An Improved RED Algorithm with Sinusoidal Packet-marking Probability and Dynamic Weight

Songpo Zhang, Jiming Sa, Jian Liu
School of Information Engineering
Wuhan University of Technology
Wuhan, PR China
e-mail: 19585994@163.com

Shaoyun Wu
Wuhan Tianhe Airport Co., Ltd.
Wuhan, PR China

Abstract—*Congestion control has become a research hotspot, because of the rapid growth of Internet. Random Early Detection (RED) algorithm is the most effective active queue management (AQM) techniques. This paper describes RED algorithm and its derivatives then presents a new algorithm. The packet-marking probability linearly with the average queue length is improper for the arrival packet at the gateway. So we present an improved algorithm named SW-RED, which can adjust weight dynamically and make the packet-marking more reasonable. Simulations by NS2 show that SW-RED has better performance and stability comparing with RED.*

Keywords: Congestion control; Active queue management; RED; TCP

I. INTRODUCTION

In the past few years, Internet has increased exponentially, bringing the subject of router's congestion avoidance [1] and control into a hot spot of researching. B. Branden, together with some researchers, came up with the idea of AQM (Active Queue Management) [2], which solved the problems of full queue, deadlock and global synchronization. In 1993, Floyd and his fellows present Random Early Detection (RED) gateways for congestion avoidance [3]. RED, is the earliest AQM that has been put forward. The RED congestion control system holds the basic idea that to monitor the congestion by observing the average queue length of each data packet reached the router. As soon as congestion is approaching, the system can choose the connection randomly to notify the congestion. So that they can reduce the CWND (congestion window) before the queue overflow leading to packet losing, and reduce data-sending speed, therefore the congestion can be relieved. However, the capability of RED is much depends on allocation of parameter, and its adaptive ability is limited in different environments. When the RED parameter allocation is irrationalized, its capability is even worse than the traditional trailing packet discard. As a result, many improved algorithms emerged after the first appearance of RED, such as NRED, ARED, BLUE and so on.

David D. Clark brought out the NRED system in 1998. Its basic idea is to regard a node and its turbulence node as a neighboring area, and think that all the queues of nodes in the neighboring area have formed a distributed queue. At the same time, we can enlarge the RED queue management system of the wired network into the distributed neighboring area queue. By monitoring the congestion in advance, informing the neighboring node, and drop the packet due to the scale of the bandwidth possessed by a current, thus to improve the fairness of TCP [4].

Sally F presented a self-adaption RED, called ARED (Adaptive RED). It can adjust the maximum packet-marking probability automatically according to the network traffic, so the queue's average length could be ranged from $\min_{th}$ to $\max_{th}$. However, the ARED can't solve the demerit of being sensitive to Parameter configuration [5].

Ott T J presented the SRED (Stabilized RED)[6], whose basic idea is that the instability of average queue length resulted from the change of Data transmission rate. Based on present data flow, present queue length and the broadband-possessing situation of the links of data packet, the probability is accountable, so we can keep the queue length stable. The SRED can maintain the stability of buffer queue, and reduce the delay variation effectively.

Nowadays, most international researches related to RED are concentrated in improving the self-adaption of packet-marking probability, but few are into the research of RED's fundamental part— average queue. We present a new RED algorithm named SW-RED, which not only inherits the improvement of drop probability, and brings up a sinusoidal Pb-avg curve, but also dynamically adjusts the average queue weight. So it deals perfectly with the slow response of the average queue length to the real-time queue length, as well as the tolerance of outburst flows. Due to the characteristics of sinusoid, that is p_b changes smaller when approaching to minimum threshold, and greater when close to maximum threshold, the improved algorithm can rationally improve the packet-marking probability, average queue length and utilization rate of date link in a way.

