

## MRED: An Improved Nonlinear RED Algorithm

Yanping Zhang<sup>a+</sup>, Jun Ma<sup>b</sup>, Yongcheng Wang<sup>c</sup> and Chao Xu<sup>d</sup>

School of Computer Science and Technology, Anhui University, Hefei, China

**Abstract.** Active queue management is the focus of current research; the linear advantages and disadvantages of the classic Random Early Detection ( i.e. RED) algorithm is introduced in this paper. The linear is simple and easy to calculate, however, when average queue size is near to the minimum and maximum threshold, the loss rate is unreasonable. In this paper, we present an improvement RED algorithm named MRED and use NS2 to simulate. Simulation studies show that the MRED algorithm improves the average throughput, decreases packet loss rate and enhances stableness and reliability of network.

**Keywords:** Random Early Detection; nonlinear RED algorithm; Average throughput

### 1. Introduction

With the development of the Internet, When the network exists excessive packets, and network resources (for example: gateway handling capacity, buffer size, link bandwidth) are limited, it will cause the network congestion which led to a sharp decline in the throughput of the network and loss massive transporting packets. Therefore congestion control[1][2]needed to adjust the message transmission rate, ease congestion and then to improve the quality of service of the network.

According to occurring position, congestion control algorithm is divided into two categories: the link algorithm and the source algorithm[3]. In the source algorithm, Transmission Control Protocol[4] is the most widely used as an indispensable factor of guaranteeing network robustness. In the link algorithm, Active Queue Management (AQM) is focus of current research. The Random Early Detection (i.e. RED)[5,6] algorithm is a representative AQM algorithm, and is also the only candidate algorithm recommended by RFC2309.

The rest of the report is organized as follows. Section 2 gives the descriptions of the RED algorithm. Section 3 presents the improved algorithm of RED named MRED. Section 4 describes the simulation results of RED and MRED using NS-2. Finally, section 5 concludes the future work.

### 2. Related Work

In this section, we focus on RED and briefly explain its parameters. RED (Early Detection Detection) is proposed by Floyd and Jacobson for congestion avoidance in packet-switch networks. This method is very simple: the router manipulates the congestion to the sources by dropping the arriving packets.

In spite of its simplicity, the optimal parameter configuration for RED remains a trouble. So several enhancements arise, such as S-RED[7], ARED[8], and BLUE[9].

The RED congestion control mechanism monitors the average queue size for each output queue with buffer size. When a new packet arrives, the router calculates the average queue size, and decides to mark by dropping probability. The drop probability  $P_b$  is a linear function of the average queue size and threshold.

---

<sup>+</sup> Email: <sup>a</sup> zhangyp@mail.hf.ah.cn <sup>b</sup> mj231251@126.com  
<sup>c</sup> luoxiying@foxmail.com <sup>d</sup> xc19860201@126.com

The RED defines two thresholds, the minimum threshold  $\min_{th}$  and the maximum threshold  $\max_{th}$ .  $Avg_q$  is average queue size,  $q$  is current size and  $\max_p$  is the maximum drop probability. If  $Avg_q$  is less than  $\min_{th}$ , drop probability equals 0 and the packet will not be dropped. If  $Avg_q$  is greater than  $\max_{th}$ , the packet will be marked. For other situations, the packet will be marked by dropping probability linear function, defined as the following equation.

$$P_b = \max_p \times (Avg_q - \min_{th}) / (\max_{th} - \min_{th})$$

The final drop-probability  $P$  is calculated as:

$$P = P_b / (1 - count \times P_b)$$

where  $count$  is the number of packets since last marked packet.

### 3. MRED Algorithm

Li cheng-huan, Zhou Hua[10] draw the conclusion that the queue of router is nonlinear. Along with the increase of the average queue size, the drop probability increases with a nonlinear relationship. Studies have shown that in the same parameter configuration, the robustness of the nonlinear RED algorithm is better than RED's. In the original RED algorithm,  $P_b$  and  $P$  calculate by linear function. Simple it is but it does not fit the nonlinear features of RED algorithm well[11,12].

When  $Avg_q$  is in the vicinity of  $\min_{th}$ , the router queue is idle. So drop probability should be slower. While  $Avg_q$  is near to  $\max_{th}$ , the router should drop packets quickly for the sake of ensuring the stability of the queue.

Based on the above conclusions, we calculate drop-probability with a nonlinear function instead of linear function. In order to take full advantage of the queue buffer, drop-probability is calculated by another linear function when average queue size is between  $\max_{th}$  threshold and twice  $\max_{th}$  threshold.

