Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 4, No. 1, February 2013), Pages: 31-38
www.jacr.iausari.ac.ir

VRED: An improvement over RED algorithm by using queue length growth velocity

Shahram Jamali[1], Bita Alipasandi[2], Neda Alipasandi[3]

(1) Computer Engineering Department, University of MohagheghArdabili, Ardabil, Iran
(2) Department of Computer Engineering, Zanjan Branch, Islamic Azad University, Zanjan, Iran
(3) Sama technical and vocational training college, Islamic Azad University,Ardabil Branch, Ardabil, Iran

jamali@iust.ac.ir; bita.alipasandi@gmail.com; alipasandi@yahoo.com

Received: 2012/08/05; Accepted: 2012/09/20

Abstract

Active Queue Management (AQM) plays an important role in the Internet congestion control. It tries to enhance congestion control, and to achieve tradeoff between bottleneck utilization and delay. Random Early Detection (RED) is the most popular active queue management algorithm that has been implemented in the in Internet routers and is trying to supply low delay and low packet loss. RED algorithm uses only the average queue length as a congestion meter to trigger packet dropping or packet marking as a congestion feedback. Since the average queue length considers only long–term behavior of any queue, this approach fails to see instantaneous changes of the queue length and hence its reaction is not fast enough. This paperincludes another meter i.e. queue length growth velocity to measure congestion level in the router. This leads to fast reaction to the congestion and hence improves the network performance. Simulation results show that the proposed algorithm outperforms RED algorithm in terms of number of dropped packets and bottleneck utilization.

Keywords: *Congestion control, active queue management, Random Early Detection, packet dropping, utilization*

1. Introduction

In the current Internet, congestion control has been a significant problem and active queue management algorithms were introduced to alleviate the problems of network congestion. The design of the AQM algorithm has received specific attention of researchers, because it is currently the most important bottleneck in the QOS performance of the best-effort Internet. The AQM for Internet routers plays an important role in congestion control and prevent congestion collapse in the network. Congestion collapse occurs when mounting levels of traffic lead to high packet loss in the network, such that few or no packets are actually delivered to their destination however each link is highly loaded. AQM algorithm controls the rate of packets arrival into the queue, by ECN marking or packet dropping to get the congestion signal that controls the source rate [1].

For the Internet to be scalable and totally distributed, the AQM algorithm should only use state information like the link capacity, packet arrival rate and the backlog in the queue. The problem in design fan AQM algorithm is the way to combine state information to generate the proper congestion feedback signal.

Reducing the number of dropped packets and increasing the utilization of the bottleneck link are the most important metrics in view point of the performance of the congestion control algorithms. Also, the performance of AQM algorithms are defined by low queuing delay, less dropped packets, suitable fairness, and high utilization.

The RED algorithm [2] is the most popular AQM mechanism which is recommended scheme of AQM by the Internet Engineering Task Force (IETF) [1]. The performance of RED is very sensitive to the average queue size and initial parameter settings [3, 4, 5, 6], which makes its behavior unpredictable. These lead to the lower utilization and higher packet lost rate.

There are several proposals on AQM by configuring the drop probability function. For example, LRED [7] takes into account the dropping probability based on loss rate and queue length rather than average queue size. This algorithm outperforms stability of original RED and archives high throughput. But, this method introduces additional parameters that are needed to be optimized to stabilize the queue length in different network scenarios.

Some other algorithms did not change the basic idea of RED, and tunes its control parameters [4, 5,8, 9, 10]. One example is Adaptive-RED [5] that by using an exponentially weighted moving average as an integral controller, attempts to stabilize router queue length at a level independent of the active connections. Proportional Derivative RED controller (PD-RED) [11] that proposed by Sun et al. is a new RED scheme based on the proportional derivative control theory that improved the performance of the AQM. However, neither Adaptive-RED nor PD-RED provides any systematic technique to configure the RED parameters. In addition, in both methods the control gain selection is only based on simulation analysis and empirical observation, which can only work well in certain given scenario.

Recently, some enhanced algorithms use the queue length and input rate jointly to attain better performance. In [8], the authors proposed PI that with attention to the queue mismatch regulates the queue length to an expected value. If the network states are known a priori, PI optimal parameters will be determined through a control theoretical model. However, in dynamic networks, PI could have to use a conservative setting to ensure stability, that yielding long response time. REM [12], in order to calculate the drop probability, uses a linear combination of the queue mismatch and input rate mismatch, that input rate mismatch, is equivalently simplified to the queue variance between two adjacent length samples.

