A Comparison of RED's Byte and Packet Modes*

Wesley M. Eddy
Ohio University
weddy@irg.cs.ohiou.edu

Mark Allman
BBN/NASA GRC
mallman@bbn.com

April 28, 2003

Abstract

Routers making use of Random Early Detection (RED) queueing take action to notify sources of growing congestion levels in the network before their resources are exhausted. The RED system hinges on two calculations: tracking the average queue size and the probability that an incoming packet is marked for congestion. These two calculations can be done in terms of the number of packets arriving at the router or in terms of the size of those packets (in bytes). Intuitively, these calculation methods offer different costs and benefits to traffic. This paper quantitatively assesses the impact of using the different queueing and marking methods on the performance of traffic traversing a RED gateway. We show that in some cases the calculation method makes a difference in the performance of the system, while in other cases the choice has little impact. We also provide a framework for rating the RED variants in particular situations in an attempt to aid in the choice of variant to use in a specific situations.

1 Introduction

Queueing schemes for Internet routers have received much attention over the last several years for a number of reasons, including: the identified shortcomings with traditional drop-tail queues [FJ93], the desire for greater fairness [DKS89] and the desire for quality of service for different types of traffic [FJ95]. Regardless of the queueing strategy used, network bandwidth and buffer space are finite and limited resources of which improper management causes suboptimal operation. Traditionally, queueing has been done in a "drop-tail" fashion. Queues are passively filled as network congestion levels increase, only dropping packets when buffer space is exhausted. In contrast, Active Queue Management (AQM) schemes (e.g., Random Early Detection (RED) [FJ93]) actively manage the queue by "notifying" sources of growing congestion levels before the queue is full. There are numerous benefits to using AQM algorithms in the presence of congestion responsive flows, which are described in both [FJ93] and [BCC+98], and include avoiding full queues, increased fairness between competing flows, avoiding global synchronization and reducing queueing delay.

In this paper we focus on the RED AQM strategy. To attempt to control incipit congestion a RED router must somehow determine the amount of congestion currently occurring in order to apply a suitable amount of backpressure on the data senders. As noted above, RED keeps an average queue length for the purposes of gauging the congestion state of a particular link. Queue length can be measured in two ways: using the number of packets awaiting service in the queue, or the number of bytes sitting in the queue waiting to be forwarded. The choice of metric has implications on the traffic shaping applied by RED.

If a router stores packets in fixed buffers regardless of packet size a 50 byte packet and a 1500 byte packet take the same amount of internal router resources (modulo the packet serialization time – which will be longer for the larger packet). On the other hand, if packets are stored in a single large memory buffer in the router and take only the amount of memory they need, then the 1500 byte packet takes 30 times more queue space than the smaller packet. Additionally, the delay through a router is dictated by the size of the packets in the queue (i.e., the number of bytes that must be serialized). There are tradeoffs to measuring the queue in terms of bytes or packets. Some of these tradeoffs involve specific router architecture (e.g., memory allocation issues), while others are more generic. In this paper we take a very generic approach to examining the biases introduced by the choice of metric (and stay away from specifics of any given architecture).

Once the RED algorithm has determined that a queue is becoming congested it must inform data sources of this incipit congestion so that they can reduce their sending rates. Rather than slowing all sources (and possibly causing oscillations and synchronization effects), RED probabilistically informs sources (with the idea being that sources sending at higher rates will have a higher chance of being asked to slow down). When RED determines

* This paper appears in Computer Networks, volume 42, issue 2, June 2003.

that the router is in a state of growing congestion each incoming packet is *marked*[1] with some probability. As with the metric for queue length, the probability of marking a packet can be based simply on the packet arrival itself or on the size of the packet arriving. Again, tradeoffs in the approach taken abound.

The following is a brief overview of the specifics of the RED queueing scheme. Readers are encouraged to review [FJ93] for more details.

- When the average queue length is less than a lower threshold, min_{th}, no packet marking occurs.
- When the average queue size is above min_{th} and below an upper threshold, max_{th}, packets are probabilistically marked. The mark rate enforced increases monotonically from zero when the average queue size is min_{th} to max_p when the average queue size is max_{th}.
- Finally, the marking rate increases linearly from max_p to 1.0 (marking all packets)[2] as the average queue length increases from max_{th} to $2 \times max_{th}$.
- When using byte-based marking RED must normalize each incoming packet to determine its chances for being marked. For this purpose, RED uses a *mean packet size* (MPS) parameter, which is static value that is intended to represent some "typical" packet size on the link. If the incoming packet is larger than the MPS the marking probability is greater, whereas if the incoming packet size is less than the MPS the marking probability is reduced. This MPS parameter is also used to convert the queue length measurement from a number of bytes to an estimated number of packets.

The pros and cons of measuring and marking in terms of packets or bytes in RED queues have received some thought within the research community [Flo97a]. [DEP00] offers a limited set of simulations comparing packet and byte marking RED variants. We expand on [DEP00] by using a wider variety of packet size mixes and traffic scenarios, as well as providing an exploration of RED's mean packet size setting and the fairness implications of the various RED modes. We are aware of no additional research to date that attempts to experimentally quantify the differences between and biases caused by using byte or packet modes in RED queues. This paper focuses on quantifying the biases offered by RED and confirming the community's intuition. Specifically out-of-scope for this paper is comparing RED byte/packet variants with alternate queueing strategies (e.g., drop-tail, BLUE [cFKSS99], REM [ALLY01], FRED [LM97], etc.). We do, however, believe that investigating biases in other queueing disciplines would be useful future work.

The remainder of this paper is organized as follows. § 2 outlines the simulation environment we use to test the performance of the RED variants. § 3 describes the results of our simulations using homogenous packet sizes, as a comparison against the results shown in § 4 which covers our experience with traffic consisting of several mixes of packet sizes. § 5 describes the results of further simulations that make use of realistic WWW traffic. § 6 explores the performance as RED's *mean packet size* setting varies. § 7 provides a system for rating RED variants and a discussion of our simulation results in the context of this rating system. Finally, § 8 summarizes our conclusions and outlines future work in this area.

2 Experimental Framework

2.1 Simulation Environment

The simulations presented in this paper were performed with the *ns-2* simulator (version 2.1b8)[3]. To evaluate RED performance, we created a network with a single bottleneck and observed how different RED calculation modes manage traffic through that bottleneck. Figure 1 illustrates our topology. The bottleneck consists of a 1.5 Mbps link with a one-way delay of 70 ms connecting two routers. Connected to each router (via Ethernet-like links) are 5 hosts that source and sink traffic.

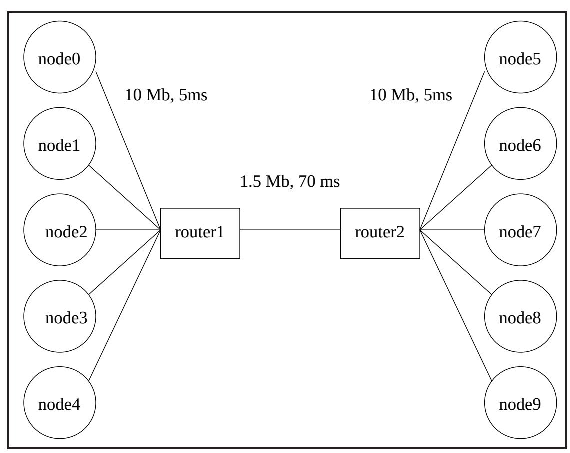


Figure 1: Simulated network topology.

To create traffic through the bottleneck, we setup an FTP sender and receiver on each of the ten hosts, so five bulk transfers compete for access to the bottleneck

[1] RED can mark packets by either dropping the packet or using an explicit signal (e.g., [RFB01]).

[2] This assumes the *gentle* variant of RED [Flo00], which we used in our simulations.

[3] http://www.isi.edu/nsnam/

in each direction. The underlying TCP connections all use selective acknowledgments (SACK) [MMFR96], delayed acknowledgments [Bra89], and a maximum advertised window set to 500 packets (simulating automatic socket buffer tuning [SMM98] since the congestion window never reaches the advertised window). The packet size mix used by each of the ten TCP senders varies (as specified in subsequent sections) to allow observations of the interaction between flows of differing packet sizes.

We hold all simulation parameters related to the network except the unit for the queueing and marking calculation and the MPS setting static in all simulations presented in this paper. We do this to focus on assessing the biases of various RED modes. Future work in varying RED parameters (e.g., min_{th} and max_{th}), bottleneck bandwidth, using multiple congested points, etc. would likely be useful. However, this work is out-of-scope for this initial study into RED's biases.

