A Hybrid Random Early Detection Algorithm for Improving End-to-End Congestion Control in TCP/IP Networks

Aun Haider, *Member, IEEE,* and Richard J. Harris, *Senior Member, IEEE* .

Abstract— **The successful operation of the present Internet depends mainly upon TCP/IP which employs end-to-end congestion control mechanisms built in the end hosts. In order to further enhance this paradigm of end-to-end control the Random Early Detection algorithm (RED) has been proposed, which starts to mark or drop packets at the onset of congestion. The paper addresses issues related to the choice of queue length indication parameters for packet marking/dropping decisions in RED-type algorithms under varying traffic conditions. Two modifications to RED are proposed: (i) use of both instantaneous queue size and its Exponential Weighted Moving Average (EWMA) for packet marking/dropping and (ii) reducing the effect of the EWMA queue size value when the queue size is less than min_{th} for a certain number of consecutive packet arrivals.**

The newly developed Hybrid RED algorithm can effectively improve the performance of TCP/IP based networks while working in a control loop formed by either dropping or marking of packets during congestion epochs. New guidelines are developed for better marking/dropping of packets to achieve a faster response of RED-type algorithms. The hybrid RED algorithm has been tested using ns-2 simulations, which show better utilization of network bandwidth and a lower packet loss rate.

Index Terms—**Congestion control, Hybrid RED algorithm, EWMA queue size, packet loss rate, link utilization.**

I. Introduction

TCP is a reliable end-to-end transport protocol, widely used in the current Internet for such diverse applications as email, Usenet news, remote login, file transfer, some streaming audio and video protocols and www. As a result of this widespread proliferation and integration of TCP/IP code in commonly used operating systems, such as Unix, Linux and Microsoft Windows, it constitutes a major portion of the current Internet traffic.

The growth of networks operating under TCP/IP has been unprecedented. The critical factor in the robustness of the current best effort Internet turns out to be end-to-end congestion control mechanisms. In this mechanism TCP constantly adapts itself to available bandwidth, slowing down the data transfer rate when it detects congestion and speeding up again when there is no congestion. Traditionally these end-to-end congestion control mechanisms were incorporated only in the end host so as to aid in upgrading the software and keeping the functions of core routers simpler. However, due to the present mammoth size of the Internet, the unreliability of developers in implementing the standard TCP congestion control algorithms in their Internet applications and the problems caused by attackers, it has been suggested to involve routers in performing the vital task of congestion control in the Internet see; e.g. [2], [3] and [4].

Presently, most widely used routers employ a Droptail policy which has the inherent limitation of inability to convey congestion information to the end host. Furthermore, they have other well known problems such as flow synchronization; see [5] and [6]. Thus, an alternative mechanism is desired, at the congestion prone routers, to detect the onset of congestion and accordingly send feedback to the end hosts, forcing reduction in their data sending rates. Thereby, congestion and consequent loss of packets is reduced. The congestion feedback from a router can be conveyed back to end hosts by Explicit Congestion Notification (ECN), to form so-called Active Queue Management (AQM); see [7]. Recently several algorithms have been proposed to address the problem of congestion control at the routers; see [8], [9] and [10].

This includes Random Early Detection (RED) algorithm, which was first proposed in [5]. Later its use was recommended in IP routers for the implementation of AQM [2]. RED has also been implemented in commercially available routers, such as Cisco AS5200, 4000, 4500 and 4700 [11]. It is believed that RED routers will solve the major problems in existing drop-tail routers such as synchronization of flows and will support quality of service by providing intelligent marking of packets [2]. The performance of RED is dependent upon the proper tuning/matching of its four intertwined parameters $\{min_{th}, max_{th}, max_p, w_q\}$, defined in Subsection 2.1, to the three parameters influencing congestion: bottleneck link bandwidth, number of TCP flows and round-trip time.

Central to the operation of RED is the EWMA mechanism, [12, Chap. 5, 55-65], for controlling the instantaneous queue size that acts as a Low Pass Filter (LPF) to absorb transient traffic bursts. It is used in conjunction with a piecewise linear marking/dropping probability function to get the desired functionality. This paper presents two modifications to the use of EWMA by RED which are shown to improve its performance under sudden change in traffic load from high to low values. Since the new algorithm combines use of both instantaneous and EWMA queue sizes, it is named Hybrid RED. The performance of the newly developed algorithm is determined by running ns-2 based simulations.

Aun Haider and Richard J. Harris are with Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand; Emails: a.haider@massey.ac.nz and r.harris@massey.ac.nz

Manuscript received on December 16, 2006; revised Feburary 04, 2008. An earlier part of this work was presented and published in Proc. of the 10th International Conference on Telecommunications (ICT) 2003, [1]

The rest of the paper is organized as follows. In Section II a brief overview of the basic RED algorithms, and a discussion of their queue behaviour is presented. Section III discusses related work. Standard RED based on EWMA queue size is analyzed in Section IV. The Hybrid RED algorithm is derived in Section V. Section VI contains simulation results and performance analysis of Hybrid RED. Finally Section VII summarizes our conclusions.

II. Background

We first briefly review the two basic forms of the existing versions of RED algorithms and then discuss how they influence the queue properties.

A. RED Algorithm

The basic version of RED uses the following LPF with a constant w_q (called queue weight, or smoothing factor, or forgetting factor), to calculate the EWMA, $\bar{q}$, of the instantaneous queue size q:

$$\bar{q} \leftarrow (1 - w_q) \cdot \bar{q} + w_q \cdot q; \tag{1}$$

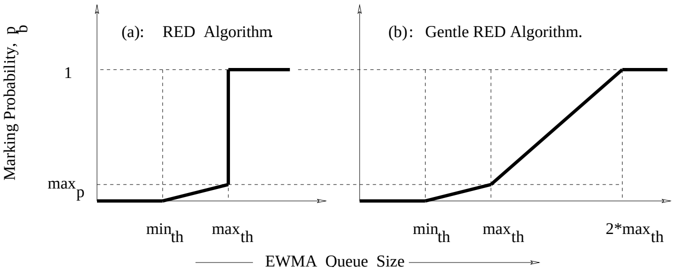


Fig. 1. Packet marking/dropping probability curves of the basic RED, Fig. 1 (a), and Gentle RED, Fig. 1 (b), [1].

see [5]. In equation (1), w_q determines the time constant of the LPF which in effect determines the length of memory used in the averaging process [13]. One can split equation (1) into the two components: (a) contribution of the present value of instantaneous queue size q, which is represented by the term $w_q \cdot q$, and (b) contribution of the previous value of EWMA queue size, given by the term $(1 - w_q) \cdot \bar{q}$. It shows that as w_q is increased, the contribution of q increases, or memory decreases, and vice versa. The linear marking/dropping probability function of the basic RED algorithm can be expressed algebraically as:

$$p_b = \begin{cases} 0, & \text{if } \bar{q} < min_{th}; \\ 1, & \text{if } \bar{q} \geq max_{th}; \\ \frac{\bar{q} - min_{th}}{max_{th} - min_{th}} \cdot max_p, & \text{if } \bar{q} \in [min_{th}, max_{th}]. \end{cases} \tag{2}$$

where min_{th}, max_{th} are minimum and maximum thresholds for EWMA queue size, respectively, and max_p is the maximum value of packet mark/drop probability; see Figure 1 (a). For a buffer size B, it is generally recommended to use $max_{th} < B$, $max_{th} = 3 \cdot min_{th}$, $max_p = 0.1$ and $w_q = 0.002$, see [5]. The discontinuity in maximum mark/drop probability, max_p, at $\bar{q} \geq max_{th}$ in equation (2) may cause violent oscillations in the instantaneous queue size q; see [14] and [15]. In general, the operation is critically dependent upon the proper selection of four intertwined parameters, $\{min_{th}, max_{th}, max_p, w_q\}$; see [5] and [16] for parameter settings of the basic RED.

