Analysis of RED-Family Active Queue Management

Jae Chung and Mark Claypool
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609, USA
1-508-831-5357
{goos|claypool}@cs.wpi.edu

***Abstract*—RED is an Active Queue Management (AQM) technique that is intended to achieve high link utilization with a low queuing delay. Recent studies show that RED is difficult to configure for some rapidly changing traffic mixes and loads [1]. Other studies show that under some conditions, the performance gains of RED and its variants over traditional drop-tail queue management is not significant given the additional complexity required for proper configuration [2], [3]. Recent variants of RED, such as Adaptive-RED [4], are designed to provide more robust RED performance under a wider-range of traffic conditions. This paper develops a general queue law for TCP-RED control systems that use packet dropping and/or Explicit Congestion Notification (ECN) marking as congestion signaling methods, and illustrates the impact of TCP traffic on the behavior of congested router queue. Furthermore, this paper provides additional analysis of RED and newer variants of RED including Adaptive-RED [4] that is designed to provide more robust RED performance under a wider-range of traffic conditions. Through careful simulation designs using the queue law and analysis, 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 ECN marking, the performance gains of AQM in terms of goodput and delay can be significant over that of drop-tail queue management.**

I. INTRODUCTION

To prevent congestion collapse, the current Internet uses end-to-end congestion control, where responsive traffic sources like TCP monitor their own transmission to detect network packet losses [1], take them as implicit congestion signals from routers in the path and reduce their transmission rate accordingly. In the network, routers use outbound queues to accommodate traffic burst 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 most widely used in Internet routers today. Unfortunately, when faced with persistent congestion, drop-tail queues that often are over-provisioned with large buffers to yield maximum throughput fill up resulting in high transmission delays. In addition, bursty packet drops due to drop-tail queue overflow can have negative impacts on system fairness and stability.

Active Queue Management (AQM) is proposed to replace drop-tail queue management targeting to improve performance of network such as delay, packet loss rate and system fairness. AQM enhances network support for end-to-end congestion control by having routers detect and notify end-systems of impending congestion earlier, allowing responsive traffic sources to reduce transmission rate before the congested router queue overflows. Thus, when properly designed and configured, AQM can reduce queuing delays by keeping a lower average queue length while achieving high link utilization. Moreover, since AQM routers are able to predict impending congestion before buffer overflows, they may explicitly signal end-systems of network congestion by marking Explicit Congestion Notification (ECN) [5] bit in the IP header rather than dropping packets, which may dramatically reduce network packet loss rate and improving goodput. Yet another gain from the early prediction is that routers may carefully select end-hosts to signal congestion improving system fairness or to possibly support diverse Quality of Services (QoS) to applications with different QoS requirements.

AQM is often synonymous with the Random Early Detection (RED) family of router queue management mechanisms, first proposed in [6]. RED monitors the outgo-

---

[1] Although not practically used in the Internet today, TCP has an option to use Explicit Congestion Notification (ECN) bit set by ECN enabled routers in congestion to detect network congestion as well.

ing queue for 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}), packets are randomly dropped with a fixed drop probability for each packet. The probability of the packet drop increases linearly from zero at the minimum threshold to a maximum drop probability (max_p) at the maximum threshold (max_{th}). When the queue average does not stay within the max_{th} but rises over it, RED drops all incoming packets to limit the queue average below max_{th} ensuring a low average queuing delay. RED also supports ECN marking congestion notification instead of packet dropping while the queue average is in between the min_{th} and the max_{th}. We'll call RED using ECN marking RED-ECN from here on.

Studies show that RED can improve throughput and fairness over drop-tail queue management while maintaining a low average queuing delay [6], [7]. However, this benefit can be achieved only for "well-configured" RED under some traffic loads, specifically when the queue average does not significantly oscillate and stays under max_{th}. Other researchers conclude that RED is too complicated to configure, and show that end-to-end performance of RED is no better than or even worse than that of drop-tail queue management in many cases [3], [2] resulting in higher packet loss rate and lower goodput. [1] shows that although offering a lower average queuing delay, RED could have negative impacts on response time for short Web transmissions. It is also shown in [1] that the end-to-end performance of RED is very sensitive to the RED parameter settings, and the gain (response time) for carefully tuned RED settings is not significant. These reports raise the concern that using RED router queue management (or AQM in a broader sense) may not be practical in a real Internet environment over a wide-range of traffic mixes and loads.

Some of the difficulties in RED configuration can be explained by TCP-RED feedback control system theory in [8]. 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: $p = g(q)$. As described above, RED active queue management control function determines the congestion notification probability as a linear function of average queue size: $p = h(q)$. [8] shows that a RED queue may be stabilized within the max_{th} if there exists a q inside the thresholds (min_{th} and max_{th}) such that $h(q) = g(q)$ as shows in Figure 1 (a). This describes the minimum requirements of a "well-configured" RED.

RED configurations that work well for one traffic mix and load, however, may not work well for another, since changes in traffic mix and load alters the queue law curve ($p = g(q)$). For example, an increase in the number of TCP flows moves the queue law curve to the upper right. When the stable state average queue length at max_p is above max_{th}, as shown in Figure 1 (b), the RED average queue often grows beyond max_{th} in order to find an equilibrium state, and results in persistent sequential packet drops larger than would drop-tail bursts, which may degrade network performances such as packet loss rate and fairness. Although RED configuration guidelines in [8] may suggest a set of RED parameters that work well for a large set of traffic load, RED configuration difficulties will remain as Internet traffic varies.