The RED queues in each of the two routers share the same configuration in all cases presented in this paper. We use the following RED parameters (on the advice of [Flo97b], which is reasonably conservative for the purposes of our study) in all our simulations: a min_{th} of 5 packets, a max_{th} of 15 packets, a queue weight (w_q) of 0.002 and a max_p of 0.10, with gentle mode [Flo00] enabled. The two aspects of the RED queue that vary in our simulations are whether the buffering is done in terms of bytes or packets and whether the drop probability is calculated in terms of bytes or packets. Our simulations do not use explicit congestion notification (ECN) [Flo94, RFB01] for marking packets, but instead drop packets to signal incipit congestion, because (i) the number of ECN aware routers in the Internet is currently small and (ii) the marking method does not greatly effect the results[4]. The length of the router buffer is set to 70 packets, and the mean packet size parameter is set to the *ns* default of 500 bytes (unless otherwise noted). When the queue is measured in terms of bytes the maximum length of the router queue is set to $70 \times MPS$ bytes (or 35,000 bytes when using the default MPS).

All simulations run for five minutes, with the ten FTP transfers starting at random times during the first 30 seconds of the simulation to minimize any synchronization effects that may be caused by starting all the flows simultaneously. We then eliminate all data pertaining to the first and last 30 seconds of the simulations, examining only the times that all 10 flows are actively competing for the scarce bottleneck resources (and, when the flows are in steady-state). All simulations with a given set of parameters (packet sizes and RED modes) are run 30 times, and all of the data we report is in the form of mean values over those 30 trials[5]. To ensure differences among the 30 trials, we seed the simulator's master random number generator with the current time at each invocation.

[4] We ran the baseline simulations presented in § 3 using ECN and the results are generally within 2% of the results obtained when dropping packets. Further, both the ECN and non-ECN simulations show the same trends.

2.2 Notation

In this paper we use several abbreviations to describe the calculation mode under consideration. The abbreviation "pq" denotes packet-based option for calculation of queue length, while "bq" denotes byte-based option for calculating the queue length. Similarly, "pm" represents the packet-based marking option while "bm" denotes the byte-based marking option. Combinations of options are then indicated by joining their two components together, as in "pq_bm" for a queue that measures its length in packets and marks packets based on their size in bytes.

2.3 Metrics

We use two principle metrics to compare the performance of the RED variants in this paper. The first metric is the utilization of the bottleneck link – indicating how well the RED variant is managing the bandwidth. The utilization is defined as all bytes in all packets that cross the bottleneck link (including data packets, acknowledgments, header bytes and data bytes). In addition to utilization we gauge the fairness of the various queueing strategies studied in this paper using Jain's fairness index [Jai91]. The fairness index is computed as:

$$f(x_1, x_2, \cdots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2} \qquad (1)$$

where x_i represents the number of total bytes received by a particular host in our simulation and n is the total number of hosts. A fairness index of 1 indicates that each host transmits the exact same number of bytes. There are alternate areas of "fairness" that our analysis does not cover. For instance, a flow using a large number of small packets may be seen as "unfair" in terms of router CPU cycles consumed when compared to a a flow that uses a small number of large packets (but sends the same amount of actual data). Our focus is on-the-wire fairness, leaving additional evaluations of fairness as future work.

Several additional metrics could be used to assess a queueing strategy's efficacy. For instance, the drop rate

[5] We report only means in this paper. However, we calculated the average standard deviation across the 30 runs for each scenario presented in this paper and found the mean to be 1.8%, the median to be 1.5% and the 95^{th} percentile to be 3.2%. Therefore, we conclude that the means presented in the paper are accurate characterizations of the behavior of the particular RED queue.

may have a direct influence on a user's perception of an audio transmission. In addition, the queueing delay can have an impact on interactive (ssh, telnet) traffic. While useful to measure we do not focus on these metrics (although this paper does include some of these results) because we believe that the RED min_{th}, max_{th} and max_p parameters are better suited for tuning RED to produce the desired queueing delay and loss rate. We do note that these metrics are effected to a small degree by the units used in the RED calculations and will present results that illustrate this later in the paper.

3 Baseline Simulations

First we present baseline simulations with all flows using the same packet size, to discover any inherent properties of the various calculation modes that appear for homogenous traffic and might confuse the results of simulations with mixed packet sizes. We use packet sizes of 296, 576, 1500, 4352, and 16000 bytes for these baseline simulations. Some of these sizes represent common MTUs of Internet links (e.g., 1500 bytes for Ethernet), while others allow for the examination of the parameter space (e.g., 16000 bytes). We use the simulation setup, parameters and topology described in § 2.1 for these simulations.

Pkt. Size	**bq_bm**	**bq_pm**	**pq_bm**	**pq_pm**
296	87.6%	83.8%	76.2%	72.2%
576	81.6%	81.6%	80.4%	80.7%
1500	76.0%	80.9%	85.1%	89.1%
4352	71.8%	79.6%	80.7%	89.6%
16000	62.2%	61.1%	59.8%	80.0%

Table 1: Percentage of available bandwidth used by various configurations.

Table 1 shows the average utilization (over 30 simulations) attained by all sources at one end of our network (including both data and ACK packets). The table shows that for the smallest two packet sizes the byte-based queueing and byte-based marking modes give better utilization than the packet modes. However, for the three larger packet sizes the reverse is true. This is explained by the mean packet size setting of 500 bytes, which affects all the byte-based calculations and is less than the three larger packet sizes. Therefore, the queue is considered to be experiencing congestion sooner when measured in terms of bytes than when measured in terms of packets, causing more packets to be marked. When byte-based marking is used, the probability of a packet being marked is multiplied by factors of roughly 3, 8.7, and 32 for packet sizes of 1500, 4352, and 16000 bytes respectively when compared to the marking probability that would be experienced in the packet-based marking mode.

An additional note about table 1 is that when using 576 byte packets, there is little observable difference in the queue's behavior across calculation modes. This is due to the proximity of this packet size to our mean packet size setting of 500 bytes. The results suggest that when a queue in bq_bm mode has a mean packet size set close to that of the actual traffic on the network, it will behave roughly the same as a queue using pq_pm, yet can be expected to deal with incoming packets during periods of link congestion more properly (i.e., based on the packet's true size). In § 6 we further investigate RED's sensitivity to the mean packet size setting.

We also note that when using 1500 byte packets the range in utilization between the RED variants is roughly 13%, showing the potential bandwidth cost of using a suboptimal RED variant.

Next we calculated the fairness index for each of our baseline scenarios. The fairness index for all but one scenario is over 0.99, indicating a high degree of fairness when considering a network with homogeneous packet sizes. The only case in our baseline simulations where the fairness index is less than 0.99 is a pq_bm queue handling 16 kB packets.

In the pq_bm case the average fairness index is 0.9065 across our 30 simulations. The reduced fairness observed in the pq_bm queue is due to the large size of the packets causing an interaction between the packet-based queue length calculation and byte-based marking. The queue size is measured in packets, thus staying relatively low compared to what its length would be if measured in bytes. The problem is that the bottleneck, in this case, is the bandwidth of the link, which is in terms of bytes (per second) and so the queue ends up filling itself to higher capacity when using a packet-based length calculation than when using a byte-based version. This is because a single packet in the byte-based queue counts as 32 (given our mean packet size setting of 500 bytes), thus increasing the average queue-length quickly, and causing the probability of packets being dropped to rise. Meanwhile, a packet-based queue waits longer before reaching comparable drop probabilities. As the queue slowly empties, many drops occur, and then the queue begins to fill again. This cycle can result in race-conditions between the flows, which must fight for the available queue space in small windows of time before the drop probability becomes high again. The pq_pm scenario doesn't experience the rapid cycling of the queue size because its probability of marking packets remains low for much longer, thus causing the queue length to change more gradually.

Since all RED calculation modes are roughly the same in terms of fairness when dealing with homogenous traffic we can assume that an increase in the range of fairness indices when using traffic with mixed packet sizes indicates biases in particular RED variants.

For comparison, our baseline tests were also conducted in a network using traditional drop-tail queueing. The drop-tail queues increased utilization in most cases, averaging roughly 91% utilization in the 1500 byte packet tests (compared to the RED queues, which achieve 89% utilization in the best case). In addition, the drop-tail queue is roughly as fair as RED, with a fairness index over 0.99. The major difference between the RED and drop-tail queues in our baseline tests is the percentage of bytes dropped. In the case of the 1500 byte packet size simulation we observed an average of 3.7% of the bytes being dropped by drop-tail routers, as compared to RED which never dropped more than 2% of the bytes. As shown in [FJ93], this result is due to RED's ability to absorb traffic bursts better than drop-tail queues.

