Diffusion Marking Mechanisms for Active Queue Management*

Gonzalo R. Arce and Rafael C. Núñez
Department of Electrical and Computer Engineering
University of Delaware
Newark, Delaware 19716
Email: {arce,nuñez}@ece.udel.edu

Abstract—Active Queue Management (AQM) schemes proposed to date use control mechanisms having fundamental weaknesses often leading to overparameterization and instabilities. This paper draws from the rich theory of dithering and quantization, extensively used in signal processing, to develop new marking and control mechanisms that overcome such limitations. In particular, a packet marking mechanism is developed through error diffusion filtering where congestion is measured and controlled through the instantaneous queue, the derivative of the queue length, and an estimate of the number of active flows. The proposed mechanism, referred to as DM (diffusion marking), is able to maintain a desired average queue length even under rapid changes in the traffic dynamics with surprisingly short lived queue size transients. DM provides a consistent low link delay without sacrificing throughput and is shown to outperform RED, REM, and AVQ type AQM mechanisms in varying network conditions with a significant improvement in stability.

I. INTRODUCTION

Active Queue Management (AQM) is an important area of interest in congestion control where new solutions to emerging challenges in traffic administration are actively sought.

Random Early Detection (RED), proposed by Sally Floyd and Van Jacobson in the early 1990s, emerged as a revolutionary concept where congestion control is achieved through packet dropping or marking before queue saturation is reached [1]. TCP responds to these packet marks as an indication of congestion and decreases the rate at which it sends data.

RED randomly marks packets in proportion to the estimated average queue length. The packet mark probability is based on three parameters: the minimum threshold $\bar{q}_{\text{min}}$, the maximum threshold $\bar{q}_{\text{max}}$, and a mark probability $p_{\text{max}}$. Although extensively studied, the precise design of the parameters in RED, $\{w, \bar{q}_{\text{min}}, \bar{q}_{\text{max}}, p_{\text{max}}\}$ is problematic. Poor choices in parameter settings results in poor performance, and therefore, do not satisfy the goals of making the network suited to delay sensitive traffic such as voice, conferencing, streaming, and achieving higher throughput.

RED’s second weakness lies in its suboptimal dithering mechanism used to mark packets [2]. As described in [1], it is desirable for packet markings to occur homogeneously and as far apart as possible to counteract global synchronization and to promote fairness and stability.

RED requires the backlog of the queue to increase for the congestion feedback signal to increase which is the third fundamental weakness. Therefore, an increasing number of flows to the queue will increase the queue’s backlog resulting in higher delay.

Random Exponential Marking (REM) [3] and Adaptive Virtual Queue (AVQ) [4, 5] belong to a class of AQM algorithms that employ rate based congestion notification. REM maintains a price associated with a link that is updated periodically. AVQ replaces the marking probability calculation with the computation of the capacity of a virtual queue.

While these and several more AQMs have propelled this area of research forward, many issues remain unsolved. Over-parameterization makes configuration for a specific network situation difficult to achieve when optimal parameter values are unknown. Suboptimal choices of parameter values most likely result in poor performance. Also, variations in traffic require different parameter choices where adaptive calculations can be complex and difficult for an AQM scheme to implement in real-time.

Most AQMs compute a packet marking probability and then decide which packets to drop based on that probability. The dithering mechanisms in current AQMs are suboptimal. The dithering mechanism is a critical part of any AQM and is directly responsible for fairness, stability, and performance.

We present a new AQM called Diffusion Marking (DM) that uses error diffusion, a technique used in digital halftoning and optimal quantization. DM uses a geometric nonlinearity with only one parameter to determine the mark probability from the estimated queue length. The decision of whether or not to mark an incoming packet is determined by an optimal dithering mechanism using an error diffusion filter. This process spreads the marks as far apart from each other as possible. By doing so, DM is fair to all flows and does not exhibit tail-drop behavior. This minimizes synchronization effects and provides increased stability.

Sections II and III describe DM, and the derivation of its mechanisms. Section IV contains optimizations to the basic design of DM. Section V shows simulations demonstrating the performance of DM compared to RED, REM, AVQ, Drop Tail and Adaptive RED. Section VI concludes the paper.

