Analysis of Active Queue Management

Jae Chung and Mark Claypool
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609, USA
{goos|claypool}@cs.wpi.edu

Abstract

Active Queue Management (AQM) is intended to achieve high link utilization with a low queuing delay. Recent studies show that RED, one of the most well-known AQMs, is difficult to configure and does not provide significant performance gains given the complexity required for proper configuration. Recent variants of RED, such as Adaptive-RED are designed to provide more robust RED performance under a wider-range of traffic conditions but have not yet been evaluated. This paper presents a router queue behavior model (a queue law) for TCP-dropping and TCP-marking control systems, and uses the queue law to illustrate the impact of TCP traffic on the load and queue behavior of congested routers. Through queue law analysis and simulation, this paper confirms that RED-like AQM techniques that employ packet dropping do not significantly improve performance over that of drop-tail queue management. However, when AQM techniques use Explicit Congestion Notification (ECN) as a method to notify TCP sources of congestion rather than packet drops, the performance gains of AQM in terms of goodput and delay can be significant over that of drop-tail queue management.

1 Introduction

To prevent congestion collapse, the current Internet uses end-to-end congestion control. At the end-host, responsive traffic flows like TCP treat network packet losses as implicit network congestion signals from routers and reduce their transmission rate. In the network, routers use outbound queues to accommodate traffic bursts and achieve high link utilization. Due to the simplicity of the FIFO queuing mechanism, drop-tail queues that drop incoming packets when the queue is full are the most widely used on the Internet today. Unfortunately, when faced with persistent congestion, drop-tail queues fill up resulting in higher delays. In addition, drop-tail queues can also result in bursty packet drops, degrading system instability and bandwidth fairness.

Active Queue Management (AQM) proposes to replace drop-tail queue management in order to improve network performance in terms of delay, link utilization, packet loss rate and system fairness. AQM enhances routers to detect and notify end-systems of impending congestion earlier, allowing responsive traffic sources to reduce their transmission rates before congested router queues overflows. AQM routers may also explicitly signal end-systems of network congestion by marking the Explicit Congestion Notification (ECN) [8] bit in the IP header rather than dropping packets, which may dramatically reduce network packet loss rate and improve goodput.

Understanding AQM congestion control requires knowledge on the response characteristics of the traffic to control. Internet traffic has been dominated by TCP [15] that responds to network congestion in an Additive Increase Multiplicative Decrease (AIMD) manner. The feedback control system theory in [6] provides insight into the impact of TCP traffic on a router with an AQM using probabilistic random drops. In [6], Firoiu and Borden used a feedback control model for long-lived TCP flows to derive a router queue behavior model called a *queue law*. Their queue law shows that a router queue at equilibrium has an average queue length as a function of the packet drop probability. The queue law is particularly useful for configuring the Random Early Detection (RED) family[1] router queue management mechanisms that use the average queue length to compute drop probability. Yet, the queue law in [6] does not clearly illustrate how different TCP traffic parameters affect the traffic control aspect of AQM. Moreover, the queue law does not fully address congestion control with ECN.

This paper develops a model for load at a router from TCP traffic and extends the queue law in [6] to support ECN traffic as well as non-ECN traffic. Using our TCP traffic

---

[1] RED [10] attempts to achieve high link utilization with a low queuing delay, and has evolved to Adaptive RED [5, 9] for more robust performance over a wider-range of traffic conditions.

load model and our more general queue law, we show the impact of a variety of TCP traffic on the load and queue behavior of congested routers. Also, we show the difference between ECN and non-ECN traffic from the viewpoint of the router, and discuss the impact of ECN traffic on AQM. In the latter half of this paper, we depict additional congestion signaling issues that cannot be covered by the queue law and carefully analyze the performance of RED-family router queue management mechanisms over a range of TCP traffic loads. We conclude by demonstrating the substantial benefits of AQM when used in conjunction with ECN over AQM with packet drops or drop-tail queue management.

We choose to study and evaluate RED-family AQM for the several reasons: First, while recently proposed AQM mechanisms such as REM [1], PI [11], and AVQ [12] do not strictly use the average queue to compute congestion, they have performance goals similar to that of RED-family AQMs. Second, since RED congestion control mechanisms are relatively well-understood and are commonly used as a benchmark for evaluation of other AQM mechanisms (including REM, PI and AVQ), further extending an understanding of RED-family mechanisms and illustrating possible outcomes of RED will help researchers to design experiments that fairly compare RED with other AQM mechanisms. Third, with the help of a general queue law, it is easy to configure RED-family AQMs to create various test scenarios that reveal interesting AQM congestion feedback control issues. In turn, a general queue law can be revalidated through evaluation with RED configurations. Fourth, since RED is already implemented in some commercial routers[2], our queue law can be used to tune these routers, thus realizing some of the potential benefits of ECN with few network infrastructure changes. Lastly, Adaptive RED [9] has not yet been thoroughly evaluated and yet shows promise as a viable AQM.