4 Mixed Packet Sizes

To gauge the fairness and performance of RED queues using different calculation modes in a network with mixed packet sizes we simulated three different situations using the setup outlined in § 2.1. The first, dubbed the *extreme* case, consists of five flows using five drastically different packet sizes going in each direction through our simulated network. This scenario proves valuable because by exaggerating the differences in packet sizes, we also exaggerate any biases that a certain calculation mode has on the traffic, thus making the biases easier to observe. We then outline two more sets of simulations using mixed packet sizes that are less varied than the extreme case. These are useful for confirming biases observed in the extreme case do appear in situations that are more realistic (in terms of packet sizes).

4.1 Extreme Case

In the simulations presented in this section we use the same set of the packet sizes used in our baseline simulations (296, 576, 1500, 4352, and 16000 bytes), providing a wide range of packet sizes. Also, the three sizes in the middle of the set represent common MTUs of different Internet links. One FTP flow in each direction uses each packet size.

Table 2 shows the mean results from 30 simulations using the extreme packet size mix. The reported values are measured by the five nodes on the right-hand side of our topology, including both data coming to the node's sink and ACKs coming to the node's sender. The value in the upper left corner of each box in the table represents the aggregate utilization of the bottleneck link. The number reported in the lower right corner of each box is the fairness attained by the given RED variant.

	pq	**bq**
pm	84.62% 0.4124	68.27% 0.5492
bm	59.83% 0.8153	57.49% 0.9346

Table 2: Overall utilization (upper left) and fairness index (lower right) for extreme simulations.

We first note that using packet marking mode provides better aggregate bandwidth utilization than byte marking mode. The table indicates that the increased utilization comes at the expense of fairness. The reason for this unfairness is that the flows consisting of 16 kB packets use a disproportionate percentage of the bandwidth and starve the flows using smaller packet sizes when using packet marking. This effect is especially evident in the pq_pm variant where the flow using 296 byte packets obtains roughly 1.5% of the bandwidth while the flow using 16,000 byte packets uses over 55% of the bandwidth. Furthermore, the pq_pm queue shows higher aggregate utilization than the other three variants. In this case the large packets aid utilization but do not increase the queue occupancy proportionally, essentially giving more bang for the buck.

Table 2 also shows that using byte modes over packet modes (for both queue length and marking probability calculations) increases fairness. Further, byte mode has a greater effect when used in the marking calculation, when compared to using byte mode for the queue length calculation. In our simulations using byte marking provides an increase of nearly 0.4 in fairness index over packet marking in a queue measured in bytes, and a nearly twofold increase over a queue measured in packets. While in packet marking queues use of byte-based queue calculation increases fairness by approximately 0.14, and in byte marking queues fairness increases by roughly 0.12.

When selecting the mode to use for RED calculations a tradeoff between fairness and utilization must be made, as Figure 2 illustrates. The figure shows that utilization and fairness are inversely related across the RED modes. This figure also shows that selection of marking mode has a greater impact on the results than selection of queue-measurement mode.

The results presented in this section show the biases RED's different calculation modes introduce. The utilization shown by the different modes has a range of more than a quarter of the bandwidth of the link, while the fairness index has a range of more than 0.5. Given the mix of packet sizes in this scenario the results are not indicative of what would be observed in a real network. However, the extreme simulations quantitatively show RED's biases do exist.

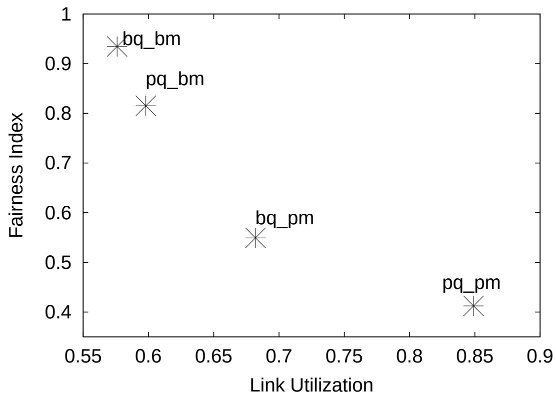


Figure 2: Utilization versus fairness relationship.

4.2 Mild Cases

We now focus on two scenarios with less radical (and more realistic) packet size mixes than the extreme case discussed above. The first case, denoted M_s, uses several small packet sizes: 350, 500, 750, 800, and 1000 bytes. The second scenario, denoted M_e, uses three flows of 1500 byte packets (the Ethernet MTU – and a common maximum packet size for Internet WWW flows [All00]), one flow using 1250 byte packets and one flow using 1750 byte packets. These packet size distributions are not meant to mirror those observed in real networks, but rather to present two mixes that are closer to reality than the extreme mix. We again use the simulation scenario outlined in § 2.1.

	pq	**bq**
pm	83.11% 0.9156	82.44% 0.9230
bm	80.88% 0.9759	81.33% 0.9787

Table 3: Average utilization and fairness in the M_s simulation.

Table 3 shows the results from our simulations of the M_s scenario. The table agrees with the main results from the last subsection (table 2) regarding both the inverse relationship between fairness and utilization and the implications of greater utilization using packet modes and greater fairness from byte modes. By using a mix of packet sizes with a small range, we note better fairness from all RED modes than observed in the extreme case. Also, note that all RED variants except pq_pm show improvements in aggregate bandwidth utilization (by upwards of 20%) when using the M_s mix of packets compared to the extreme packet mix. The pq_pm variant differs by less than 2% between the M_s and extreme simulations. In addition, the results of the M_s set of simulations show more uniformity across RED variants than observed in the extreme case. For instance, the difference between the best utilization and the worst is just over 2%. Meanwhile the range of the fairness index deviates by roughly 0.06 between the best case and the worst case. These results are explained by the tighter distribution of packet sizes employed for this set of simulations.

The major reason why the selection of byte and packet modes does not seem to matter as much in the M_s scenario as in the extreme scenario is the packet sizes do not vary as much in the M_s case. In addition, the packet sizes are close to RED's fixed mean packet size of 500 bytes, causing the packet modes to be more fair while allowing the byte modes to better utilize the bottleneck link.

Table 3 shows that the bq_bm and pq_bm mode combinations achieve comparable utilization to each other, as do the bq_pm and pq_pm modes. This again indicates that the selection of marking mode has greater impact than the selection of queue measurement strategy. In addition, when the mean packet size setting is properly tuned the byte modes can obtain utilization near that of packet modes, thus reducing the degree of trade-off between utilization and fairness. Setting the mean packet size parameter is considered in more detail in § 6.

In addition, the setting of the mean packet size parameter influences how the queues using byte marking behave. RED with byte marking consistently marks packets that are bigger than the mean packet size (MPS) with a higher frequency than those packets that are smaller than the MPS. In particular, consider the packet measured queues in the M_s simulation. The two flows using packet sizes at or below the MPS have 28% and 11% *less* bytes marked respectively by the byte marking queue than when packet marking is used. In addition, the three flows above the MPS have 10%, 7%, and 15% *more* bytes marked respectively when calculating the marking probability in terms of bytes.

	pq	**bq**
pm	88.88% 0.9950	80.88% 0.9940
bm	85.11% 0.9991	75.77% 0.9998

Table 4: Average utilization and fairness in the M_e simulation.

Table 4 summarizes the results of the M_e simulations. These results show roughly the same trends we noted in the extreme and M_s simulations. The inverse relationship between utilization and fairness is not as noticeable because all of the queues achieved high fairness indices (above 0.99) due to the tight distribution of packet sizes.

We note, however, that the packet modes once again performed better in terms of utilization than the byte modes. We also again note that the range in utilization is roughly 13%, indicating the importance of choosing the right calculation modes.

4.3 Summary

We draw the following conclusions from the bulk transfer experiments presented in this section:

- The extreme set of simulations confirm that biases are, in fact, present in various RED variants.
- The choice of marking mode has a larger impact on performance than the choice of queueing mode.
- The choice of RED mode changes aggregate bottleneck utilization by varying amounts depending on the mix of packet sizes (e.g., by roughly 3% in the M_s experiments and by roughly 13% in the M_e simulations).
- The choice of RED variant can also have fairness implications (ranging from nearly no difference in the M_e simulations to a range of 0.06 in the fairness index in the M_s experiments).