B. Gentle RED Algorithm

The gentle version of RED was proposed to avoid the sudden jump in probability function in (2) by modifying it as follows:

$$p_b = \begin{cases} 0, & \text{if } \bar{q} < min_{th}; \\ 1, & \text{if } \bar{q} \geq 2 \cdot max_{th}; \\ \frac{\bar{q} - min_{th}}{max_{th} - min_{th}} \cdot max_p, & \text{if } \bar{q} \in [min_{th}, max_{th}]; \\ \frac{\bar{q} - max_{th}}{max_{th}} \cdot (1 - max_p) + max_p, & \text{otherwise}; \end{cases} \tag{3}$$

see [17]. The probability that the intermarking time, X, is equal to number of packets, n, that arrive after a marked/dropped packet (until the next packet is marked/dropped) is given by: Prob$[X = n] = (1 - p_b)^{n-1} \cdot p_b$. It shows that X is a random geometric variable with $E(X) = 1/p_b$, which can mark/drop packets at irregular time intervals and may leads to global synchronization of TCP flows, [5].

Thus, both (2) and (3) will result in an undesirable geometrical pattern for marking/dropping of incoming packets. Therefore, it is desired that X should be a uniform random variable, which can be achieved by using the following expression:

$$p_a = \frac{p_b}{1 - count \cdot p_b}, \tag{4}$$

where $count$ is the number of unmarked packets that have arrived since the last marked/dropped packet; for more details see [5]. It is clear that the operation of both versions of RED algorithm, as given by (2) and (3), is crucially dependent on the use of the EWMA queue size $\bar{q}$. In a later section it will be shown that $\bar{q}$ is not by itself an ideal indicator of congestion events in a router and thus leads to non-optimal performance, due to unnecessary packet marking/dropping.

The instantaneous queue size, q, in RED routers varies between zero and the buffer size B. When q hits zero, a period of under-utilization of the bottleneck link bandwidth begins. On the other hand, when q reaches the upper limit B, all of the incoming packets will be dropped, and such forced drops will require retransmissions. This process will lead to congestion. Transient periods of bursty traffic can cause wide fluctuations of q, which can be diminished by introducing the EWMA mechanism.

C. EWMA Tuning

The use of EWMA for packet marking/dropping in conjunction with a simple linear function requires an optimal value of queue weight, w_q [12] and [5]. If w_q is too large then transient congestion will not be filtered out, and if it is too small then $\bar{q}$ will respond too slowly to enforce the desired changes in p_b. Larger values of w_q also encourage oscillations in the queue size. Assuming that the queue was initially empty and has grown to an instantaneous size of L packets, i.e. $q = L$, over the arrival of L consecutive packets the following expression of EWMA queue size, $\bar{q}$, has been obtained

$$\bar{q} = L + 1 + \frac{(1 - w_q)^{L+1} - 1}{w_q}, \tag{5}$$

see [5]. The upper limit for w_q is derived from $\bar{q} < min_{th}$, i.e.:

$$L + 1 + \frac{(1 - w_q)^{L+1} - 1}{w_q} < min_{th}. \tag{6}$$

The lower limit for w_q is computed by the simple observation that it takes $\frac{-1}{\ln(1-w_q)}$ packet arrivals to change $\bar{q}$ from 0 to 0.63 while q stays at 1 packet. As stated in the previous subsection, for RED type algorithms the value of w_q is usually chosen small, of the order of 0.002, so as to avoid a bias against sudden bursts of incoming packets. In [18] a good value of w_q for a bottleneck link of capacity C and round trip time RTT, was determined as:

$$w_q = 1 - e^{\frac{-1}{10 \cdot C \cdot RTT}} \approx \frac{1}{10 \cdot C \cdot RTT}, \tag{7}$$

based on a control-theoretic analysis of RED type algorithms; and is a general form of the expression given in [19]. Using (7), the value of w_q can be automatically adjusted according to the network scenario.

The effects of using a longer memory (smaller values of w_q) in equation (1) were further investigated in [13] by means of simulation studies. It was found that if w_q is decreased below a certain critical value then the performance of the RED algorithm degrades as the coefficient of variation of queue occupancy increases. Also, the use of a longer memory in the EWMA process and an increase in the round trip time delay in feedback loop of TCP/IP connections often drive a stable process into oscillatory mode, which is inconsistent with the theoretical results obtained in [20]. It was suggested that generally w_q should be chosen in the range of 0.5 to 1 (note that w_q=1 stands for case when $\bar{q}(k) = q(k)$, i.e. no averaging by the EWMA process) rather than the current practice of using values in the range of 0.001 to 0.002. However, it was also predicted that for queue size in the buffers of backbone routers, using RED algorithm, with small values of w_q (0.001 to 0.002) might not be practically detrimental, [13].

In [20], the EWMA process is modeled as an *Ornstein-Uhlenbeck* process with delayed feedback which proves that for sufficiently long delay in feedback the process becomes unstable. It is conjectured that widespread use of the EWMA process, at routers equipped with RED-type algorithms, might compromise stability of the Internet. In order to offset these problems, the use of both instantaneous and EWMA queue size for packet mark/drop has been suggested in the literature, see [14].

III. Related Work

V. Jacobson and co-workers have suggested that the gain of the filter or queue weight w_q should be different for building queues and draining queues [21]. For building queues, the response of RED to a unit step function is investigated with w_q chosen to be equal to the inverse of the number queue samples in a round trip time, so that the filter approximates the mean over a Round Trip Time (RTT). The analysis shows that it takes 2 RTTs for RED to reach 90% of the persistent queue level. For draining queues, they suggest that w_q be set to one, i.e. EWMA queue size should be equal to the instantaneous queue size, thus, reinforcing the results reported in [20]. Despite of improvements in [5], the trade-off between the ability to absorb transient burst of packets and tracking the instantaneous queue size, still exists. More work is required to investigate the effects of a persistent queue and the selection of queue sampling interval.

In [14], the use of both instantaneous and EWMA queue size for packet marking decisions was investigated on a testbed. In one of the tests, instantaneous queue size was used in a gentle RED version [17]. These experiments showed that EWMA hinders the anticipation of congestion and better congestion indicators (like instantaneous queue size) are needed to detect incipient congestion in a network.