As an easy fix to the RED configuration problem, the "gentle" modification to RED was proposed [7], which replaces the packet drop behavior when the average queue size is over max_{th} as shown in Figure 2 (a). 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 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. In other words, gentle-RED may find a stable state drop probability over max_p that may stabilize the queue at some point greater than max_{th}. Unfortunately, the "gentle" modification is not a gentle solution and may result in a very unstable queue oscillations due to stiff slope of the "gentle" portion of RED control function ($q > max_{th}$). This is shown and discussed in Section V.

Recently, researchers proposes Adaptive RED (A-RED) [4] to make RED well tuned under a wider range of conditions. A-RED tries to adapt to changing traffic load by slowly adjusting max_p as shown in Figure 2 (b). A-RED tries to dynamically configure itself to a well-configured state by defining a target region for the average queue within min_{th} and max_{th}. A-RED seeks the average queue target region by additively increasing max_p up to a limit (0.5 in default) if the average queue size goes above the region and multiplicatively decreasing max_p down to a limit (0.001 by default) in case the average queue size goes below the region. In short, A-RED tries to find a slope for the dropping probability that can intersect the queue law curve to make the feedback control system stable for current traffic load. However, A-RED still does not guarantee that it will find a slope within the range given by the limit for the max_p, in which case an unstable queue oscillation will take place as the case of

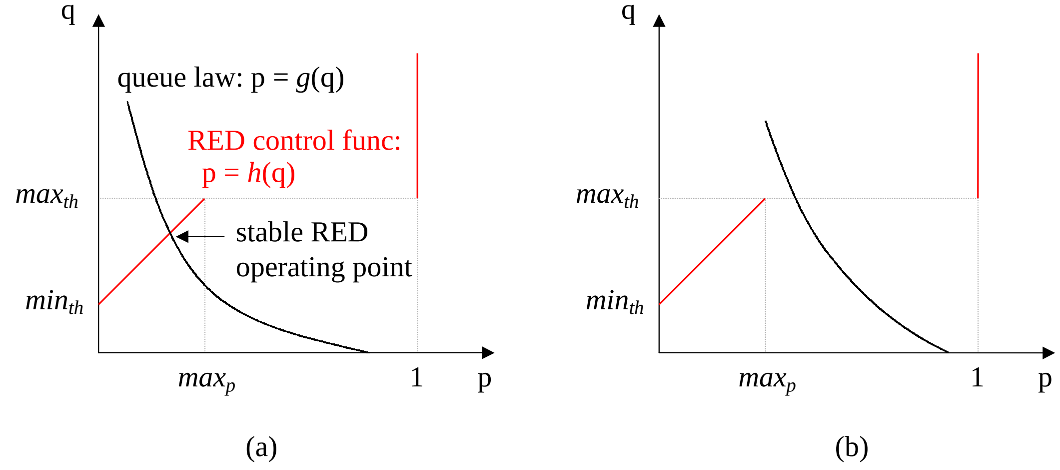

Fig. 1. Well-configured RED (a) and poorly configured RED (b)

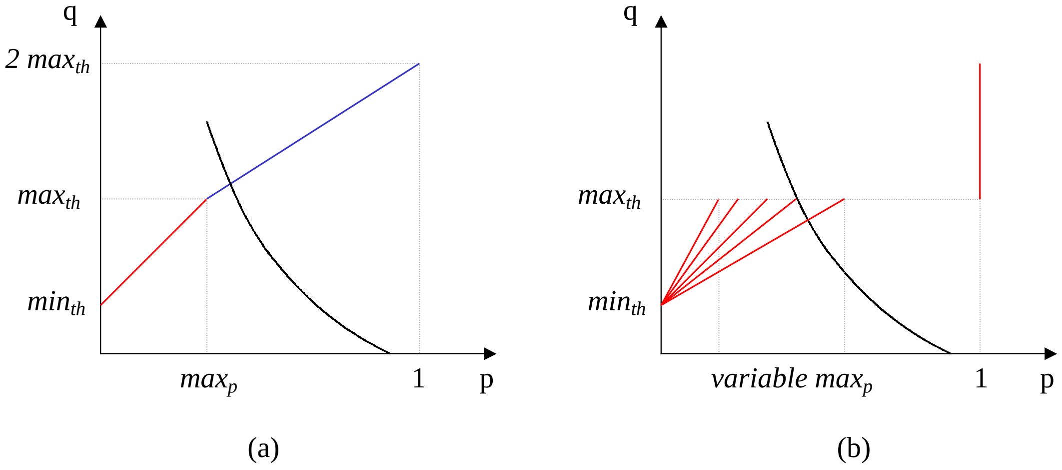

Fig. 2. RED in Gentle Mode (a) and Adaptive RED (b)

original RED. For this reasons, it is also recommended to use the "gentle" setting with A-RED.

Although need thorough evaluation, which is partially done in this work in Section V, A-RED seems to be a novel solution to the RED configuration issue. Yet, there still are concerns that RED family AQM may not be practical since the performance gain over drop-tail queue management is little significant given the complexity of implementation [2], [3]. This paper confirms this argument by showing that even "well-configured" RED family AQM mechanisms yield a higher packet loss rate (or lower goodput) to achieve a lower average queuing delay than drop-tail under fairly heavy traffic loads. Although it is attractive that RED and its variants give us control over average queuing delay, especially when considering QoS for interactive multimedia applications, improving goodput is critical, since it is a measure of how efficiently network resources are used without wasting bandwidth.

While RED family AQM only tradeoff one network performance for another, the potential performance gain of RED family AQM is magnificent when using ECN marking rather than packet dropping. This is because ECN marking brings down the cost of congestion notification in terms of packet loss rate to zero. Thus, RED-ECN can offer a very small packet loss rate as well as a low average queuing delay while achieving high link utilization, if RED-ECN can stay "well-configured" for a relatively wide range of traffic loads. This paper seeks to show and demonstrate that Adaptive RED [4] using ECN (or A-RED-ECN) can be "well-configured" for a wide range of traffic mixes, achieving significant performance gains over drop-tail queue management.

