Mathematical and Computational Applications, Vol. 14, No. 1, pp. 65-72, 2009.

AN ACTIVE QUEUE MANAGEMENT ALGORITHM FOR REDUCING PACKET LOSS RATE

Babek Abbasov
Department of Computer Engineering
Qafqaz University, Baku, Azerbaijan
babek@qafqaz.edu.az

Serdar Korukoğlu
Department of Computer Engineering
Ege University, 35100
Bornova, İzmir, Turkey
serdar.korukoglu@ege.edu.tr

Abstract-Among various active queue management schemes (AQM), random early detection (RED) is probably the most extensively studied. Unlike the existing RED enhancement schemes, we use hazard rate estimated packet dropping function in RED. We call this new scheme HERED. The underlying idea is that, with the proposed packet dropping function, packet dropping becomes gentler than RED at light traffic load but more aggressive at heavy load. Simulations demonstrate that HERED achieves a higher and more stable throughput and performs better than current active queue management algorithms due to the lowest packet drops.

Keywords- Active Queue Management, Congestion, Random Early Marking, Hazard Function.

1. INTRODUCTION

Congestion control is one of the most important problems in the Internet. Most of the existing Internet routers play a passive role in congestion control, and are known as droptail routers. A droptail router discards packets when its FIFO queue is full. It was shown in Zhang et al. [1] that under heavy load conditions, droptail routers cause global synchronization, a phenomenon in which all senders sharing the same bottleneck router/link shut down their transmission windows at almost the same time, thereby causing a sharp drop in the bottleneck link utilization. It was also found in Mahajan et al. [2] that droptail routers are biased against bursty sources. This is because, when a burst of packets from a sender arrives at a fully occupied queue, a sustained packet drop within the same window of data occurs. Research in Floyd et al. [3] and Xu and Ansari [4] showed that the dominant transport layer protocol, transmission control protocol TCP [5], lacks the ability to recover from such multiple packet losses within the same window of data. Therefore, the TCP sender has to rely on retransmission timeouts to recover the lost packets. Retransmission timeout significantly slows down the transmission rate of a TCP flow, because almost no data will be sent when the sender waits for the retransmission timer to expire. A good congestion control scheme should therefore avoid triggering unnecessary timeouts.

On the other hand, as more and more multimedia applications running on top of UDP are being deployed, the traditional approach of relying solely on TCP's end-to-end congestion control algorithms will no longer be viable [6]. The network, in particular the routers in the network, should play an active role in its resource allocation, so as to effectively control/prevent congestion. This is known as active queue management (AQM) [7]. The essence is that an AQM router may intelligently drop packets before the queue overflows.

Among various AQM schemes, Random Early Detection (RED) [8] is probably the most extensively studied. RED is shown to effectively tackle both the global synchronization problem and the problem of bias against bursty sources. Due to its popularity, RED or its variants has been implemented by many router vendors in their products. For example Cisco implemented using weighted random early detection (WRED). On the other hand, there is still a hot on-going debate on the performance of RED. Some researchers claimed that RED appears to provide no clear advantage over droptail mechanism [9]. But more researchers ([2 10]) acknowledged that RED shows some advantages over droptail routers but it is not perfect, mainly due to one or more of the following problems.

- RED performance is highly sensitive to its parameter settings ([10]). In RED, at least 4 parameters, namely, maximum threshold (max_{th}), minimum threshold (min_{th}), maximum packet dropping probability (max_p), and weighting factor (w_q), have to be properly set.
- RED performance is sensitive to the number of competing sources/flows.
- RED performance is sensitive to the packet size.
- With RED, wild queue oscillation is observed when the traffic load changes.

As a result, RED has been extended and enhanced in many different ways ([2 10]). It can be found that a common underlying technique adopted in most studies is to steer a router to operate around a fixed target queue size (which can either be an average queue size or an instantaneous queue size). There are some concerns on the suitability of this approach, since the schemes thus designed are usually more complicated than the original RED. This renders them unsuitable for backbone routers where efficient implementation is of primary concern. In some schemes, additional parameters are also introduced. This adds extra complexity to the task of parameter setting.

Unlike the existing RED enhancement schemes, we propose to simply replace the hazard rate estimated packet dropping function in RED. The rest of the original RED remains unchanged. We call this new scheme HERED. The underlying idea is that, with the proposed nonlinear packet dropping function, packet dropping is gentler than RED at light traffic load but more aggressive at heavy load. Therefore, at light traffic load NLRED encourages the router to operate in a range of average queue sizes rather than a fixed one. When the load is heavy and the average queue size approaches the maximum threshold max_{th} – an indicator that the queue size may soon get out of control, HERED allows more aggressive packet dropping to quickly back off from it. Simulations demonstrate that NLRED achieves a higher and more stable throughput than RED and other efficient variants of RED. Since HERED is fully compatible with RED, we can easily upgrade/replace the existing RED implementations by HERED.