II. RANDOM EARLY DETECTION (RED)

RED was an active queue management technology presented by Floyd and Jacobson in 1993. RED calculates the average

---

Identify applicable sponsor/s here. *(sponsors)*

queue length by exponential weighted average algorithm, and uses the average queue length to adjust the packet-marking probability, in order to inform the transmitting end to reduce the transmit bit rate at the very beginning of net congestion , finally realize congestion avoidance. The basic idea of RED is to detect the congestion by testing the average length of the queue in router output port. It can select the connection randomly to notify congestion if the congestion approaches and makes them to reduce CWND (congestion window) and lower the data-transmitting speed before the queue-full result in package-lost, which can relieve the congestion. It is comparatively easy to implement RED because it schedules strategy based on FIFO queue and only drop the data-package entering the router.

A. *Average Queue Length*

Linking data is paroxysmal. That is to say, if a normally empty queue is filled and emptied in a short time, the alarm of router congestion for the transmitting end is unnecessary. So the calculation of average queue length adopted the weighted average algorithm. The formula is as follows:

$$avg_i = (1-w_q)avg_{i-1} + w_q q_i \qquad (1)$$

avg : Average queue length. q_i : start of the queue idle time. w_q : Queue weight.

B. *Packet-marking Probability*

RED sets two control threshold of average queue length: maximum threshold and minimum threshold. The packet-marking probability gets higher with the increase of the average queue's length. When the average queue length less than the minimum threshold, all the clusters get access to the queue without any marks. And when the average queue length is greater than the maximum threshold, all the successor clusters will be marked. If the length is between $\min_{th}$ and $\max_{th}$, the successor clusters will be dropped or marked by probability, which formula is as follow:

$$p_b = \max{}_p \times \frac{avg - \min_{th}}{\max_{th} - \min_{th}} \qquad (2)$$

Where $\max_p$ means maximum packet-marking probability, $\min_{th}$ and $\max_{th}$ stands for the threshold level setted before. p_b obeys the linear distribution between $\min_{th}$ and $\max_{th}$ as seen in Figure 1. We illustrate the relationship in p_b and avg according to Table 1.

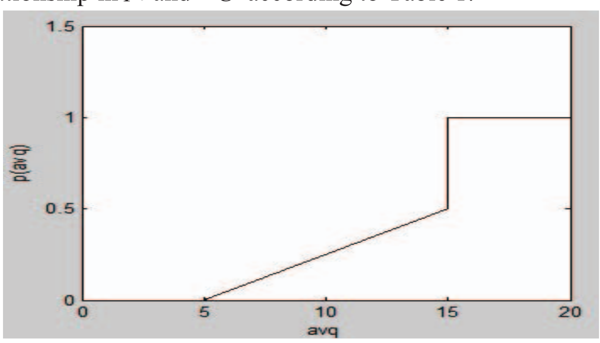

Figure 1 P_b curve graph of RED

Table 1 Classical parameters

Min_th	***Max_th***	***Max_p***	***w_q***
5 packet	15 packet	0.1	0.002

C. *Amendment of Drop Probability*

The final packet-marking probability p_a increase slowly as the count increase since the last marked packet:

$$P_a = \frac{P_b}{(1 - count \times P_b)} \qquad (3)$$

III. Our Proposed Algorithm

The main improved method of SW-RED for RED is in the dynamic adjustment of weight and the adaptive ability of packet-marking probability.

A. *The sinusoidal curve of the packet-marking probability*

The packet-marking probability linearly with the average queue length is improper for the arrival queue at the gateway because the nonlinear distribution of the queue length at the routing buffer dose not match the linearly packet-marking probability of RED algorithm. The average queue length is approaching the $\min_{th}$ when the network is relatively idle, so the packet-marking probability should be as small as possible to allow the router accommodates more arriving packets. Being affected by burst flow, the average queue approaching the maximum threshold can easily surpass $\max_{th}$. This may easily forces the router to mark the packets and leads to global synchronization, which will decrease the utilization of the link, so the packet-marking probability as high as possible is needed in such situation. The sinusoid curve mode can solve the problem effectively.

We illustrate a sinusoidal curve between two adjacent extreme points which depicts the relationship between average queue length and packet-marking probability as seen in Figure 2.