Also, it has been pointed out in [15] that the RED algorithm needs modification by employing both q and $\bar{q}$ for packets mark/drop decisions. The following major drawbacks are inherent in RED type algorithms due to the use of only EWMA queue size:

- Since EWMA queue size is computed at each packet arrival, it can grow at any rate depending on the arrival rate. On the other hand, the maximum decrease rate in the EWMA queue size is limited by the bottleneck bandwidth (or even less, since recomputation is only triggered by packet arrivals, not departures). Thus EWMA shows averaging of packet arrival rate rather then the true buffer occupancy.
- During packet bursts, the packet arrival rate is greater than the packet departure rate at the bottleneck routers. Due to this difference in the two rates, equation (1) resembles more a peak detector than an EWMA filter. Hence it has a bias towards large values of queue size.

Such problems in the EWMA queue size were also considered in [22] and a new algorithm, named Flow Random Early Detection (FRED), has been proposed. However, FRED has several performance limitations [23], and also it maintains state variables for all active flows which makes it less scalable.

The LPF/ODA algorithm, [24] and [25], aims to improve link utilization when EWMA is at a high value ($min_{th} < \bar{q} < max_{th}$ or $\bar{q} > max_{th}$) but the instantaneous queue size is lower than min_{th}. The algorithm can be summarized as follows:

- During normal operation the queue size is controlled by the EWMA value, which is the default control mechanism of the RED algorithm. During this phase, called *Active Drop Phase or LPF*, the router will behave as normal RED.
- If the instantaneous queue size stays below min_{th} for three consecutive packet arrivals, then it indicates the onset of the *Over Drop Avoidance (ODA) Phase* during which the EWMA is reduced to half of its present value.
- After this reduction, updating of the EWMA queue size is continued as in normal RED.

In [24] and [25], the number of ODA-phase triggering packets,

$\mathcal{N}_o$, and amount of reduction in EWMA queue size (during that ODA-phase) were arbitrarily chosen as 3 and 0.5, respectively. In contrast, this paper determines the effects of changes in the number of ODA triggering packets, and the amount of reduction in EWMA queue size, on throughput and loss rate at the bottleneck link while employing the Hybrid RED. Furthermore, the results presented in both [24] and [25] provide no information about the error statistics. In contrast, this paper reports the results for Hybrid RED obtained after multiple runs of ns-2 based simulations.

In general, our work supports the concept of employing both instantaneous and EWMA queue size during packet marking/dropping decisions in RED type algorithms which is in line with the findings of [15], [20], [21], [14], [24] and [25].

IV. Analysis of RED with EWMA queue size

Consider a bottleneck link of capacity C being served by a RED router, having a small value of w_q, at a rate of μ where $\mu < C$. Thus, the instantaneous queue size, q, has not built up and the EWMA queue size, $\bar{q}$, is small in value. Now, let another traffic flow join the session, making the aggregate packet arrival rate at router ports greater than C. Now q is large and $\bar{q}$ is computed on packet arrivals, which makes $\bar{q}$ to grow much larger than max_{th}, causing all of the new incoming packets to be dropped.

This dropping of packets will continue for a long time even after the queue has become empty, because $\bar{q}$ will take a long time to return below max_{th}. During this epoch of time, the transmission rate of packets from the queue will reach a very low value as almost all of the incoming traffic will be dropped. Such cases can be identified in Figures 3 and 9 of [5]. The situation will momentarily return to normal after the value of $\bar{q}$ drops below max_{th}. However the consequent increase in sending rate will cause $\bar{q}$ to increase again. Thus, we will reach a state in which $\bar{q}$ oscillates around max_{th} and packets will be again dropped forcibly [26]. Hence, depending upon the rate of incoming packets, bottleneck link rate, and responsiveness of flows, there might be epochs when all of the packets will be discarded while the link is idle.

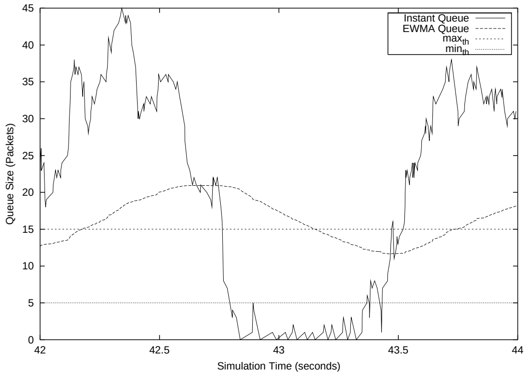


Fig. 2. Queue snapshot when $q < min_{th}$ and $\bar{q} > max_{th}$, [1].

Two such cases are illustrated in Figures 2 and 3, where q is well below min_{th} but $\bar{q} > max_{th}$ and $min_{th} < \bar{q} < max_{th}$ respectively, causing unwanted dropping of packets. It indicates that EWMA by itself does not always provide a true picture of the backlog in buffers of RED queues. There can be instants when EWMA gives false indications of buffer occupancy which will lead to wastage of precious network bandwidth. Thus, in the following section we present a new algorithm, called Hybrid RED, aimed at overcoming the above mentioned problems by utilizing both instantaneous queue size, q, and its EWMA value, $\bar{q}$.

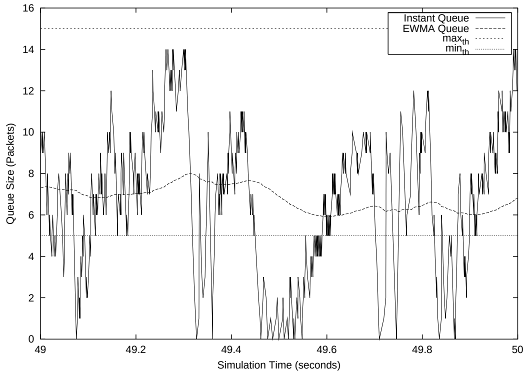


Fig. 3. Queue snapshot when $q < min_{th}$ and $min_{th} < \bar{q} < max_{th}$, [1].

V. Hybrid RED

The new Hybrid RED algorithm adjusts $\bar{q}$ during $q < min_{th}$ periods and also takes the value of q into account during enqueue operation of incoming packets. It is derived as follows.

A. Decreasing $\bar{q}$

Discretizing equation (1) at time instant k and converting the assignment operation to equality yields the following equation:

$$\bar{q}(k) = (1 - w_q) \cdot \bar{q}(k-1) + w_q \cdot q(k); \tag{8}$$

see [27]. By rearranging terms in equation (8) we get:

$$q(k) = \left(\frac{1 - w_q}{w_q}\right) \cdot \bar{q}(k-1) + \frac{\bar{q}(k)}{w_q}. \tag{9}$$

Equations (8) and (9) indicate the interdependency of q and $\bar{q}$ created through the use w_q in the EWMA process given in (1).

As $q(k)$ varies between 0 and B, it also traverses the regions between 0 to min_{th} and between min_{th} and max_{th}, see Fig 1. If $q(k) \leq min_{th}$ for time intervals greater than a single packet arrival time (or, for $\theta \geq 1$ consecutive packet arrivals), then we need to decrease the first term on the RHS of equation (8) by a factor ξ so that $\bar{q}$ is either less than min_{th} or close to it. In discrete terms, we can write this sub-algorithm as:

$$\bar{q}(k) = \begin{cases} (1 - w_q) \cdot \bar{q}(k-1) + w_q \cdot q(k), & \text{for } q > min_{th}; \\ \left(\frac{1-w_q}{\xi}\right) \cdot \bar{q}(k-1) + w_q \cdot q(k), & \text{for } q \leq min_{th}, \ \ \theta > 1. \end{cases} \tag{10}$$

The usefulness of equation (10) depends upon the proper choice of the pair $\{\theta, \xi\}$, otherwise it can degrade the performance of Hybrid RED and even cause stability problems by making wide variations in $\bar{q}$. Thus there must be a balance between the values of ξ and θ. For large values of θ, we should choose larger values of ξ and for smaller values of θ, ξ must be smaller.

