

Receiver-Window Modified Random Early Detection (RED-RWM) Active Queue Management Scheme: Modeling and Analysis

Visvasuresh Victor Govindaswamy,
University of Texas at Arlington
Texas, USA
victor@uta.edu

Gergely Záruba
University of Texas at Arlington
Texas, USA
zaruba@uta.edu

G.Balasekaran
University of Pittsburgh
Pennsylvania, USA
bala@hgen.pitt.edu

Abstract – In this paper, we present a mathematical model for a novel TCP congestion control approach called Receiver-Window Modification (RWM). RWM could be used with any of the present Active Queue Management (AQM) schemes, such as Random Early Detection (RED), Adaptive Random Early Detection (ARED), and BLUE queues to reduce congestion at the ingress and gateway routers. We extend the work done on stochastic modeling of RED-ECN gateways to RED-RWM gateways. Using this model, we provide a weak convergence theorem for the number of connections; thus leading the way to more relaxed configuration of RED-RWM gateway parameters. We use a Monte Carlo simulation method to compare outcomes of our mathematical model with those of NS2 simulation experiments.

I. INTRODUCTION

TCP is the connection oriented transport layer protocol of the Internet. One of the services TCP provides is the cutting back on the transmission rate of flows whenever a congestion is encountered along the way of the packet flow. Such congestion at routers may be caused by too many sources trying to send an excessive amount of data with a rate too high for the network to handle. Although the above used de-facto mechanism works in controlling the rate thus reducing congestion in the network, it is rather ad hoc and has been shown to produce unexpected behavior, such as introducing chaotic behavior [1].

Modeling and analysis of such TCP connections and their interaction with queues has been an active research area. In [2] a Markov chain based model is analyzed for N TCP connections using either Tail-Drop or RED gateways, where the state space is the vector of window size of all of the N connections, the queue size is a function of the sum of the size of congestion windows and state transition probabilities are dependent on packet loss probability at the router buffer, and the slow start (SS) and congestion avoidance (CA) phases of TCP. Many of the past work assume that the packet loss probability is constant but it is in fact dependent on the buffer size and packet discarding discipline at the router and the window size of the transmission. In [3], the authors use a previously developed nonlinear dynamic model of TCP to

analyze and design RED by relating its parameters such as the low-pass filter break point and loss probability profile to the network parameters. Finally, Tinnakornsriruphap et al. [4] have modeled a RED queue with TCP connections that explicitly incorporate complete packet level operations in TCP, probabilistic packet marking mechanism in RED, heterogeneous round trip times (RTTs) of TCP flows and session-layer dynamics, *i.e.*, connection arrivals and departures. Their model avoids the state space explosion of [5] as well, assuming that the RED queue does not drop any packets.

In this paper, we present a mathematical model for the Receiver Window Modification (RWM) scheme introduced in [6] that can be used to complement the RED scheme at the ingress and gateway routers. Our model extends the modeling work done in [4] to the RWM extension, henceforth known as RED-RWM, relaxing the assumption that the buffer size is infinite. Here, upon selection of packets, the RED-RWM queue, instead of marking them, sets the receiver window field to one maximum segment size (MSS) in the ACK packets that are going towards the sender from the receiver. The main idea of the RWM approach is to restrict the TCP transmission window with the flow control window instead of the congestion control window, thus controlling the transmission window with a finer granularity. Results from simulations [6] show that RWM extended queues outperform their counterparts (functioning with and without ECN) in terms of packet drops, packet delay, delay jitter and throughput.

The rest of the paper is organized as follows: Section II covers the necessary background information and previous work. In Section III, the RWM model is introduced; simulation details, results as well as analysis are covered in Section IV. The paper is concluded in Section V.

II. BACKGROUND INFORMATION

This section gives an overview of the various major topics involved in this research. Subsections A and B provide an overview of the RED AQM and RWM mechanisms respectively.