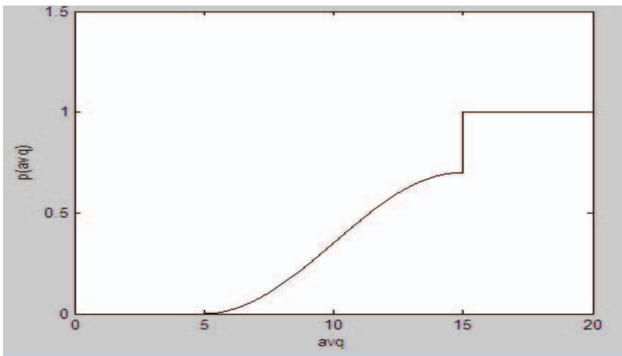

Figure 2 P_b curve graph of improved RED

The improved packet-marking probability as bellow:

$$p_b = \frac{\max_p}{2}\sin\frac{\pi}{\max_{th}-\min_{th}}(avg \times \frac{\max_{th}+\min_{th}}{2}) + \frac{\max_{th}}{2} \quad (4)$$

According to the formula (4), when the average queue length is small, the increase of corresponding packet-marking probability is little (When average queue length approach to the minimum threshold, the packet-marking probability speed decreases) so that as many packet to services.

When the average queue length is great, it range from $(\max_{th}+\min_{th})/2$ to $\max_{th}$. The packet-marking probability of new algorithm is greater than the overall packer-marking probability. Particularly when the average queue length close to the maximum threshold, the packet-marking probability is very close to $\max_p$ .This design considers the security of average queue length, so as to force the router sudden packet loss resulting from the situation greatly reduced.

B. *Dynamic-weight*

As a new-type dynamic weight w_q tuning algorithm RED-DWRED was presented in [7], SW-RED improves this algorithm.

It'll send a mass of Packets to the network as soon as the TCP connection successfully established, which may easily lead to the exhaust of router's buffer memory space in the network, and furthermore, the congestion. The average queue length of RED was too long to bringing about the hoik during the slow start phase, very easily lead to mark the packets forcibly. When TCP flows enter into congestion avoidance status, the real-time queue length would slump. The average queue cannot plunge so that to reduce needless packet-marking if w_q unable to reflect this alteration sensitively (or the settings of weight is too low.). There may come a time that the real-time queue has little length because of the high packet-marking probability when the average queue length is long enough. To solve the problem, this paper presents a dynamic weight tuning algorithm.

As it is relatively difficult to directly use the average queue length to judge the trend of queue changed, we take averaging method together with range ability to make the variation trend of RED average queue length clear. Here we'd like to declare the variable first for the sake of convenient calculation.

$$avg_N = \sum_{i=1}^{n} \frac{avg_i}{N} \quad (5)$$

Where avg_N is the average value of N average queue length.

$$D_j = avg_{jN} - avg_{(j-1)N} \quad (6)$$

$$D_{(j-1)} = avg_{(j-1)N} - avg_{(j-2)N} \quad (7)$$

$w_{\min}$, $w_{\max}$ separately indicate minimum and maximum of dynamic weight tuning. We take the value of $w_{\min}$ as classic setting 0.002. $w_{\max}$ is determined by the minimum threshold of RED and periodic bursts L that router allowed. It satisfies the equation below:

$$l+1+\frac{(1-w)^{l+1}}{w} < \min_{th} \quad (8)$$

The real-time queue length rises when TCP flow is still in the slow start phase, $D_j > 0$, $D_{j-1} > 0$ and $D_j > D_{j-1}$. We've learned the experience of self-adapting filter to adjust w_q self-adapting. At this moment, we set a threshold of n to 1.8, as a way to distinguish the two adjustment methods .The adjustment formula is as follows:

A:When $D_j / D_{j-1} > n$ Then $w_q = w_{\min} \times (D_j / D_{j-1})$ (9)

B: When $n > D_j / D_{j-1} > 1$ Then $w_q = w_{\max}$ (10)

In the case A, queue length increase rapidly. But we can not determine whether the mutations caused by burst traffic .So we adjust the weight increase from $w_{\min}$ to $w_{\max}$, and set D_j / D_{j-1} as coefficients; In the case B , we have identified the increase in queue length is not caused by the burst traffic. We set w_q to the maximum value, so that the average queue length can reflect the real-time queue length sensitively. It has a preferable control of packet-marking probability in the slow start phase.