If ξ is chosen such that $min_{th} < \bar{q} < max_{th}$ while $q(k) \leq min_{th}$, then we will still have some probabilistic packet drops as governed by equations (2) or (3). These drops will be fewer in number and thus will cause less bandwidth wastage than in the previous state of $\bar{q} > max_{th}$.

B. Packet marking/dropping based on q and $\bar{q}$.

The enqueuing and marking/dropping instances of Hybrid RED as functions of q and $\bar{q}$ are shown in Figure 4 divided into regions labeled from 1 to 11. Regions 1 and 11 correspond to forced packet dropping and 2, 3, 4, 6, 7 correspond to probabilistic or random packet marking/dropping, [26]. The RED type algorithms derived from either its basic or gentle version do forced marking/dropping of packets in all regions for which $\bar{q} > max_{th}$ (i.e. regions 4, 7 and 10). By contrast, in Hybrid RED we enqueue packets in regions 7 and 10 which are ($min_{th} < q < max_{th}$) and ($q < min_{th}$), respectively. A more conservative approach, adopted in this paper, would be to use region 10 only and not use region 7 for enqueuing operations while $\bar{q} > max_{th}$. Also skipping of region 10 will turn this sub-algorithm OFF and we will default to RED regions only.

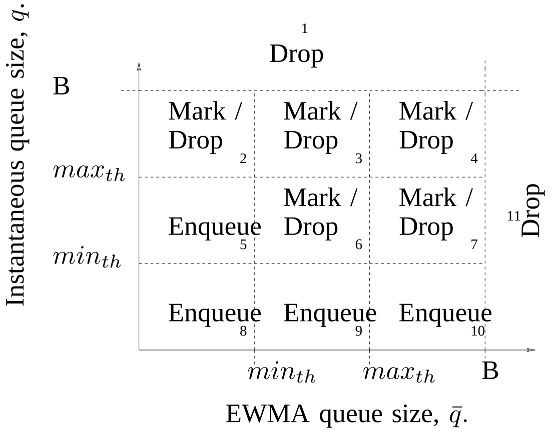


Fig. 4. Packet enqueuing and marking/dropping logic for conservative Hybrid RED algorithm, [1].

C. Pseudocode of Hybrid RED

The sub-algorithms described in the previous two subsections can be combined together into the following composite and coherent Hybrid RED algorithm:

Algorithm 1 Hybrid RED algorithm

```
procedure ComputeEWMA($w_q$, $q(k)$, $\bar{q}(k-1)$)
    for all $k$ do
        $\bar{q}(k) \leftarrow (1 - w_q) \cdot \bar{q}(k-1) + w_q \cdot q(k)$
    end for
end procedure
procedure ComputeProbability($min_{th}$, $max_{th}$, $max_p$, $\bar{q}(k)$)
    if $\bar{q}(k) < min_{th}$ then
        $p_b \leftarrow 0$
    else if $\bar{q}(k) > 2 * max_{th}$ then
        $p_b \leftarrow 1$
    else if $\bar{q}(k) \in [min_{th}, max_{th}]$ then
        $p_b \leftarrow \frac{\bar{q}(k) - min_{th}}{max_{th} - min_{th}} \cdot max_p$
    else
        $p_b \leftarrow \frac{\bar{q}(k) - max_{th}}{max_{th}} \cdot (1 - max_p) + max_p$
    end if
end procedure
procedure UniformlyDistributeProbability($p_b$, count)
    $p_a \leftarrow \frac{p_b}{1 - p_b \cdot count}$
end procedure                    ▷ RED's procedures completed
procedure ReduceEWMA($\bar{q}(k), \bar{q}(k-1), \theta, min_{th}, \xi$)
    for $\theta$ consecutive packets do
        if $\bar{q}(k) < min_{th}$ then
            $\bar{q}(k) \leftarrow \frac{\bar{q}(k-1)}{\xi}$
        else
            Proceed as normal RED
        end if
    end for
end procedure
procedure EnqueuePackets($q(k), \bar{q}(k), min_{th}, max_{th}$)
    if $(\bar{q}(k) > max_{th}) \&\& (\bar{q}(k) < 2 * max_{th}) \&\& (q(k) < min_{th})$ then
        Enqueue Packet
    else if $(\bar{q}(k) > max_{th}) \&\& (min_{th} < q(k) < max_{th})$ then
        Enqueue Packet
    else
        Proceed as normal RED
    end if
end procedure
```

VI. Performance Evaluation

The network simulator ns-2.27 [28] is used in this study. We first describe the simulation set up and the performance metrics and then we analyze the results.

A. Simulation set-up

The first simulation scenario is illustrated in Figure 5, which consists of a simple dumbbell topology consisting of three main links, a-d, b-d and c-d, each having a capacity of 100 Mbps and propagation delays of 1, 3 and 5 ms, respectively. The bottleneck link d-e has a capacity of 10 Mbps with a delay of 5 ms. Each of the nodes a, b and c is sending FTP traffic during the simulation period of 50 s. Hence, it constitutes a network with fixed traffic scenario.

The second simulation topology is shown in Figure 6. It consists of four main links, labelled as a-c, b-c, e-d and f-d, each having a capacity of 10 Mbps and delays of 2, 3, 4 and 5 ms, respectively. The bottleneck link, c-d, has a capacity 1.5 Mbps with a delay of 20 ms. The forward traffic consists of two long lived TCP flows (from a to e and b to f) and the reverse traffic consists of one long lived TCP flow (from e to b). The presence of reverse data traffic will cause ACK compression (loss of ACK packets) which will increase the burstiness in data traffic on the forward paths. At a time of 25 s twenty new TCP flows start one after the other at intervals of 0.1 s. This models a sharp change in the level of contention among competing users. Thus, this network represents a dynamic traffic scenario.

The bottleneck link is using one of the algorithms: RED [5], Adaptive RED [19], LPF/ODA [25] or Hybrid RED router and all other links are using Droptail routers. Packet size is 1000 bytes and buffer size (B) of all routers is 50 packets. We use $min_{th} = 5$ and $max_{th} = 15$ packets and $w_q = 0.002$ for Hybrid RED, which are also the default values of RED in [28]. Thus, router queues are ensured to be under active queue management as in [25]. We use all algorithms in dropping rather than in packet marking mode as we are not employing Explicit Congestion Notification (ECN), [7].

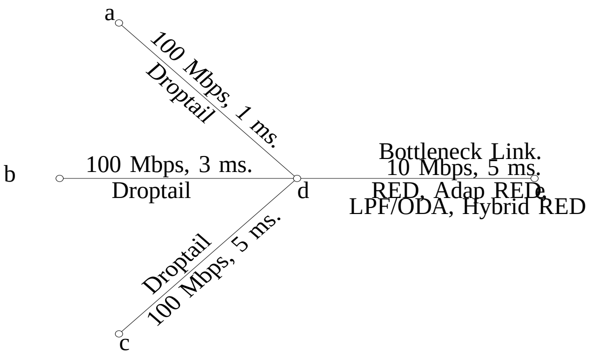


Fig. 5. Network topology 1, [1].