*This work was supported in part by the National Science Foundation under grants 0322744 and 0312851 and by U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011
Fig. 1. Packet marking in DM using the error diffusion filter

II. CONTROLLING QUEUES WITH ERROR DIFFUSION

Diffusion Marking directly addresses the problems stated with existing AQMs. DM makes use of the instantaneous queue length instead of RED’s average queue length. This feature allows DM to react faster to rapid changes in the incoming traffic. Moreover, the relation between the probability of marking a packet and the queue length is simplified and optimized to eliminate parameters, and the number of flows is used in the calculation to eliminate any backlog increase. DM uses these features in conjunction with an optimal dithering mechanism to decide which packets to mark.

As described in [1], it is desirable for packet markings to occur homogeneously and as far apart as possible to counteract global synchronization and to promote fairness and stability. It turns out that the probabilistic markings used in RED are suboptimal and that significantly better marking mechanisms exist. Notably, this same problem has been studied extensively in the quantization of digital images. In the image processing literature, this problem is referred to as spatial dithering for digital halftoning where grey-level images are to be represented by either black (marked) pixels or white (unmarked) pixels [6, 7]. The marking method used to assign black or white markings to pixels determines the type of spatial dithering used in the halftone process. Thus, packet dropping in congestion control is analogous to the grey-scale value of digital images – darker grey-level image areas should be represented by a higher percentage of black markings.

Error diffusion is an adaptive algorithm that uses error feedback to produce markings. The concept of error feedback traces back to Inose and Yasuda’s 1963 sigma delta modulation method for the analog to digital conversion of audio signals [8]. Error diffusion, was used in digital halftoning by Floyd and Steinberg in 1976 [9]. A generic form of this algorithm is illustrated in Fig. 1. At time \( n \), error diffusion outputs \( D[n] \), which determines whether the incoming packet is marked (dropped) \( (D[n] = 1) \) or not \( (D[n] = 0) \). Since \( P[n] \) is the initial drop probability variable inputted to error diffusion, \( D[n] \) can be thought of as a quantized representation of \( P[n] \). \( D[n] \) is determined by adjusting and thresholding \( P[n] \) as

\[
D[n] = \begin{cases} 
1 & \text{if } P[n] - P^e[n] \geq 0.5 \\
0 & \text{otherwise}
\end{cases} 
\]

(1)

where \( P^e[n] \) is the diffused quantization error accumulated from previous iterations as

\[
P^e[n] = \frac{2}{3}D^e[n-1] + \frac{1}{3}D^e[n-2]
\]

(2)

and where \( D^e[n] = (P[n] + P^e[n]) - D[n] \). Thus, the error associated with each incoming packet, \( D^e[n] \), is diffused to

the processing of the next two packets arriving immediately after. Notice the inherent memory structure through which the diffused error is used to modify the mark probability for the incoming packets. In essence, the \( n \)th probability mark is corrected with the errors from its preceding packet marks. The corrected probability value, not the original probability is then thresholded.

III. THE PROBABILITY OF MARKING A PACKET

With a method of deciding which packets to mark, a probability of marking must be determined. We base the probability of marking on the piecewise linear function used in Gentle-RED. It closely follows the nonlinear function

\[
P_n = P(q_n) = \left( \frac{q_n}{S} \right) ^ \alpha
\]

(3)

where \( S \) is the queue buffer size and \( \alpha \) controls the concavity. By adjusting \( \alpha \), the desired packet drop rate can be attained. In order to keep the backlog of the queue from increasing, DM uses the estimate of the number of active flows described in section IV. With the number of flows, optimal choices of \( \alpha \) to maintain a constant average queue length can be calculated. Consider the usual bottleneck network topology shown in Fig. 2. We assume there are \( N \) FTP connections using TCP/Reno with unlimited data to transmit, and with router \( R \) implementing DM. We assume the network is in a steady state with each TCP window now in congestion avoidance. The number of packets from a particular TCP flow that the DM router forwards between successive marks is \( \frac{1}{p} \). \( p \) is the probability of DM marking a packet. Then,

\[
\frac{1}{p} = \frac{3}{8}W^2
\]

(4)

With a constant average queue length, the congestion window can be calculated as:

\[
\frac{3}{4}WN = \frac{BD}{MSS} + q_n
\]

(5)

where \( N \) is the number of active flows, \( B \) is the bandwidth, \( MSS \) is the maximum segment size, and \( D \) is the propagation delay.