5 Realistic WWW Traffic

While the simulations presented in the last two sections are useful as baselines, the traffic pattern is not realistic. Studies of wide-area traffic show that the majority of Internet traffic consists of HTTP transfers for World Wide Web (WWW) resources [TMW97, MC00]. Generally, WWW transfers consist of transfers of small amounts of data. The goal of the simulations presented in this section is to determine whether our conclusions from the previous sections hold when a WWW traffic pattern is employed[6].

To simulate RED behavior with a WWW traffic mix, we use the topology and RED setup given in § 2.1, except we replace the FTP senders on each host with HTTP servers and use multiple HTTP clients on each host node instead of the single FTP client. The HTTP clients and servers are standard *ns* HTTP entities that are configured with a page pool whose average page size is 1024 bytes. The clients request resources from the servers across the bottleneck at random exponentially distributed intervals throughout the simulation (from a distribution with a mean of 0.01 seconds). The TCP model used in these simulations is *ns*' FullTcp which allows for bi-directional data flow. The maximum packet size for two of the nodes on each side of the network is set to 576 bytes. Another node on each side of the network uses a maximum packet size of 4352 bytes (to simulate FDDI) and the remaining two nodes have maximum packet sizes of 1500 bytes (ala Ethernet). We note that we set the *maximum* packet size, but in HTTP client requests generally do not require a full-sized packet and so there are variable size packets transmitted in these simulations. Three different traffic loads are simulated by varying the number of HTTP clients per host. We use 10 clients on each node for *light* traffic, 20 clients for a *moderate* traffic load, and 40 clients for a *heavy* load. In all cases, the simulations run for five minutes and the reported values represent the average of 30 random simulation runs.

[6] We do not claim this scenario is completely *realistic*, since a realistic traffic mix would include long bulk transfers, rate-based UDP traffic and many other applications. We are using the WWW traffic mix as a second data point in studying the biases present in the RED variants under study.

	pq	**bq**
pm	68.44% 0.9952	66.00% 0.9853
bm	69.11% 0.9777	68.88% 0.9893

Table 5: Utilization and fairness for light WWW load simulations.

5.1 Light Load

Table 5 shows the bandwidth utilization and fairness results for the simulations involving a light traffic load (10 clients per host). All RED modes performed similarly in terms of both aggregate utilization and fairness in these simulations. Since this simulation places a light load on the network the number and percentage of packet drops is small (roughly 0.05%). Therefore, these simulations do not offer insight into the biases of the various RED mechanisms, but rather show that in lightly loaded networks the choice of RED variant does not have a significant impact on utilization or fairness.

5.2 Moderate Load

Table 6 summarizes the results of simulations involving a moderate load (20 clients per host). The results show more divergence in the performance of the various RED modes than noted in the light load simulations. The results show that using packet mode for marking increases utilization, while byte-based marking variants show increased fairness. Aside from the pq_pm combination, all modes once again perform similarly in terms of fairness.

We also note that the packet marking variants are marking small packets more often than large packets. The implication of marking small packets is that a larger number

	pq		bq	
pm	79.1%	0.9349	83.8%	0.9826
bm	74.7%	0.9886	80.9%	0.9873

Table 6: Utilization and fairness for moderate WWW load simulations.

need to be marked before congestion is reduced. In our simulations we note that the packet marking variants drop 2–3 times as many packets as the byte marking variants. This situation may have detrimental effects on both real and user-perceived performance of network applications and lead to unfair bandwidth sharing between flows. This effect can be seen in table 6 which shows packet marking queues to be less fair than their byte marking counterparts.

We also find that measuring the queue in terms of bytes reduces the percentage of bytes marked at the expense of utilization. This indicates that less bytes are being carried through the bottleneck link, but we also note that a greater than proportional number of bytes are being marked. The results, therefore, imply that the average congestion level of the bottleneck is lower when measuring the queue in terms of bytes. The plots in figure 3 show the average queue length kept by RED and the maximum instantaneous queue length recorded for each second of the simulation[7]. The y-axis is in terms of the mean packet size (500 bytes) for the bq mode. As shown in the plots, both the average queue length and the instantaneous queue length are generally lower when using byte-based queue measurement. This is a direct result of the HTTP traffic pattern in which request packets are typically much smaller than the responses and the mean packet size. Thus several of them can fit into a single mean packet size when the queue is measured in bytes but will count as several packets if the queue is measured in packets.

Another reason the queue length is lower when using byte-based queueing is that the unit of queue measurement matches the resource being managed. In other words, the size of the packet determines how quickly it will be transmitted onto the link[8]. When queueing in terms of packets the average queue size is not directly matched to the serialization time of the packets in the queue, hence the increased difficulty in managing the queue size and the variability in the length of the measured queue.

[7]The maximum instantaneous queue size for each second of the simulation is intended to show the variability in the actual queue size over time without overly cluttering the plot with every instantaneous queue size measurement taken by the RED gateway.

[8]As elsewhere, we ignore router processing, given that our simulator does not provide a means to model the cycles required on a per-packet basis.

5.3 Heavy Load

	pq		bq	
pm	78.9%	0.7720	82.7%	0.8289
bm	71.6%	0.9909	77.6%	0.9894

Table 7: Utilization and fairness for heavy WWW load simulations.

Table 7 summarizes the results of our WWW simulations using a heavy traffic load (40 clients per node). As in the previous set of WWW simulations, we observe that queues measured in bytes are able to utilize a greater percentage of the available bandwidth than those measured in packets. Additionally, queues marking in terms of packets show better utilization than queues marking in terms of bytes. These results are different from our previous bulk transfer simulations and therefore warrant further discussion.

Our previous simulations (see § 3) suggest that the performance of the byte modes depends on the mean packet size parameter. (This will be examined further in § 6.) If the MPS is accurate for the actual traffic traversing the link, then RED in byte marking mode should behave more appropriately when marking packets by concentrating on those packets that are larger than the MPS (i.e., the "heavy hitters") during periods of congestion. Also, the MPS affects how quickly a queue measured in bytes detects incipit congestion. Since the capacity of the link is measured in bytes (per some time unit) rather than packets a queue measured in bytes more accurately reflects both the imposed queueing delay and the congestion level of that link[9]. In the previous subsection we present figures that show that byte-based queue measurement provided an increased ability to absorb bursts of traffic when compared to packet based queue measurement.

Figure 4 illustrates the difference in both the instantaneous[10] and average queue lengths maintained by byte-based and packet-based queue measurement strategies. Observe that in figure 4(a) the instantaneous queue length measured in bytes never grows larger than $50 \times MPS$, while in figure 4(b) the queue length peaks at roughly $70 \times MPS$, which is the maximum queue length configured so packet dropping behavior is forced at that point. Also notice that the average queue length, as calculated by RED, tends to fluctuate near a slightly lower value in the bq_bm plot than it does in the pq_bm queue.

[9]The implicit assumption is that packet processing time is negligible. If a router is CPU limited then the router itself, rather than the link, becomes the bottleneck.

[10]Again, we report the maximum instantaneous queue length for each second of the simulation.

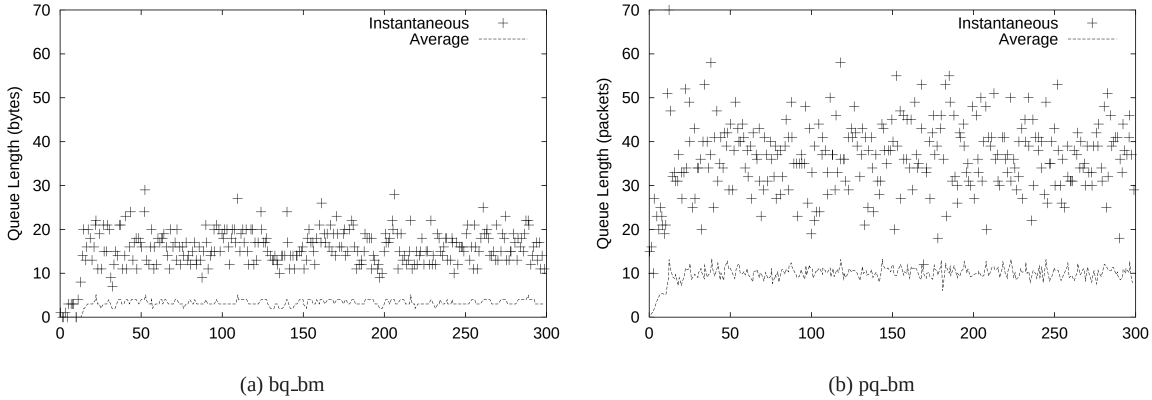


Figure 3: Queue lengths for byte-based marking variants in the moderate load web simulations.