The rest of the paper is organized as follows: Section 2 derives a general queue law for systems that use implicit congestion notification and ECN; Section 3 illustrates the impact of TCP traffic on router queue behavior and discusses the effect of short-lived flows on end-to-end congestion control; Section 4 compares and contrasts our queue law for packet dropping and ECN marking systems; Section 5 describes RED-family AQM mechanisms; Section 6 carefully configures RED family AQMs for both ECN and non-ECN traffic using our queue law and compares the performance of the RED family AQMs in terms of packet loss rate, delay and queue oscillation; and Section 7 concludes the paper.

2 Router Load and the Queue Law

In this section, we develop a simple model of traffic load at a router with persistent congestion, apply it to TCP traffic, and derive a general and complete queue law that works both for packet dropping and ECN marking systems.

Assuming a TCP only network with one congested router that uniformly notifies traffic sources of congestion with a probability, there exists a relationship between the average queue size, congestion notification probability and capacity of the router, and the TCP traffic parameters such as the number of flows and the average round trip time. Firoiu and Borden [6] define this relationship as a“queue law” and apply it to systems that use packet drops for congestion notification. Their queue law can be used to estimate a router's congestion notification probability in order to provide a target average queue size (or vice versa) for a given TCP traffic mix, and can be used to more easily configure a RED router.

More generally, traffic load at a router queue (L) can be expressed as the ratio of the packet arrival rate (AR) over the link service rate (SR) (usually the capacity of the outgoing link). When the router faces persistent congestion, the traffic load times one minus the dropping probability (p_d) is one if the system stable. Applying this stable load equation to a TCP-only traffic mix, AR can be expressed in terms of the number of TCP flows (N), average TCP window size (tcp_wnd) and the average round trip time (RTT) of all flows traversing the router. The RTT can be further decomposed of the queuing delay (q) at the congested router and the average round trip link delay ($RTLD$). Thus:

$$\begin{aligned} L(1-p_d) &= \frac{AR}{SR}(1-p_d) \\ &= \frac{N \times tcp\_wnd(p_d+p_m) \times (1-p_d)}{RTT \times SR} \\ &= \frac{N \times tcp\_wnd(p_d+p_m) \times (1-p_d)}{(RTLD + q/SR) \times SR} = 1 \qquad (1) \end{aligned}$$

Note in Equation 1 that the average TCP window size of flows traveling through the router is the function of only the implicit (p_d) and explicit ECN (p_m) marking probabilities (or $p = p_d + p_m$, in general), as shown in [14]. Re-writing Equation 1 for q, we get a general queue law which implies that the queue of the congested router is linearly proportional to N, inversely proportional to $RTLD$ and SR, and shows that q is function of only p_d and p_m, $q = f(p_d + p_m)$, for a given TCP traffic mix:

$$q = N \times tcp\_wnd(p_d + p_m) \times (1 - p_d) - RTLD \times SR \qquad (2)$$

To validate this new queue law, we ran a series of NS[3] simulations using the network setup shown in Figure 1,

---

[2] Cisco GSR, http://www.ieng.com/warp/public/cc/pd/rt/12000/index.shtml

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

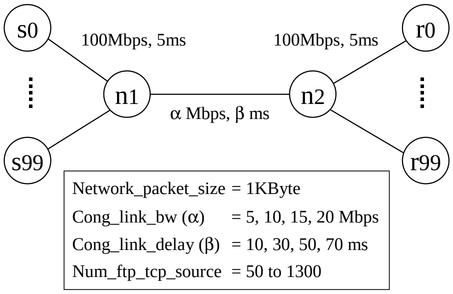


Figure 1. Simulation Network Setup.

and compared the results with the corresponding theoretical queue law curves. For the TCP window, we used [14], which models long-lived TCP sources that have no congestion window limit (*cwnd_limit*) nor receiver window limit, and always have data to transmit. For the simulated network traffic sources, we used bulk transfer FTP applications on top of TCP NewReno and set the *cwnd_limit* of all TCP agents to infinite. For the congested router queue, we implemented a theoretically infinite queue that randomly drops or ECN marks incoming packets with a fixed congestion notification probability.

Figure 2 shows simulated and theoretical queue law curves for both packet dropping and ECN marking systems for 300 TCP flows with a service rate (SR) of 20 Mbps and a round trip link delay ($RTLD$) of 80 ms. Figure 2 illustrates that the new queue law predicts the average queue size for given congestion notification probability very well for ECN marking systems. The queue law precition is not as accurate for the packet dropping systems as for the ECN marking systems, and the precision decreases slightly as p_d increases. We believe that this is because the TCP model used does not accurately model the TCP window behavior for TCP timeout behaviors. Results from other configurations [3] show similar accuracy of our new queue law but are not presented here due to space limitations.

