SREM an Active Queue Management Scheme for High Performance Computer Communication System

[1]Dr. S.N. Awashti, [2]Dr. Anil Kumar, [3]Gautam Kumar Rajput
[1]Ex - H.O.D. of Mathematics & Statistics, S.M. College, Chandausi (U.P.)
[2]Lecturer, Dept. of Mathematics & Statistics, R. S. M. College, Dhampur (U.P.)
[3]Research Scholar, Dept. of Mathematics & Statistics, S. M. College, Chandausi (U.P.)

ABSTRACT

Active queue management scheme used to improve the throughout and to achieve high link utilization with low queuing delay. However in our study we reveal that REM Random exponential marketing has limited improvement such as through put and low delay for heavy load of computer system. The aim of REM was to achieve both high utilization and negligible loss and delay in a simple and scalable manner. The key idea is to decouple congestion measure from performance measure such as loss, queue length or delay while congestion measure indicates excess demand for band with and must track the number of users. Performance measure should be stabilized around their targets independently of the number of users.

KEY WORDS: *AQM, SREM, EREM*

INTRODUCTION

Active Queue Management Schemes (AQM) with explicit congestion notification, which support the end to end congestion control mechanism of computer system by performing congestion control have been actively studied by many researcher. The enormous growth of the internet and increased demand to use the internet for time sensitive voice and video application necessitate the design and utilization of new efficient with effective active queue scheme. Many researchers have been obtained the several AQM such as RED, PRED; PBRED, SRED, PEM and EREM etc. and now are doing some better.

RELATED WORK

The throughput of PED has been designed by [May99]. They have shown that under heavy load, the throughput is inversely proportional to the load. Throughput can be increased by tracking their state of the individual connection or by selecting appropriate parameters for PED [FlJa93]. [Lin99] at proposed fair RED which relies on usage of buffer spaces by the different flows to determine the draft rate of the flow. Although it achieves as a fair drop rate for different flow, it needs to track the state of each flow with result in scalability problems similar to Thone in [Low2000].

To solve the scalability problem of RED [Low2000] and PRED [RaFl99], [Qtt99] proposed SRED, like RED, preemptively discards packets with load dependent drop probability when the buffer in a router gets congested. If however suffer flow low throughput as shown in their simulation results where the normalized throughput is very low even with small number of traffic flows.

[AtLi2001], [AtLo2000] studied the decouple congestion measure from performance measure (loss and queue length/delay) of REM. It the found that it is time consuming process for calculating the marking probability though exponential method and has more dropping probability for higher queue length.

To solve above problem [SaSl2003] proposed EREM, which used fraction method to calculate marking probability and hence it has more throughput and less time to calculate marking probability than REM.

Now we have designed SREM Stabilized Random Exponential Marking which achieves more throughputs and less delay than EREM. It is found that SREM has higher throughput with low delay for heavy load of networks.

RED

A main purpose of active queue management is to provide congestion information for sources to set their rates. The design of active queue management algorithms must answer three questions, assuming packets are probabilistically marked.

1. How is congestion measured?
2. How is the measure embedded in the probability functions?
3. How is it feedback to users?

RED answers there questions as follows.

First RED measures congestions by (exponentially weighted average) queue length. Importantly, the choice of congestion and hence it affects the user utility functions being implicitly optimized by [Low2000] second the probability functions is a price wire linear and increasing functions of the congestion measure, as illustrated in figure. Finally, the congestion information is converged to the users either by dropping a packet or setting as ECN but probabilistically. In fact, RED only decides the first two questions. The third question is largely independent.

RED interacts with TCP as source rates increase, queue length grows, more packet are market, prompting the sources to reduce their rates and the cycle repeat TCP defines precisely how the source rates are adjusted while active queue management defines how the congestion measure is updated. For RED the congestion measure is queue length and it is automatically updated by the buffer process. The queue length is the next period of equals the current queue length plus aggregate input minus output.

$$b_1(t+1) = [b_1(t) + x_1(t) - c_i(t)] \quad (1)$$

Where [z] = max{z,o}

Here $b_i(t)$ is the aggregate queue length at queue 1 in period t, xl(t) is the aggregate input rate to queue etc. in period t, and cl(t) is the output rate in period t.

REM [Random Exponential Marking]

REM differs from RED only in the first two design questions, it used a different definition of congestion measure and different marking probability functions. These differences lead to the two key features mentioned in the last section. Detail derivation and justification. A prejudice implementation and much more extensive simulations can be found in [AtLo2000], [ALLY2000].

MATCH RATE CLEAR BUFFER

The first idea of REM is to stabilize both the input rate around link capacity and the queue around a small target, regardless of the number of users sharing the link.