A. Random Early Detection

Random Early Detection (RED) [7] employs a dual-threshold approach defining a minimum threshold value (Th_{min}), a maximum threshold value (Th_{max}), and a variable dropping probability $P_a \leq P_{max}$. There are two parts to the RED algorithm; the estimation of average queue size and the decision on which packet to drop. If the average queue size is smaller than Th_{min} then the arriving packet is accepted. If the average queue size is between Th_{min} and Th_{max} , then the packet is dropped with probability P_a . (P_a is a linear function of the average queue size, possibly increasing to a pre-determined maximum value P_{max} .) However, if the calculated average queue size is larger than Th_{max} , then the incoming packet is dropped with probability one.

In [4], Peerapol et al. have presented a RED model that deals with TCP transmission rate in terms of packet-level operations. Our RED-RWM model is an extension of this model; hence it has the following adopted features:

1. Incorporates a more accurate Additive Increase / Multiplicative Decrease (AIMD) window mechanism of TCP. The AIMD algorithm controls the window size in response to the modified acknowledgement packet from the bottleneck RED-RWM gateway.
2. Provides a more realistic session-level dynamics, *i.e.*, arrivals and terminations of flows, and the variability of round-trip delays.
3. Provides a realistic interaction between TCP and RED-RWM gateway.

B. Receiver-Window Modification (RWM)

AQM solutions, such as RED, use packet drops at router queues to manipulate the TCP sender into decreasing its transmission rate. With ECN-compliant RED queues, packets are not unnecessarily dropped but marked as if they were dropped. However, as a downside: i) ECN messages may get lost or dropped by downstream routers; and ii) TCP implementations at both the source and the destination have to be ECN-compliant (which presents a significant problem in today's implementations). Currently, there is no practical benefit in setting ECN bits in Internet packets; as this requires ECN capable routers (at least at bottleneck points), and servers and clients throughout a network. In [8], experiments were conducted using TBIT (the TCP Behavior Inference Tool) showing that in September 2000 only 21 out of 26447 (0.07%) sites positively responded with an appropriate SYN/ACK. In March 2002, only 7 out of 12364 sites (0.05%) responded positively. These tests included sites running nearly all types of operating systems.

In [6], we had proposed in using flow control feedback to reduce congestion at ingress and gateway routers. Instead of setting ECN related bit in the chosen packets of RED queues, we proposed the use of receiver window modification (RWM). RWM sets the receiver window field to one maximum segment size (MSS) in the ACK packets that are going towards the sender from the receiver. (Note that whenever the TCP header of an ACK packet is

modified, the checksum in the TCP header needs to be adjusted for error control.)

III. RED-RWM MODEL

This section presents the model for the RED-RWM gateway with N TCP connections competing for bandwidth. We have expanded on the model developed for the ECN/RED gateway in [4].

We will discretize time T with a uniform duration ΔT and make use of timeslots represented by $[t*\Delta T, (t+1)*\Delta T)$ where t is an integer starting at 0 (thus we will be talking about timeslot t or time $t*\Delta T$). The round-trip times (RTTs) of TCP connections are approximated as integer multiples of the length of timeslots (ΔT). We define a logical indicator function we denote as: $1[L]$ which evaluates to 1 if L is true and 0 if L is false. We also define the following variables:

1. $f_m^{(N)}(t) : \mathbf{R}_+ \rightarrow [0,1]$: the feedback (segment (or packet) acknowledgement modifying) probability function in the RED-RWM gateway when there are x number of packets in the queue being used by N number of sources. Hence, the probability of acknowledgements being modified by the RED-RWM gateway is $f_m^{(N)}(t)$. ($f_m^{(N)}(t)$ should scale with N .)
2. $f_d^{(N)}(t) : \mathbf{R}_+ \rightarrow [0,1]$: the segment dropping probability function in the RED-RWM gateway when there are x number of packets in the queue being used by N number of sources. The same rules apply to $f_d^{(N)}(t)$ as to $f_m^{(N)}(t)$.
3. $Q^{(N)}(t)$: the queue size at time $\Delta T*t$ with N number of TCP connections. During timeslot t , this size will increase or decrease if the number of incoming packets at the queue is greater or less (respectively) than the number of outgoing packets.
4. $A_i^{(N)}(t)$: the number of TCP segments received by the RED-RWM gateway from TCP connection i ($i \in \{1, \dots, N\}$) within timeslot t ; thus, the total number incoming segments at the RED-RWM queue is:
$$A^{(N)}(t) = \sum_{i=1}^N A_i^{(N)}(t). \quad (1)$$
 $A_i^{(N)}(t)$ can be modeled by an integer-valued random variable that encodes the transmission window size (in packets) at time $\Delta T*t$; we assume that $A_i^{(N)}(t) \in \{0, \dots, W_{max}\}$, where W_{max} represents the receiver advertised window size of the connection (the transmission window size of an idle connection is zero).
5. $C^{(N)}(t)$: the number of TCP segments served by the RED-RWM gateway during timeslot t .
6. Since the queue is always non-negative, using the above variables, the queue dynamics can be represented by the following recursive function:

$$Q^{(N)}(t+1) = [Q^{(N)}(t) + A^{(N)}(t) - C^{(N)}(t)]^+ \quad (2)$$

In addition, both $A^{(N)}(t)$ and $C^{(N)}(t)$ follow some stochastic recursions which depend on the current queue size thereby forming a stochastic feedback system. If we denote the average queue serving capacity per connection by C , then $C^{(N)}(t) = NC$, thus (2) becomes:

$$Q^{(N)}(t+1) = [Q^{(N)}(t) + A^{(N)}(t) - CN]^+ \quad (3)$$

7. $W_i^{(N)}(t)$: the transmission window size at connection i at the beginning of timeslot t . The transmission windows size is the minimum of $cwnd$ and $rwnd$. We assume that each connection will transmit all TCP segments that it can during slot t . Note, that $A_i^{(N)}(t) \leq W_i^{(N)}(t)$. (More precisely if $\Phi_i^{(N)}(t)$ is the number of pending TCP segments for connection i at time $\Delta T * t$ and connections have an unlimited amount of segments to transmit, then $A_i^{(N)}(t) = W_i^{(N)}(t) - \Phi_i^{(N)}(t)$.)

8. $X_i^{(N)}(t)$: the remaining workload (in packets) at the TCP sending process of connection i at the beginning of timeslot t ; thus, $A_i^{(N)}(t) \leq \min(W_i^{(N)}(t), X_i^{(N)}(t))$. (More precisely, $A_i^{(N)}(t) = \min(W_i^{(N)}(t) - \Phi_i^{(N)}(t), X_i^{(N)}(t))$)

9. $D_i^{(N)}(t)$: the round trip time (RTT) at time $t * \Delta T$ as observed by the TCP sending process of connection i ; $D_i^{(N)}(t)$ will be given in integer multiples of ΔT . We model $D_i^{(N)}(t)$ as a random variable of possible outcomes: $\{2, \dots, D_{max}\}$.

10. We will assume that any congestion-control action occurs at the end of a round-trip. We also simplify the segment transmissions to the queue by assuming that all segments allowed within an RTT are transmitted in the same slot to the queue. Let $\beta_i^{(N)}(t)$ represent the number of timeslots (at timeslot t) since the last time connection i has transmitted a segment to the queue; then:

$$\begin{aligned} \beta_i^{(N)}(0) &= \mathbb{1}[X_i^{(N)}(0) > 0], \\ \beta_i^{(N)}(t+1) &= (1 + \beta_i^{(N)}(t) \mathbb{1}[\beta_i^{(N)}(t) < D_i^{(N)}(t)]) \times \\ &\quad \mathbb{1}[X_i^{(N)}(t) > 0]. \end{aligned} \quad (4)$$

With these assumptions our formula for $A_i^{(N)}(t)$ will evolve to:

$$A_i^{(N)}(t) = \min(W_i^{(N)}(t), X_i^{(N)}(t)) \mathbb{1}[\beta_i^{(N)}(t) = 1], \quad (5)$$

where the indicator function $\mathbb{1}[\beta_i^{(N)}(t) = 1]$ is to ensure that the connection transmits only once per RTT. (Note, that the above assumptions eliminate the need for variable $\Phi_i^{(N)}(t)$.) Now our formula for the queue evolution can be rewritten as in (6).

11. Let us introduce function $G_{i,t}(a,b)$ evaluating to a if the last transmission action of connection i at time $t * \Delta T$ was within the then applying roundtrip time, and evaluating to b otherwise. G will be a useful tool to change variables only after a RTT has passed. It is shown in (7).

