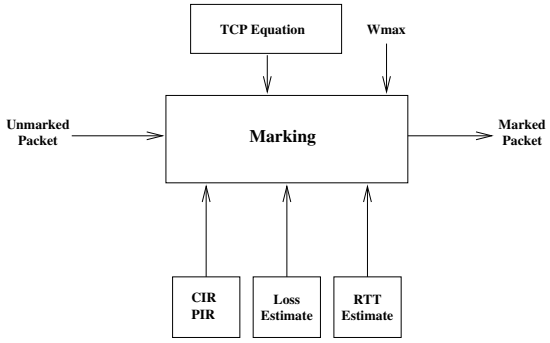

Fig. 3. Main blocks of the EBM

A. Calculation of the target loss probability

Eq. (5) indicates a one-to-one relationship between throughput, r_t, and loss probability, p, for given maximum window size, RTT, and mean packet size. EBM uses this relationship to derive the loss probabilities corresponding to the target throughput rates, CIR and PIR. As mentioned earlier, these loss probability values are called *target loss probabilities*, $p_{cir} = T_p^{-1}(CIR, RTT, T_o, W_{max})$ and $p_{pir} = T_p^{-1}(PIR, RTT, T_o, W_{max})$. This calculation requires the inverse of the TCP equation with respect to loss probability, p,

[2]We call T_p^{-1} the inverse of T w.r.t. p.

```
Each periodic interval:

1.  Use current estimate of round-trip time and time-out,
    RTT and T_o, respectively
2.  Use current estimate of loss probability, p_curr
3.  Use value of the maximum TCP window size, W_max
4.  Calculate maximum achievable throughput, p = 0,
    r_max = T(0, RTT, T_o, W_max)
5.  if (CIR > r_max)
6.    Exit with error ``Not able to achieve CIR and PIR''
7.    if (PIR > r_max)
8.      Warning ``Not able to achieve PIR''
9.    end if
10. end if
11. Calculate p_cir = T_p^-1(CIR, RTT, T_o, W_max)
          and p_pir = T_p^-1(PIR, RTT, T_o, W_max)
12. Calculate packet marking probabilities P_yellow and P_red
13. Mark packets
```

Fig. 4. EBM algorithm

but, unfortunately, there does not exist a closed form for this inverse. So, EBM numerically calculates the required values. The calculation process is tuned by changing the rate and/or accuracy of calculation.

One important step to be taken by the EBM algorithm is to ensure that the numerical iterations will converge to a correct value; otherwise, there is no need to iterate in the first place. This is the role of Steps 4-10, where the algorithm calculates the maximum achievable TCP throughput, r_{max}, with the current values of RTT, T_o, and W_{max} by putting $p = 0$[3] when evaluating T. Then, it compares the target throughput rates, CIR and PIR, to this maximum value, and if it is smaller than any of the target rates, then this target rate can not be achieved under the current circumstance. Accordingly, the numerical iteration will not converge to a correct value. The proof to this convergence criterion is given in Section IV-C.

B. Estimation of RTT and T_o

This module uses a method similar to the one used in TCP's estimation of RTT with two differences. First, the estimation procedure uses the timestamp option in the TCP header in order to estimate RTT and T_o with a high resolution. Second, the module is located outside the sender TCP, so it reads and modifies the packets' TCP headers in order to use the timestamp option. The estimation procedure is depicted in Figure 5, where *srtt* is the estimated RTT and T_o the estimated retransmit time-out.

C. Estimation of the current loss probability

Using the average loss interval method in [23], EBM estimates the current loss probability of the network seen by this flow in particular.

[3]In fact, there is no value for T at $p = 0$, but we use a very small value like 10^{-15}, instead.

```
At each estimation interval
Record and timestamp a packet of the TCP flow
Upon receiving an acknowledgment
/*Check if it is for the previously-recorded packet*/
if (seqno of the ack ≥ recorded seqno)
  current rtt := now − recorded timestamp
  /*Calculate the WMA of the measured RTT values*/
  srtt = w × rtt + (1 - w) × srtt
  delta = |rtt - srtt|
  rtt_var = w_rto × delta + (1 − w_rto) × rtt_var
  T_o = srtt + 4 × rtt_var
end if
```

Fig. 5. Estimation of RTT

For convenience, we describe the method here and give the details of detecting loss events[4] from the marker side, which is located outside the TCP source. The loss interval, s_i, is defined as the number of packets transmitted correctly between two loss events $(i, i-1)$. The estimated loss interval $\hat{s}_{(1,n)}$ is calculated as the weighted average of the last n intervals:

$$\hat{s}_{(1,n)} = \frac{\sum_{i=1}^{n} w_i s_i}{\sum_{i=1}^{n} w_i}$$

for weights w_i:

$$w_i = \begin{cases} 1 & 1 \le i \le n/2 \\ 1 - \frac{i-n/2}{n/2+1} & n/2 < i \le n \end{cases}$$

For the reasons mentioned in [23], EBM uses $n = 8$, giving weights of 1, 1, 1, 1, 0.8, 0.6, 0.4 and 0.2 for w_1 through w_8, respectively. The reported loss probability will be *loss_rate* = $1/\hat{s}$.