The MRED scheme can be described by the following pseudo code:

Initialization:

$Avg_q = 0$ ;

$count = -1$ ;

for each arrival packet, calculate the average queue size  $Avg_q$

if  $\min_{th} \leq Avg_q < \max_{th}$  calculate the drop-probability  $P_b$

$$P_b = \max_p \times (Avg_q^2 - \min_{th}^2) / (\max_{th}^2 - \min_{th}^2)$$

mark the arriving packet with probability  $P$

else if  $\max_{th} \leq Avg_q < 2 \max_{th}$  calculate the drop-probability  $P_b$

$$P_b = (1 - \max_p) \times (Avg_q - \max_{th}) / \max_{th} + \max_p \quad \text{mark the arriving packet with probability } P$$

else if  $2 \max_{th} \leq Avg_q < Buffize$

mark the arriving packet

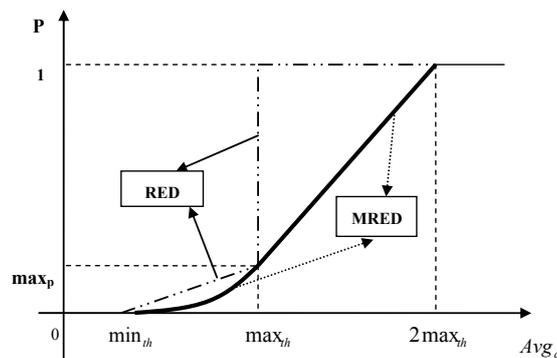


Fig. 1 the relationship between drop probability and the average queue.

## 4. Simulation and Evaluation

In section 4, firstly we describe the network topology and related parameter value, then describe the series of simulated experiments, and finally give the evaluation results. Our experimental environment is Microsoft windows XP Sp3, Cygwin2. 510. 2. 2 and NS-2. 31.

Fig. 2 shows the network topology that represents a simple network bottleneck configuration with two routers and number of subnet nodes. Each subnet has a number of TCP and UDP sources. The bottleneck-sharing link between Router1 and Router2 is 10Mbps with 10ms round trip time. The source subnet use 20Mbps link with 10ms round trip time and the destination use 20Mbps link with 15ms round trip time.

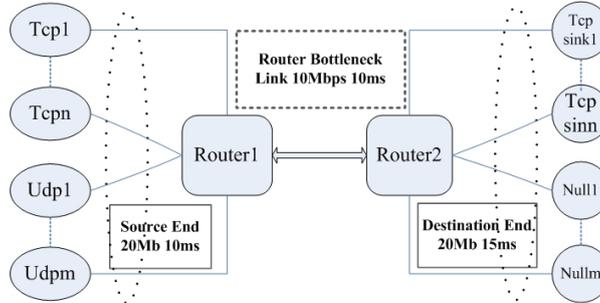


Fig.2 Network topology for RED and MRED.

Other configurations are as follows:

	Parameters	Values
<b>RED</b> <b>and</b> <b>MRED</b>	Queue size	100
	Minimum threshold ( $\min_{th}$ )	10
	Maximum value for $P_b$ ( $\max_{th}$ )	30
	Queue weight ( $W_q$ )	0.002
	Maximum value for $P$ ( $\max_q$ )	0.1
	Simulation time (sec)	50

As throughput is one of the most important indicator of network performance, and the average queue length and its jitter affect the stability of the network, the average throughput, packet loss rate and the average queue length as the indicators are used to measure network performance in this paper.

First we adopt the sole TCP flows to carry on the experiment, in this scenario, the traffic source is only TCP traffic with the number of sources from 5 to 50. The values of  $n$  are 5, 10, 20, 30, 40, 50 and  $m$  is 0.

Next, we use the mixed flows to simulate the two algorithms. The mixed flows are composed of TCP and UDP. TCP flows' settings are unchanged. We add 2 UDP flows. A short-term high-speed flow with 10Mbps transmission rate and 1000byte for each packet starts from the 3rd sec to 10th sec. The other is a long-term low-speed flow, the transmission rate of which is 1Mbps with 1000byte of a packet, starting from the 6th sec to 40th sec.

The simulation data of average throughput and packet loss rate presents separately in Table1, Table2, Table3, Table4.