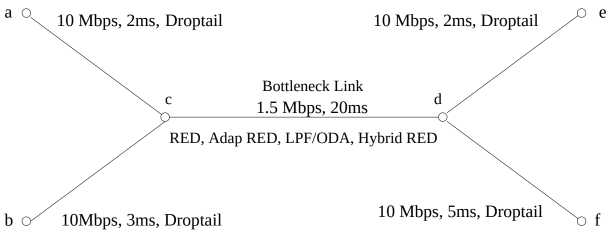


Fig. 6. Network topology 2.

The performance metrics used for the evaluation of Hybrid RED are loss rate (δ) and link utilization (η). EWMA variations are also shown. The values of δ and η are computed from the following relations:

$$\delta = \frac{\sum Packets\ Dropped\ at\ Queue}{\sum Packets\ Enqueued}; \quad (11)$$

$$\eta = \frac{\sum Packets\ sent\ over\ Bottleneck\ Link}{Maximum\ Packets\ sent\ on\ Bottleneck\ Link}; \quad (12)$$

see [29].

B. Simulation Results

We performed simulations on the networks shown in Figures 5 and 6. We employed the RED, LPF/ODA and Hybrid RED algorithms at the bottleneck link of both networks. In order to operate the Hybrid RED algorithm in an optimal fashion, we need to select suitable values of θ and ξ. In [24] [25] the values of θ and ξ are arbitrarily chosen as 3 and 2, respectively. However, this paper presents a spectrum of simulation results for different combinations of θ and ξ for the Hybrid RED algorithm.

A quantitative comparison (for both networks) between average link utilization (η_{av}) with associated error ($\Delta_{\eta_{av}}$), and average loss rate (δ_{av}) with associated error ($\Delta_{\delta_{av}}$) for the RED, Adaptive RED and LPF/ODA algorithms, after multiple runs (100) of simulation, are presented in Table I. It shows that for the static scenario, Figure 5, LPF/ODA has a lower link utilization but also a lower loss rate as compared to both RED and adaptive RED. Whereas for the dynamic scenario, Figure 6, LPF/ODA has the highest link utilization and also the lowest loss rate, as compared to RED and its adaptive version. In the case of Hybrid RED these values depend upon the selection of ξ and θ. Employing Hybrid RED at the bottleneck link of the networks shown in Figures 5 and 6, we performed simulations for all combinations of $\theta = \{1, 2, 3, 4, 5\}$ and $\xi = \{1.1, 1.25, 1.5,\ 1.75, 2.0, 2.5, 3.0\}$. Quantitative results showing the performance of Hybrid RED are given in Table II.

A comparison of the average loss rate, δ_{av}, at the bottleneck link in Figure 5, for Hybrid RED using different values of θ, is presented in Figure 15. It shows that the average loss rate starts decreasing with an increase in ξ, i.e. decrease from 1.016 % to values close to 0.8 % which is less than that for both RED and LPF/ODA. Also a comparison between average link utilization, for different values of θ, is shown in Figure 16 where with an increase in ξ it first increases from 98.7 % to 98.97 % and later decreases to lower values. Therefore, until $\xi \leq 1.1$ a tradeoff exists between lower average loss rate and higher average link utilization and for $\xi > 1.1$ both decrease together (ignoring the fluctuations due to statistical errors).

Next, a comparison between the average loss rate and average link utilization for different values of θ for Hybrid RED employed at bottleneck in Figure 6 (dynamic scenario) are shown in Figures 17 and 18, respectively. In this case, with an increase in ξ, the average loss rate starts to decrease from 8.21 % to a value close to 8.1 %. On the other hand, average link utilization starts to increase from 88.18 % to a value close to 90.5 %. In contrast to the previous static case, Figure 5, the average link utilization in this dynamic case does not decrease with an increase in ξ. Thus, in a dynamic scenario, higher values of ξ are required for attaining higher values of η_{av}. Also, a tradeoff between average loss rate and average link utilization is absent in this case. However, η_{av} increases to maximum value for $\theta = 1$ as in the static case. Hence, it is proposed that $\theta = 1$ and $\xi < 1.5$ should be chosen for nearly optimal operation of Hybrid RED as the average loss rate and average link utilization curves level out for $\xi > 1.5$.

Further, it is important to consider the effects of variations in ξ and θ on the stability of EWMA queue size. Therefore, the EWMA queue sizes for RED, LPF/ODA and Hybrid RED for the topology shown in Figure 5 have been plotted in Figures 7, 8, 9 and 10, respectively. It can be seen that the EWMA queue size for RED fluctuates between $min_{th} = 5$ and $max_{th} = 15$ packets, whereas for LPF/ODA it varies rapidly between 1 and 12 packets. In the case of Hybrid RED, the range of abrupt

TABLE I

A COMPARISON BETWEEN RED, ADAPTIVE RED AND LPF/ODA ALGORITHMS. THE VALUES OF AVERAGE LINK UTILIZATION (η_{av}) WITH ASSOCIATED ERROR ($\Delta_{\eta_{av}}$) AND AVERAGE LOSS RATE (δ_{av}) WITH ASSOCIATED ERROR $\Delta_{\delta_{av}}$ ARE EXPRESSED IN %.

Algorithm	Topology 1 (Fig. 5)				Topology 2 (Fig. 6)			
	η_{av}	$\Delta_{\eta_{av}}$	δ_{av}	$\Delta_{\delta_{av}}$	η_{av}	$\Delta_{\eta_{av}}$	δ_{av}	$\Delta_{\delta_{av}}$
RED	98.73	0.23	1.02	0.09	90.38	3.42	8.15	6.76
Adapt RED	95.38	0.36	0.97	0.06	88.93	2.22	8.25	6.75
LPF/ODA	94.93	1.04	0.84	0.18	90.67	2.04	7.93	6.80

TABLE II

PERFORMANCE OF HYBRID RED FOR $\theta = 1$ PACKET. THE VALUES OF AVERAGE LINK UTILIZATION (η_{av}) WITH ASSOCIATED ERROR ($\Delta_{\eta_{av}}$), AVERAGE LOSS RATE (δ_{av}) WITH ASSOCIATED ERROR ($\Delta_{\delta_{av}}$) ARE EXPRESSED IN %.

Hybrid RED	Topology 1 (Fig. 5)				Topology 2 (Fig. 6)			
(θ, η)	η_{av}	$\Delta_{\eta_{av}}$	δ_{av}	$\Delta_{\delta_{av}}$	η_{av}	$\Delta_{\eta_{av}}$	δ_{av}	$\Delta_{\delta_{av}}$
(1, 1.10)	98.96	0.79	0.87	0.11	89.25	2.56	8.13	6.79
(1, 1.25)	98.92	0.82	0.79	0.15	89.80	2.40	8.12	6.69
(1, 1.50)	98.86	0.90	0.77	0.14	90.24	1.73	8.10	6.64

fluctuations of EWMA queue sizes depends upon the choice of ξ.