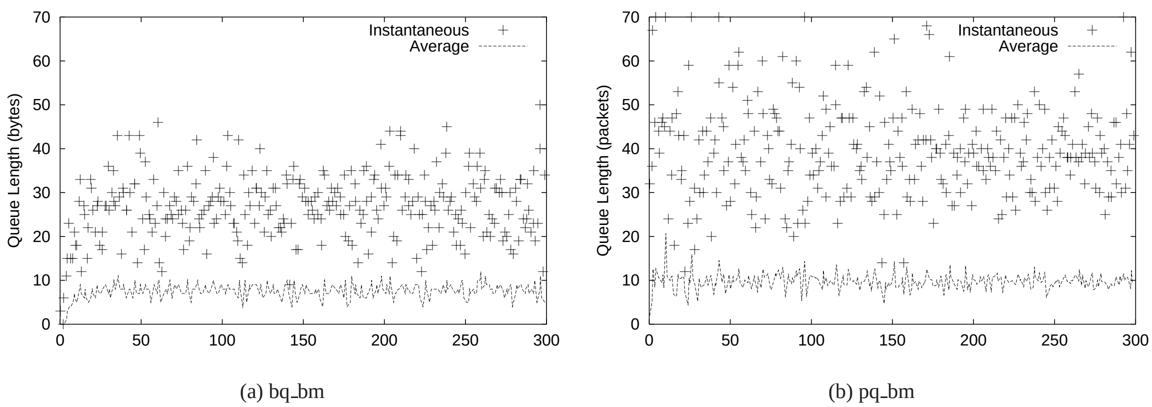


Figure 4: Queue lengths for byte-based marking variants in the heavy load web simulations.

Table 7 shows a difference in fairness between queues of different marking modes (of roughly 0.16–0.22), yet varying the units of measurement for the queue length has little impact on fairness. As in the previous simulations, byte mode for marking packets outperforms the packet-based variant. Byte-based packet marking continues to attain fairness indices of 0.98 and greater even under high levels of congestion. These simulations serve as a strong argument for using byte-based marking in environments where fairness among competing flows is a desirable property.

5.4 Summary

We draw the following conclusions from the WWW simulations presented in this section:

- The importance of choosing the correct RED variant is proportional to the level of congestion across the bottleneck link. Therefore, when the link is lightly loaded the choice has little impact. However, as the level of traffic increases we see non-negligible ranges in utilization (roughly 11%) and fairness (roughly 0.22) between RED modes.
- We also find that byte-based queue measurement offers smaller queue sizes than the packet-based approach when using byte-based marking.

6 The Mean Packet Size Parameter

While exploring the variants of RED queue measurement and marking behavior in the above sections, we held the mean packet size constant (at the *ns* default of 500 bytes). In this section, we explore RED behavior as a function of the MPS setting. Incoming packets that are smaller than the MPS are less likely to be marked until the average queue length reaches $2 \times max_{th}$ (since gentle mode is used in our simulations) when a queue uses byte-based marking. Similarly, enqueuing packets with size less than MPS will have less effect on the measured queue length when measuring in terms of bytes. This is problematic for routers whose processing power is limited in packets per second because it allows the congestion level to grow faster than the probability of signaling congestion. The opposite is true if incoming packets are larger than the MPS – when the likelihood of being dropped increases when the average queue length is above min_{th} in byte marking modes. In addition, the large packets have the ability to increase the measured congestion level more rapidly when using byte measuring modes.

In this section we first briefly present some real-world data showing the diversity of packet sizes on the Internet. We then present simulations to assess how sensitive RED performance is to the setting of the MPS parameter.

6.1 Real-World Packet Size Distributions

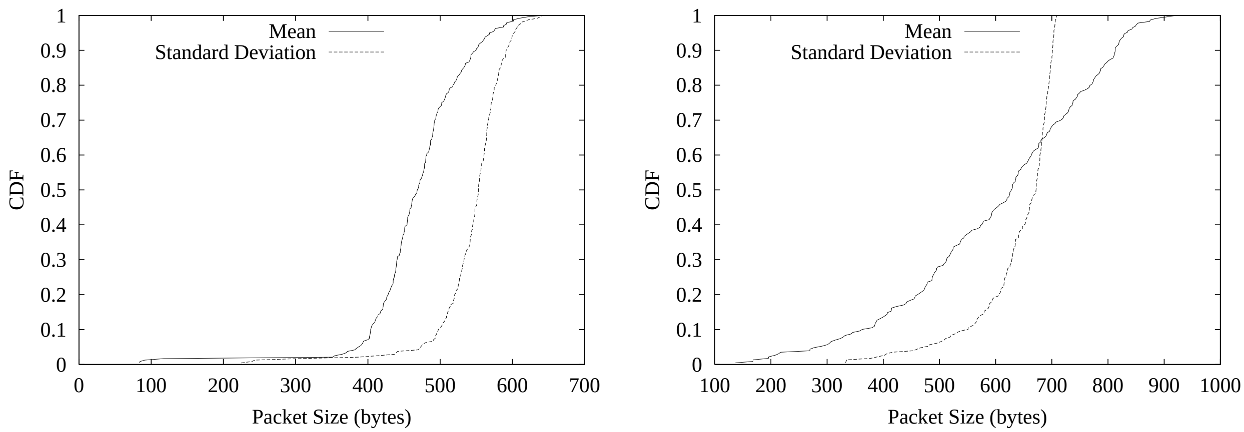
In this section we examine the packet sizes found in two sets of traces taken at different locations in the network. While an in-depth study of packet sizes in the Internet is beyond the scope of this paper we provide a simple analysis of two sets of traces to (i) provide context for the simulations presented in the remainder of this section and (ii) to attempt to gain an understanding of the difficulty involved in choosing a static *mean packet size* parameter. For our analysis we use two datasets from NLANR[11], as follows:

- **ADV dataset.** This set of traces was taken at the WAN link of Advanced Networks and Services during November 2001. The set consists of 8 traces for each day of the month. Each trace contains the headers of all packets observed in a 90 second interval. On average, each trace contains approximately 184,567 packets with a standard deviation of roughly 175,946 (suggesting a wide range of traffic patterns, likely caused by diurnal activity patterns).
- **BUF dataset.** This set of traces was taken at the WAN link of the University of Buffalo during January 2002. The set consists of 229 trace files. While the goal of the researchers at NLANR is to collect 8 traces per day, on 4 days in the BUF dataset not all of these attempts were successful. As in the ADV dataset, each trace contains all packets observed during a 90 second interval. On average, each trace consists of roughly 363,463 packets with a standard deviation of approximately 365,851 (showing a wide range of traffic patterns, as noted in the ADV dataset).

Figure 5 shows the distribution of packet sizes found in each dataset. As shown, the BUF dataset has greater fractions of larger packets than the ADV dataset. For instance, in the BUF dataset roughly 40% of the packets are approximately 1500 bytes. Meanwhile, in the ADV dataset less than 20% of the packets consist of 1500 bytes. This plot clearly shows that determining a globally applicable "typical" packet size for RED's mean packet size is likely difficult at best.

Next we analyze how the packet sizes change over time. Figure 6 shows the distribution of the mean packet size and standard deviation calculated for each 90 second trace for both datasets. The plot illustrates the heterogeneous nature of traffic (and therefore packet size). The mean packet sizes observed in the BUF dataset are more spread out than those in the ADV dataset. A wide range of packet sizes are present within each sample as the distribution of the standard deviation illustrates.

[11] The traces are available from `http://pma.nlanr.net/PMA/`.



(a) ADV dataset.

(b) BUF dataset.

Figure 6: Distribution of the mean packet size and standard deviation per 90 second trace.

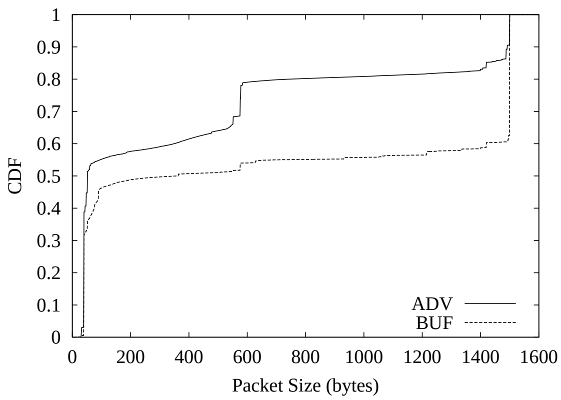


Figure 5: Distribution of packet sizes in the ADV and BUF datasets.