12. Let the modifying and dropping probability functions be given by (8) and (9) respectively.

$$\begin{aligned} Q^{(N)}(t+1) &= \\ &[Q^{(N)}(t) - NC + \sum_{i=1}^N \min(W_i^{(N)}(t), X_i^{(N)}(t)) \times \\ &\quad \mathbb{1}[\beta_i^{(N)}(t) = 1]] \end{aligned} \quad (6)$$

$$G_{i,t}(a,b) = a \cdot \mathbb{1}[\beta_i^{(N)}(t) > 1] + b \cdot \mathbb{1}[\beta_i^{(N)}(t) = 1] \quad (7)$$

$$f_m^{(N)}(\bar{Q}^{(N)}(t)) = \begin{cases} 0 & Th_{min} > \bar{Q} \\ P_{max} \left(\frac{\bar{Q} - Th_{min}}{Th_{max} - Th_{min}} \right) & Th_{min} \leq \bar{Q} \leq Th_{max} \\ 0 & \bar{Q} > Th_{max} \end{cases} \quad (8)$$

$$f_d^{(N)}(\bar{Q}^{(N)}(t)) = \begin{cases} 0 & Th_{max} > \bar{Q} \\ 1 & \bar{Q} > Th_{max} \end{cases} \quad (9)$$

where P_{max} is the maximum marking probability, Th_{min} is the minimum threshold, Th_{max} is the maximum threshold.

13. Since the RED-RWM scheme complements the RED queue, it uses the same queue averaging filter as the RED queue (known as Exponentially Weighted Moving Average -EWMA) to evaluate the average queue size based on the instantaneous queue size with a past-weighting parameter α ($0 < \alpha \leq 1$). Let $\hat{Q}^{(N)}(t)$ represent average queue at time $t * \Delta T$, then:

$$\hat{Q}^{(N)}(t+1) = (1 - \alpha) \hat{Q}^{(N)}(t) + \alpha Q^{(N)}(t+1) \quad (10)$$

14. Let $z_{i,j}^{(N)}(t+1)$ represent random indicator variables for dropped packets, where $j \in \{1, \dots, A_i^{(N)}(t)\}$; $z_{i,j}^{(N)}(t+1)$ is zero if the j^{th} packet of connection i is dropped in timeslot t (and 1 otherwise). Thus:

$$z_{i,j}^{(N)}(t+1) = \mathbb{1}[(L_{i,j}^{(N)}(t+1) > f_d^{(N)}(\hat{Q}^{(N)}(t))] \quad (11)$$

where $L_{i,j}(t)$ s are I.I.D. (in all their dimensions) uniform random variables taking values from $[0, 1]$. Hence, the indicator function of the event that no packet of connection i is dropped in timeslot t can be written as:

$$\begin{aligned} z_i^{(N)}(t+1) &= \begin{cases} \prod_{j=1}^{A_i^{(N)}(t)} z_{i,j}^{(N)}(t+1) & A_i^{(N)}(t) \geq 1 \\ 1, & A_i^{(N)}(t) = 0. \end{cases} \\ z_i^{(N)}(0) &= 1 \end{aligned} \quad (12)$$

If $z_i^{(N)}(t+1) = 0$, then at least one packet from connection i in timeslot t has been dropped by the RED-RWM gateway. This information may only be used one RTT later to halve

the value of the congestion window size, thus we need to propagate a zero value to the end of the roundtrip time. We will use variable $Z_i^{(N)}(t)$ for such propagation. $Z_i^{(N)}(t)$ thus evolves according to:

$$\begin{aligned} Z_i^{(N)}(t+1) &= G_{i,t+1}(Z_i^{(N)}(t) * z_i^{(N)}(t+1), 1) \\ Z_i^{(N)}(0) &= 1 \end{aligned} \quad (13)$$

Now we can define variable $\hat{Z}_i^{(N)}(t)$ to reflect that there was a drop during the previous RTT. $\hat{Z}_i^{(N)}(t)$ evolves as:

$$\hat{Z}_i^{(N)}(t+1) = G_{i,t+1}(\hat{Z}_i^{(N)}(t), Z_i^{(N)}(t)) \quad (14)$$