Detecting loss events from the marker side and outside the TCP sender requires some knowledge of packet losses. In TCP, a packet loss is detected when the sender receives three duplicate ACKs, or when the retransmit time-out expires, whichever occurs first. In both cases, and according to the TCP Reno's fast retransmit procedure [24], the lost packet is retransmitted. Detecting three duplicate ACKs at the marker is not a problem by using similar state variables used in TCP, but detecting packet loss that causes a time-out is a little bit harder, especially when a time-out timer is not used. Also, we ignore any loss within one RTT from a previous one, as mentioned in [23]. After taking several tuning procedures, a reasonable behavior of the loss estimator has been reached.

D. Marking Function

Based on the target loss probabilities, EBM uses a linear function to calculate the marking probabilities, P_{yellow} and P_{red} for the three-color marking case, proportional to p_{cir} and p_{pir} as shown in Figure 6. This marking function is similar to the one used in the TSWTCM marking scheme, but with throughput replaced by loss rate, and CIR and PIR replaced by p_{cir} and p_{pir}. The *YScale* and *RScale* are design parameters that take values which depend on the current network conditions.

```
if(loss_rate ≥ p_cir)
  Mark packet as GREEN
else if (p_cir > loss_rate ≥ p_pir)
  calculate P_yellow = (p_cir − loss_rate)/p_cir × (p_cir × YScale)
  with probability P_yellow mark packet as YELLOW and
  with probability (1 - P_yellow) mark packet as GREEN
else if (loss_rate < p_pir)
  calculate P_red = (p_pir − loss_rate)/p_pir × (p_pir × RScale)
  calculate P_yellow = (p_cir − p_pir)/p_pir × (p_pir × YScale)
  with probability P_red mark packet as RED and
  with probability P_yellow mark packet as YELLOW and
  with probability (1-(P_yellow + P_red)) mark packet as
  GREEN
```

Fig. 6. Marking function

To show how the marking function works, we present the case of four similar TCP flows but with different RTTs. We plot their p_{cir} and p_{pir} in Figure 7 and the resulting marking probabilities for the three-color case in Figure 8. We identify here that the marking function tries to adjust the packet marking probabilities to equalize the loss probabilities for the four different flows, and hence equalizing throughput (as there is a one-to-one relationship between throughput and loss probability); the flow with a higher percentage of Yellow, and Red will have more losses, and hence less throughput.

One should see the same behavior for the other factors, or for a combination thereof as well. This validates the operation of EBM in providing the required fairness among heterogeneous TCP flows or aggregates.

IV. Analysis of EBM Operation

In this section, we analyze the operation of EBM to show the correctness of the operation and examine the convergence of the numerical iterations taken to calculate the target loss probabilities. Here we chose a steady-state analysis, and will consider the dynamics of EBM in our future work. First, we present a correctness proof based on the feedback model of the marker with both TCP and the drop gateway. The approach to this proof has been inspired by the study in [25]. We present the proof here for the case of flows with different RTTs only. Using a similar procedure, the other cases of different target rates or different packet sizes can also be proved.

A. Assumptions

In order to make the proof tractable and simplify the mathematics involved, we make the following assumptions.

A1. We consider a system of n TCP flows passing through a common link ℓ with capacity C as shown in Figure 11, except for using a single-hop network. In Section V, we use a multi-hop network as a generalization, and the results are valid for that case as well. All

[4] A loss event is different from packet loss, as the former may consist of several packet losses within a round-trip time.