In the ADV dataset the mean packet size is between 400 and 600 bytes in just over 90% of the traces. This is an indication that in *some networks* and at *some times* determining the mean packet size parameter for RED to within a couple hundred bytes may be straightforward. On the other hand, the BUF dataset shows that the difference between the 5^{th} and 95^{th} percentiles of the mean packet size distribution is over 540 bytes. The BUF dataset suggests that at *some times* in *some networks*, determining the mean packet size parameter for RED may be difficult (at least for an MPS that would be valid over long timescales).

Note that the two datasets are for *different* months and so the difference in packet sizes could be caused by either the difference in the networks observed or the difference in the observation period. We do not delve into this to determine the exact cause, but rather use these two datasets to illustrate that determining some notion of a "typical packet size" is challenging (as is finding "typical" properties of network in general [PF01]).

The data we present in this section is illustrative and not conclusive. We show the heterogeneous nature of packet sizes in different networks at different times. In the context of the RED MPS parameter the packet size distributions we show indicate that either RED needs to be robust to gross settings for the MPS or that RED needs to be able to dynamically set the MPS parameter based on measurements taken in the network. Further, we note that a more in-depth study into the causes of different packet size distributions and changing packet size characteristics is an area that deserves additional effort from the research community (but, is tangential to the goals of this paper).

6.2 Bulk Data Transfer

To explore the influence the MPS setting has on the three modes that make use of the MPS parameter in their calculations we run several simulations. For these simulations we use the network traffic and topology described in § 2.1. We use two flows of 576 byte packets, two flows using 1500 byte packets and one flow using 4352 byte packets. We then run sets of 30 simulations with MPS settings in the routers between 250 and 2000 bytes with a step of 125 bytes as well as at 40 bytes and 4352 bytes. For comparison, table 8 provides data on queues that do not use the MPS parameter. In the drop-tail simulations the mean size of packets that arrive at the bottleneck is 985 bytes, with a median of 576 bytes. Similarly, in the simulations with a pq_pm queue we observed a mean packet size of 1038 bytes, and a median of 576 bytes. Also note that the utilization and fairness are similar between drop-tail queueing and the pq_pm variant of RED.

Table 9 contains the data from our simulations with an

Metric	drop-tail	pq_pm
Bandwidth Util.	91.6%	90.0%
Fairness Index	0.6757	0.6640
Kilobytes Marked	741	583
Packets Marked	789	462
KB/Packets	0.94	1.26
% bytes marked	1.75%	1.54%
Obs. Mean Pkt. Size	985	1038

Table 8: Results of baseline simulations using drop-tail and RED pq_pm queues.

Metric	bq_bm	bq_pm	pq_bm
Bandwidth Util.	27.7%	24.9%	90.9%
Fairness Index	0.8865	0.9045	0.9812
Kilobytes Marked	1287	1297	1479
Packets Marked	1540	1537	875
KB/Packets	0.84	0.84	1.69
% bytes marked	10.3%	10.4%	4.06%
Obs. Mean Pkt. Size	487	472	782

Table 9: Results of the RED byte-based variants using an MPS of 40 bytes.

MPS of 40 bytes. An MPS of 40 bytes is small given the observed mean and median packet sizes from the baseline simulations (and also that 40 bytes is the minimum TCP segment size). We observe that in the pq_bm simulations the small MPS made little difference, as the bandwidth utilization is comparable to that of the drop-tail and pq_pm queues (table 8), while showing greater fairness than the pq_pm case. The main difference from the baselines is that there is more marking because the small MPS causes the probability that a packet will be marked to increase (when compared to the case of a larger MPS). The two RED variants using byte mode for the queue calculation fared worse than the pq_bm case both in terms of aggregate utilization and fairness. Both bq modes transmit less than a third of the bytes across the bottleneck when compared to the pq_pm variant and their fairness indices are roughly the same (and less than the pq_bm case). This shows that the byte marking mode is more robust against low mean packet size settings than the byte-based queue measurement mode.

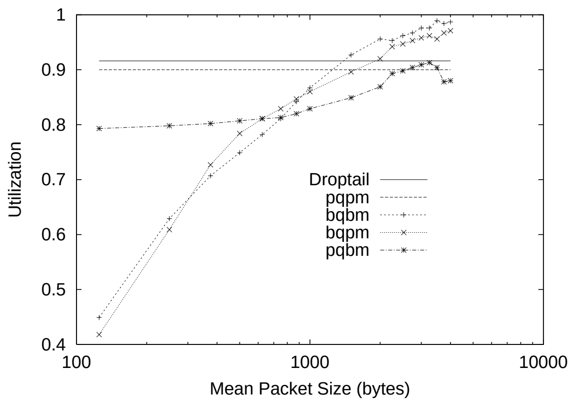
Table 10 contains a summary of the results of our simulations with an MPS of 4352 bytes, which is the largest MTU of any of the links feeding into the bottleneck in our topology. In addition, we use 4352 bytes as a large value to explore the behavior of RED when the MPS is larger than the actual mean size of packets arriving at the router. The pq_bm statistics shown in table 10 are similar to results of the simulations involving an MPS of 40 bytes, with the exception that a lower percentage of bytes are marked. This result confirms that the byte marking mode is robust to poor selection of the mean packet size parameter in either direction and the increased level of fairness and decreased percentage of bytes marked indicate that in the bq_ modes it is better to set MPS too high rather than too low. All three modes shown in table 10 have observed mean packet sizes close to those in the baseline simulations with drop-tail and pq_pm queues (roughly 1000 bytes). The two variants using byte mode for measuring the queue length perform better in terms of utilization than predicted by the results in the previous sections. When measuring the queue in terms of bytes we observe that utilization exceeds both the drop-tail and pq_pm baselines. We also observe that when using byte-based queueing the calculated average queue length stays lower than the queue length when measured in terms of packets because of the high mean packet size setting. Therefore, when using byte-based queueing the queue uses more of the available buffer space, increasing queueing delay and decreasing the ability to deal with burstiness (all working against the overall goals of RED queueing). Neither of these disadvantages are reflected in table 10.

Metric	bq_bm	bq_pm	pq_bm
Bandwidth Util.	99.8%	96.6%	91.0%
Fairness Index	0.9312	0.6668	0.9567
Kilobytes Marked	353	372	644
Packets Marked	236	451	459
KB/Packets	1.50	0.82	1.40
% bytes marked	0.79%	0.85%	1.55%
Obs. Mean Pkt. Size	897	1072	830

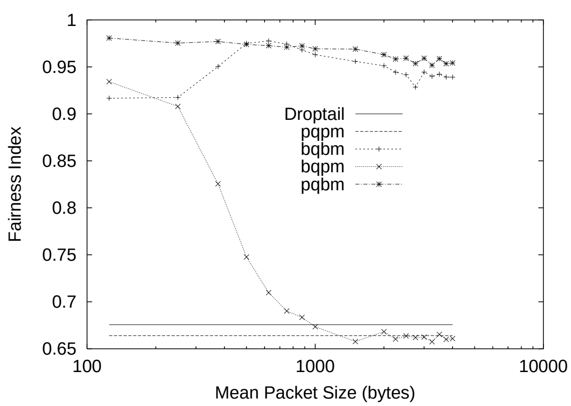
Table 10: Results of the RED byte-based variants using an MPS of 4352 bytes.

Figure 7 illustrates the bandwidth utilization and fairness indices attained by the three RED modes using byte calculations as a function of the MPS setting and compares them to the performance of drop-tail and pq_pm RED queues. Figure 7 is consistent with the results presented above for the cases when the MPS is set to the extremes of 40 and 4352 bytes. The plot shows that the MPS has a greater effect on queue measurement when compared to the marking probability because the queue length is constantly being calculated while the marking probability is only calculated when the average queue length lies within a certain range. Therefore, invocation of the marking function is not only less frequent than queue measurement, but its invocation is dependent upon the queue measurement's result.

The plots also show that utilization improves when using byte measured queues as the value of the MPS setting grows larger than the actual mean packet size! This is ex-



(a) Utilization



(b) Fairness

Figure 7: Utilization and fairness as a function of the MPS setting.

plained by considering an example with an MPS setting of 2000 bytes and the actual mean size of packets traversing the link is near 1000 bytes. When using bq the queue will be able to store twice as many packets before signaling congestion when compared to the case when the measurements are conducted in terms of packets. The disadvantages of the situation include an increase in the queueing delay, reducing RED's ability to absorb burstiness, and potentially creating a problem when processing power in terms of packets per second is the limit, rather than the bandwidth of the bottleneck link. Using an MPS setting that is large compared to the actual packet size of the packets traversing the queue is re-introducing the drawbacks of drop-tail queueing.