15. Similarly, let the modification of acknowledgement packets by the RED-RWM gateway be represented by an indicator random variable $m_{i,j}^{(N)}(t+1)$ where $j \in \{1, \dots, A_i^{(N)}(t)\}$; $m_{i,j}^{(N)}(t+1)$ is zero if the j^{th} packet of connection i causes the acknowledgement packet to be modified by the RED-RWM gateway to one MSS in timeslot t :

$$m_{i,j}^{(N)}(t+1) = \mathbb{I}[(V_{i,j}(t+1) > f_m^{(N)}(\hat{Q}^{(N)}(t)))] \quad (15)$$

where $V_{i,j}(t)$ s are I.I.D. (in all their dimensions) uniform random variables taking values from $[0, 1]$. Hence, the indicator function of the event that no packet of connection i is RWM-modified in timeslot t can be written as

$$\begin{aligned} m_i^{(N)}(t+1) &= \begin{cases} \prod_{j=1}^{A_i^{(N)}(t)} m_{i,j}^{(N)}(t+1) & A_i^{(N)}(t) \geq 1 \\ 1, & A_i^{(N)}(t) = 0. \end{cases} \\ m_i^{(N)}(0) &= 1 \end{aligned} \quad (16)$$

If $m_i^{(N)}(t+1) = 0$, then at least one packet from connection i in timeslot t has been RWM-modified by the RED-RWM gateway. This information may only be used one RTT later to modify the transmission to 1 MSS, thus we need to propagate a zero value to the end of the roundtrip time. We will use variable $M_i^{(N)}(t)$ for such propagation. $M_i^{(N)}(t)$ thus evolves according to:

$$\begin{aligned} M_i^{(N)}(t+1) &= G_{i,t+1}(M_i^{(N)}(t) * m_i^{(N)}(t+1), 1) \\ M_i^{(N)}(0) &= 1 \end{aligned} \quad (17)$$

Now we can define variable $\hat{M}_i^{(N)}(t)$ to reflect that there was a drop during the previous RTT. $\hat{M}_i^{(N)}(t)$ evolves according to:

$$\hat{M}_i^{(N)}(t+1) = G_{i,t+1}(\hat{M}_i^{(N)}(t), M_i^{(N)}(t)) \quad (18)$$

16. Connection i , after receiving the modified acknowledgement from the RED-RWM gateway, will transmit only one MSS for one RTT. After the RTT the source will transmit the minimum of the previous congestion window before modification and the new advertised receiver window (unless it receives more modified acknowledgements from the RED-RWM gateway). Thus, we need to save the transmission window's

size before the RWM modification has happened using variable $\Omega_i^{(N)}(t)$. $\Omega_i^{(N)}(t)$ evolves according to:

$$\begin{aligned} \Omega_i^{(N)}(t+1) &= G_{i,t+1}(\Omega_i^{(N)}(t), W_i^{(N)}(t+1)\hat{M}_i^{(N)}(t+1) \\ &\quad + \Omega_i^{(N)}(t)(1 - \hat{M}_i^{(N)}(t+1))) \\ \Omega_i^{(N)}(0) &= 1 \end{aligned} \quad (19)$$

17. We also need an indicator variable at each roundtrip's end time, showing whether there has been a marking two roundtrips ago while there has been none in the roundtrip that was just finished.

$$E_i^{(N)}(t+1) = G_{i,t+1}(0, (1 - \hat{M}_i^{(N)}(t))M_i^{(N)}(t)) \quad (20)$$

18. Each connection may be in one of two basic modes i.e., either in an active or an idle mode. At the beginning of timeslot t if a session has no packet to transmit then it is in the idle mode. An idle connection may become active by the beginning of the next timeslot with probability p_{ar} independently of previous events. Hence the duration of an idle period is *geometrically* distributed with parameter p_{ar} (and a mean of $1/p_{ar}$). We can use this to capture the dynamics of connection arrivals, where the interarrival times are exponentially distributed (as the geometric distribution is the discrete equivalent to the exponential distribution), we let i) $U_i(t)$ ($i \in \{1, \dots, N\}$) represent a collection of I.I.D. uniform random variables on $[0, 1]$ and ii) $\mathbb{I}[U_i(t+1) \leq p_{ar}]$ represent an indicator function for an idle connection i for the event of the arrival of a new packet in timeslot $t+1$.