2. THE HAZARD FUNCTION

The hazard function is the conditional density function of failure at time *t*, given that the unit has survived until time *t*. Therefore, letting *X* denote the random variable and *x* denote the realization,

$$\begin{aligned} f(x|X \geq x) &= h(x) \\ &= F'(x|X \geq x) \\ &= \lim_{\Delta x \to 0} \frac{F(x+\Delta x|X \geq x) - F(x|X \geq x)}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{F(x \leq X \leq x+\Delta x|X \geq x)}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{F(x \leq X \leq x+\Delta x, X \geq x)}{\Delta x P\{X \geq x\}} \\ &= \lim_{\Delta x \to 0} \frac{F(x \leq X \leq x+\Delta x)}{\Delta x[1-F(x)]} \\ &= \frac{f(x)}{1-F(x)} \end{aligned} \tag{1}$$

It turns out that specifying a hazard function completely determines the cumulative distribution function (and vice-versa).

2.1. The Failure Rate for the Weibull Distribution

For the Weibull distribution, the hazard function is

$$h(x) = \frac{f(x)}{1-F(x)} = \frac{(\beta/\theta)(x/\theta)^{\beta-1} e^{-(x/\theta)^{\beta}}}{e^{-(x/\theta)^{\beta}}} = \frac{\beta}{\theta}\left(\frac{x}{\theta}\right)^{\beta-1} \tag{2}$$

Note that if $\beta = 1$ the Weibull hazard function is constant. This should be no surprise, since for $\beta = 1$ the Weibull distribution reduces to the exponential. When $\beta > 1$, the Weibull hazard function increases, approaching ∞ as $\beta \to \infty$. Consequently, the Weibull is a fairly common choice as a model for components or systems that experience deterioration due to wear-out or fatigue. For the case where $\beta < 1$, the Weibull hazard function decreases, approaching 0 as $\beta \to 0$.

For comparison purposes, note that the hazard function for the gamma distribution with parameters *r* and λ is also constant for the case *r* = 1 (the gamma also reduces to the exponential when *r* = 1). Also, when *r* > 1 the hazard function increases, and when *r* < 1 the hazard function decreases. However, when *r* > 1 the hazard function approaches λ from below, while if *r* < 1 the hazard function approaches λ from above. Therefore, even though the graph of the gamma and Weibull distributions look very similar, and they can both produce reasonable fits to the same sample of data, they clearly have very different characteristics in terms of describing survival or reliability data.

Finally we can simplify hazard function for Weibull distributions as $h(x) = cx^{c-1}$. Figure 1 illustrates h(x) function

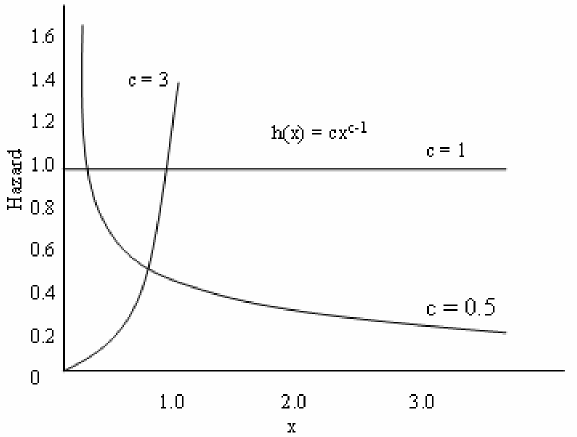

Figure 1. Hazard function for the Weibull distribution

3. RELATED WORKS

Lots of proposed queue management schemes are RED oriented. The difference of these schemes is mainly in the parameter-adjusting method. Their mechanisms for adaptation of dropping-probability in response to network condition are different. In this section, brief descriptions of these queue management schemes are given. These representative queue management schemes include droptail RED [8], BLUE [11], REM [12], SRED [13], and DRED [14].

For droptail, arrival packets are dropped when queue overflow occurs. Droptail incurs large queue length and high packet loss rate at congested links. Especially, droptail results in a phenomenon, called global synchronization [4], when a lot of TCP flows compete in a bottleneck. The total throughput of TCP flows decreases when global synchronization occurs. It is really a simple queue management and does nothing to prevent congestion. However, droptail is the most widespread queue management scheme due to its simplicity.

Random early detection is a queue management scheme that is intended to remedy the shortcomings of droptail. RED implicitly notifies one of sources of congestion by randomly dropping an arriving packet. The selected source is informed by the packet loss and its sending rate is reduced accordingly. Consequently, congestion is alleviated.

The dropping probability of RED is decided by the queue length. An arriving packet may be dropped before the queue is full. It is an early congestion notification. The dropping probability is a function of average queue length. When the queue occupancy grows, congestion builds up. Then, the dropping probability increases in order to provide enough early congestion notifications.