3 Analysis of Queue Law

In this section, we discuss the impact of TCP traffic on a congested router queue using our general queue law. Figure 3 illustrates the relationship between q at the router and N, SR and $RTLD$ shown in Equation 2 through simulation. Note that the congested router is configured to use packet drops for this section; the congested router is configured to use marking for the next section.

Figure 3 (top) shows the measured queue law curves for systems that have settings that differ only by the number of TCP connections: 100, 200 and 300. At a given drop rate where all of the queue averages are greater than 0, the average queue size increases linearly with the number of

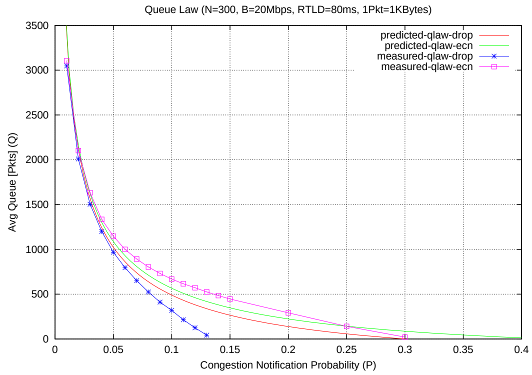


Figure 2. Queue Law: Theory and Simulation Results Comparison

flows. Figure 3 (middle) shows the queue law of the congested router for simulations that differ only in the average round trip link delay of each source: 40, 80, 120 and 160 ms. At a given drop rate where all of the queue averages are greater than 0, the average queue size increases linearly with the round trip link delay. Figure 3 (bottom) shows the queue law of the congested router for simulations that differ only by the link bandwidth: 5, 10, 15 and 20 Mbps. At a given drop rate where all of the queue averages are greater than 0, the average queue size decreases linearly with the link bandwidth.

So far, the queue law has been examined with long-lived TCP flows in which the size of the congestion window (or receiver window) is unlimited and the traffic sources have an infinite amount of data to transmit. In this case, the average TCP window is the function of the congestion notification (drop/mark) probability only and behaves identically under different the network and traffic configurations. However, in real networking environments the TCP congestion window (or receiver window) and/or data object size impose window limits and alter the congestion response behavior of TCP flows. Thus, the average TCP window may not behave the same from one TCP traffic mix to another. As a typical example, consider a TCP traffic mix that consists entirely of short-lived Web flows in which small Web objects limit the window growth before the transmission ends. The TCP window size averaged over all connections will often be less than in the case of unlimited window growth under the same drop/mark rate, especially for a low drop/mark rates.

We illustrate the effect of limited TCP window growth by setting all the TCP congestion window limits to first 12 packets, and then 6 packets. The number of TCP connections used in the simulations is 700 and 1300 correspondingly in order to have the same queue average at a drop rate of 0.01. For this set of simulations, the congested link bandwidth and the round trip link delay is set the same as in the

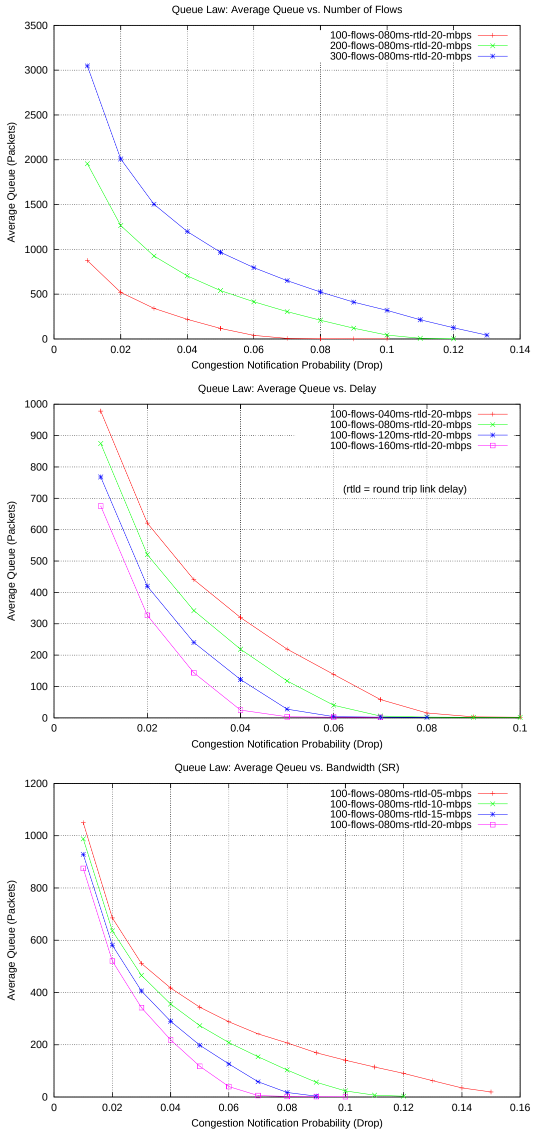


Figure 3. Queue Law: q vs. N (top), $RTLD$ (middle) and SR (bottom)