In Section II, we develop a simple model for load on the router, apply it to TCP traffic and derive general queue law that works both for systems that use packet dropping and ECN marking congestion notification. Then, we verify the general queue law through simulations with ideal long-lived TCP flows.

In Section III, we illustrate the relationship between average queue size and other parameters of queue law such as number of TCP flows, service rate (or link bandwidth), and round trip link delay to help understanding the impact of TCP traffic on the router queue behavior. Then, we extend our discussion to the effect of short-lived flows on end-to-end congestion control.

In Section IV, we compare and contrast the queue law for packet dropping and ECN marking system, and discuss key router configuration characteristics in order to be well-configured in the presence of ECN traffic. We show that an ECN enabled router may improve average queue oscillation with presence of ECN traffic, and provide support for recommendation that RED family AQM mechanisms (or AQM in general) should apply a much higher marking rate for ECN traffic than for TCP traffic [9].

In Section V, this paper determines a set of RED and RED-ECN configurations that illustrate the behavior of RED, RED-ECN and their variants well using queue law, and measure the performance of RED, gentle-RED, A-RED, RED-ECN, gentle-RED-ECN and A-RED-ECN over a continuum of TCP traffic loads. We compare the performance of the RED family AQMs with one another in terms of packet loss rate, delay and queue oscillation, also compared with that of drop-tail queue management.

In Section VI, this paper concludes that RED family AQMs, particularly Adaptive RED using ECN (A-RED-ECN), can, indeed, be "well-configured" for variety of TCP traffic mixes, achieving both a very low network packet drop rate and a low queuing delay, which can never achieved with drop-tail queue management alone.

II. LOAD AT ROUTERS AND QUEUE LAW

Assuming a TCP only network with one congested router that uniformly notify traffic sources of congestion with a probability, there exists a relationship among average queue size (q), congestion notification probability (p) and service rate (SR) of the router, and TCP traffic parameters such as the number of flows (N) and average round trip link delay ($RTLD$). This relationship is referred to as "queue law" and introduced in [8] for system that uses packet drops for congestion notification. Queue law can be used to estimate a router's congestion notification probability that will give a targeted average queue size (or vice versa) for a given TCP traffic mix, and is useful when configuring a RED router.

In this section, we develop a simple model of traffic load at router in 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. The new queue law distinguishes and takes packet dropping notification probability (p_d) and ECN marking notification probability (p_m) separately, where the total congestion notification probability at the router $p = p_d + p_m$. We validate the correctness of the general queue law through simulation. Note that the system to model assumes a single congested router that applies p uniformly to incoming packets to notify congestion.

In general, traffic load at a router queue (L) can be expressed as the ratio of the packet arriving rate (AR) over the service rate (SR) that is usually the bandwidth of the outgoing link. When the router is in persistent congestion, the load minus dropping probability is 1 ($L - p_d = 1$) to make the system stable. Applying this stable load equation to TCP only traffic mix, AR can be expressed in terms of the number of TCP flows (N), average TCP window size and average round trip time (RTT) of all flows traveling though the router, where RTT can be further decomposed of queuing delay (q) at the congested router and average round trip link delay ($RTLD$). Thus,

$$\begin{aligned} L - p_d &= \frac{AR}{SR} - p_d \\ &= \frac{N \times avg\_tcp\_wind(p_d + p_m)}{RTT \times SR} - p_d \\ &= \frac{N \times avg\_tcp\_wind(p_d + p_m)}{(RTLD + q/SR) \times SR} - p_d \\ &= 1 \end{aligned} \tag{1}$$

Note in Equation 1 that the average TCP window size for average flows traveling through the router is the function of only p_d and p_m (or p in general), which is shown in TCP throughput model [10] and verified later in this section through simulation. Re-writing Equation 1 for q, we get the general queue law which implies that q of the congested router is linearly proportional to N, negative linearly 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:

$$q = \frac{N \times avg\_tcp\_wind(p_d + p_m)}{p_d + 1} - RTLD \times SR \tag{2}$$

To validate the new queue law, we ran a series of NS [11] simulations using the network setup shown in Figure 3, and compare the results with the corresponding theoretical queue law curves. For the TCP window model for the new queue law, we used one from [10], which models after ideal long-lived TCP sources that have no congestion window limit (*cwnd_limit*) nor receiver window limit, and always have data to transmit. Therefore, for

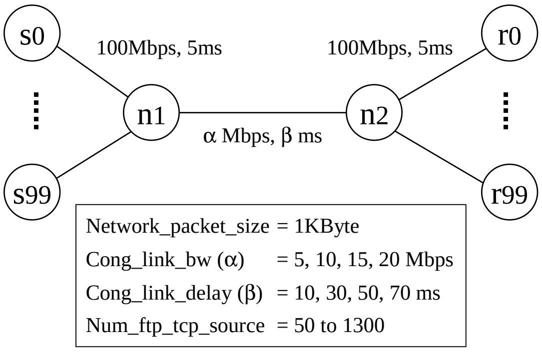

Fig. 3. Simulation Network Setup.

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 an infinite queue that randomly drops or ECN mark incoming packets with a given fixed congestion notification probability.

Figure 4 shows simulated and theoretical queue law curves for both packet dropping and ECN marking systems for 100, 200 and 300 ideal TCP flows while fixing SR (congested link bandwidth) to 20Mbps, RTLD to 80ms. It shows that the new queue law predicts the average queue size for given congestion notification probability very well for ECN marking systems. However, the prediction is not as accurate for the packet dropping systems as for the ECN marking systems, and the precision decreases as p_d increases. We believe that this is because the TCP model used does not accurately model the TCP window behavior for packet drop congestion notification, particularly for TCP fast retransmission timeout behavior.