When CWND (congestion window) surpasses slow start threshold (SSThresh), the slow start procedure comes to an end, turning into congestion avoidance phase, and the congestion window halves. Though the real-time queue length is low, and the average queue length drops, the probability of packet losing stays high yet. As a result of the average queue length still stays high enough. Therefore we adjust $w_q = w_{\max}$ to reflect this change in time, to avoid these problems, and to reduce packet loss ratio in a long-time congestion stage.

The visualized optimum effects can be demonstrated by graphs. The red line represents real-time queue length, green line represents the original RED average queue length, and blue the line shows the SW-RED average queue length.

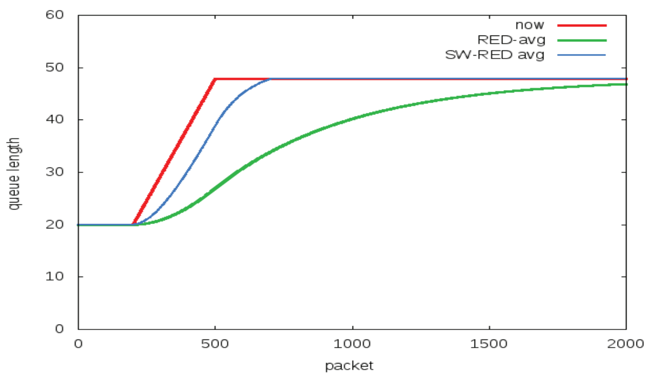

Figure.3 Queue length abruptly up

When the queue length increase suddenly, we found that SW-RED detected congestion approximation faster by comparing the curves of RED with the curves of SW-RED as seen in figure 3. In SW-RED, avg can more quickly rise to real queue length than origal RED.

When the real-time queue length suddenly decreases, SW-RED would monitor this situation rapidly. At this moment, the average queue length of SW-RED become less immediately as seen in figure 4. This make the link utilization higher.

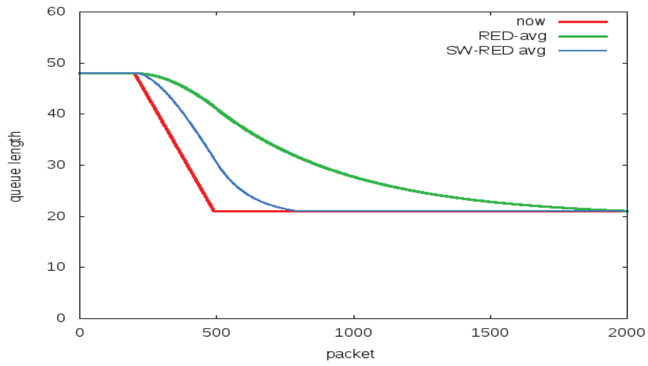

Figure.4 Queue length abruptly down

IV. Simulation and Analysis

We verify our propositions via a set of ns simulations [8]. However, our proposition is based on a liberalized model while ns simulation is actually non-linear in nature.

We look at a single bottle-link router with DropTail, RED and SW-RED application. The topology of the network is shown in Figure. 5. A number of TCP flows are also included in this simulation. In our simulation experiment, we look at a queue with n TCP flows. The bottle link bandwidth is 56kbps, and the propagation delays for the flows are 10ms. The minimum threshold and maximum threshold are set as 5 and 15 packets, with average packet size being 500 Bytes. The queue has a 50 packets buffer, and all of the other parameters are set to be default. In order to better compare SW-RED with RED and DropTail, We simulate them in two different situations in which the number of TCP flows is 10, 20 respectively.