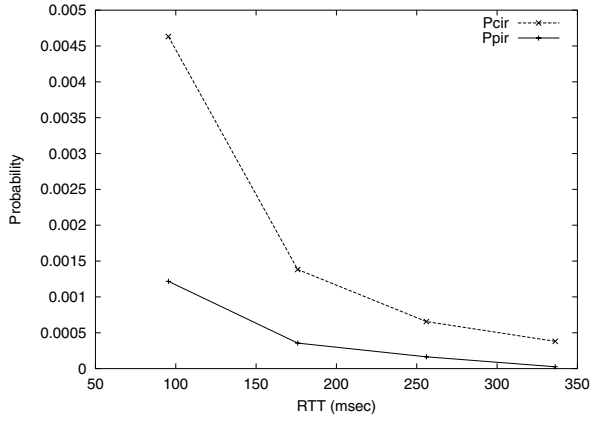

Fig. 7. Pcir and Ppir

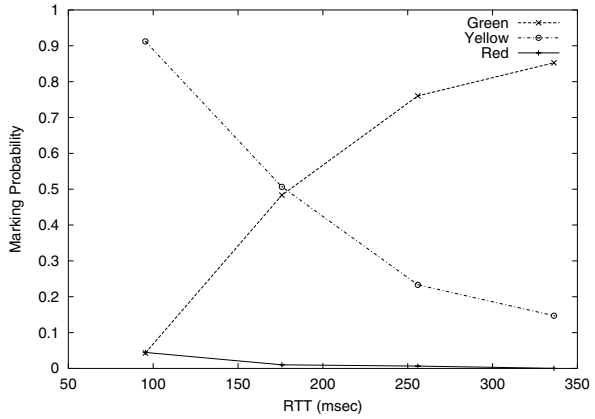

Fig. 8. Marking probabilities

other access links have enough capacity, so that link ℓ is the only bottleneck for all flows.

A2. All TCP flows have the same parameters except for RTTs, i.e., each flow sees a different RTT. We assume that the system is in the steady state (no slow start), and all TCP flows have unlimited data to send. We also assume that the system is under-subscribed, i.e., there is a surplus of bandwidth that can be allocated to each flow.

A3. Without loss of generality, we will use a two-color version of the RED drop gateway, called RIO [5], to make the proof simpler and tractable, but the results work for the three-color case as stated in Section V.

A4. Unlike the study in [25], we do not use the average queue size model of the RIO. Instead, we assume that a typical RIO drop gateway gives a fixed loss probability per class (IN and OUT) for all flows passing through this gateway. So, all flows see the same value of loss probabilities, $loss_{in}$, for IN packets, and $loss_{out}$ for OUT packets. Note that different flows may still see different total losses, depending on their parameters, but we only assume that loss probabilities in a class are the same for all flows. A justification for this assumption is given in AppendixB.

As mentioned before, the TCP model in AppendixA does not have a closed-form inverse with respect to loss probability p; neither does the approximate model given in [25], so we use numerical methods for this proof and discretize the problem as explained next.

B. Proof of Correctness

Theorem 1—Fairness: For a set of TCP flows, $f_i, 1 \leq i \leq n$, with same parameters and same target rate, *CIR*, but with different RTTs, RTT_i, when EBM is used for AF marking, they will get approximately the same goodputs, r_{t_i}. In other words, for all $i \neq j, 1 \leq i, j \leq n$, if $RTT_i \neq RTT_j$, then using EBM will result in $r_{t_i} = r_{t_j} \ \forall\, i$ and j.

Proof: Listed below are the steps taken for the proof.

1) For each value of RTT_i, calculate the TCP throughput $r_{t_i}(L_i, RTT_i)$ from Eq. (5), using the loss probability, L_i, seen by flow f_i.
2) The value of L_i is calculated from the RIO loss probabilities for each flow, depending on its packet marking probability, P_{m_i}.

$$L_i = P_{m_i} \times loss_{out} + (1 - P_{m_i}) \times loss_{in} \quad (1)$$

3) The value of the marking probability, P_{m_i}, is calculated for each flow using a similar marking function to the one in Figure 6 but for two colors only (IN and OUT).

$$P_{m_i} = \frac{p_{cir_i} - L_i}{p_{cir_i}} \times (p_{cir_i} \times Scale) \quad (2)$$

4) The value of p_{cir_i} is calculated numerically for each flow using its own RTT_i from the inverse of Eq. (5) with respect to p as:

$$p_{cir_i} = T_p^{-1}(CIR, RTT_i) \quad (3)$$

5) Solving for L_i using Eqs. (1) and (2), we get:

$$L_i = \frac{p_{cir_i} \times (loss_{out} - loss_{in}) \times Scale + loss_{in}}{1 + (loss_{out} - loss_{in}) \times Scale} \quad (4)$$

Then, substituting in Step 1 for each value of RTT_i, we get the final TCP throughput, r_{t_i}, for each flow f_i.

One instance of the results using these steps is shown in Figure 9 for $M = 576bytes$, $W_{max} = 256$, $Scale = 5000$, $C = 15Mbps$, $CIR = 0.5Mbps$, $n = 20$, $loss_{in} = 0.000066$, $loss_{out} = 0.00066$, and RTT is uniformly-distributed from 200ms to 900ms. The figure also shows the throughput values without using EBM, as well as the CIR value. The figure clearly shows the correct operation of EBM with respect to the required fairness among TCP flows with different RTTs. ■

C. Convergence of EBM Iterations

As mentioned in Section III-A, Eq. (5) describes a continuous and one-to-one relationship between r_t and p as shown in Figure 10, where T is plotted as a function of p for different RTTs ranging from 0.07s to 0.7s, $M = 576bytes$, and