Table 1 Average throughput of the sole Tcp flows

Flows n	RED Average Throughput (Mbs)	MRED Average Throughput (Mbs)
5	9.598	9.682
10	9.810	9.814
20	9.808	9.841
30	9.867	9.852
40	9.926	9.929
50	9.949	9.956

Table 2 Packet loss rate of the sole Tcp flows

Flows n	RED Packet loss rate (%)	MRED Packet loss rate (%)
5	0.00091756	0.00089242
10	0.00189579	0.00182793
20	0.00435723	0.00446327
30	0.00867089	0.00861511
40	0.01282072	0.01278644
50	0.01675472	0.0167259

Table 3 Average throughput of the mixed flows

Flows n	RED Average throughput (Mbs)	MRED Average throughput (Mbs)
5	9.588	9.590
10	9.725	9.755
20	9.847	9.862
30	9.873	9.870
40	9.901	9.910
50	9.927	9.929

Table 4 Packet loss rate of the mixed flows

Flows n	RED Packet loss rate (%)	MRED Packet loss rate (%)
5	0.00578594	0.00547634
10	0.0081852	0.0081597
20	0.0122582	0.0120873
30	0.0180835	0.0175759
40	0.0226475	0.0218074
50	0.0270019	0.0264582

We can draw Fig. 3 and Fig. 4 from Table 1 and Table 3.

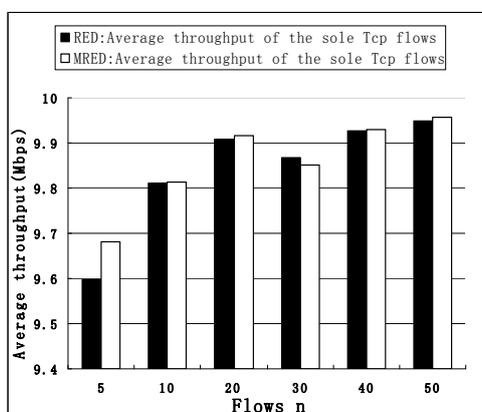


Fig. 3 Average throughput of the sole Tcp flows

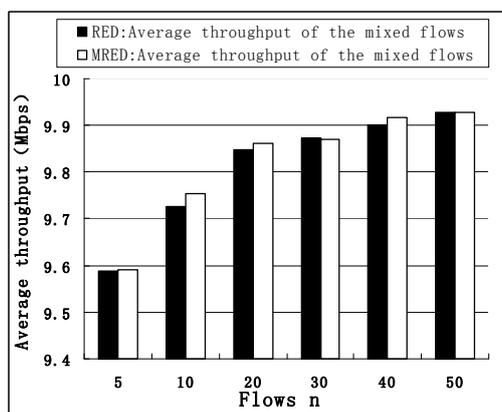


Fig. 4 Average throughput of the mixed Tcp flows

Fig. 3 and Fig. 4 illustrate that when  $n$  is 30, MRED's average throughput is worse than RED's; However, in general, the average throughput of MRED is better than RED.

With the similar method, we get Fig. 5 according to Table 4.

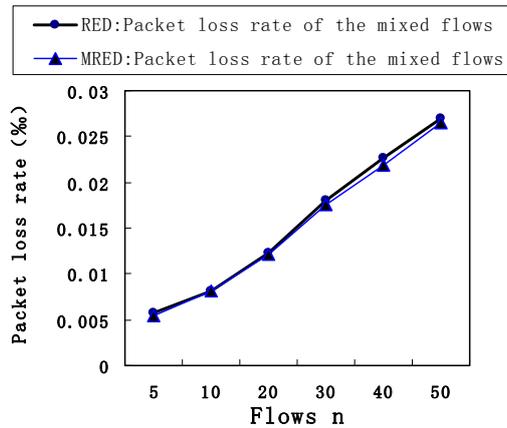


Fig. 5 Packet loss rate of the mixed Tcp flows

In Fig. 5 the packet drop rate of the mixed TCP flows is presented. With the number of flows increase, the packet loss rate increases correspondingly. The packet drop rate of MRED is lower than RED. The same conclusion is also shown in Table 2.