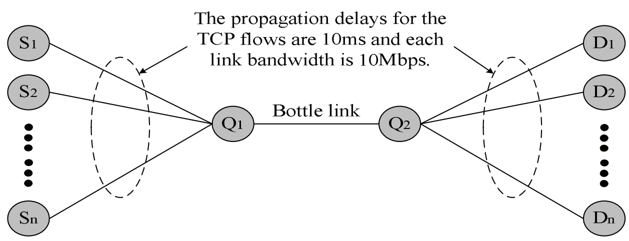

Figure 5 Simulation network topology

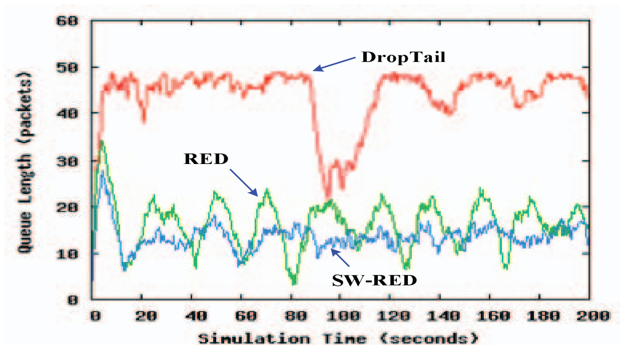

Figure.6 Queue length vs. simulation time (10 TCP flows)

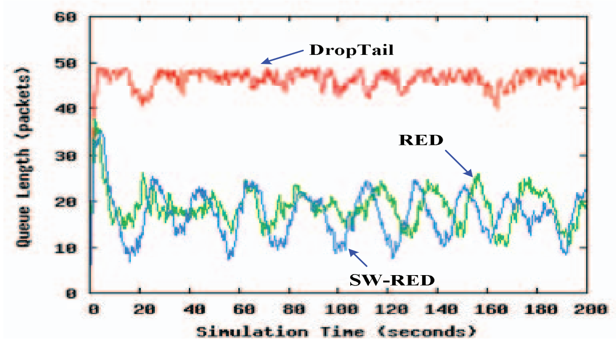

Figure.7 Queue length vs. simulation time (20 TCP flows)

Figure 6 and Figure 7 are the simulation results. They compared the trend of SW-RED real queue packets length with DropTail and RED when the simulation time grows up. We found that SW-RED is better than RED in packet-marking probability and it make the queue packets length more stable and less in the queue buffer from the simulation diagram.

V. Conclusions

This paper outlines RED and its derivatives then present SW-RED in allusion to the lack of RED. SW-RED has been improved in both packet-marking probability and weight adjustment. Sinusoidal packet-marking probability is more reasonable than the linear packet-marking probability of RED. What's more, dynamic adjustment of weights makes the average queue length more adjustment to real-time queue length. We carried out a simulation by NS2. The simulation results show that SW-RED has better performance and stability comparing with the original RED.

Acknowledgment

The author would like to acknowledge the support of "the Fundamental Research Funds for the Central Universities" (Grant No. 2010-la-054) for funding this work.

References

[1] Jacobson V. Congestion avoidance and control [J] .ACM Computer Communication Review, 1988, 18(4):314-329.

[2] BRADEN B, et al. IETF RFC 2309, Recommendations on Queue Management and Congestion Avoidance in the Internet [S] .April 1998.

[3] Floyd S ,Jacobson V. Random early detection gateways for congestion avoidance [J] IEEE/ ACM Transaction on Networking , 1993 , 1 (4):397-413

[4] David D. Clark. Explicit allocation of best-effortpacket delivery service[J]. IEEE/ACM Transactions on Networking, 1998, 6(4): 362-373.

[5] Sally F, Ramakrishna C, Scott S. Adaptive RED: an algorithm for increasing the robustness of red's active queue management[EB/OL].http://www.ieir.org/floyd/papers/adaptive Re − d.pdf.

[6] Ott T J, Lakshman T V, Wong L H. SRED: stabilized RED[C]//Proc IEEE INFOCOM.1999.

[7] Hang Jiang.A dynamic-weight tune RED gateway[J].ACTA elecrtonica sinica,2005,33(3);574-577.

[8] The Network Simulator ns-2 homepage [EB/OL]. http//www. isi. edu/ nsnam/ ns.