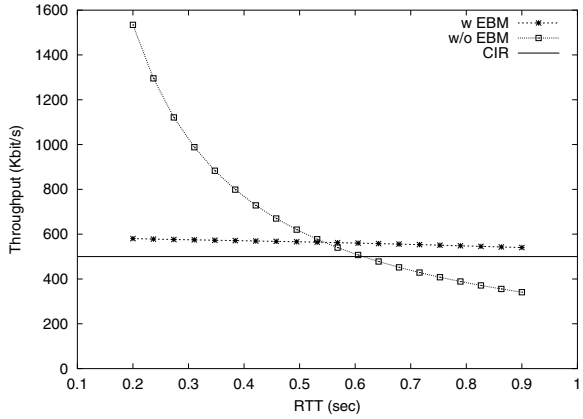

Fig. 9. Output of the analytical proof — throughput vs. RTT

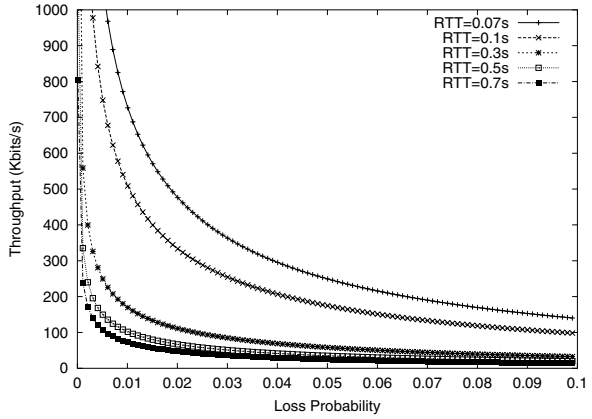

Fig. 10. Throughput vs. loss probability for different RTT

$W_{max} = 256$. We use the bisection method for finding the inverse of T w.r.t. p numerically [26], and to iterate for the values of p_{cir} and p_{pir}, we start from $p = 0$ and $p = 1$. These two values bound the whole scale of possible loss probability, meaning that a valid value of r can be calculated by the Intermediate Value Theorem [26]. However, T is a function of other variables like RTT, M, and W_{max}, which impose other limitations on the throughput achieved at certain p as shown in Figure 10. So, given some values for RTT, M, and W_{max}, there is a maximum value for the throughput, r_{max}, that can be achieved. This is the value calculated in Section III-A to compare with. By definition, r_{max} should be found at very small p, and this value is bounded for certain values of RTT, M, and W_{max}. Therefore, we first test for this condition in the EBM algorithm, and if the required target rates are below this maximum value, then the numerical iteration will always converge.

V. Evaluation

We use `ns-2` [27] to evaluate the performance of EBM and compare it with other marking algorithms. The network in Figure 11 is used for our evaluation.

Different scenarios have been simulated on this network to measure the performance for different parameters. In all these scenarios $n = 10$ TCP sources are used along with two UDP sources as background traffic generators. This background traffic represents the best-effort traffic in the Internet and uses CBR (Constant Bit Rate) and Red-colored (lowest-priority) packets.

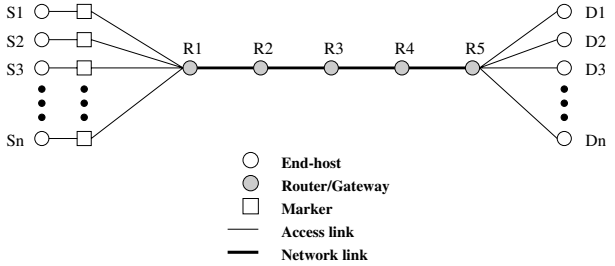

Fig. 11. Simulation topology

Edge routers do the metering and packet marking (represented as boxes in Figure 11) and core routers implement the Multi-RED (MRED) buffer management as a realization of the AF PHB. MRED uses a single average, non-overlapped (staggered) model and has the parameters listed in Table I where thresholds relative to the total queue length L are used.

TABLE I
MRED PARAMETERS

Parameters for	Green	Yellow	Red
Queue length	L	L	L
$\mathtt{Max}_{\mathtt{th}}$	0.875L	0.625L	0.3125L
$\mathtt{Min}_{\mathtt{th}}$	0.625L	0.3125L	0.025L
$\mathtt{Max}_{\mathtt{p}}$	0.02	0.05	0.1
$\mathtt{w}_{\mathtt{q}}$	0.002	0.002	0.002

Each TCP source generates an infinite FTP bulk data transfer, with its own target rates, CIR and PIR. The subscription level of the network is set by properly adjusting CIRs and the background rate.

All access links have a capacity of 100Mbps and a latency adjustable to the simulation scenario, while the links between core routers have a 10Mbps capacity and a 10ms latency.