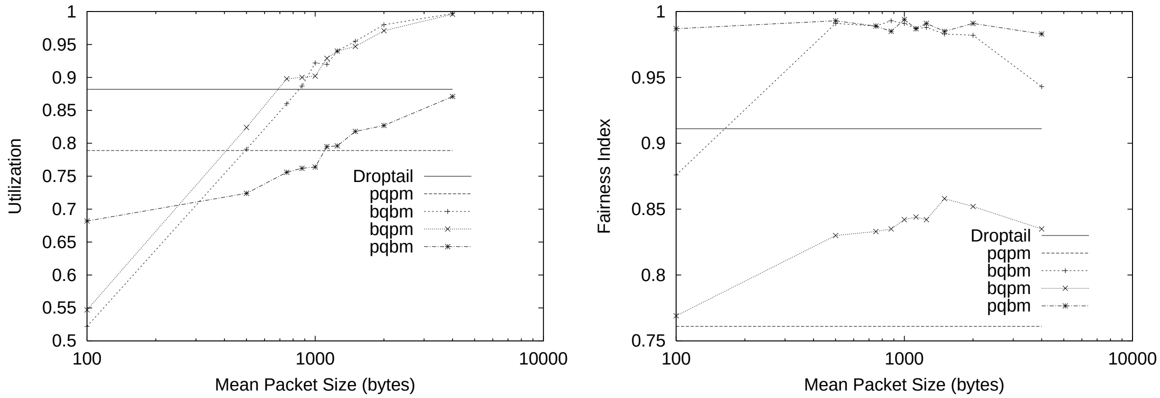
Figure 7(b) agrees with our previous simulations, indicating the marking mode is the key ingredient for fairness and that byte marking produces greater fairness. The main conclusion we can draw from figure 7(b) is that in terms of fairness, byte-based marking is robust to poor selection of the MPS parameter across a range of choices. The bq_pm variant, however, suffers a decrease in the fairness level as the MPS setting increases. The bq_pm variant obtains a fairness index on par with drop-tail and pq_pm queues when the MPS setting is too high, and achieves better fairness when the setting is too low. The results presented in this section indicate that it is possible in the bq_bm mode to increase bandwidth utilization greatly without drastically decreasing fairness by merely tuning the MPS setting.

6.3 WWW Traffic

Figure 8 shows the utilization and fairness of WWW traffic through different RED variants as a function of the MPS setting. For these simulations we use 200 HTTP client/server pairs (as in the heavy WWW simulations outlined in § 5.3) and the points on the plot represent the average of 30 simulation runs. As in previous simulations we use 5 nodes on each side of our network to source and sink data. Two of the nodes use 576 byte packets, two use 1500 byte packets and the remaining node uses 4352 byte packets.

Figure 8(a) shows that the utilization in the WWW traffic simulations increases with the mean packet size setting. The plot shows that with an MPS of roughly 1000 bytes or greater (near the actual MPS of the network being simulated) the utilization obtained by the two byte queueing RED variants is greater than the utilization of drop-tail queues. The utilization of both variants grows as the MPS setting increases to the maximum packet size transmitted in the network.

However, figure 8(b) shows the downside of increasing the MPS setting too much. The plot shows that both byte marking RED variants achieve a higher degree of



(a) Utilization

(b) Fairness

Figure 8: Utilization and fairness of RED byte-based variants as a function of the MPS setting.

fairness than drop-tail queueing, as well as the packet marking RED variants. However, when the MPS setting is 4352 bytes (the maximum packet size sent in our network) the fairness achieved by the bq_bm RED variants is reduced when compared to a more accurate MPS setting.

In sum, the results indicate that the bq_bm variant of RED can achieve both high utilization and a high degree of fairness when the MPS setting is approximately correct (within roughly ± 250 bytes of the actual mean packet size). Future work in this area could include attempting to derive a simple scheme to determine and set the MPS setting in bq_bm RED queues in networks with a dynamic packet size mix (including reseting the MPS setting on the fly based on empirical measurements).

6.4 Summary

We draw the following conclusions from the experiments presented in this section:

- We show that RED is robust to MPS settings that are off by roughly ± 250 bytes from the actual mean packet size observed at the queue.

- We show that diversity in packet size distributions occurs in the Internet. However, the majority of the distribution of average packet sizes in our two datasets falls within a range of 500 bytes suggesting that a static MPS could be derived for a particular output link. Another approach to the problem of choosing an MPS parameter is to derive it from the traffic flowing through the bottleneck dynamically (much like [FKSS99] suggests other RED parameters be derived). While this approach may yield a more accurate MPS setting there are details that need worked out (e.g., how to average packet sizes and over what timescales). In addition, a dynamic calculation costs additional CPU cycles. Without further study we cannot say definitively that a static MPS would be preferable to the cost of deriving a dynamic estimate, only that a static MPS offers reasonable performance assuming somewhat proper tuning).

7 Discussion

The results presented in the previous sections quantify the behavior of various RED queueing strategies in several scenarios. Based on the results it is still difficult to ascertain which RED variant (if any) is the "best" across a variety of situations. To further aid our understanding we have developed a method for rating the RED variants within each simulation scenario, using the following equation:

$$R = \prod_{i=1}^{n} f_i^{\alpha_i} \tag{2}$$

where each $f_i \in (0, 1]$ represents a *factor* in the rating and α_i is the weight for the i-th factor. The α_i terms sum to 1.0. Example factors include fraction of the bottleneck link utilized, fairness index and average fraction of the queue occupied. Larger values of α_i can be used to give more weight to one factor over others when analyzing queue behavior.

We chose to use the product of the weighted factors to favor variants of RED that are *balanced*. Another way to define the rating would be to sum $f_i \cdot \alpha_i$ terms. However, consider an example with equal weighting for each factor. In this case, $f_1 = \frac{1}{3}x, f_2 = \frac{2}{3}x$ yields the same rating as $f_1 = f_2 = \frac{1}{2}x$. Whereas, when using equation 2 the case when f_1 and f_2 are the same gives a higher rating than the case when $f_2 \neq f_1$. That is, when one factor excels at the expense of the other factors the rating is not as good as if all factors performed similarly.

From the generic framework defined above we define a more specific rating to look at the results of our simulations. For each variant and each simulation scenario we calculate a rating, R', as follows:

$$R' = U^{\alpha} \cdot F^{(1-\alpha)} \tag{3}$$

where U is the average observed fraction of the link utilized and F is the average observed fairness index.

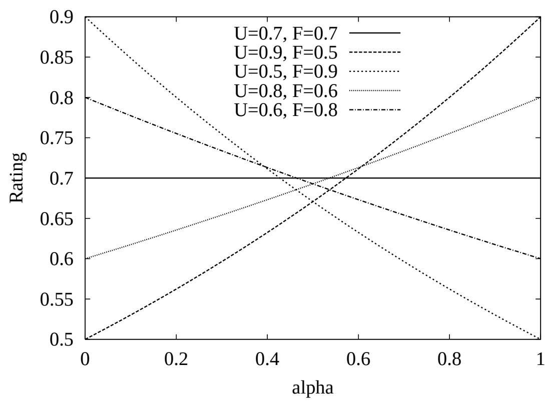


Figure 9: Examples of the rating defined in equation 3 as a function of the weight assigned to the utilization.

Figure 9 shows several examples of different U and F values over the range of choices for α. The figure highlights several properties of the rating system:

- When $U = F = c$ the assigned weights have no impact on the rating which is c.
- The rating ranges from $min(U, F)$–$max(U, F)$ depending on α.
- When the relative weights of utilization and fairness are equal ($\alpha = \frac{1}{2}$) and $U + F = c$ we observe higher ratings for more balanced situations. That is, when $U = F = 0.7$ we obtain a higher rating than when $U = 0.8, F = 0.6$.