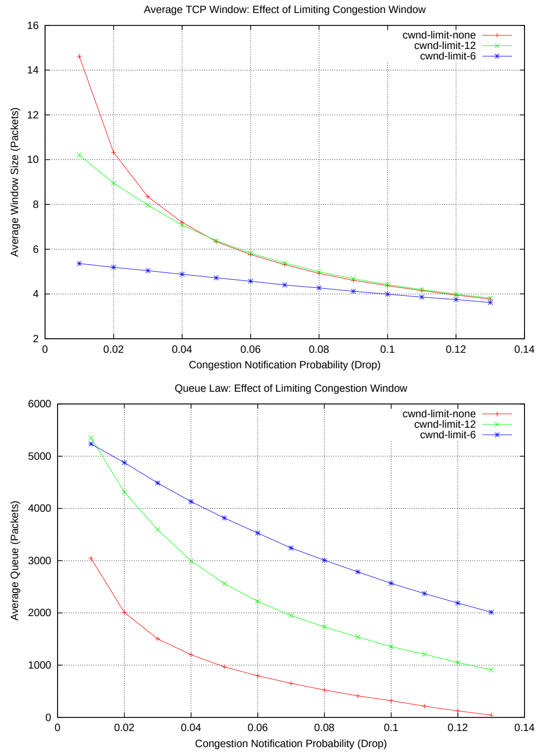


Figure 4. Effect of Limiting TCP Congestion Window: Average TCP Window (top) and Queue Law (bottom)

previous unlimited congestion window simulation that had 300 TCP connections. Figure 4 (top) shows the average TCP window behavior, and (bottom) shows the corresponding queue law curve.

Figure 4 (top) shows that as the congestion window (*cwnd*) limit decreases, the average congestion window curve flattens. The small average windows make TCP connections much less responsive in that a change in the drop rate has less effect on queue size than does the same change in the drop rate for TCP flows with large windows. This reduced responsiveness is especially true at relatively low drop rates. Thus, a router that is congested with many short Web flows will need to apply a relatively high drop rate to keep the average queue length within a certain range since the short TCP flows are less responsive than are longer TCP flows. For example, the simulation with smaller *cwnd*s (6 packet limit) has to apply about twice as high a drop rate as in the simulation with the larger *cwnd*s (12 packet limit) to maintain an average queue length of 3000 packets.

The queue law illustrates that a router can compute an optimal congestion notification probability and better manage queue size in congestion if it is informed of the number of flows (N), round-trip link delay for each flow ($RTLD$), service rate (SR), and average TCP window size. Practically, SR is known to the router, counting N can be done by techniques such as in [13], but obtaining $RTLD$ (or RTT) and the TCP window size requires an Internet structure change. Such changes are part of our ongoing work.

4 Feedback Method: Drop vs. Mark

In this section, we briefly compare and contrast our queue law using ECN with our queue law using packet drops. Figure 5 shows the measured queue law curves for drop and mark systems with from 100 to 300 flows. For the

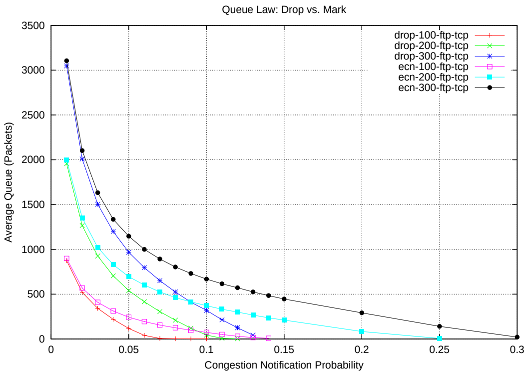


Figure 5. Queue Law: Drop vs. Mark

same number of flows, the average queue lengths for TCP and TCP with ECN are almost the same when congestion notification probability is low. However, as the congestion notification probability increases, the average queue length for TCP with ECN decreases noticeably less and at a steadier rate than the average queue length with TCP.