The bottleneck is made to occur at the links between the core routers. Unless otherwise stated, a packet size of 576 bytes is used. In all the simulations, we measure the goodput achieved by each TCP flow using the Weighted Moving Average (WMA) technique with a 1-second window and a weight of 0.5–0.8. We then calculate the average goodput over the whole simulation period. Each simulation scenario is repeated 10 times, and then an average is taken over all runs. We use the following notations in the graphs:

- TCM Token bucket Three Color Marker.
- TSW Time Sliding Window three color marker.
- ETSW Enhanced Time Sliding Window marker.
- RPM Random Packet Marking.
- APM Adaptive Packet Marking.
- EBM Equation-Based Marking.
- CIR Committed Information Rate.
- PIR Peak Information Rate.

The EBM uses a 10sec interval between two successive calculations of p_{cir} and p_{pir}, and an accuracy of 0.005%, while using a *YScale* of 500 and a *RScale* of 1000. These values have been set empirically for the best performance of EBM with the current network configuration used in the simulation.

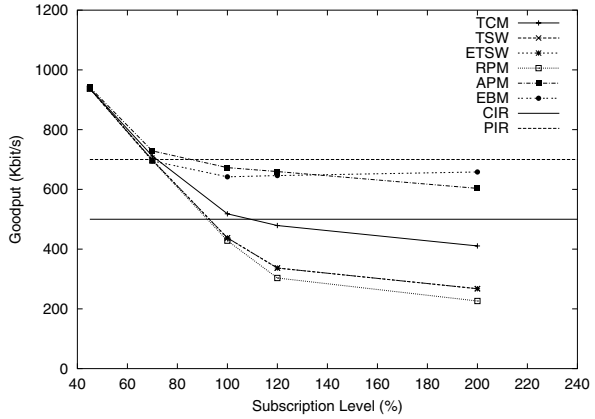

Fig. 12. Goodput vs. subscription level

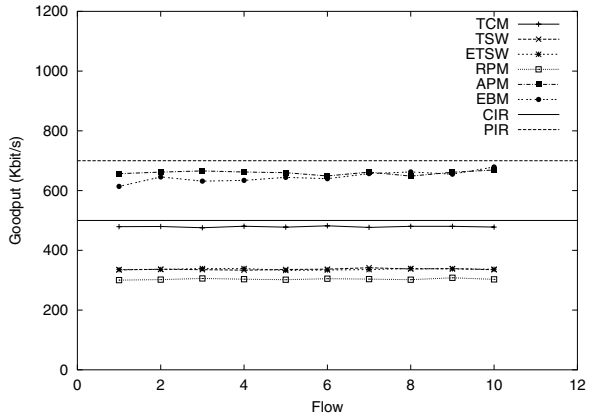

Fig. 13. Goodput per flow — over-subscription

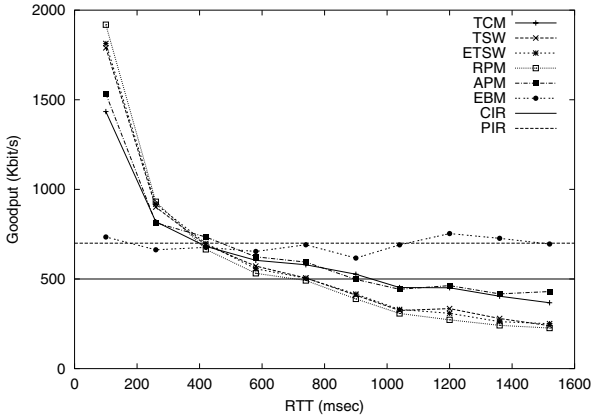

Fig. 14. Goodput vs. RTT — under-subscription

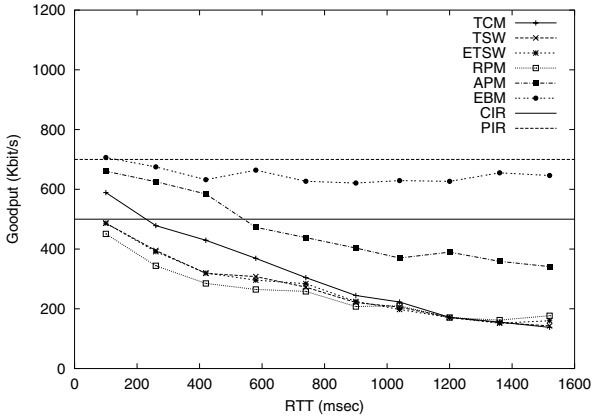

Fig. 15. Goodput vs. RTT — over-subscription

A. Subscription Level

In the first scenario, we investigate the effect of the network subscription level, or load, on the performance of EBM, comparing it with other marking schemes. The subscription level is changed from a light load (45%) to a heavy overload (200%), and the results are plotted in Figure 12. A CIR value of 500Kbps and a PIR value of 700Kbps are used for all markers, and by changing the background rate, we achieve the required subscription level in the network.