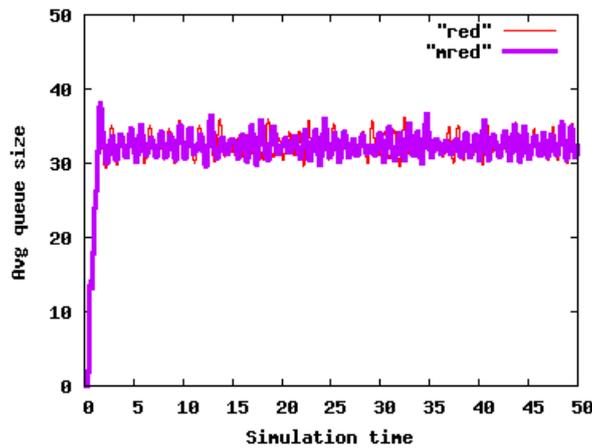


Fig. 6 Average queue size of the sole Tcp flows

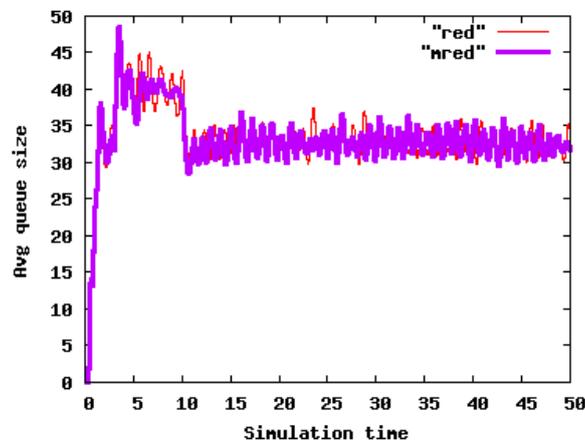


Fig. 7 Average queue size of the sole Tcp flows

When  $n$  is 40, we get the average queue size of RED and MRED in Fig. 6 and Fig. 7. For the sole TCP flows, the average queue of both algorithms is stable. In terms of the jitter, the MRED is better than RED. In Fig. 7, from 3rd sec to 10th sec, we can see that the average queue's jitter is heavy for mixed flows, because

of the UDP flows. Generally speaking, the average queue size of MRED is stabler than RED's, so MRED's jitter is better than RED's.

## 5. Conclusions and Future Work

In this paper, we introduce the classic Random Early Detection (i.e. RED) algorithm. The linear drop probability function is simple and easy to calculate, but it doesn't conform the relationship between the drop probability and the average queue size.

When average queue size is near to the minimum and maximum threshold, the loss rate is unreasonable. Based on RED algorithms, we propose an improved AQM algorithm MRED, which uses the nonlinear function to calculate the drop probability.

Simulation results confired that MRED achieves high performance even under heavy congestion and its performance is quite good in terms of goodput, queue length and packet loss ratio.

There are many possible future works for enhancing the MRED algorithm. Such as "How to decrease the delay of the end to end?" A quantitative analysis of MRED's response time is also very important to evaluate MRED's performance more precisely.

In this paper, we do not cover the fairness issue for the queuing algorithm, which is actually another important factor of queuing algorithm. So the fairness issue will be our future work too.

## 6. Acknowledgment

This research was supported by the National Natural Science Foundation of China (Grant Numbers 60675031), National 973 Key Project of China (2007CB311003), and Ministry of Education Youth Social Science Research Fund (07JC870006).

## 7. References

- [1] R. Gurin and V. Peris, "Quality-of-Service in Packet Networks: Basic Mechanisms and Directions," *Computer Networks*, vol. 31, no. 3, pp. 169-179, Feb. 1999.
- [2] Y.-T. Hou, D. Wu, B. Li, T. Hamada, I. Ahmad, and H. J. Chao, "A Differentiated Services Architecture for Multimedia Streaming in Next Generation Internet," *Computer Networks*, vol. 32, no. 2, pp.185-209, Feb. 2000.
- [3] Steven H Low. "TCP Congestion Controls: Alorithms and Models," [C] IEEE ICC, 2000.
- [4] Luo Wangming, Lin Chuang, Yan Bao-Ping. "A Survey of Congestion Control in the Internet," [J]. *Chinese Journal of Computer* 2001, 24(1)1-18.
- [5] Floyd S, Jacobson V. "Random Early Detection Gateways for Congestion Avoidance," [J]. *IEEE/ACM Transactions on Networking*, 1993, 1(4): 397-413
- [6] FloydS. "RED: Discussions of Scetting Parameters," [J]. *IEEE Journal on Selected Areas in Communications* 1999, 17(6): 1159—1169.
- [7] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "A Self-Configuring RED Gateway," *Proc. of IEEE INFOCOMM*, vol. 3, pp. 1320-1328, Mar. 1999.
- [8] Floyd S, Gummadi R, Shenker S. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. 2001. <http://www.icir.org/~floyd>.
- [9] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "The Blue ActiveQueue Management Algorithms," *IEEE/ACM Trans. Networking*, vol. 10, no. 4, pp. 513-528, Aug. 2002.
- [10] Li Cheng-huan, Zhou Hua, Chen, Dong. "Improved Random Early Detection Algorithm".[J]. *Computer Engineering*, 2008, 34(3): 139-144.
- [11] Ren Feng-Yuan, Lin Chuang, Wang Fu-Bao Stability of RED Algorithm: Analysis Based on Nonlinear Control Theory. [J]. *Chinese Journal of Computer*, 2002, 25(12) : 1302-1307.
- [12] Priya Ranjan, Eyad H. Abed, and Richard J. La, "Nonlinear Instabilities in TCP-RED".[J].*IEEE/ACM Trans. Networking*, vol. 12, pp. 1079–1092, Dec. 2004.