It follows that several significant points can be made:

First, an ECN enabled AQM should be configured to apply a significantly higher marking rate than the same AQM using packet drops in order to operate with a reasonably low queuing delay. We believe that a common mistake that many researchers make is in using the same AQM settings for both packet drops and ECN marks, resulting in a mark rate that is too low for ECN enabled traffic.

Second, for a reasonable average queue length target (for example, 500 packets in Figure 5), as traffic load increases linearly, to maintain the queue length at the same level the difference between the stable state mark rate and the stable state drop rate must increase exponentially, which indicates that ECN should increase its mark rate exponentially above any drop rate. However, the queue law for ECN converges smoothly towards an average queue size of 0 for a mark probability of 1, suggesting that there exists a mark rate that can keep the average queue length low even for a highly loaded situation. Thus, the benefits of ECN should still be effective, even under a heavy TCP traffic load.

Third, the slowly and steadily decreasing average queue length curves of ECN compared to that of packet drops as the random drop/mark rate increases indicates that the average queue length can be more easily stabilized for AQMs with ECN than for AQMs with drops.

We illustrate the 2nd and 3rd points further in Section 6.

5 RED Family AQM Mechanisms

Random Early Detection (RED) [10] monitors impending congestion by keeping an exponential weighted moving average of the queue (q). When congestion is detected, indicated by the queue average rising above a fixed minimum threshold (min_{th}), incoming packets are randomly marked (drop or ECN) with a probability. The probability of the mark increases linearly from zero at min_{th} to a maximum mark probability (max_p) at the maximum threshold (max_{th}). When the queue average rises above max_{th}, RED drops all incoming packets to ensure a low average queuing delay. RED supports ECN marking while the queue average is in between the min_{th} and the max_{th}. We will call RED using ECN marking RED-ECN hereafter. The performance benefits of RED can be maximized when RED uses ECN marking and is “well-configured” under the traffic load, in which the queue average does not significantly oscillate and stays under max_{th}.

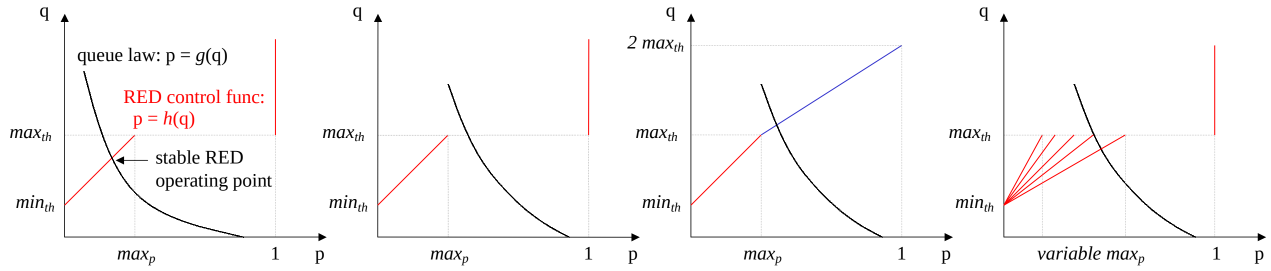
Unfortunately, studies show that RED is hard to configure for a wide-range of traffic mixes and loads [4, 2]. Some of the difficulties in configuring RED can be explained by the TCP-RED feedback control system theory in [6]. Firoiu and Borden derive a queue law and feedback control law for long-lived TCP flows to show that a router queue at equilibrium has a congestion notification probability (random packet drop probability) as a function of the average queue size, q: $p = g(q)$. RED’s feedback control function is a linear function of average queue size: $p = h(q)$. [6] shows that a “well-configured” RED queue may be stabilized if there exists a q inside the thresholds (min_{th} and max_{th}) such that $h(q) = g(q)$, as depicted in Figure 6 (a).

However, RED configurations that work well for one traffic mix and load may not work well for another, since, as shown in Section 3, changes in traffic mix and load alters the queue law curve ($p = g(q)$). Figure 6 (b) shows a RED queue that is “ill-configured” since the stable state average queue is above max_{th}. For this RED configuration and traffic, there would be persistent and large sequential packet drops, degrading network performance in terms of packet loss rate and fairness. Although it is possible to make an ill-configured RED queue a well-configured RED queue for a certain range of traffic, RED configuration difficulties will remain as long Internet traffic mixes and loads vary.

As an attempted to fix the problems with RED configuration over a wide-range of traffic conditions, the “gentle” modification to RED (gentle-RED)[4] was proposed, which modifies the packet drop behavior when the average queue size is over max_{th}, as shown in Figure 6 (c). Instead of setting the drop probability to 1 after the average queue size goes over max_{th}, gentle-RED linearly increases the drop probability from max_p to 1 as the average queue size grows from max_{th} to $2 \times max_{th}$. This modification loosens the bound on the average queue length for a continuous probabilistic drop behavior. Yet, as shown in Section 6, the gentle slope can still be too steep, making the control system unstable.