For the dynamic network topology, shown in Figure 6, EWMA queue sizes for RED, Adaptive RED, LPF/ODA and Hybrid RED have been plotted in Figures 11, 12, 13 and 14, respectively. For Hybrid RED, the magnitude of abrupt fluctuations in EWMA queue size depends upon the choice of θ and ξ, which is less than that of LPF/ODA.

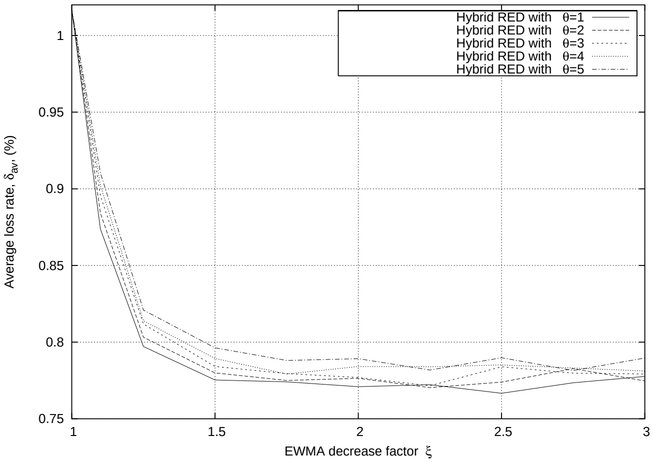


Fig. 15. Comparison of average loss rate, δ_{av}, of Hybrid RED algorithm, for different values of θ, for simulation topology shown in Figure 5.

VII. CONCLUSIONS

The limitations of using only the EWMA queue size for packet marking/dropping in RED type algorithms at time of congestion are discussed. A Hybrid RED algorithm is proposed which combines EWMA queue size with the instantaneous queue size for making packet marking/dropping decisions and which also incorporates the adjustments to the EWMA queue size when $q < min_{th}$ over a certain number of consecutive packet arrivals. We experimented with a reduction in the EWMA queue size when $q < min_{th}$ for 1 to 5 packet arrivals and the results are presented.

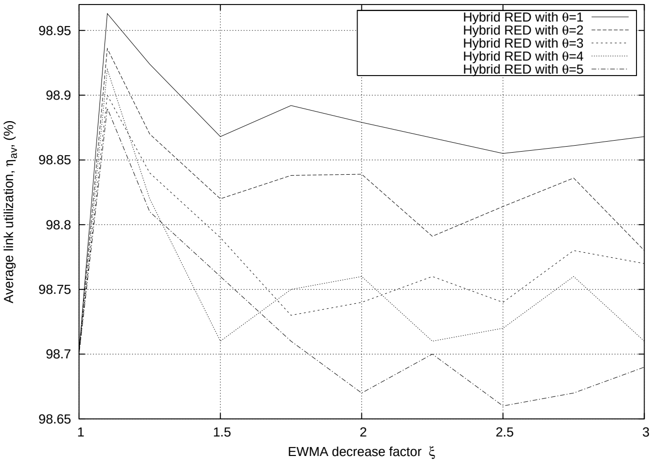


Fig. 16. Comparison of average link utilization, η_{av}, of Hybrid RED algorithm, for different values of θ, for simulation topology shown in Figure 5.

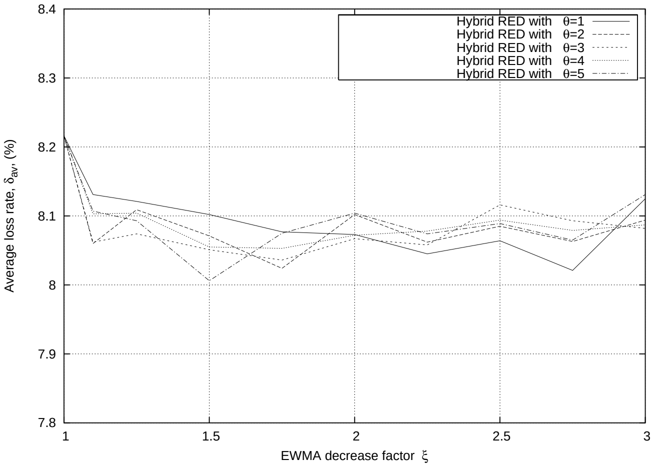


Fig. 17. Comparison of average loss rate, δ_{av}, of Hybrid RED algorithm, for different values of θ, for simulation topology shown in Figure 6.

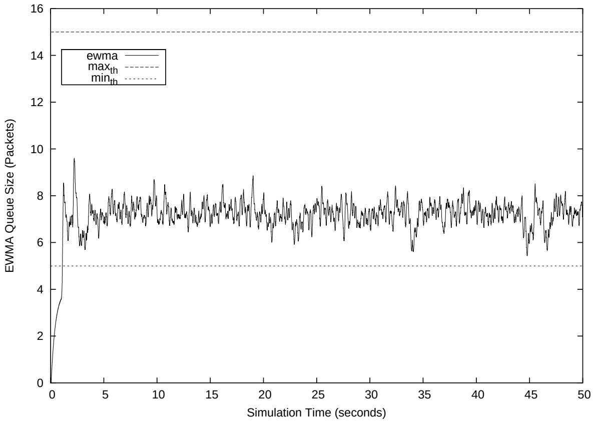


Fig. 7. EWMA queue size of RED algorithm for simulation topology shown in Figure 5.

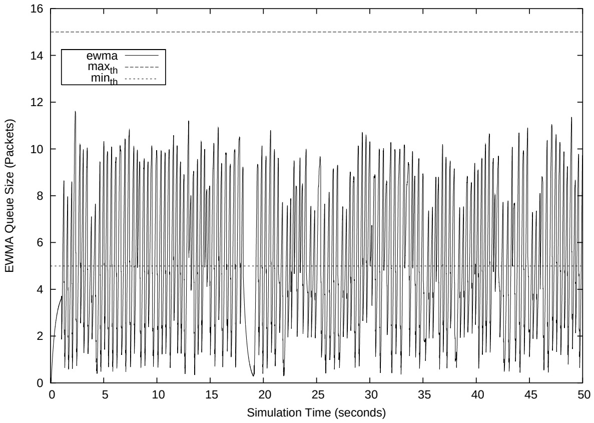


Fig. 8. EWMA queue size of LPF/ODA algorithm for simulation topology shown in Figure 5.

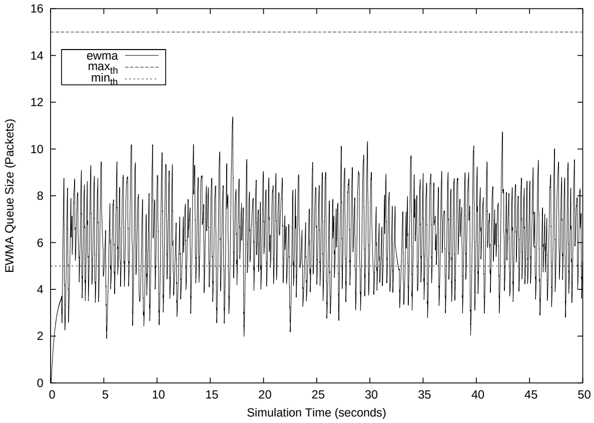


Fig. 9. EWMA queue size of Hybrid RED algorithm with $\{\theta, \xi\} = \{1, 1.25\}$ for simulation topology shown in Figure 5.