In this section, we developed and verified general queue law that models the average queue behavior of a congested router well for both drop and mark. In the next section, we discuss the impact of TCP traffic on congested router queue using the general queue law.

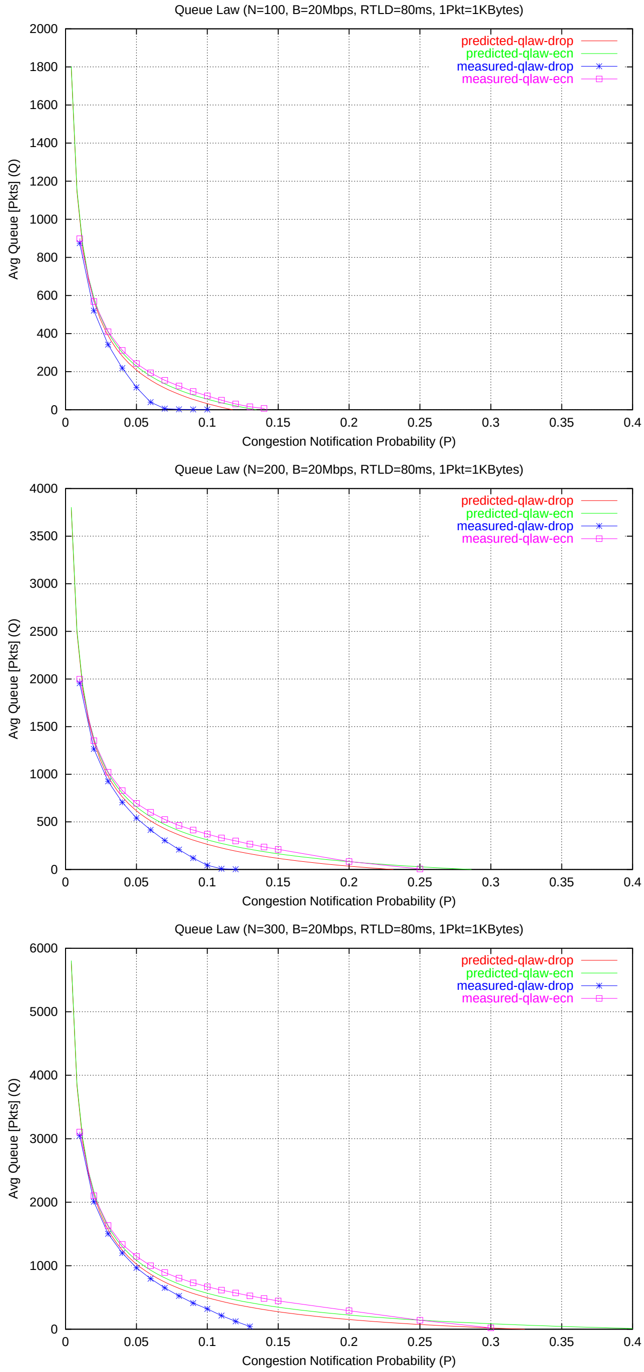

Fig. 4. Queue Law: Theory and Simulation Results Comparison

III. ANALYSIS OF QUEUE LAW

To further help understanding the impact of TCP traffic on the router queue, Figure 5 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 these illustrations.

Figure 5 (top) re-displays the measured queue law curves for packet dropping system in Figure 4 in one graph (simulation settings differ only by the number of TCP connections: 100, 200 or 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 flows.

Figure 5 (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 5 (bottom) shows the queue law of the congested router for simulations that differ only by the link band-