The RED dropping probability function is a piece-wise linear function. It is defined by a triplet ($min_{th}, max_{th}, p_{max}$). Upon a packet arrival, the packet is admitted into the queue if the average queue length q_a is less than min_{th}, or is dropped if larger than max_{th}. If q_a is between min_{th} and max_{th}, the arrival packet is randomly dropped with the probability defined by a linear function evaluated at q_a. However, queue length is not a good indicator of severity of congestion and issues of congestion notifications may be too bursty, leading to excessive loss, or too mild, failing to stop congestion building. Several queue management schemes, such as BLUE, REM, SRED, and DRED, were proposed to improve the performance of RED.

BLUE uses queue length and link utilization as indicators of traffic load and congestion. Drop rate is adjusted by the indicators. If the current queue size exceeds a predetermined threshold L, dropping probability is raised by a very small fixed amount d_1. Conversely, if the link is idle, dropping probability is reduced by a small fixed amount d_2.The parameters d_1 and d_2 are configurable for BLUE. To avoid aggressive packet dropping, BLUE keeps a minimum interval, called "*freeze time*", between two successive updates.

However, if dominant round-trip time (RTT) of flows or number of flows through the queue changes, the parameter settings may not be suitable and queue length oscillates.

To deploy Random Exponential Marking (REM), routers must perform exponential marking and sources must be REM aware. REM gives no incentive to cooperative sources. Furthermore, a properly calculated and fixed design control constant must be known globally.

Stabilized RED estimates number of flows and adapts dropping probability accordingly. The estimation is mainly based on a zombie list which first initializes the Hit parameter equal to 0 and creates an empty list. Source–destination addresses of each incoming packet are stored in the list. Once the list is full, an entry is randomly chosen from the list for each subsequent packet arrival, and its content is compared with those of the newcomer packet. If there is a match, Hit is set to 1. Otherwise, Hit is set to 0, and with a certain probability, the content of this entry may be replaced by the source–destination addresses of the newcomer packet. By this match–mismatch way, an approximate number of flows are estimated. However, the estimation converges very slowly and is heavily dependent on the size of zombie list. The estimated number of flows in SRED is inaccurate in most of time in a varied network environment.

4. HERED ALGORITHM

We make some major changes to the RED algorithm and call the new algorithm HERED. An important advantage of our algorithm HERED is that one can change the probability value p via the use of the hazard function $h(x) = cx^{c-1}$. Let T and q_{len} denote the target value and the queue length, respectively. In order to stabilize the queue length against high congestion levels, we use c=3, c=2 and again c=3 in the hazard function for the cases where $min_{th} < q_{\text{len}} < T$, $T < q_{\text{len}} < max_{th}$ and $max_{th} < q_{\text{len}} < 2*max_{th}$, respectively. The pseudo code for HERED algorithm is illustrated in Figure 2.

```
for each packet arrival
   calculate new prop
   probability $p$ using hazard
   fuction h(x) = $cx^{c-1}$:
$if(qlen < \min_{th})$
   c = 0
   no drop
else $if(\min_{th} \le qlen < T)$
   c = 3
   calculate probability $p_a$:
   with probability $p_a$:
      mark the arriving packet
else $if(T \le qlen < \max_{th})$
   c = 2
   calculate probability $p_a$:
   with probability $p_a$:
      mark the arriving packet
else $if(\max_{th} \le qlen < 2 * \max_{th})$
   c = 3
   calculate probability $p_a$:
   with probability $p_a$:
      mark the arriving packet
else $if(qlen \ge 2 * \max_{th})$
   mark the arriving packet
```

Figure 2. Pseudo code for HERED algorithm.

5. SIMULATION RESULTS

In this section, we compare the simulation results of our proposed HERED with the existing queue management schemes, REM and RED.

All simulations are performed using NS-2 simulator. In all of our simulations, we use the topology shown in Figure 3, which consists of n senders and one sink, connected together via router N, with one of the queue management schemes. S_n denotes the source of TCP flows. By varying flow number in each source S_n, we produce different levels of traffic load and thus different levels of congestion on the bottleneck link. N has a queue buffer size of 50 packets.

The parameters used for the simulations are presented in Table 1.

In the simulations, 10 to 200 TCP sources are run for 100 seconds from all nodes in (S_1, S_2… Sn) to the destination node D. Figure 4 illustrates Loss Rate of RED, REM and HERED algorithms. It is obvious that HERED has the lowest packet drops.

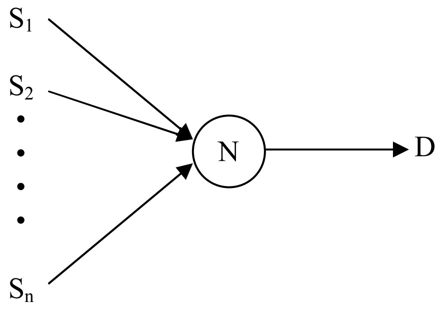

Figure 3. Simulation topology.