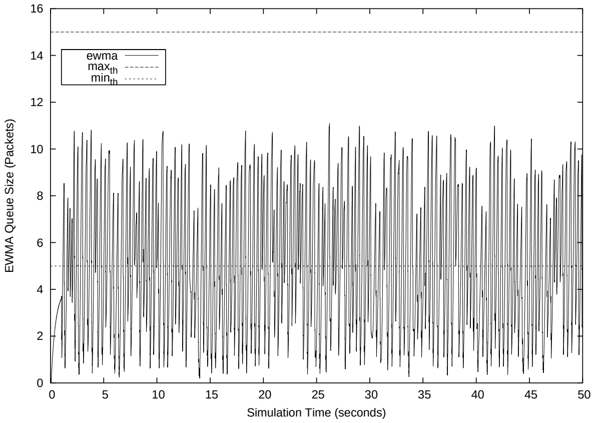


Fig. 10. EWMA queue size of Hybrid RED algorithm with $\{\theta, \xi\} = \{1, 2.0\}$ for simulation topology shown in Figure 5.

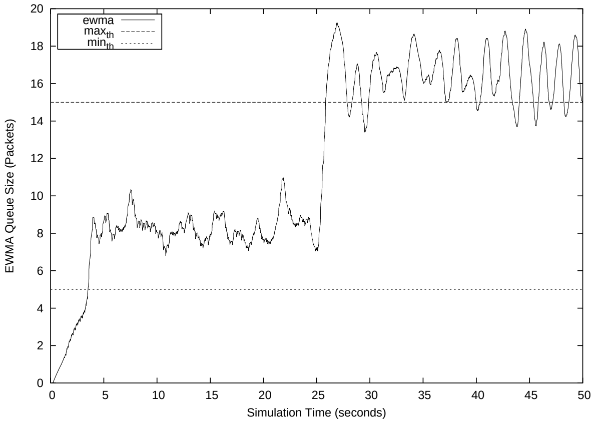


Fig. 11. EWMA queue size of RED algorithm for simulation topology shown in Figure 6.

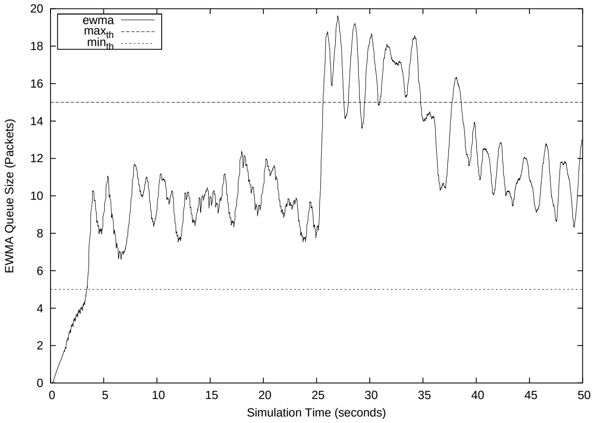
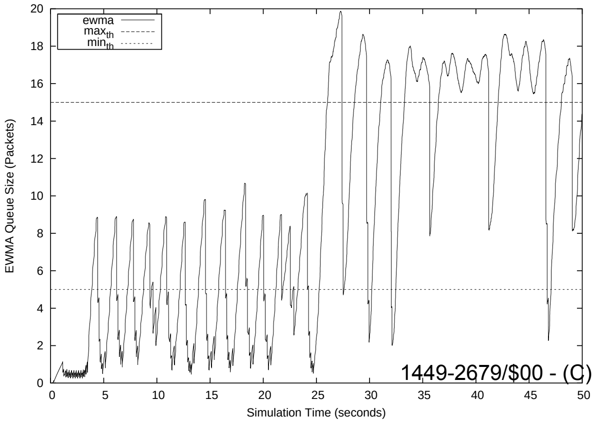


Fig. 12. EWMA queue size of Adaptive RED algorithm for simulation topology shown in Figure 6.



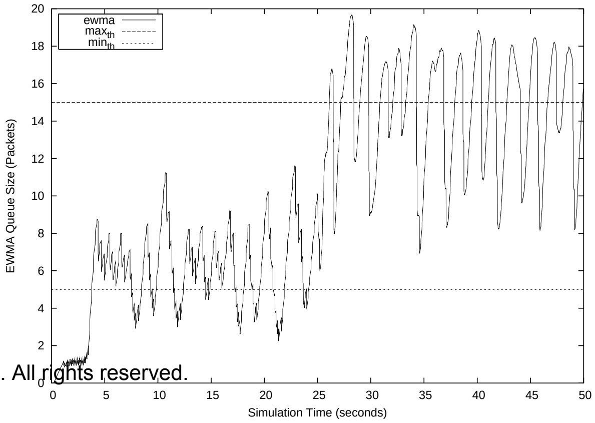


Fig. 13. EWMA queue size of LPF/ODA algorithm for simulation topology shown in Figure 6.

Fig. 14. EWMA queue size of Hybrid RED algorithm with $\{\theta, \xi\} = \{1, 1.25\}$ for simulation topology shown in Figure 6.

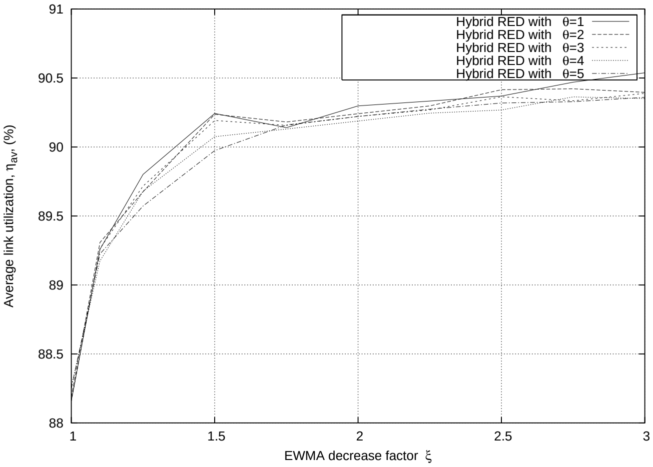


Fig. 18. Comparison of average link utilization, η_{av}, with Hybrid RED algorithm for simulation topology shown in Figure 6.

In order to study the queue dynamics, we have plotted EWMA queue size graphs for networks with both static and dynamic traffic scenarios. Generally, for higher values of ξ, the EWMA queue size becomes noisy and thus fluctuates rapidly, which might cause non-convergence of queue size and stability problems. On the other hand, higher values of ξ do improve bottleneck link loss rate. Therefore, we conclude that there exists a trade-off between stable queue behaviour and better link utilization, as well as lower loss rate.

These simulation studies show that for a static network scenario the new Hybrid RED algorithm gives a better loss rate and link utilization compared to the existing RED and LPF/ODA algorithms. In the case of dynamic traffic scenarios the performance of Hybrid RED depends upon the choice of θ and ξ.