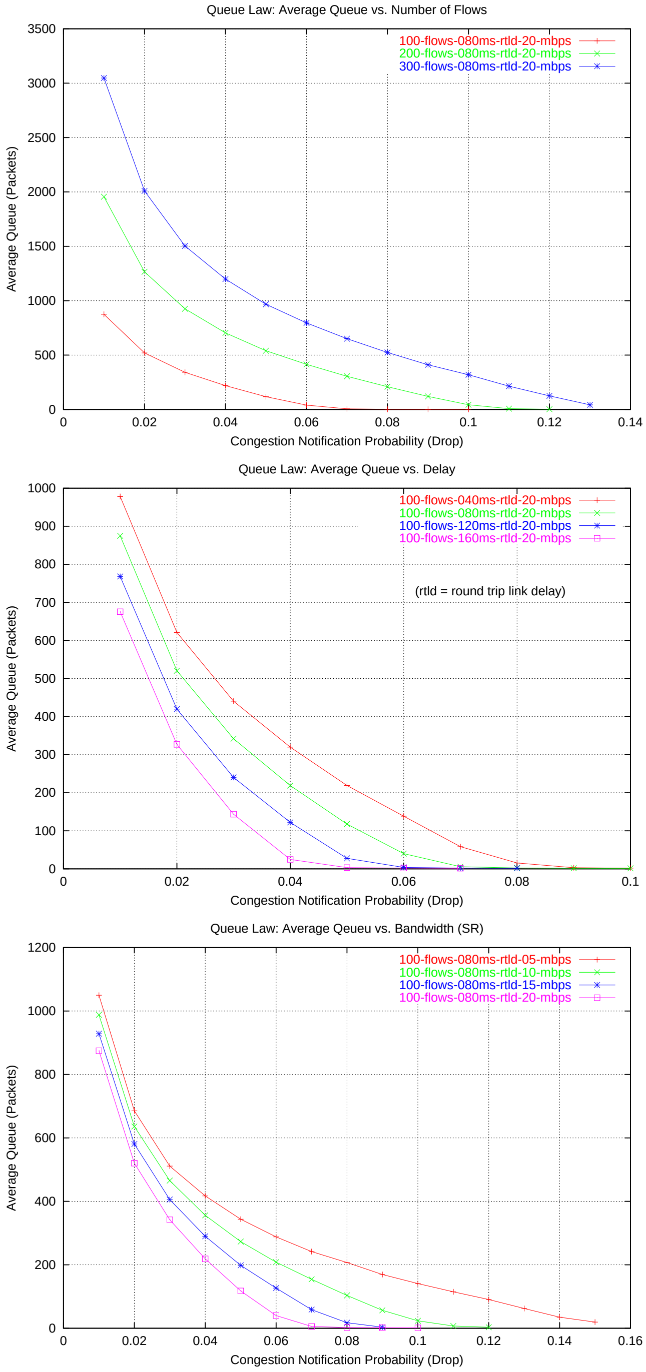

Fig. 5. Queue Law: q vs. N (top), $RTLD$ (middle) and SR (bottom)

width: 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.

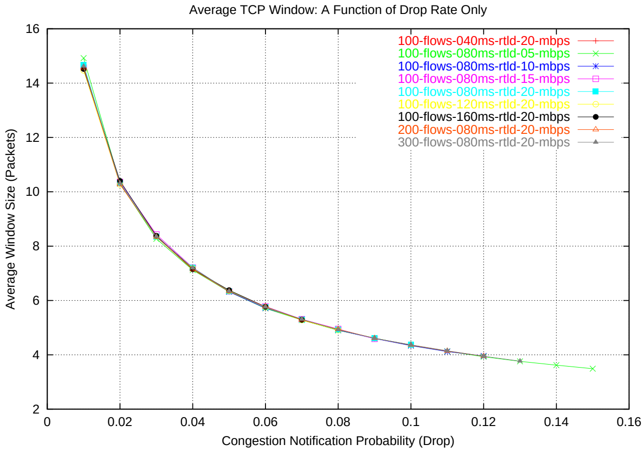

Fig. 6. Average Window Size of All TCP Flows (for Drop)

So far, queue law was examined with ideal long-lived TCP flows in which the size of congestion window (or receiver window) is unlimited and the traffic sources have an infinite amount of data to transmit. In this case, as shown in Figure 6, the average TCP window is the function of congestion notification (drop/mark) probability only and behaves identical for different the network and traffic configurations. However, in real networking environments where factors such as the TCP congestion window (or receiver window) limit and data object size impose different window operation limits and alter the congestion response behavior of TCP flows, the average TCP window may not behaves same from one TCP 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 in average will often be less than in the case of unlimited thresholds given the same drop/mark rate, especially for a low drop/mark rates.

We illustrate the effect of limited TCP window growth by setting the TCP congestion window limits to a low value. The congestion window sizes for all TCP sources are set 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 previous unlimited congestion window simulation that had 300 TCP connections. Figure 7 shows the average TCP window behavior, and Figure 8 shows the corresponding queue law curve.

Figure 7 shows that as the congestion window (*cwnd*) limit decreases, the average congestion window curve flattens. This means that small average window limits make TCP connections much less responsive, especially for changes in relatively low drop rates. For the simulations with the TCP sources limited to a *cwnd* of 6 packets, we had to approximately double the number of TCP flows to achieve an equivalent queue average at a drop rate of 0.01. Thus, a router that is congested with many of short Web flows will need to apply a relatively high drop rate to keep the average queue length within a certain range since

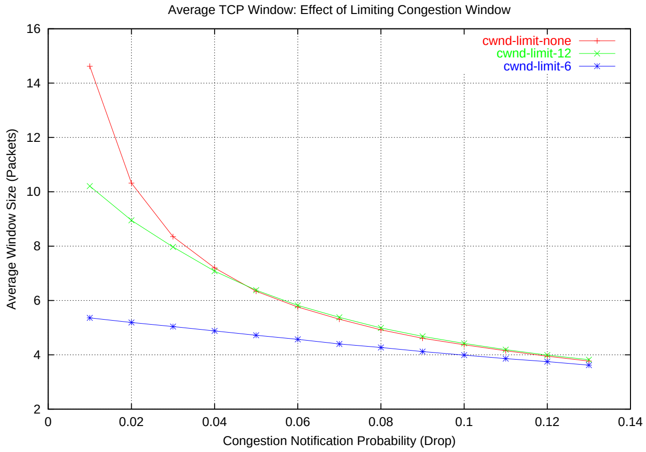

Fig. 7. Average Window: Effect of Limiting TCP Congestion Window

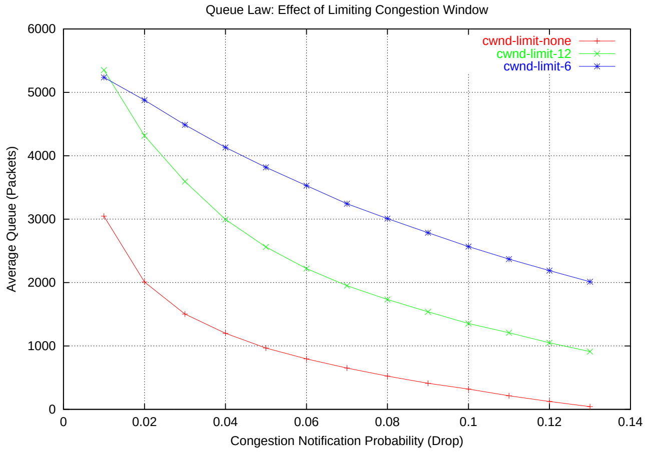

Fig. 8. Queue Law: Effect of Limiting TCP Congestion Window (Drop)

the short TCP flows are less responsive. For example, the simulation with smaller *cwnd*s (6 packet limit) has to apply about twice as high a drop rate as the simulation with the larger *cwnd*s (12 packet limit) to maintain an average queue length of 3000 packets.