From the figure, we see that EBM has better protection and adherence to CIR than the other marking schemes under a broad range of network loads. Of course, under a very light load, all the markers can achieve good throughput, as there is a large surplus of capacity in the bottleneck links. On the other hand, for heavy network loads, EBM is superior to others in protecting the AF traffic from background traffic to sustains its CIR and get as much bandwidth as they can from the extra capacity towards PIR. To show the fairness among different flows, Figure 13 plots the goodput per flow for a 120% subscription level. All marking schemes have the same fairness in this case for similar flow parameters.

Note that APM also yields similar performance on a large time scale, but on a small time scale, APM causes more fluctuations in the achieved throughput than EBM. This characteristic is also experienced in all other scenarios.

B. RTT

Using a range of RTTs from 100ms to 1520ms for different TCP flows, by changing the latencies of access links, we evaluate the fairness and performance of EBM against the other marking schemes in moderately- and heavily-loaded networks. The results are plotted in Figures 14 and 15 for 80% and 130% load, respectively.

One can see form these figures how EBM equalizes the throughput among different TCP flows while satisfying the CIR requirement under both conditions, whereas the other marking schemes can not even reach CIR under heavy loads and large RTTs.

C. Target Rate

TCP flows with different target rates should get proportional shares of excess bandwidth in under-subscribed networks [7], [19]. We evaluate EBM for this case using a range of CIR from 0.5Mbps to 2.3Mbps and a PIR equal to twice the CIR. Each flow has a different CIR and PIR values. A link capacity of 20Mbps is used between core routers instead of 10Mbps. Results are plotted in Figure 16 for 80% load and in Figure 17 for 120% load. One can see that EBM provides each connection its proportional share of excess bandwidth for both under-subscribed and over-subscribed cases.

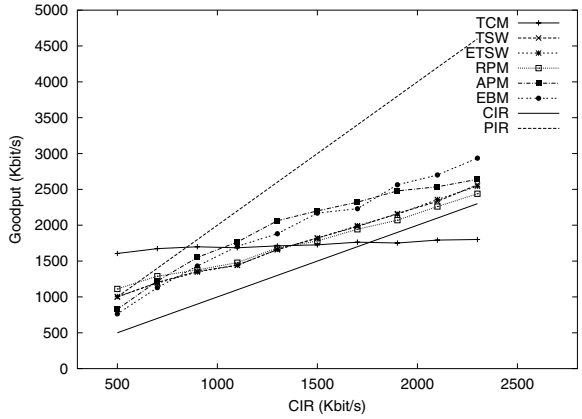

Fig. 16. Goodput vs. target rate — under-subscription

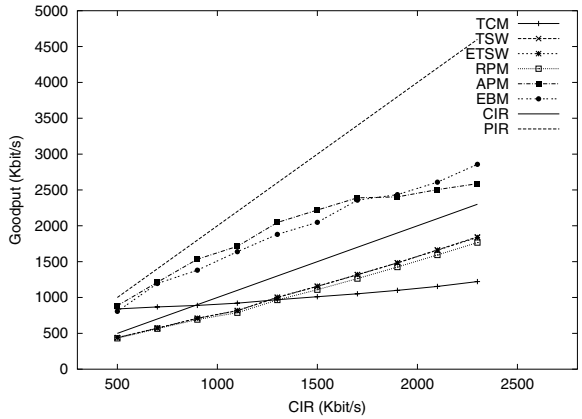

Fig. 17. Goodput vs. target rate — over-subscription

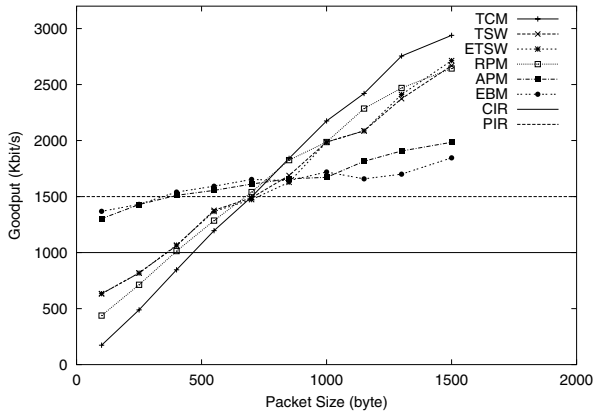

Fig. 18. Goodput vs. packet size — under-subscription

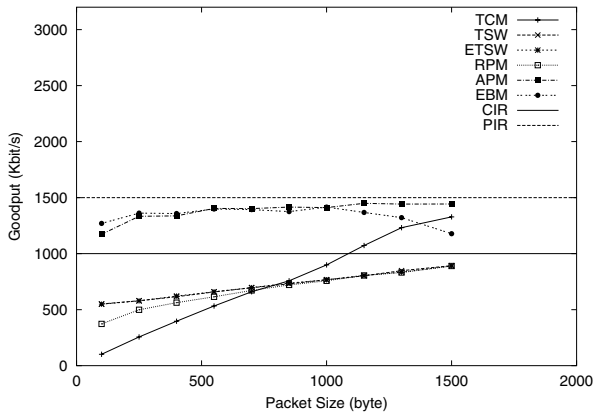

Fig. 19. Goodput vs. packet size — over-subscription

D. Packet Size

We now evaluate the performance of EBM along with the other marking schemes for flows of different packet sizes. We use a range of packet sizes from 100 bytes to 1500 bytes and show the results for 65% load (under-subscribed) in Figure 18, and 120% load (over-subscribed) in Figure 19. CIR and PIR values are 1Mbps and 1.5Mbps, respectively, and we use a link capacity of 20Mbps between core routers. Again, EBM succeeds in providing the required fairness among the heterogeneous TCP flows providing almost an equal bandwidth from the network for equal target rates, CIR and PIR.