---

[4] http://www.icir.org/fbyd/red/gentle.html



(a) (b) (c) (d)

Figure 6. Well-configured RED (a), poorly configured RED (b), Gentle RED (c) and Adaptive RED (d)

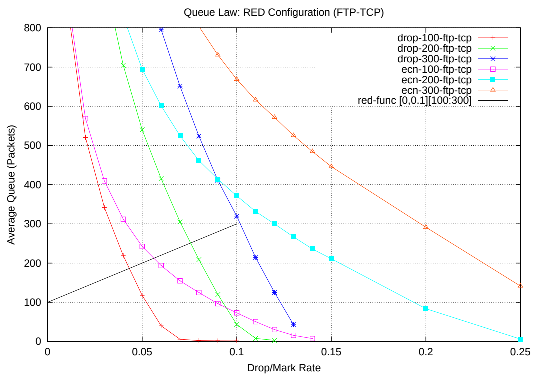


Figure 7. Queue Law: RED Configuration

.

Adaptive RED (A-RED) [5, 9] was proposed to make RED well-tuned over a wider range of traffic conditions. A-RED tries to adapt to changing traffic load by slowly adjusting max_p as shown in Figure 6 (d). A-RED attempts to find a feedback control slope that can intersect the queue law curve to make the system stable for the current traffic load by adjusting max_p up to a limit (0.5, by default). Beyond the max_p limit, A-RED recommends the use of the gentle mode for a continuous probabilistic drop behavior. In Section 6, we provide a more thorough analysis of A-RED than has been presented previously in [9].

6 Analysis of RED Family AQM

This section evaluates RED, gentle-RED, Adaptive RED, RED-ECN, gentle-RED-ECN and Adaptive RED-ECN using the queue law curves for both packet dropping and ECN marking and verifies the effectiveness of the queue law in characterizing RED-family AQM performance. This section also compares the performance of RED family AQMs with one another and also with that of drop-tail queue management in terms of queue size stability, throughput, and packet loss rates to see how RED and its variants behave as they are pushed out of a well-configured state as the offered traffic load increases.

As in the previous sections, we use the network configuration shown in Figure 1 with a congested link bandwidth of 20 Mbps and a round trip time link delay of 80 ms. Each simulation starts with 50 FTP-TCP flows, and 50 more FTP-TCP flows are added every 50 seconds. The physical queue length is set to 500 packets, with the packet size set to 1 Kbyte. For the RED parameter settings, max_p is 0.,, min_{th} is 100 packets and max_{th} is 300 packets, based on recommendations in [7]. Although not shown in Figure 7, the limit of max_p for Adaptive RED is set to 0.5 (the default value), which gives the router queue a chance to be well-configured for all the given TCP traffic loads.

Comparing the queue behavior of each RED family AQM with the queue law shown in Figure 7 demonstrates that the queue law is effective for predicting RED behavior. For RED, the queue law indicates that RED will stably manage TCP traffic up to about 200 flows. In Figure 8 (left), RED's average queue was stable up to a traffic load of 150 flows, but at 200 flows it hit the maximum threshold and becomes increasingly unstable. Gentle-RED, shown in Figure 8 (middle), was able to manage load up to 200 flows since there is no longer a sudden increase in drop probabilities from the max_p of 0.1 to 1 at max_{th}. For RED-ECN, shown in Figure 9 (left), the average queue becomes unstable at a load of 150 flows, as the queue law indicated. And as is the case of gentle-RED, gentle-RED-ECN, shown in Figure9 (middle), also gets the benefit of the gentle behavior for 200 flows.

Our results show that the gentle setting for RED is beneficial when the offered TCP traffic load is slightly greater than the stable target load for a given configuration. However, the benefit of the gentle setting is not as clear in terms of queue oscillations when a RED router is highly overloaded (250+ flows, in our simulations), although the gentle behavior does reduce the packet loss rate somewhat, as shown in Figure 10.