Let $F_i(t)$ represent the total number of TCP segments (the session size) connection i will have to transmit if it becomes active at the beginning of timeslot t ; $F_i(t)$ can be modeled as a collection of I.I.D. non-negative integer-valued random variables with some general probability mass function (usually a geometric or a discretized exponential pmf).

Recall, that we used $X_i^{(N)}(t)$ to denote the remaining workload (expressed in TCP segments) of connection i at the beginning of timeslot t , thus in (21).

19. RTT will not change for a connection during the same RTT, thus we can formulate it using the $G_{i,t}(a,b)$ function. At the beginning of time slot $t+1$ (if $t+1$ is the timeslot for connection i to send its data) the expected roundtrip time is determined by the proportion of how many segments are in the queue at time $\Delta T^*(t+1)$ plus how many are expected during slot $t+1$ (which is $A_i^{(N)}(t+1)$) and the rate with which the queue is served, as in (22).

Note, that this can lead to synchronization if we assume that all connections arrive at the same time. This synchronization will be removed by the random marking and dropping as t grows. The above formulation also need

$D_i^{(N)}(t)$ to be at least two as we expect the acknowledgments to come back in a different time slot than what was used to

$$X_i^{(N)}(t+1) = 1[X_i^{(N)}(t) > 0][X_i^{(N)}(t) - A_i^{(N)}(t)] + 1[X_i^{(N)}(t) = 0][U_i(t+1) \leq P_{ar}]F_i(t+1) \quad (21)$$

$$D_i^{(N)}(t+1) = \max(G_{i,t+1} = (D_i^{(N)}(t), (Q^{(N)}(t+1) + A^{(N)}(t+1))/(N * C)), 2) \quad (22)$$

send their segments.

The evolution of the window mechanism for connection i can now be described through recursion. The congestion-control mechanism of TCP operates either in the *slow start* (SS) or the *congestion avoidance* (CA) phase. A new connection starts in SS in order to probe the available bandwidth of the network (the queue in our case). While in SS, the congestion window size is doubled every round-trip time until the receiver window field of one or more acknowledgement packets are modified by the RED-RWM gateway to one MSS or one or more packets are dropped by the RED-RWM gateway. Note that the transmission window size is limited by the receiver advertised window size W_{\max} . Thus, if the connection is in SS then the transmission window of connection i evolves according to (24).

20. In CA, the congestion window size in the next timeslot ($t+1$) is incremented if no modified acknowledgement packet was received in timeslot t , while if one or more segments are modified in timeslot t the congestion window in the next timeslot is set to one segment. The congestion window size in CA is then given by (25).

Let the $S_i^{(N)}(t)$ random indicator variable denote the state of TCP connections ($S_i^{(N)}(t)$ is zero if connection i is in CA and one if it is in SS). Therefore, combining (24) and (25), the complete recursion of the congestion window size can be written as:

$$W_i^{(N)}(t+1) = (S_i^{(N)}(t)W_{i,SS}^{(N)}(t+1) + (1 - S_i^{(N)}(t))W_{i,CA}^{(N)}(t+1)) \times 1[X_i^{(N)}(t) - A_i^{(N)}(t) > 0] \quad (23)$$

The last indicator function is used to reset the congestion window size to zero when connection i runs out of data and returns to its idle state.

Finally, the evolution of $S_i^{(N)}(t)$ can be described as in (26) where connection i will be in the SS state during timeslot t if there is either i) no packet left to transmit (so the connection resets) at the beginning of the timeslot or ii) the connection was active and in the SS state then a packet drop or marking leads to the CA state. Equation (26) assumes that a new TCP connection, in the SS state, is set up one timeslot after the previous connection is torn down upon finishing its workload, and the new TCP connection becomes active when a new file/object arrives, initiating a