E. Overhead

We evaluate the overhead caused by the operation of EBM which is expressed as two terms. The first is the packet classification and marking, and state variable updates. This occurs in all other packet markers and is not unique to EBM.

The second overhead is for the calculation of p_{cir} and p_{pir}. We measured this overhead to be 1.5 to 2ms on a Pentium II, 450MHz processor with 192MB RAM machine. This calculation is performed every 10 seconds resulting in 0.015% to 0.02% overhead.

VI. Conclusions

In this paper we introduced a new marking algorithm called EBM that can be used to serve the AF service class in the Diff-Serv framework. The EBM marking scheme solves the problems associated with previous schemes regarding fairness between heterogeneous TCP flows and protection of target rates under diverse network conditions. We evaluated the performance of EBM through simulation demonstrating its superior performance to the other marking schemes.

The dynamics of the EBM operation need to be evaluated in order to make sure that there will be no large fluctuations in the performance. The difficulty associated with this analysis is that we have to model all systems involved using proper transfer functions, including the random drop module, in order to analyze the transient and dynamic operation of EBM quantitatively. The behavior of EBM with short-lived TCP flows should be also considered. These issues are matters of our future inquiry.

Acknowledgments

The authors would like to thank Sugih Jamin, Haining Wang, Padmanabhan Pillai, and Hani Jamjoom of the University of Michigan, and Nabil Seddigh, Peter Pieda and Victor Firoiu of Nortel Networks for their assistance.

References

[1] S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, IETF, December 1998.

[2] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the Internet," Internet-draft, draft-nichols-diff-svc-arch-02.pdf 2475, IETF, April 1999.
[3] J. Padhye, V Firoiu, D Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *Proc. of ACM SIGCOMM '98*, Oct. 1998.
[4] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," RFC 2597, IETF, June 1999.
[5] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, August 1998.
[6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.
[7] N. Seddigh, B. Nandy, and P. Pieda, "Bandwidth assurance issues for TCP flows in a differentiated services network," *Proc. of IEEE GLOBECOM '99*, March 1999.
[8] R. Makkar et al., "Empirical study of buffer management schemes for diffserv assured forwarding PHB," Technical report, Nortel Networks, May 2000.
[9] J. Rezende, "Assured service evaluation," *Proc. of IEEE GLOBECOM '99*, March 1999.
[10] J. Ibanez and K. Nichols, "Preliminary simulation evaluation of an assured service," Internet-draft, work in progress, draft-ibanez-diffserv-assured-eval-00.txt, IETF, August 1998.
[11] J. Heinanen and R. Guerin, "A two rate three color marker," RFC 2698, IETF, September 1999.
[12] W. Fang, N. Seddigh, and B. Nandy, "A time sliding window three colour marker (TSWTCM)," Internet-draft, work in progress, draft-fang-diffserv-tc-tswtcm-01.txt, IETF, March 2000.
[13] E. Kusmierek and R. Kooldi, "Random packet marking for differentiated services," UMN Technical Report TR-00-020, Dept. of Comp. Science & Eng., University of Minnesota, 2000.
[14] W. Lin, R. Zheng, and J. Hou, "How to make assured services more assured," *Proc. of ICNP '99*, 1999.
[15] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firiou, "On achievable service differentiation with token marking for tcp," *Proc. of ACM Sigmetrics'00, Santa Clara, CA. Also in Performance Evaluation Review*, vol. 28, no. 1, Jun. 2000.
[16] A. Feroz, A. Rao, and S. Kalyanaraman, "A TCP-friendly traffic marker for IP differentiated services," *Proc. of IWQoS '00*, 2000, Also as IETF internet-draft, draft-azeem-tcpfriendly-diffserv-00.txt.
[17] H. Kim, "A fair marker," Internet-draft, work in progress, draft-kim-fairmarker-diffserv-00.txt, IETF, April 1999.
[18] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Adaptive packet marking for maintaining end-to-end throughput in a differentiated-services Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 685–697, Oct 1999.
[19] B. Nandy, N. Seddigh, P. Pieda, and J. Ethridge, "Intelligent traffic conditioners for assured forwarding based differentiated services networks," *Proc. of High Performance Networking 2000 Conference, Paris, France*, May 2000.
[20] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review, ACM SIGCOMM*, vol. 27, no. 3, July 1997.
[21] I. Yeom and A. Reddy, "Modeling TCP behavior in a differentiated services network," TAMU ECE Technical Report, Dept. of Electrical Engineering, Texas A & M University, May 1999.
[22] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," *Proc. of IEEE NOSSDAV '99*, June 1999.
[23] S. Floyd, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," *Proc. of ACM SIGCOMM '00*, 2000.
[24] V. Jacobson, "Congestion avoidance and control," *Proc. of ACM SIGCOMM '88*, pp. 314–332, Aug. 1988.
[25] V Firoiu and M. Borden, "A study of active queue management for congestion control," *Proc. of IEEE INFOCOM '00*, 2000.
[26] Kendall E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley and Sons, New York, 2nd ed., 1989.
[27] Network simulator, ," Ns-2, University of California at Berkeley, CA. Version ns-2.1b6, http://www.isi.edu/nsnam/ns/, Jan. 2000, and Diffserv additions to ns-2 by S. Murphy, available from http://www.teltec.dcu.ie/ murphys/ns-work/diffserv/, May 2000.
[28] N. Seddigh, B. Nandy, and P. Pieda, "Study of TCP and UDP interaction for the AF PHB," Internet-draft, work in progress, draft-nsbnpp-diffserv-tcpudpaf-01.txt, IETF, August 1999.
[29] S. Sahu, D. Towsley, and J. Kurose, "A quantitative study of differentiated services for the Internet," CMPSCI Technical Report 99-09, Dept. of Computer Science, University of Massachusetts, 1999, Also in *Proc. of IEEE GLOBECOM '99*, pp. 1808-1817, Dec. 1999.