We believe that the steepness of the gentle RED con-

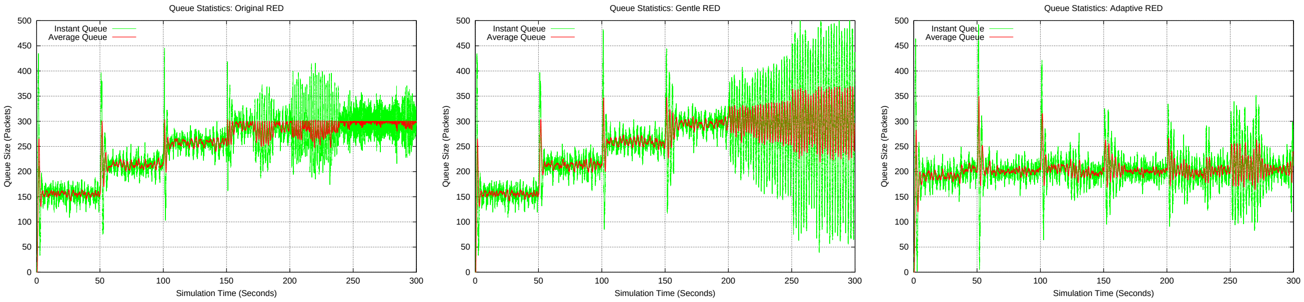


Figure 8. Queue Statistics: RED (left), Gentle RED (middle) and Adaptive RED (right)

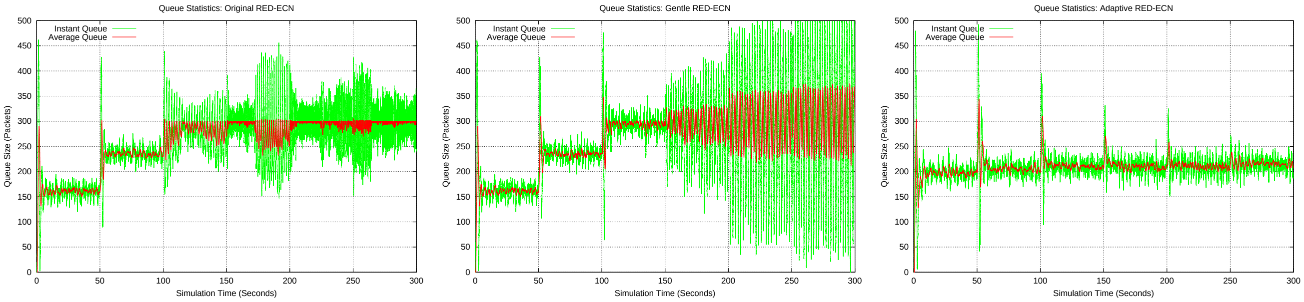


Figure 9. Queue Statistics: RED-ECN (left), Gentle RED-ECN (middle) and Adaptive RED-ECN (right)

trol function ($q > max_{th}$) causes the unstable queue oscillation. When configuring a RED router, max_p is commonly set to a low value (0.1 in our case) to achieve a high throughput, which makes the gentle portion of RED control function significantly steeper than the control slope between min_{th} and max_{th}. When the stable RED operating q is over max_{th}, a small change in q results in a large change in the notification probability (p), which will again cause a large change in q. This process repeats, causing a large and unstable q oscillation.

Comparing the queue behavior of Adaptive RED, shown in Figure 8 (right), and Adaptive-RED-ECN, shown in Figure 9 (right), with non-adaptive versions of RED clearly shows the benefits of adjusting max_p. That is, by finding the proper drop/marking slope for changing traffic load conditions, Adaptive RED can stably handle a wide range of TCP traffic.

The above RED queue stability analysis gives additional information on AQM feedback control that the queue law does not address. That is, the granularity of congestion notification probability adjustment is another sensitive AQM parameter that affects the stability of the control system. It is desirable to consider congestion level information such as number of active flows and average RTT in determining the adjustment granularity. AQM mechanism may need to use heuristics to estimate and reflect the congestion level to calibrate the notification probability adjustment granularity. Adaptive RED implicitly performs this estimate by adjusting max_p based on the current difference between the target queue and the current average queue.

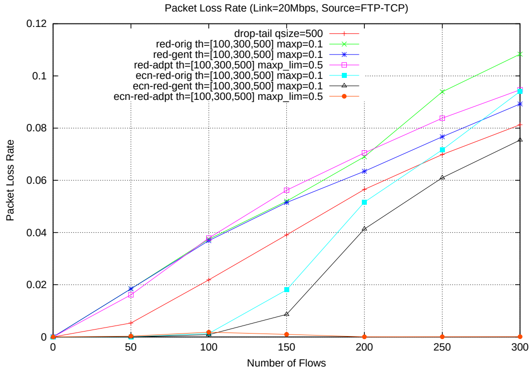


Figure 10. Packet Loss Rate

We next analyze the delay-loss tradeoffs between drop-tail and RED. The bottleneck link in all simulations was fully utilized for all TCP traffic loads[5], thus goodput is affected by packet loss only, in our simulations. Figure 10 shows the packet loss rates at the routers, and suggests that all the RED family queue mechanisms that use drops for congestion notification have consistently higher packet loss rates than do drop-tail routers. Drop-tail does not actively drop packets, so the drop distribution that results from buffer overflow at a drop-tail queue may be bursty.

---

[5] Shown only in [3] due to the space limitations.