three-way handshake. Moreover, it implies that the state is updated in

$$W_{i,SS}^{(N)}(t+1) = G_{i,t+1}(W_{i,SS}^{(N)}(t), \min(\max(2W_i^{(N)}(t), 1), W_{\max})) \times \hat{Z}_i^{(N)}(t+1)(1 - E_i^{(N)}(t+1))\hat{M}_i^{(N)}(t+1) + E_i^{(N)}(t+1) * \Omega_i^{(N)}(t) + (1 - \hat{M}_i^{(N)}(t+1)) + \left\lfloor \frac{W_i^{(N)}(t)}{2} \right\rfloor \hat{M}_i^{(N)}(t+1)(1 - \hat{Z}_i^{(N)}(t+1)) \times (1 - E_i^{(N)}(t+1)) \quad (24)$$

the next timeslot following a transmission despite the window size being updated one RTT after transmission using the appropriate SS/CA state as in the correct operation of TCP.

$$W_{i,CA}^{(N)}(t+1) = G_{i,t+1}(W_{i,CA}^{(N)}(t), \min(W_i^{(N)}(t) + 1, W_{\max})) \times \hat{Z}_i^{(N)}(t+1)(1 - E_i^{(N)}(t+1))\hat{M}_i^{(N)}(t+1) + E_i^{(N)}(t+1) * \Omega_i^{(N)}(t) + (1 - \hat{M}_i^{(N)}(t+1)) + \left\lfloor \frac{W_i^{(N)}(t)}{2} \right\rfloor (1 - \hat{Z}_i^{(N)}(t+1))(1 - E_i^{(N)}(t+1)) \times \hat{M}_i^{(N)}(t+1) \quad (25)$$

$$S_i^{(N)}(t+1) = 1[X_i^{(N)}(t) \leq W_i^{(N)}(t)] + 1[X_i^{(N)}(t) > W_i^{(N)}(t)] \times S_i^{(N)}(t) \hat{Z}_i^{(N)}(t+1) \quad (26)$$

(Initial condition 1) For each $N = 1, 2, \dots$ and $i = 1, \dots, N$, the initial conditions of random variables in the model are:

$$Q^{(N)}(0) = \hat{Q}^{(N)}(0) = W_i^{(N)}(0) = E_i^{(N)}(0) = \beta_i^{(N)}(0) = 0;$$

$$D_i^{(N)}(0) = 2$$

and

$$z_i^{(N)}(0) = \hat{Z}_i^{(N)}(0) = Z_i^{(N)}(0) = S_i^{(N)}(0) = \Omega_i^{(N)}(0) = m_i^{(N)}(0) = \hat{M}_i^{(N)}(0) = M_i^{(N)}(0) = 1. \quad (27)$$

We denote the vector of state variables for connection i in timeslot t by

$$Y_i^{(N)}(t) := (W_i^{(N)}(t), X_i^{(N)}(t), E_i^{(N)}(t), S_i^{(N)}(t), D_i^{(N)}(t), \beta_i^{(N)}(t), \Omega_i^{(N)}(t), m_i^{(N)}(t), \hat{M}_i^{(N)}(t), M_i^{(N)}(t), z_i^{(N)}(t), \hat{Z}_i^{(N)}(t), Z_i^{(N)}(t)). \quad (28)$$

(Initial condition 2) All TCP connections are responsive to congestion control and the number of connections N is greater than the value of Th_{min} .

Theorem 1. Assume that both Initial conditions (1 and 2) are valid. Let us use the notation: \Rightarrow_N to denote a weak convergence in N to a steady state. Then:

$$\hat{Q}^{(N)}(t) \Rightarrow_N N. \quad (29)$$

The proof of Theorem 1 [9] is omitted here due to space reasons. Intuitively, since RWM complements RED, it inherits RED's property of choosing a particular flow, to mark or drop during congestion, which roughly is proportional to that flow's share of the bandwidth at the router. However, the ideal packet distribution in a RWM queue is one per flow for N TCP flows in steady state (which is indicated by simulations as t grows), especially for large N such that $N > Th_{min}$ and $N < Th_{max}$. Hence, RWM's potential is limited by the RED algorithm.