The research reported in this paper was aimed at finding the new approach to meet all requirement of an efficient AQM scheme.

The rest of this paper is organized as follows. Section 2 presents motivation of this research. In section 3, the RED algorithm is introduced. In section 4, a new improved AQM algorithm based on RED is proposed and the functionality of new AQM mechanism is discussed in detailed. Section 5 presents implementation and simulations results and compares the performance of RED and new proposed AQM mechanism. Finally, the conclusions are provided in section 6.

2. Motivation

The demand for the Internet-based services has exploded over the last decade. Many organizations use the Internet and particularly the Wide Web as their primary medium for communication and business. To have a high-performance Internet, a good congestion control system is essential for it. Congestion control area regards active queue management as an important mechanism to notify traffic sources about the early stages of congestion and thereby avoid the need for strong source reaction due to heavy overload. There are many reasons to worry that the current AQM schemes in the Internet may bereaching its limits. Despite the RED AQM algorithm considered as a most popular AQM mechanism, this algorithm has disadvantages that reduce its performance. The RED algorithm uses only the average queue length as a congestion meter to trigger packet dropping or packet marking as a congestion feedback. Since the average queue length considers only long–term behavior of any queue, this approach fails to see instantaneous changes of the queue length and hence its reaction is not fast enough. This paper offers a new approach to measure congestion level in router. This leads to fast reaction to the congestion and hence improves the network performance.

3. RED active queue management algorithm

RED is the most well-known AQM algorithm that developed by S. Floyd and V. Jacobson in 1993 [2], and it is widely implemented in routers today. The RED algorithm is a basis for many other AQM proposals which seek to enhance the performance of the basic RED structure. The primary goal of RED AQM is to achieve low average delay and high throughput. In order to achieve theses goals, the RED gateway directs queue by monitoring the average queue size (avg). And when the avg exceeds the expected lower threshold (min_{th}), arriving packets are randomly dropped or marked with a particular probability, in order that some connections can sense the early congestion, and then regulate their own window sizes to avoid serious congestion and packet lost. This probability linearly increases from 0 to p_{max}, where p_{max} is the maximum packet dropping probability. Once the average queue size is larger than the expected upper threshold(max_{th}), the RED mechanism drops or marks every arriving packet as shown in Figure 1.

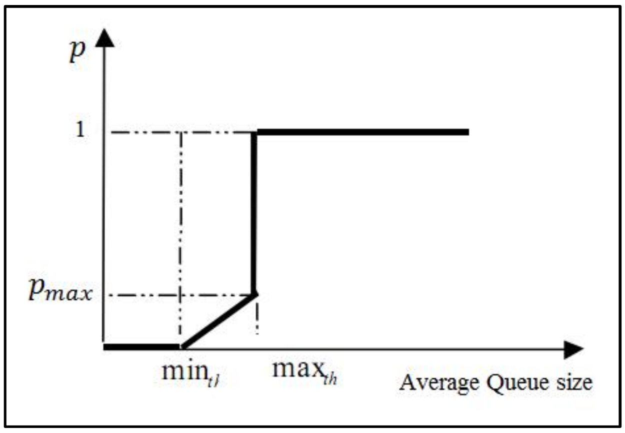

Figure1.RED Dropping Probability vs. Queue Size [2]

Therefore, the RED algorithm contains two computational parts that are computation of the average queue length avg and calculation of the drop probability p.

The average queue length can be got with the equation (1):

$$avg = (1 - w)\ avg + wq \tag{1}$$

Which avg is average queue size at the previous time, q is instantaneous queue length and w is exponential averaging weight which is limited at [0, 1]. avg is updated at the time of packet arrival [2].

The drop probability calculated according the equation (2):

$$p = \frac{avg - min_{th}}{max_{th} - min_{th}} p_{max} \tag{2}$$

Which p_{max} is the drop probability when avg is equal to max_{th}. max_{th} and min_{th} are the expected upper and lower thresholds. General algorithm for RED gateways is shown in figure 2.

```
For each packet arrival
   1. Calculate the average queue size $\boldsymbol{avg}$
   2. If $\boldsymbol{min_{th}} \leq \boldsymbol{avg}\ \ \boldsymbol{max_{th}}$
   3. Calculate probability $\boldsymbol{p}$
   4.      Drop/mark the arriving packet with probability $\boldsymbol{p}$
   5. Else if $\boldsymbol{avg} \geq \boldsymbol{max_{th}}$
   6.      Drop/Mark the arriving packet
```

Figure2. The RED algorithm [2]

RED controls the average queue size within the expected range [min_{th},max_{th}], even in the lack of cooperating sources.