Solving (3), (4), and (5) simultaneously for \( \alpha \) yields:

\[
\alpha(N) = \frac{\log \left( \frac{3}{2} \right) - 2\log \left( \frac{BD}{MSS} + q_n \right) + \log \left( N^2 \right)}{\log \left( \frac{q_n}{S} \right)}
\]

(6)

Using the properties of the logarithm and constant values from the network, (6) reduces to a single, memoryless, adaptive calculation of the form

\[
\alpha(N) = a + b \log(N)
\]

(7)

where \( a \) and \( b \) are constants. This equation is simple for a router to implement; therefore, \( \alpha \) is no longer a parameter that must be manually configured.
While (7) is a straightforward calculation for an AQM, it does contain some flaws and can be further improved with a few observations. First, replacing \( \alpha \) in (3) with the RHS of (7) yields

\[
P_n = \left( \frac{q_n}{S} \right)^{\alpha + \log(N)} = \left( \frac{q_n}{S} \right)^{\alpha} \left( \frac{q_n}{S} \right)^{\log(N)}
\]

Note that when \( \alpha + \log(N) > 1 \), \( P_n \) is convex. However, if that is not the case, the curve becomes concave and eventually has values that are greater than one. Using logarithm laws, we can approximate the probability of marking a packet as:

\[
P_n = \begin{cases} (\frac{q_n}{S})^{N^2} & \text{if } q_n > S(N^2)^{-\frac{1}{\alpha}} \\ 1 & \text{otherwise} \end{cases}
\]

where

\[
\alpha = \log \left( \frac{1}{2} \right) - 2 \log \left( \frac{BD_{\text{MSS}} + q_d}{S} \right)
\]

In (10), \( q_d \) is the desired queue length. Figure 3 plots (9) for \( N = 1 \), \( N = 5 \), and \( N = 10 \).

IV. OPTIMIZING THE CONTROL MECHANISM

A. Adaptive Threshold Control

Many modifications to error diffusion [9] have been proposed in the last 25 years to improve its quality. One of the most effective methods is to dynamically change the threshold of the algorithm. The threshold in Equation (1) is 0.5, but it can be generalized with an arbitrary function \( f(x) \). The quantizer threshold thus becomes an algorithm parameter. Such modification receives the name of threshold modulation [10].

Low and Kim suggest proportional and derivative actions for TCP stability with a minimum cost and maximum throughput [2]. DM already has proportional control since the drop probability is derived directly from the estimated queue length. Derivative control is provided through input dependent threshold modulation. Then, \( D[n] \) changes not only because of variations in \( P[n] \) but also because of the rate of change of \( P[n] \). This capability allows DM to respond to changes in traffic load more efficiently.

The threshold function in DM has the form:

\[
f(P[n]) = kP[n]
\]

where \( k \) is some positive constant. Selection of \( k \) is a tradeoff between faster response and stability. Many simulations have been performed to find the appropriate \( k \), and it was found that \( k = 5 \) reflects an intermediate point between faster response and stability. Then, (1) changes to

\[
D[n] = \begin{cases} 1 & \text{if } P[n] - P^c[n] \geq 5P[n] \\ 0 & \text{otherwise} \end{cases}
\]

B. Dynamic Estimation of Active Flows

With the purpose of obtaining a precise estimator for Diffusion Marking we make use of the stochastic models of TCP. The estimator needs to capture the effect of all the flows that are sending packets and increasing their congestion windows (not in timeout), as well as the flows that are entering in timeouts not longer than the round trip time. The number of flows in timeout during less than RTT is important because their packets are still in the pipe. Any other flow not having packets in the channel is not included in the estimate. The number of active flows \( N \) is given by:

\[
\hat{N} = \begin{cases} \frac{1}{\hat{W}} \left(1 + \frac{2}{\hat{W}} \left(\hat{q} + \beta\right) \right) & \text{if } \hat{W} \geq 3 \\ \frac{2}{\hat{W}}(\hat{q} + \beta) & \text{otherwise} \end{cases}
\]

where \( \hat{W} \) is an estimate of the congestion window in a single flow, \( \hat{q} \) is an estimate of the queue length, and \( \beta = \frac{BD_{\text{MSS}}}{3} \) is the bandwidth-delay product. In the current implementation, we calculate \( \hat{q} \) as the sample mean of the queue size, and \( \hat{W} \) is computed based on the stochastic model of the congestion window provided by Padhye et al. [11]:

\[
\hat{W} = \frac{2 + b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2 + b}{3b}\right)^2}
\]

where \( b \) is the number of packets that are acknowledged by a received ACK, typically 2. \( p \) is calculated in two steps: first, the drop probability is measured as the average of the marking signal (Drop = 1, Not-Drop = 0). After the first stage, and given the spikes that can be present with traffic alterations, the value passes through a median filter whose sampling window is five times the used in the previous measure. This process guarantees a good and stable approximation of the drop probability.