One way to optimize θ and ξ would be to employ the active queue management model, [30], and control theoretic techniques, [31], [32] and [18]. The Hybrid RED algorithm will effectively change the transfer function of the overall control loop and thus the stability of network. Thus, feedback controllers may be designed which can adjust these parameters according to the desired stability margins, [33], and quality of service policy, [34].

In the future, we intend to study Hybrid RED under different traffic mixes such as exponential ON/OFF traffic over UDP with TCP, and also under ECN based packet marking.

References

[1] A. Haider, H. Sirisena, and K. Pawlikowski, "Improved Congestion Control with Hybrid RED," *Proceedings of the 10th International Conference on Telecommunications*, vol. 2, pp. 923–928, 2003, papeete, Tahiti.

[2] B. Braden, D. Clark *et al.*, "Recommendations on Queue Management and Congestion Avoidance in the Internet," *Network Working Group, Request for Comments: 2309*, April 1998, category: Informational.

[3] M. Allman, V.Paxon, and W. Stevens, "TCP Congestion Control," *Network Working Group, RFC 2581*, April 1999.

[4] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, August 1999, simulation scripts available at http://www.icir.org/fbyd/end2end-paper.html.

[5] 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, http://www.icir.org/fbyd/red.html.

[6] ——, "The Synchronization of Periodic Routing Messages," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 122–136, April 1994.

[7] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," *Network Working Group, RFC 3168*, September 2001, category: Standards Track.

[8] C.-Q. Yang and A. V. S. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks," *IEEE Network*, vol. 9, no. 4, pp. 34–45, July-August 1995.

[9] A. Haider, H. Sirisena, K. Pawlikowski, and M. J. Ferguson, "Congestion Control Algorithms in High Speed Telecommunication Networks," *Proceedings of 36th Annual ORSNZ, Christchurch, New Zealand.*, pp. 88 – 97, November-December 2001, http://www.mang.canterbury.ac.nz/orsnz/conf2001/papers/Haider.pdf.

[10] C. Zhu, O. W. W. Yang, J. Aweya, M. Ouellette, and D. Y. Montuno, "A Comparison of Active Queue Management Algorithms Using the OPNET Modeler," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 158–167, June 2002.

[11] S. Floyd, http://www.icir.org/fbyd/red.html, 2002.

[12] P. Young, *Recursive Estimation and Time-Series Analysis*. Springer-Verlag Berlin, Heidelberg, Germany: Sringer-Verlag, 1984, ISBN 3-540-13677-0.

[13] A. Misra, T. Ott, and J. Baras, "Effect of Exponential Averaging on the Variability of a RED Queue ," *Proceedings of IEEE International Conference on Communications, ICC 2001.*, vol. 6, pp. 1817–1823, June 2001.

[14] M. May, C. Diot, B. Lyles, and J. Bolot, "Influence of Active Queue Management Parameters on Aggregate Traffic Performance," INRIA, France, Tech. Rep. RR-3995, August 2000, ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-3995.pdf.

[15] L. Rizzo, "RED and non-responsive fbws," end2end-interest mailing list, ftp://ftp.isi.edu/end2end/end2end-interest-1997.mail;type=a, June 1997.

[16] S. Floyd, "RED: Discussions of setting parameters," Email available at http://www.aciri.org/fbyd/REDparameters.txt, November 1997.

[17] ——, "Recommendation on using the gentle_ variant of RED," Technical note available at http://www.aciri.org/fbyd/red/gentle.html, March 2000.

[18] H. Sirisena, A. Haider, and K. Pawlikowski, "Auto-Tuning RED for Accurate Queue Control," *Proceedings of the IEEE GLOBECOM'02*, vol. 2, pp. 2010–2015, November 2002, taipei, Taiwan. http://www.globecom2002.com.tw/.

[19] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," Unpublished draft available at http://www.icir.org/fbyd/papers/adaptiveRed.pdf, August 2001.

[20] T. Ott, "On the Ornstein-Uhlenbeck Process with Delayed Feedback," Unpublished draft available at http://web.njit.edu/∼ott/Papers/, December 1999.

[21] V. Jacobson, K. Nichols, and K. Poduri, "RED in a Different Light," Unpublished draft available at http://www.cnaf.infn.it/∼ferrari/papers/ispn/red_light_9_30.pdf, September 1999.

[22] D. Lin and R. Morris, "Dynamics of Random Early Detection," *ACM Computer Communication Review*, vol. 27, no. 4, pp. 127–136, October 1997, cannes, France.

[23] I. Stoica, S. Shenker, and H. Zhang, "Core Stateless Fair Queueing: Achieving Approximatley Fair Bandwidth Allocations in High Speed Networks," *Proceedings of the ACM SIGCOMM'98*, vol. 28, no. 4, pp. 118–130, September 1998.

[24] B. Zheng and M. Atiquzzaman, "Low Pass Filter/Over Drop Avoidance (LPF/ODA): An algorithm to improve the performance of RED gateways," University of Oklahoma, USA, Tech. Rep. CS-TR-01-001, February 2001, http://www.icir.org/fbyd/red.html.

[25] ——, "Low pass filter/over drop avoidance (LPF/ODA): an algorithm to improve the response time of RED gateways," *Wiley International Journal of Communication Systems*, vol. 15, no. 10, pp. 899–906, December 2002, http://www3.interscience.wiley.com/cgi-bin/jissue/102522518.

[26] S. Floyd and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control," Network Research Group, Lawrence Berkeley National Laboratory, Berkeley, CA, USA, Tech. Rep., February 1997, simulation scripts available at http://www.icir.org/fbyd/papers/collapse/ns2/mechanisms.html.

[27] G. H. Hostetter, *Digital Control System Design*. 111 Fifth Avenue, New York 10003, USA: Holt, Rinehart and Winston, Inc., 1988.

[28] DARPA/VINT, LBNL, Xerox, UCB, and USC/ISI, “Network simulator ns 2.27,” Information available at http://www.isi.edu/nsnam/ns/, 2004, code available at http://www.isi.edu/nsnam/dist/.

[29] W.-C. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, “A Self Configuring RED Gateway,” *Proceedings of the IEEE INFOCOM 1999*, pp. 1320 –1328, March 1999, eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies.

[30] V. Misra, W.-B. Gong, and D. Towsley, “Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED,” *Proceedings of the ACM SIGCOMM’00*, pp. 151–160, 2000.

[31] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, “A Control Theoretic Analysis of RED,” *Proceedings of the IEEE INFOCOM 2001*, vol. 3, pp. 1510–1519, April 2001, Anchorage, Alaska, USA.

[32] ——, “On Designing Improved Controllers for AQM Routers Supporting TCP Flows,” *Proceedings of the IEEE INFOCOM 2001*, vol. 3, pp. 1726–1734, April 2001, Anchorage, Alaska, USA.

[33] K. J. Astrom and B. Wittenmark, *Computer Controlled Systems*, 2nd ed. Englewood Cliffs, N.J. 07632, USA.: Prentice-Hall International, Inc, 1990.

[34] B. Wydrowski and M. Zukerman, “QoS in Best-Effort Networks,” *IEEE Communications Magazine*, vol. 40, no. 12, pp. 44–49, December 2002.