4. Proposed active queue management algorithm: VRED

Queue length growth is a most important factor to take into account that considered in this paper. The new algorithm that called VRED uses the queue length growth velocity to measure the congestion level in router, and triggers the drop probability according to this meter. This leads to fast reaction and hence improves the network performance. The underlying idea of the VRED, is that packet dropping becomes gentler than RED at light traffic load but more aggressive at heavy traffic.

V is the queue length growth velocity and it can be got with the Equation (3):

$$V = \Delta q / \Delta t \tag{3}$$

Δq is the differences of the current queue length and previous queue length and Δt is the differences of current time and previous time.

It is clear that in heavy congestion state, time required for the specified change of queue length is lower than the time in light congestion state and so the value of V in heavy congestion is higher than light congestion. The following algorithm shows how the VRED works. Base on this discussion, the following algorithm shows the VRED algorithm.

```
For each packet arrival
1.  Calculate the average queue size (avg)
2.  $q_r$=current queue length
3.  $\Delta q = q_r - q_{r-1}$
4.  If $\Delta q >= q\_th$
5.      $Time_r$=current time
6.      $\Delta t = Time_r - Time_{r-1}$
7.      Calculate $V$
8.      r=r+1
9.  If $\mathbf{min}_{th} \le \mathbf{avg} \le \mathbf{max}_{th}$
10.     Calculate probability p
11.     If $V > 0$
12.         Increment p according to the V
13.     Else if $V < 0$
14.         Decrement $p$ according to the $V$
15.     Drop/Mark the arriving packet with probability p
16. Else if $\mathbf{max}_{th} \le \mathbf{avg}$
17.     Drop/Mark the arriving packet
```

Figure3. The VRED Algorithm

Where *q_th* is the threshold of queue length growth that its default value is 5 packets and other parameters are the same as those with RED. If queue length growth velocity is positive then the drop probability should be increased, and conversely if it is negative then the drop probability should be decreased. Several level of the congestion is defined according to the severity of queue length growth velocity. Also the percentage of increase or decrease of the drop probability is dependent on these levels. It is clear that this method of adjusting the drop probability leads to lower queue oscillation.

5. Implementation and simulation results

In order to study performance of the proposed mechanism, VRED implemented by making some modification on RED module of ns-2 software package. This section focuses on the following key performance metrics: packet loss ratio and utilization. Consider a network with a set of 20 source nodes and a set of 20 destination nodes. The network has a simple dumbbell topology as shown in figure 4.The bottleneck link capacityis 2 Mbps and all sources have the same RTT of 10 ms.The simulations use RED with NS's default values. The simulation experiments were run for 100 seconds.

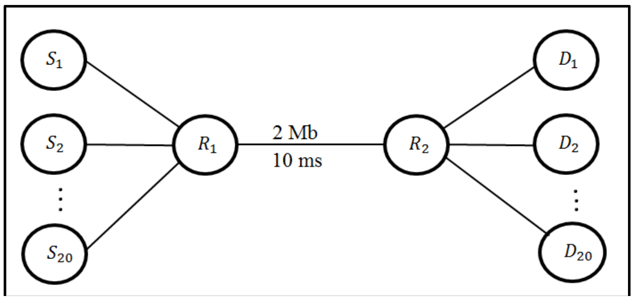

Figure 4.Structure of the network (n=20)

In this section, the *simulation results of two different scenarios* are presented to compare the performance of the VRED and RED.

In the first scenario, only five sources are active at time 0,at time t=20 s ten sources and at times t=40 s five sources become active. In the second scenario, for modeling the burst traffic, five sources are active at time 0, at time t=15 sfifteen sources become active. Table 1 compares RED and VRED algorithms in term of drop count.

Table 1.Comparison of packet losses inRED and VRED

Scenario	Drop Count
Scenario1	● RED : 2974 ● VRED : 2511
Scenario2	● RED : 3333 ● VRED : 2747

Table 1 shows that while both RED and VRED encountered with the same congestion, VRED avoids the RED's higher packet loss. Figure 5 and figure 6 show simulation results of the scenario 1.