Each output queue that implements REM maintains a variable. We call 'price' as a congestion measure. This variable is used to determine the marking probability, as explained in the next subsection. Price is updated periodically or around nearly, band on rate mismatch (i.e. difference between input rate and link capacity. The price is incremented if the weighted sum of these mismatches is positive and decremented otherwise. The weighted sum is positive when either in input rate exceeds the link capacity or there is excess backlog to be cleaned and negative otherwise.

When the number of users increases, the mismatches in rate and in queue grow, pushing up price and hence marking probability. This send a stronger congestion signal to the source which then reduce their rates, when the source rates are too small, the mismatch will be negative, pushing down price and marking probability and raining sources rates.

When the sources are too small the mismatches will be negative, pushing down price and marking probability and raining source rates, until eventually, the mismatches are driven to zero, yielding high utilization and negligible loss and delay in equilibrium. The buffer will be cleaned in equilibrium if the target queue is not to zero.

SREM explicitly controls the update of its price to bring about its first property. Precisely for queue l, the price $P_1(t)$ in period t is updated according to -

$$P_1(t+1) = [P_1(t) + r(\alpha_1(k_1(t) - b_1) + x_1(t) - c_1(t)] \quad (2)$$

Where r > 0 and $\alpha_1 > 0$ are small constants

$$[Z^t] = \max\{z,0\}$$

Here k_1 is the aggregate buffer occupancy at queue l in period t and $k_1 > 0$ is target queue length, $x_1(t)$ is the aggregate input rate to queue l in period t, and $c_1(t)$ is the available bandwidth to queue l in period t. The difference $X_1(t)$ - $C_1(t)$ measures rate mismatch and the difference $k_1(t)$ - k_1 measure queue mismatch. The constant α_1, can be set by each queue individually and trade off utilization and queuing delay during transient. The constant r controls the responsiveness of SREM to change in network conditions.

Hence from [LPWV2001] the price is increased if the weighted sum of rate and queue mismatches weighted by α_1 is the positive, and decreased otherwise. In equilibrium, the price stabilizes and this weighted sum must be zero i.e. $\alpha_1(b_1\text{-}b_1)+x_1\text{-}c_1 = 0$.

This can hold only if the input rate equals capacity (x_1=c_1) and the backlog equals its target ($b_1 = b_1$*).

We make two remarks on implementation. First SREM uses only local and aggregate information in particular no per flow information is needed and works with only work conserving service discipline. It updates its price independently of other queues or routers. Hence its complexity is independent of the number of users or the size of the network or its capacity.

Second its is usually easier to sample queue length than rate in practice, when the target queue length b_1* is nonzero, we can bypass the measurement of rate mismatch $x_1(t)$ - $c_1(t)$ in the price update (1). Notice that $x_1(t)$ - $c_1(t)$ is the rate at which the queue length occurs, when the buffer is nonempty.

Hence we can approximate this term by the change in backlog $b_1(t+1)$ - $b_1(t)$. Then the update rule (1)

$$P_1(t+1) = [P_1(t) + r(b_1(t+1) - (1\text{-}\alpha)b_1(t) - \alpha b)] \quad (3)$$

That is the price is updated based only on the current and previous queue length. The update rule (3) were queue length to update a price at which is then used to determine the marketing probability. Hence under SREM, the price stability increases while the mean queue length is stabilized around the target b_1, as the number of users increase

PRICES

The second ideas of SREM is to use the sum of the link prices along a path as a measure of congestion in the path and to embed it into the end to end marking probability that can be observed at the source.

The output queue marks each arrival packet that is not already marked at an upstream queue, with a probability that is exponentially increasing in the current price. This making probability is illustrated in figure (1). The exponential from of the marking probability in critical in a large network where end to end marking for a packet that transverse multiple congested links from source to destination depends on the link marking probability at every link in the path when and only when individual link marking probability is exponential in its link price, this end to end marking probability, it can be easily estimated by sources from the fraction of their own packets that are marked and used to design their rate adoption.

Suppose a packet transverse links l = 1, 2-L that have prices $P_1(t)$ in period t. Then the marking probability $M_1(t)$ at queue l in period t is

$$m_1(t) = 1 - \phi^{-P_1(t)}$$

(4)

When $\phi > 1$ is a constant, the end to end marking probability for the packet then -

$$1 - \underset{e=1}{\overset{L}{\pi}} (1 - M_1(t)) = 1 - \phi^{-\sum_e P_e(t)}$$

(5)

RULES AND SETS

In this rule we calculate the dropping probability behavior based on the average on the average queue size. The dynamic way of calculating the dropping probability comes from the fact that according to the rate of change of queues, the current buffer size belong a different set of rules applies. Based on there rules the dropping probability calculated.