From the analysis on queue law, one may see that a router can compute an optimal congestion notification probability and better manage it's queue in congestion if it is informed of N, $RTLD$, SR, and average TCP window size. Usually, SR can be known without any price as it's the bandwidth of the outgoing link. However, counting N or obtaining $RTLD$ (or RTT) and TCP window size from traffic sources require a price and network structure change. Recently, studies suggest that DiffServ [12] architecture has potentials to obtain these useful information in a relatively cheap price using edge-core router architecture. However, how to collect, distribute and utilize these information securely and effectively is little known and requires more study.

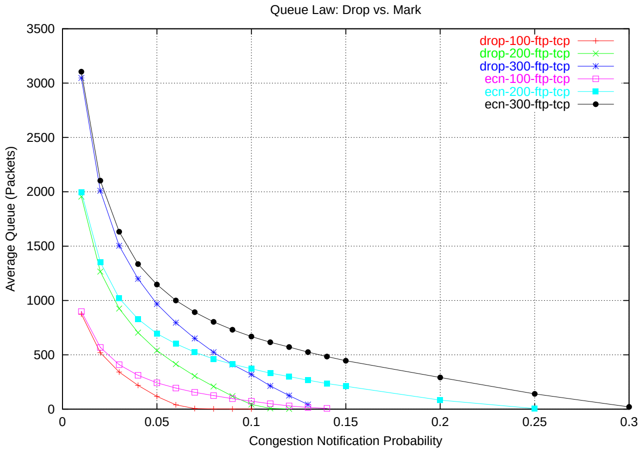

Fig. 9. Queue Law: Drop vs. Mark

IV. FEEDBACK METHOD: DROP VS. MARK

In previous sections, we derived a general queue law and illustrated the impact of TCP traffic on congested router queue behavior. In this section, by comparing and contrasting the ECN queue law and 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.

Figure 9 re-displays the measured queue law curves in Figure 4 in a graph to compare the queue laws for drop and mark notification systems. For the same number of flows, the average queue lengths for TCP and TCP with ECN are almost the same when the congestion notification probability at the router is low. However, as the notification probability is increased, the average queue length of the queue with TCP with ECN decreases noticeably slower and steadier 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.

Second, for a reasonable average queue length target (for example, 500 packets in Figure 9), as traffic load increases linearly, the difference between the stable state mark rate and the stable state drop rate to maintain the queue length at the same level increases exponentially, which indicates that ECN should increase its mark rate exponentially above any drop rate. However, the queue law for ECN converges 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 at a reasonable low target even for a highly loaded situation.

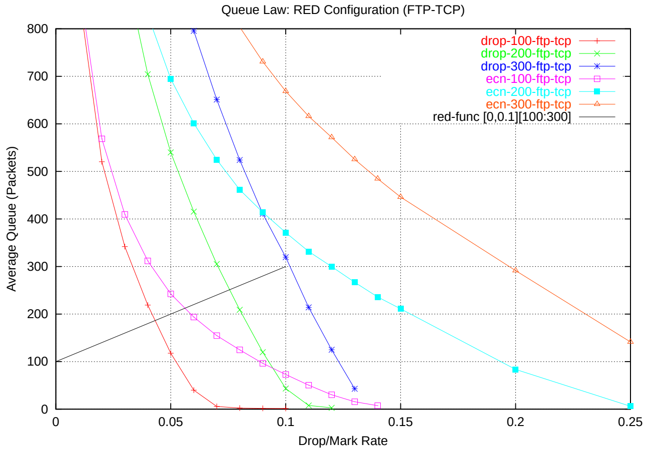

Fig. 10. Queue Law: RED Configuration

.

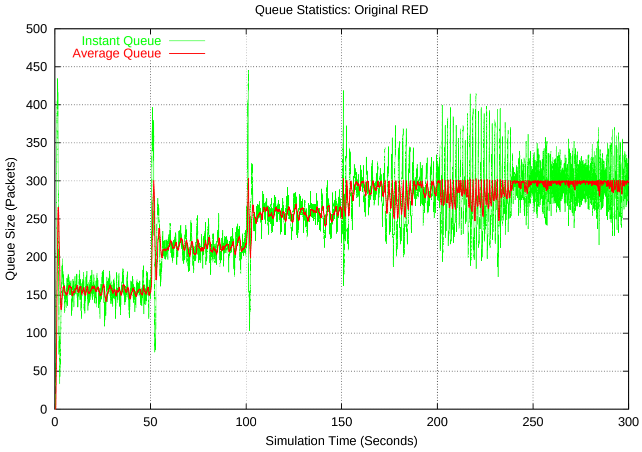

Fig. 11. Queue Statistics: RED

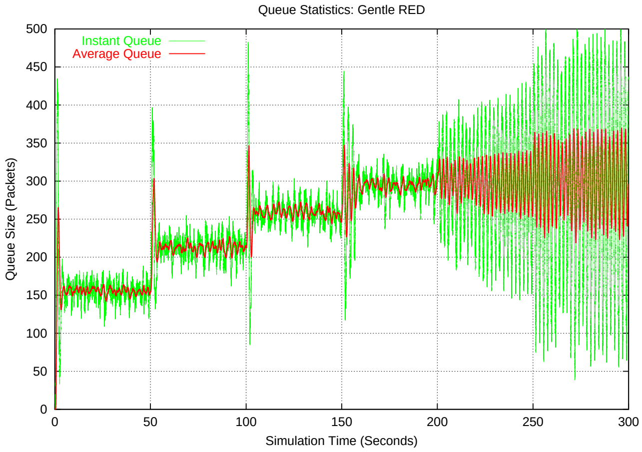

Fig. 12. Queue Statistics: Gentle RED

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 curve 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. We illustrate this further in Section V.