V. PERFORMANCE RESULTS

In this section we use the ns-2 environment to simulate DM and to compare it Drop Tail, RED, Adaptive RED, REM, and AVQ, in two different scenarios: long lived flows, and HTTP flows. We use the network topology shown in Fig. 2. TCP/Reno is the transport protocol and TCP packet size set to 1000 bytes.

A. Long lived flows

In this simulation, \( B = 10Mbps \). The bottleneck router implements each AQM with \( S = 100 \)packets. The AQMs drop packets. 20 FTP connections are initially active and 20 more are added every 50 seconds. \( D = 50ms \). DM is
configured for a queue length $q_d = 30$. The RED parameters are $q_{min} = 20$, $q_{max} = 40$, $w = 0.001$, and $p_{max} = 0.1$. The REM parameters are $\gamma = 0.001$, $\phi = 1.001$, and $b^* = 30$. AVQ is configured with $\gamma = 0.98$.

Figure 4(a) shows the average queue size as a function of the number of users. The transients are not taken into account for the calculation of the averages. Note that the average queue length remains for almost all AQMs. Drop Tail queue is increasing with traffic. AVQ shows a slightly decreasing shape, while RED tends to increase with the number of users.

Figure 4(b) contains the end to end delay. The average end to end delay exhibit the same shape than the average queue size plots. Any oscillation in the queue length is undesirable since it cause packets to have different queueing delays, which makes QoS more difficult. The oscillations are captured in Fig. 4(c). DM and AVQ show the smallest end to end delay. In addition, the variance of the delay using DM remains small and stable when the number of users increase. Note that the variance for the remaining AQMs is not as predictable as in DM.

The stability of DM is also visible in the throughput. The throughput is always high in a router implementing DM. AVQ has a stable throughput, independent of the number of users, but is smaller than the other three AQMs. REM presents an unpredictable behavior, while Adaptive RED increases its throughput up to DM’s level, when the number of users increases.

### B. HTTP traffic

In this experiment we use the module PackMime-HTTP [12]. The web traffic in the network is modeled as flowing between a set of clients (browsers) and a set of servers. A fundamental parameter for the module is the rate at which new TCP connections are initiated by the cloud of web clients. We use a client cloud and a server cloud, connected by the bottleneck link of 10Mbits and $D = 2$ms. The connection rate is 100 connections/second, and is increased 100c/s every 50s. The initial stage of 100c/s simulates a low traffic condition, while the 400c/s simulates an extremely high traffic condition.

Figure 5(a) shows the average queue size for the AQMs. If the traffic increases, RED, Adaptive RED, and REM cannot control the queue in the appropriate manner. AVQ reacts by emptying the queue because the virtual queue decreases as the incoming rate increases. Then, AVQ keeps the queue size very low and the utilization is reduced. DM is able to keep the queue in the user-defined queue size value, with any traffic condition. Note that, except for AVQ, all the other AQMs increase their average queues. Fig. 5(b) contains the average delay as a function of the connection rate. DM keeps a reduced average delay. In addition, DM keeps a low and stable variance, as is depicted in Fig. 5(c).

Figure 5(d) contains the throughput for all the AQMs. It
Fig. 5. AQM’s performance with HTTP flows

shows that when the traffic increases, the throughput rises and converges to the same value for DM, RED, Adaptive RED, and REM, while AVQ converges to a lower throughput.

On the whole, DM shows an increment on the performance of the analyzed AQMs. This is achieved by keeping a stable queue. The end to end delay is small and stable, and the throughput is maximized.

VI. CONCLUSIONS

We presented a new AQM scheme called Diffusion Marking. The main goal of DM is to maintain the desired average queue length even under rapid variations in the traffic. DM is based on optimal dithering with error diffusion. Error diffusion is a powerful method extensively used in signal quantization. The use of optimal quantization techniques guarantees fairness and leads to improved stability of DM when compared with RED, REM, and Adaptive RED. According to our simulations, DM presents higher throughput and less variance in the queue size when compared with AVQ. DM keeps a high throughput. The small delays and increased stability of DM make this algorithm suitable for applications requiring low jitter/delay, as multimedia transmissions.

Our current work includes the design of an improved version of DM that does not require the estimation of the number of flows. An extended stability analysis of DM is also being developed. Notably, the performance of the new version is comparable to the scheme presented here, and the results will be presented soon.

REFERENCES