We calculate R' for each RED variant in each simulation scenario with $\alpha = \frac{1}{2}$ and then rank the variants within each scenario and report these rankings in table 11. The "MPS" lines in the table are taken from § 6 and use an MPS setting of 1000 bytes (approximately the measured mean packet size in those scenarios). The table shows that the _bm modes obtain the highest two ranks most often, with the bq_bm variant garnering the highest rank in 6 out of the 8 scenarios. These results suggest that the bq_bm variant is the most *well rounded* variant, given that utilization and fairness are taken with equal weights and in such a situation a well balanced variant is also desirable according to equation 3. However, while bq_bm is the highest rated variant most of the time the variant shows the lowest rating in the M_e scenario. Therefore, it is important to note that the rating of a variant is dependent on the network conditions and just because bq_bm is often highly ranked does not imply that it will always perform better than the alternate variants. Determining a worst variant when $\alpha = \frac{1}{2}$ is more difficult. Both _pm modes are generally ranked low, but neither distinguishes itself over the other enough to determine a consistent trend across the scenarios.

Table 12 shows the rankings (based on R') for the RED variants when $\alpha = \frac{4}{5}$ (i.e., utilization is deemed more important than fairness). As in the case when $\alpha = \frac{1}{2}$ the only clear trend from the table is that the bq_bm variant is generally ranked either 1 or 2.

While both tables suggest that the bq_bm variant is a reasonable choice across a variety of scenarios it is not always the highest ranked version of RED. Therefore, we encourage operators investigating RED to verify that the variant choice made is appropriate for the given network using the rating system given in this section.

Finally, we note that there are many additional ways to rate the performance of a queueing scheme to compare across variants. We believe the rating system outlined in this section is a reasonable method. However, the system should not be considered *the* way to rate queueing schemes, but rather *a* way to derive useful high-level information about the multiple factors a queueing scheme tries to optimize.

8 Conclusions and Future Work

In this paper we explore the various biases exhibited by four RED variants. This paper makes several contributions to the community's understanding, as follows.

- We *quantitatively confirm* the biases in RED when using various calculation strategies that have been widely conjectured.
- We show that the choice of calculation unit has more impact on the marking strategy when compared to the queueing strategy.
- The choice of RED variant can have a large impact on both the utilization and fairness of the bottleneck (e.g., in the heavily loaded WWW simulations presented in § 5.3), but the choice also sometimes matters little (e.g., the lightly loaded WWW simulations presented in § 5.1).

Experiment	**Section**	**bq_bm**	**bq_pm**	**pq_bm**	**pq_pm**
Extreme Mix	4.1	1	3	2	4
M_s	4.2	1	4	2	3
M_e	4.2	4	3	2	1
Light Load	5.1	1	4	3	2
Moderate Load	5.2	2	1	4	3
Heavy Load	5.3	1	3	2	4
MPS-Bulk	6.2	1	4	2	3
MPS-WWW	6.3	1	2	3	4

Table 11: Ranking of the ratings (from equation 3) across RED variants and simulation scenarios with $\alpha = \frac{1}{2}$.

Experiment	**Section**	**bq_bm**	**bq_pm**	**pq_bm**	**pq_pm**
Extreme Mix	4.1	4	2	3	1
M_s	4.2	1	3	4	2
M_e	4.2	4	3	2	1
Light Load	5.1	1	4	2	3
Moderate Load	5.2	2	1	4	3
Heavy Load	5.3	2	1	4	3
MPS-Bulk	6.2	1	4	2	3
MPS-WWW	6.3	1	2	3	4

Table 12: Ranking of the ratings (from equation 3) across RED variants and simulation scenarios with $\alpha = \frac{4}{5}$.

- When using byte-based marking we found that byte-based queueing leads to generally smaller queues when compared to packet-based queueing.

- We show that the distribution of packet sizes on real networks is heterogeneous. This means that no one particular value for RED's mean packet size (MPS) is likely to be good across networks. We also show that performance of RED is robust to MPS settings that are off by roughly $\pm$ 250 bytes. Future work could include an in-depth study of packet sizes on a wide range of networks to determine whether a static value can be derived periodically to use as the MPS or whether a more costly process within the RED queue may be needed to make the MPS setting highly dynamic to accommodate different networks' needs.

- We define a generic rating system that takes various queueing factors into account, as well as a notion of balance between the factors. This rating system offers a method for easily comparing RED variants across various scenarios.

- We define the generic rating in terms of utilization and fairness and find that across a number of simulation scenarios using byte mode for both queueing and marking of traffic offers the highest rating in most instances. This suggests that the bq_bm variant is likely a reasonable choice in a variety of networks (but, not all networks since the bq_bm variant is also the lowest ranked variant in some situations).

Future work in this area should include verifying the results with testbed and/or live network experiments. In addition, studying the interactions between the RED variants and the RED parameters (e.g., min_{th}) would be useful.

Acknowledgments

We thank the following people for their insights in discussions about this work: Fred Baker, Brian Frantz, Bob Dimond, Jim Griner, Joseph Ishac, Will Ivancic. We also thank the anonymous Computer Networks reviewers for valuable feedback.

References

[All00] Mark Allman. A Web Server's View of the Transport Layer. *Computer Communications Review*, 30(5):10–20, October 2000.

[ALLY01] Sanjeewa Athuraliya, Victor Li, Steven Low, and Qinghe Yin. REM: Active Queue Management. *IEEE Network*, 15(3):48–53, May/June 2001.

[BCC+98] Robert Braden, David Clark, Jon Crowcroft, Bruce Davie, Steve Deering, Deborah Estrin, Sally Floyd, Van Jacobson, Greg Minshall, Craig Partridge, Larry Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and Lixia Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet, April 1998. RFC 2309.

[Bra89] Robert Braden. Requirements for Internet Hosts – Communication Layers, October 1989. RFC 1122.

[cFKSS99] Wu chang Feng, Dilip Kandlur, Debanjan Saha, and Kang Shin. Blue: A New Class of Active Queue Management Algorithms. Technical Report CSE-TR-387-99, University of Michigan, April 1999.

[DEP00] Stefann DeCnodder, Omar Elloumi, and Kenny Pauwels. Effect of Different Packet Sizes on RED Performance. In *Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC)*, 2000.

[DKS89] Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *ACM SIGCOMM*, pages 1–12, September 1989.

[FJ93] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[FJ95] Sally Floyd and Van Jacobson. Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386, August 1995.

[FKSS99] Wu-Chang Feng, Dilip Kandlur, Debanjan Saha, and Kang Shin. A Self-Configuring RED Gateway. In *IEEE InfoCom*, 1999.

[Flo94] Sally Floyd. TCP and Explicit Congestion Notification. *Computer Communications Review*, 24(5):10–23, October 1994.

[Flo97a] Sally Floyd. RED: Discussions of Byte and Packet Modes, March 1997. http://www.icir.org/floyd/REDaveraging.txt.

[Flo97b] Sally Floyd. RED: Discussions of Setting Parameters, November 1997. http://www.icir.org/floyd/REDparameters.txt.

[Flo00] Sally Floyd. Recommendation on using the ”gentle_” variant of RED, March 2000. http://www.icir.org/floyd/red/gentle.html.

[Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. Wiley, 1991.

[LM97] Doug Lin and Robert Morris. Dynamics of Random Early Detection. In *ACM SIGCOMM*, September 1997.

[MC00] Sean McCreary and K. Claffy. Trends in Wide Area IP Traffic Patterns A View from Ames Internet Exchange. In *Proceedings of ITC*, May 2000. http://www.caida.org/outreach/papers/AIX0005/.

[MMFR96] Matt Mathis, Jamshid Mahdavi, Sally Floyd, and Allyn Romanow. TCP Selective Acknowledgement Options, October 1996. RFC 2018.

[PF01] Vern Paxson and Sally Floyd. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.

[RFB01] K.K. Ramakrishnan, Sally Floyd, and David Black. The Addition of Explicit Congestion Notification (ECN) to IP, September 2001. RFC 3168.

[SMM98] Jeff Semke, Jamshid Mahdavi, and Matt Mathis. Automatic TCP Buffer Tuning. In *ACM SIGCOMM*, September 1998.

[TMW97] Kevin Thompson, Gregory Miller, and Rick Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, 11(6):10–23, November/December 1997.

Wesley M. Eddy is a graduate student at Ohio University working as a member of the Internetworking Research Group advised by Shawn Ostermann. He graduated with a BS in computer science from Ohio University in 2002. The work described here was sponsored by the NASA Glenn Research Center where he has interned for several summers.

Mark Allman is a computer scientist working for BBN Technologies at NASA's Glenn Research Center. His current research interests are in the areas of transport protocols, congestion control and measuring network dynamics. Mark is involved in the Internet Engineering Task Force, where he has chaired several working groups and BoFs and is currently a member of the Transport Area Directorate. Mark also chairs the Internet Measurement Research Group within the Internet Research Task Force. Mark holds B.S. and M.S. degrees in computer science from Ohio University.