V. 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 packet dropping and ECN marking systems in the previous section, and verifies the effectiveness of queue law in characterizing RED performance. At the same time this section compares the performance of RED family AQMs with one another and also with that of drop-tail queue management in terms of throughput and packet loss rate 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 3 setting the congested link bandwidth to 20 Mbps and the round trip time link delay to 80 ms. Each simulation starts with 50 FTP-TCP flows, with 50 more FTP-TCP flows added every 50 seconds. The physical queue length is set to 500 packets, with the packet size set to 1 Kbyte. For RED parameter settings, max_p is set to 0.1, min_{th} is 100 packets, and max_{th} is 300 packets, based on recommendations [13]. Although not shown is Figure 10, 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 the given TCP traffic loads.

In general, comparing the queue behavior of each RED family AQM with the queue law shown in Figure 10, demonstrates that the queue law indeed works very well predicting RED behavior. For RED, the queue law indicates that RED will be stably manage TCP traffic up to about 200 flows. In Figure 11, 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 12, was able to manage load up to 200 flows since there no longer a sudden increase in drop probabilities from the max_p 0.1 to 1 at max_{th}. For RED-ECN, shown in Figure 14, 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 Figure15, 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 some-

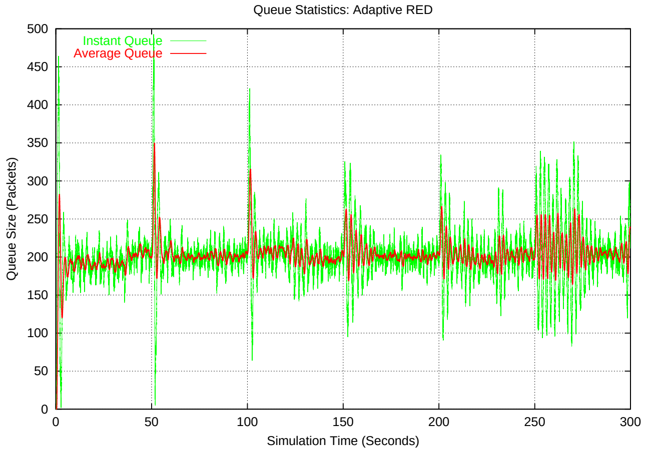

Fig. 13. Queue Statistics: Adaptive RED

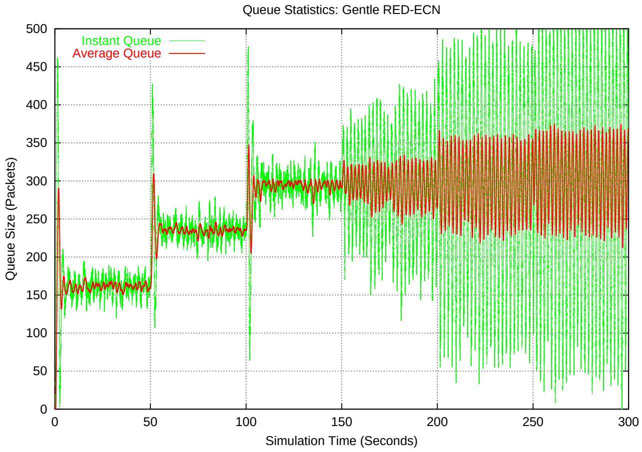

Fig. 15. Queue Statistics: Gentle RED-ECN

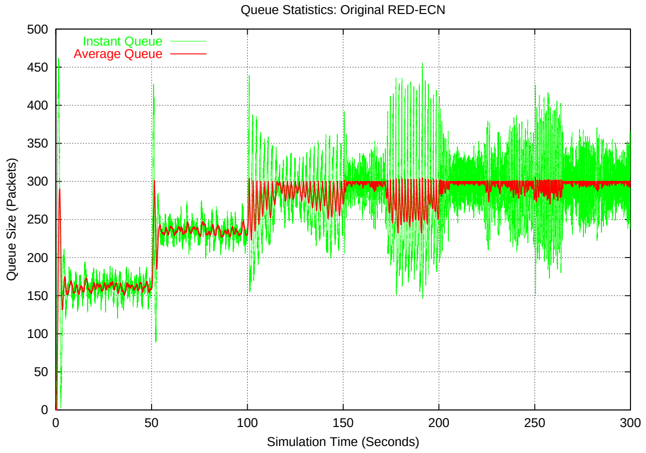

Fig. 14. Queue Statistics: RED-ECN

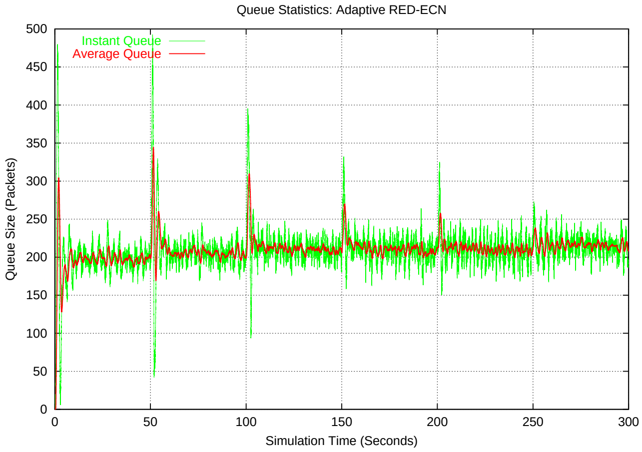

Fig. 16. Queue Statistics: Adaptive RED-ECN

what, as shown in Figure 18.

We believe that the stiffness of the “gentle” RED control function ($q > max_{th}$) causes the unstable q 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 stiff. When the stable RED operating q exists 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 shortly. This process repeats causing a large and unstable q oscillation.