Table 1. Simulation Parameters

Algorithm	*Parameter Settings*
RED	min_{th} = 5, max_{th} = 15, (3 × min_{th}), w_q = 0.002, max_p = 1/150, (as recommended in RED [3]), gentle version used
REM	$\phi = 1.001$, $\gamma = 0.001$, $w_q = 0.002$
HERED	min_{th} = 5, max_{th} = 15, (3 × min_{th}), w_q = 0.002, max_p = 1/150

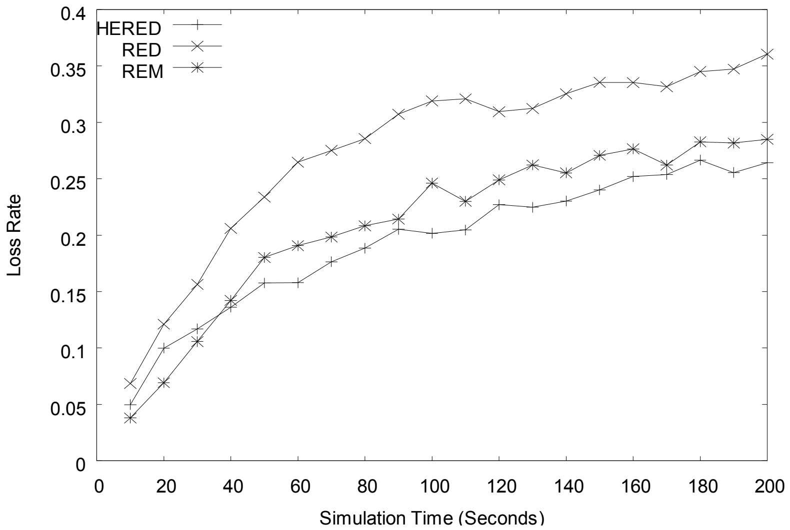

Figure 4. Loss Rate of RED, REM and HERED algorithms.

6. CONCLUSIONS

This paper has demonstrated the active queue management algorithms that use different complex algorithms. In this study, a new active queue management algorithm called HERED has been designed and evaluated. HERED performed better than current active queue management algorithms due to the lowest packet drops.

REFERENCES

1. L. Zhang, S. Shenker, and D.D. Clark, Observations on the dynamics of congestion control algorithm: the effects of two-way traffic, *Proceedings of the Conference on Communications Architecture &Protocols*, 133-147, Zurich, Switzerland, 1991.
2. R. Mahajan, S. Floyd, and D. Wetherall, Controlling High-Bandwidth Flows at the Congested Router, *Proceedings of the Ninth International Conference on Network Protocols*, 192, November 11-14, 2001
3. S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, An Extension to the Selective Acknowledgement (SACK) Option for TCP, RFC Editor, 2000.
4. K. Xu, and N. Ansari, Stability and fairness of rate estimation based AIAD congestion control in TCP, *IEEE Communications Letters*, 9, 378-380, 2005.
5. Information Science Institute, Transmission Control Protocol, in: RFC793, 1981.
6. S. Floyd and K. Fall, 1999. Promoting the use of end-to-end congestion control in the Internet, *IEEE/ACM Transactions on Networking (TON),* 7(4), 458-472, 1999.
7. B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski and L. Zhang, Recommendations on Queue Management and Congestion Avoidance in the Internet, *RFC Editor*, 1998.
8. S. Floyd, and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking (TON),*1(4),397-413,1993.
9. C. Brandauer, G. Iannaccone, C. Diot, T. Ziegler, S. Fdida, and M. May, Comparison of Tail Drop and Active Queue Management Performance for Bulk-Data and Web-Like Internet Traffic, *ISCC* 122-129,2001.
10. W. Feng, D. Kandlur, D. Saha, and K. Shin, Blue: A new class of queue management algorithms, *U. Michigan CSE-TR*-387-99,1999.
11. W. Feng, D. Kandlur, and D. Saha, A self-configuring RED gateway, *Proceeding of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, (INFOCOM'99), 3, 1320-1328, NY, USA, March 21-25, 1999.
12. S. Athuraliya, SA.H.Low, V.H. Li and Q. Yin, REM: Active queue management, *IEEE Network,* 15(3), 48-53, 2001.
13. T.J. Ott, T.V. Lakshman and L.H.Wong, SRED: Stabilized RED, *in Proc. IEEE INFOCOM'99*, NY, 1346–1355, 1999.
14. J. Aweya, M.Ouellette and D.Y. Montuno, A control theoretic approach to queue management, *Computer Networks J.* 36(2–3) 203–235, 2001.