However, with many TCP sources, the drops are uniform across flows, resulting in a well-configured state matching the queue law near the drop-tail buffer size. Thus, the delay-loss tradeoff between drop-tail and RED is clear in that RED, using drops as a congestion notification method, pays the price in terms of higher packet drop rates over that of drop-tail to maintain the lower average queue size.

We next consider the benefits of marking (via ECN) over dropping as an indicator of congestion. One of the main issues that discourages deployment of RED (or AQM in general) is that the complexity for AQM design is high compared with the potential gain of a lower average queue size [4, 2]. However, even with the required higher ECN congestion notification rate, the "price" of the notification in terms of packet loss rate or reduced goodput is zero compared to the "price" for dropping packets. Figure 10 shows this clearly. ECN enabled RED and its variants in a well-configured state can bring down the packet loss rate to zero. Furthermore, Adaptive RED-ECN is able to achieve a packet loss rate very close to zero for the entire range of traffic loads (depicted in the very bottom line near zero, nearly on top of the x-axis). In addition, as mentioned in Section 4, ECN enabled AQM can be more stable than AQMs without ECN as the queue law curve decreases far more slowly and steadily under high loads than when using drops. This is shown by by comparing the average queue of Adaptive RED and Adaptive RED-ECN, where the average queue oscillation of the ECN enabled AQM remains more stable even at a high traffic load compared to the AQM that does not use ECN.

7 Summary

In this paper, we develop a model for load on a router during congestion due to TCP traffic, derive a general queue law both for packet drop and ECN mark notification, and use the queue law to illustrate the impact of TCP traffic on congested router queues. We show that the three TCP traffic parameters that affect the behavior of router queues during congestion are the number of flows (N), average round trip link delay ($RTLD$) and average TCP window size.

By comparing and contrasting the ECN queue law and packet drop queue law, we discuss characteristics of ECN traffic and key router configuration issues in order to be well-configured in the presence of ECN traffic. We confirmed that ECN marking routers should apply a significantly higher marking rate than packet drop routers in order to operate with a reasonably low queuing delay. In addition, we found that ECN traffic may help routers during congestion by stabilizing average queue oscillations.

We configured RED family AQMs using our queue law, compared the performance of the RED family AQMs and drop-tail queue management in terms of packet loss rate, queuing delay and queue oscillation, and demonstrated the trade-offs between drop-tail queue management and RED family AQMs. We conclude that RED family AQMs using ECN can be well-configured over a wide-range of TCP traffic mixes, achieving both a low network packet drop rate and a low queuing delay, often far superior to that of drop-tail queue management.

Future work includes extending our study to a mixture of ECN and non-ECN TCP flows. In addition, future work includes building an adaptive AQM technique that makes use of our queue law to more quickly adapt to a well-configured state in the presence of changing network loads.

References

[1] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. REM: Active Queue Management. *IEEE Network*, May 2001.

[2] M. Christiansen, K. Jeffay, D. Ott, and F. Smith. Tuning RED for Web Traffic. In *Proceedings of ACM SIGCOMM*, Aug. 2000.

[3] J. Chung and M. Claypool. Analysis of RED-Family Active Queue Management Over a Wide-Range of TCP Loads. Technical Report WPI-CS-TR-02-05, CS Department, Worcester Polytechnic Institute, Feb. 2002.

[4] C. Diot, G.Iannaccone, and M. May. Aggregate Traffic Performance with Active Queue Management and Drop from Tail. Sprint ATL Technical Report TR01-ATL-012501, 2001.

[5] W. Feng, D. Kandlur, D. Saha, and K. Shin. A Self-Configuring RED Gateway. In *Proceedings of IEEE INFOCOM*, Mar. 1999.

[6] V. Firoiu and M. Borden. A Study of Active Queue Management for Congestion Control. In *Proceedings of IEEE INFOCOM*, Mar. 2000.

[7] S. Floyd. RED: Discussions of Setting Parameters. http://www.icir.org/floyd/REDparameters.txt.

[8] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, 24.

[9] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. Under submission, http://www.icir.org/floyd/papers/adaptiveRed.pdf.

[10] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, Aug. 1993.

[11] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. In *Proceedings of IEEE INFOCOM*, pages 1726–1734, 2001.

[12] S. Kunniyur and R. Srikant. Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management. In *Proceedings of ACM SIGCOMM*, San Diego, CA, USA, Aug. 2001.

[13] R. Morris. Scalable TCP Congestion Control. In *Proceedings of IEEE INFOCOM*, Mar. 2000.

[14] J. Padhye, V. Firoiu, D.Towsley, and J.Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings of ACM SIGCOMM*, 1998.

[15] K. Thompson, G. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, 11(6):10–23, November/December 2000.