Comparing the queue behavior of Adaptive RED, shown in Figure 13, and Adaptive-RED-ECN, shown in Figure 16, 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 very wide range of TCP traffic.

We next analyze the delay-loss tradeoffs between drop-tail and RED. Starting with link utilization, Figure 17 shows that the bottleneck link was fully utilized for all TCP traffic loads and thus goodput is affected by packet loss rate only in our simulations. Figure 18 shows the packet loss rates at the routers, which suggests that all the RED family queue mechanisms that use drops for congestion notification have consistently higher packet loss rate than does drop-tail queue management. Drop-tail does not actively drop packets, so the drop distribution that results from buffer overflow at a drop-tail queue may be bursty. 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 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 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 price for AQM design is too high compared with the potential gain of a lower average queue size [2], [3]. However, even with the required higher ECN

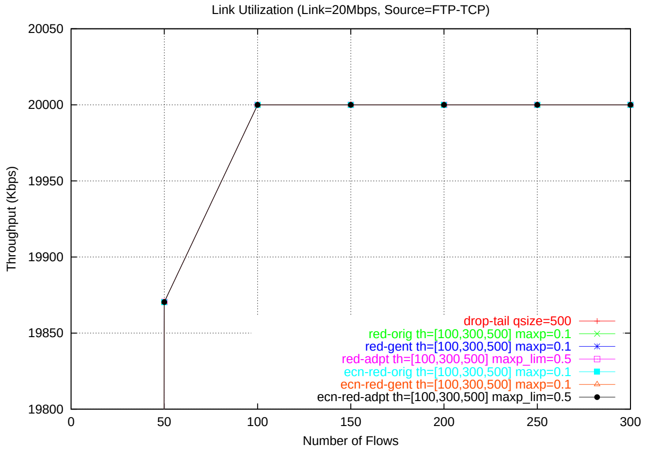

Fig. 17. Link Utilization

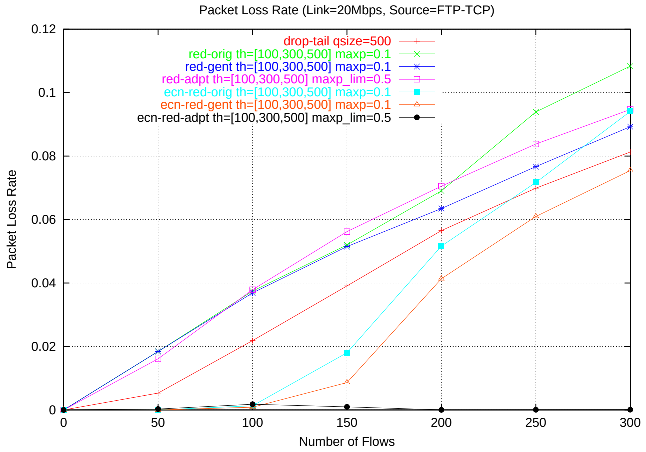

Fig. 18. Packet Loss Rate

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 18 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. In addition, as mentioned in Section IV, 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 one remains more stable even at a high traffic load compare to the one that does not not use ECN.

VI. SUMMARY

In this paper, we developed a model for load on router in congestion, applied it to TCP traffic, derive general queue law that works both for drop and ECN mark notification, and illustrated the impact of TCP traffic on congested router queue behavior. We showed that the 3 TCP traffic parameters that affect the behavior of router queue in congestion are the number of flows (N), average round trip link delay ($RTLD$) and average TCP window size.

Also, by comparing and contrasting the ECN queue law and drop queue law, we discussed 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 enabled routers should apply a significantly higher marking rate than RED routers in order to operate with a reasonably low queuing delay. In addition, we found that ECN traffic may help routers in congestion stabilizing average queue oscillation.

Then, we configured RED family AQMs using queue law, compared the performance of the RED family AQMs with one another and drop-tail queue management in terms of packet loss rate, delay and queue oscillation, and demonstrated the trade-offs between drop-tail queue management and RED family AQMs. This paper concludes that RED family AQMs, particularly Adaptive RED using ECN, can, indeed, be well-configured for variety of TCP traffic mixes, achieving both a very 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 of to a mixture of ECN and non-ECN TCP flows. In addition, we intend to build an adaptive AQM technique that makes use of queue law to more quickly adapt to a well-configured state in the presences of changing network load.

ACKNOWLEDGMENTS

Authors would like to thank to professor Robert Kinicki of WPI CS department for reviewing this paper.

REFERENCES

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

[2] M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED," in *Proceedings of the IEEE/IFIP International Workshop on Quality of Service (IWQoS'99)*, June 1999.

[3] 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.

[4] 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.

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

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

[7] V. Rosolen, O. Bonaventure, and G. Leduc, “A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic,” *ACM Computer Communication Review*, Jul 1999.

[8] V. Firoiu and M. Borden, “A Study of Active Queue Management for Congestion Control,” in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Mar 2000.

[9] Z. Zheng and R. Kinicki, “Adaptive Explicit Congestion Notification (AECN),” in *Proceedings of Seventh IEEE Symposium on Computers and Communications (ISCC02)*, Jul 2002.

[10] 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.

[11] VINT, “Virtual InterNetwork Testbed, A Collaboration among USC/ISI, Xerox PARC, LBNL, and UCB,” http://www.isi.edu/nsnam/vint/.

[12] Y. Bernet, J. Binder, S. Blake, M. Carlson, B. E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss, “A Framework for Differentiated Services,” Internet Draft draft-ietf-diffserv-framework-02.txt, IETF, February 1999.

[13] S. Floyd, “RED: Discussions of Setting Parameters,” http://www.icir.org/floyd/REDparameters.txt.