The main objective of the SREM is to reduce the dropping of the packets and at the same time to increase the throughput. In this paper we have developed new rules called SREM. As shown figure (2) the classification is made as minimum M_1 low medium (Me, Medium Me, low high LH and High (H). The figure (2) illustrate that, if the queue size is less then or equal to a_2, then the packet dropping is minimum. If the queue is between a_2 and b_2, than the packet dropping in low medium. If the queue size between b_2 and c_2 then dropping is medium. If the queue size c_2 and d_2 then the dropping is low high. If the queue size is in between d_2 and e_2, then the dropping is high and if the queue size exceeds to e_2. Then almost all the packets are dropped. The membership for dropping probability is given by

$$MX_2(x) = \begin{cases} \dfrac{X.Ta_2}{a_2} & \text{for } 0 < x \leq a_2 \\ Ta_2 + \dfrac{x - a_2}{b_2 - a_2}(Tb_2 - Ta_2) & \text{for } a_2 < x \leq b_2 \\ Tb_2 + \dfrac{x - b_2}{c_2 - b_2}(Tc_2 - Tb_2) & \text{for } b_2 < x \leq c_2 \\ Tc_2 + \dfrac{x - c_2}{d_2 - c_2}(Td_2 - Tc_2) & \text{for } c_2 < x \leq d_2 \\ Td_2 + \dfrac{x - d_2}{e_2 - d_2}(Te_2 - Td_2) & \text{for } d_2 < x \leq e_2 \\ Te_2 & \text{for } x > e_2 \end{cases}$$

Where X is the average queue length and Ta_2, Tb_2, Td_2 and Te_2 are the tolerances limit of a_2, b_2, c_2, d_2 and e_2 respectively. The linguistics rules can be framed according to the above dropping membership functions.

DROPPING PROBABILITY

If queue capacity φ_e, queue size φ_s and dropping probability DP.

If φ_c is empty and φ_s is decreasing the DP is low.
If φ_c is empty and φ_s is zero then DP is low.
If φ_c is empty and φ_s is increasing then DP is low medium.

If φ_c is medium and φ_s is decreasing then DP is low high.

If φ_c is full and φ_s is decreasing then DP is low high.
If φ_c is full and φ_s is zero, then DP is low medium.
If φ_c is full and φ_s is increasing the DP is high.

SIMULATION RESULTS

We performed a set of simulation. The simulation configuration is shown in to investigate the performance of proposed scheme in terms of throughput and queuing delay, the proposed scheme has tested under low load and heavy load.

QUEUEING DELAY AND TIME AVERAGE OF PROPOSED TECHNIQUE

The dropping probability, throughput performance time average queuing delay for SREM has been studied throughput.

CONCLUSION

In this paper, efficient linguistic rule based new Active Queue Management scheme of High Performance Network has been proposed and tested. For analyzing output the proposed scheme performed quite well when compared with EREM in terms of throughout and delay of heavy load of networks. This proposed technique can improve the performance of internet router for all network and computer communication system such as low and high column of packets.

REFERENCES

1. S.Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking. 1(4): 397-413, August 1993. fttp://ftp.ee.ibi.gov/papers/early.ps.gz.
2. K.K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP. RFC 2481, January 1999.
3. Steven H. Low. A duality model of TCP flow controls. In Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, September 18-22, 2000, http://netlab.caltech.ed.
4. Sanjeeva Athuraliya and Steven H. Low. Optimization flow control, II: Implementation. Submitted for publication, http://netlab.caltech.ed., May 2000.
5. Sanjeeva Athuraliya, Victor H. Li, Steven H. Low and Qinghe Yin. REM : Active Queue Management (extended version). Submitted for publication, http://netlab.caltech.edu, October 2000.
6. Steven H. Low, Larry Peterson and Limin Wang. Understanding Vegas : a duality model. In Proceedings of ACM Sigmetrics, June 2001, http://netlab.caltech.edu/pub.html.
7. Sally Floyd and Eddle Kohler : "Research in AQM Scheduling" http://www.icir.org, October 2002.
8. 8T.J. Ott, D.V. Lakshman and L. Wong, "SRED Stabilized RED" in IEEE INFOCOM 1999, 1346-1355, March 1999.
9. M. May, J.C. Bolot, A. Jean-Mary and C. Diot : "Simple Performance models of differentiated Services Schemes for the internet" in IEEE INFOCOM99, New York, March, 1999.
10. Sanjeeva Athuraliya, Victor H.Li. : Steven H. Low, Qinghe Yin, "REM Active Queue Management", IEEE Network, June 2001.
11. Sakthivel N.K., Sivaramakrishnan T.R. and Ravichandran K.S. : "EREM, An Active Queue Management Scheme for High Speed Networks" ACCST Research Journal, Siwan (Bihar), India, July 2003, pp. 166-173.
12. Jianping Yin, Zhiping Cai, and Weifeng Chen, RRED: Robust RED Algorithm to Counter Low-rate Denial-of-Service Attacks, IEEE Communications Letters, vol. 14, pp. 489-491, 2010.