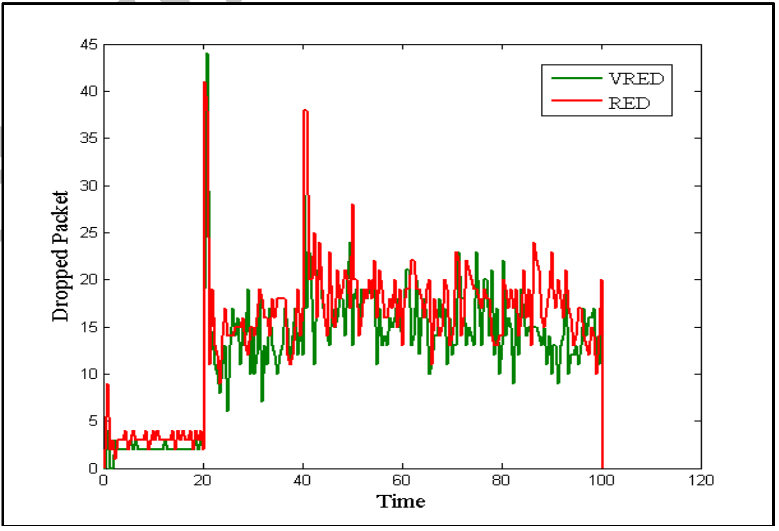

Figure 5. Comparison of VRED and RED algorithms in term of Dropped Packets in the scenario 1

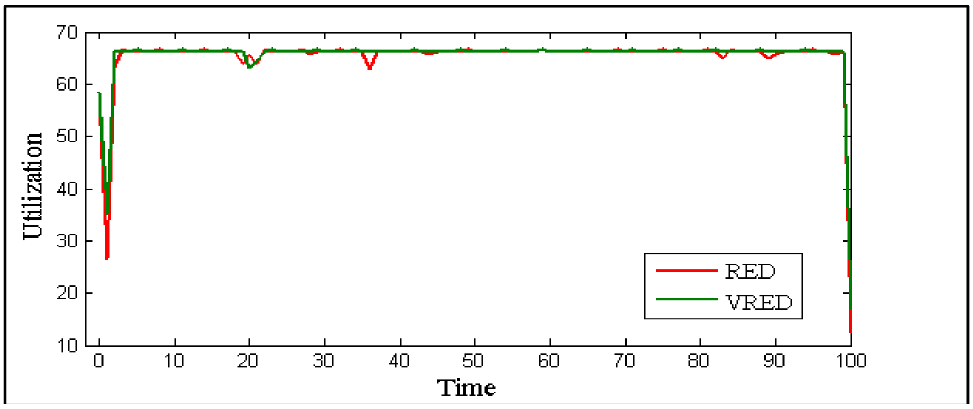

Figure 6. Comparison of VRED and RED algorithms in term of Utilization in the scenario 1

According to these figures, VRED achieves low packet loss, and its utilization remains more than RED's utilization. Figure 7 and figure 8 show simulation results of the scenario 2.

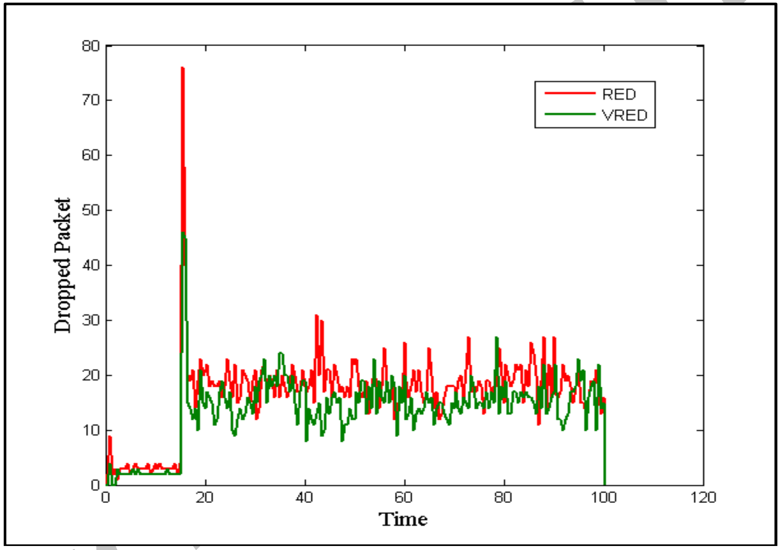

Figure 7. Comparison of VRED and RED algorithms in term of Dropped Packets in the scenario 2