IV. SIMULATION RESULTS

We have created an NS2 [10] model of the RED-RWM scheme as described in [6]. (The NS2 TCP-Reno model had to be corrected as it does not follow a common linux-type TCP implementation with the interaction of restricted contention and flow control windows). We used a simple barbell topology with a single congested link of 10 Mbps and varied the number of connections N in $\{10, 20, 30\}$ with each link at 5Mbps. All connections were modeled as "File Transfer Protocol (FTP)" flows, i.e., as constant and inexhaustible source of information. We used the following RED parameters: Queue weight = 0.002, $Th_{min} = N-1$, $Th_{max} = 3N$, Max drop probability = 1/3 and queue size = 4N.

To verify our formulation, we have also created a custom Monte Carlo simulation implementing the analytic equations. We have performed simulation with parameters similar to those of the discrete event simulation. The average queue sizes were observed with both methods and are depicted in Fig 1; curves with a DE suffix are results of the discrete event simulations (NS2) while curves with an MC suffice show the Monte Carlo results (analytic). (Note that the time frame of these two simulations is not identical but plotting them on the same figure, i.e., their variation may not be on the same time scale). We can observe a good match, validating our mathematical model (note, that more extensive simulations have been performed with changes in other parameters, topologies and those results underline this conclusion as well).

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a mathematical model for the RED-RWM queue by extending the work done previously for RED-ECN queues. We have verified our model by comparing Monte Carlo simulations of our model

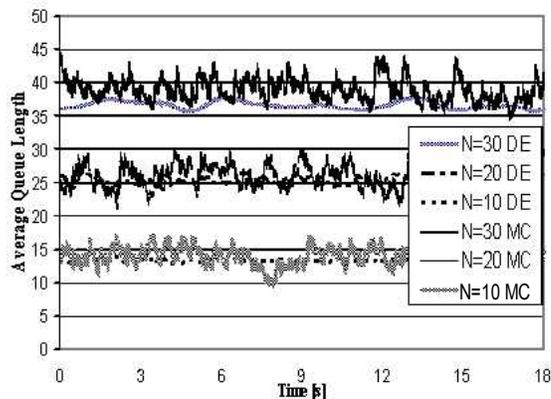


Figure 1. Average queue length at the RED-RWM queue.

to discrete event simulations of an NS2 model. We "theoretized" that the RWM modified RED queue will weakly converge to a steady state and that as the number of clients (TCP connections) grows, the queue size will weakly converge to the number of clients, assuming that the minimum marking threshold (Th_{min}) is less than the number of clients.

Our current work is focused on proving lower and upper limits on the instantaneous and average queue sizes with respect to various serving rates and marking threshold values (Th_{min} , and Th_{max}) in complex network topologies.

REFERENCES

- [1] A. Veres and M. Boda, "The Chaotic Nature of TCP Congestion Control," *Proceedings of the IEEE INFOCOM 2000, Tel Aviv, Israel*, pp. 1715-1723, March 2000.
- [2] G. Hasegawa and M. Murata, "Analysis of dynamic behaviors of many TCP connections sharing Tail-Drop/RED routers," *Proceedings of IEEE GLOBECOM*, November 2001.
- [3] C. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," *Proceedings of IEEE INFOCOM*, 2001.
- [4] P. Tinnakornsrisuphap, "Dynamics of Random Early Detection Under Large Number of TCP Flows," *Ph.D. dissertation University of Maryland, College Park* (professors: A.M. Makowski, J.S. Baras, R.J. La, P. Narayan, A.U. Shankar), 2004.
- [5] M. May, T. Bonald and J. Bolot, "Analytic Evaluation of RED Performance," *Proceedings of INFOCOM 2000*, pp. 1415 - 1424, vol. 3, March, 2000.
- [6] V. V. Govindaswamy, G. Záruba and G. Balasekaran, "Complementing Current Active Queue Management Schemes with Receiver-Window Modification (RWM)," *UTA Technical Report 2005-6*, April, 2005.
- [7] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, pp. 397-413, vol. 1, no. 4, August, 1993.
- [8] TBIT, the TCP Behavior Inference Tool. <http://www.icir.org/tbit/ecn-tbit.html>.
- [9] V. V. Govindaswamy, G. Záruba and G. Balasekaran, "Modeling and Analysis of Receiver-Window Modified Random Early Detection (RED-RWM) Queues," *UTA Technical Report 2006-7*, February, 2006.
- [10] NS simulator <http://www.isi.edu/nsnam/ns/>