APPENDIX

A. TCP Model

$r_t = T(p, RTT, T_o, W_{max}) =$

$$\begin{cases} M \frac{\frac{1-p}{p} + W(p) + \frac{Q(p,W(p))}{1-p}}{RTT(\frac{b}{2}W(p)+1) + \frac{Q(p,W(p))F(p)T_o}{1-p}} & \text{if } W(p) < W_{max} \\ M \frac{\frac{1-p}{p} + W_{max} + \frac{Q(p,W_{max})}{1-p}}{RTT(\frac{b}{8}W_{max} + \frac{1-p}{pW_{max}} + 2) + \frac{Q(p,W_{max})F(p)T_o}{1-p}} & \text{otherwise} \end{cases} \quad (5)$$

where

$$W(p) = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2}$$
$$Q(p,w) = min\left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{(w-3)}))}{1-(1-p)^w}\right)$$
$$F(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6$$

where
r_t: is the sending rate of the TCP flow in bits/sec.
M: is the average packet size in bits.
p: is the loss probability (i.e., probability of loss).
RTT: is the average round-trip time.
T_o: is the typical value of the retransmit timeout (typically $5RTT$).
W_{max}: is the maximum receiver window size enforced by the receiver in packets.
b: is the average number of packets acknowledged by an ACK, (usually 2).

B. Justification for the RIO Assumption

In RED and similar Active Queue Management (AQM) schemes like RIO, the drop rate is a linear function of the average queue size. So, for example, for a two-color drop gateway based on RIO, the drop probabilities (or drop rates) are given as follows. For the IN packets:

$loss_{in} =$

$$\begin{cases} 0, & 0 \le \bar{q}_{in} < min_{th_{in}} \\ \frac{\bar{q}_{in} - min_{th_{in}}}{max_{th_{in}} - min_{th_{in}}} p_{max_{in}}, & min_{th_{in}} \le \bar{q}_{in} < max_{th_{in}} \\ 1, & max_{th_{in}} \le \bar{q}_{in} \le B \end{cases} \quad (6)$$

where
B: is the maximum queue size or buffer size.
$\bar{q}_{in}$: is the average queue size for the IN packets calculated as WMA of the instantaneous queue samples.
$p_{max_{in}}, min_{th_{in}}, max_{th_{in}}$: are the RIO parameters for IN packets.

A similar formula exists for OUT packets obtained simply by replacing every *in* with *out*.

The loss probabilities, $loss_{in}$ and $loss_{out}$, will be in the range $(0, p_{max_{in}})$, and $(0, p_{max_{out}})$, respectively. It is important to note that the calculation does not depend on per-flow information, so packets from different flows will see the same

instantaneous values of loss probabilities for IN and OUT packets as functions of the same average queue size. However, total losses for IN and OUT packets for each flow depends on the number of IN and OUT packets in the flow's packet stream, which is directly related to the marking technique used. The values of $loss_{in}$ and $loss_{out}$ are proportional to the average traffic load on this gateway. If the load is high, the loss probabilities will be high; and if the load is low, the loss probabilities will be low, but will still be confined in range $(0, p_{max_{in/out}})$.