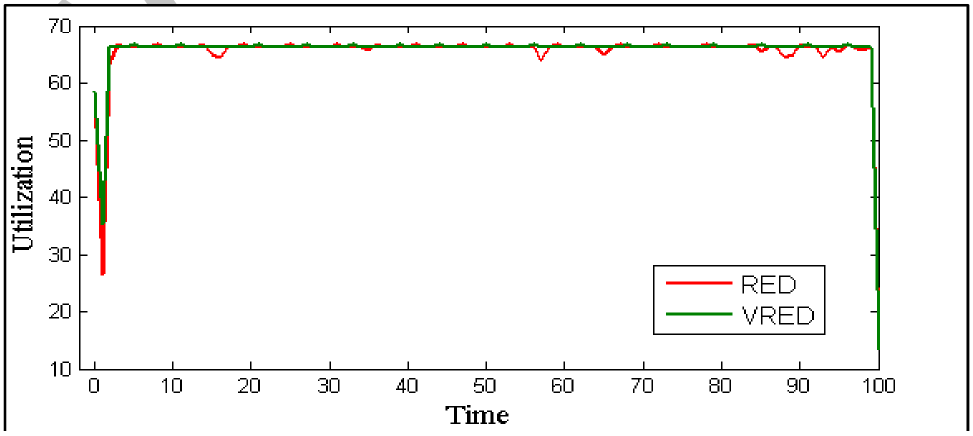

Figure 8. Comparison of VRED and RED algorithms in term of Utilization in the scenario 2

Simulations demonstrate that VRED achieves low packet loss and high utilization. Whereas packet bursts are an unavoidable aspect of packet networks, the main advantage of VRED appears in dealing with burst traffic and VRED by fast reaction improves the network performance.

6. Conclusion

This paper proposed VRED algorithm as an improvement over RED algorithm. Queue length growth velocity used to measure congestion level. The VRED algorithm implemented in ns2 simulator that the simulation results imply that the VRED significantly achieves low packet loss and high utilization. VRED in burst traffic properly diagnoses the congestion and by fast reaction improves the network performance. In addition, the VRED does not change the basic principle of the RED algorithm, and can be simply applied in practice.

7. References

[1] 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 2309*, 1998.

[2] S. Floyd, V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE-ACM Transactions on Networking*, 1993.

[3] M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED," *in Proceedings of the 7th International Workshop on Quality of Service*, 1999.

[4] W. Feng, D. Kandlur, D. Saha, and K. G. Shin, "A self-configuring RED gateway," *in Proceedings of the IEEE Conference on Computer Communications*, 1999.

[5] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," http://www.icir.org/floyd/papers/adaptiveRed.pdf, 2001.

[6] R. J. La, P. Ranjan, and E. H. Abed, "Analysis of adaptive random early dection (Adaptive RED*)," in Proceedings of the 18th International Teletraffic Congress*, 2003.

[7] C. Wang, J. Liu, B. Li, K. Sohraby, and Y. T. Hou, "LRED: a robust and responsive AQM algorithm using packet loss ratio measurement," *IEEE Transactions on Parallel and Distributed Systems*, 2007.

[8] C.V. Hollot, V.Maisra, D. Towsley,W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," *in Proceedings of IEEE Conference on Computer Communications*, 2001.

[9] L. Tan, W. Zhang, G. Peng, and G. Chen, "Stability of TCP/RED systems in AQM routers," *IEEE Transactions on Automatic Control*, 2006.

[10] B. Zheng andM. Atiquzzaman, "A framework to determine bounds of maximum loss rate parameter of RED queue for next generation routers," *Journal of Network and Computer Applications*, 2008.

[11] J. Sun, K. Ko, G. Chen, S. Chan, and M. Zukerman, "PD-RED: to improve the performance of RED," *IEEE Communications Letters*, 2003.

[12] S. Athuraliya, S. Low, V. Li, and Q. Yin, "REM: Active Queue Management," *IEEE Network Magazine*, 2001.