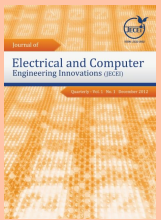
SRTTU

Journal of Electrical and Computer Engineering Innovations

JECEI, Vol. 2, No. 2, 2014

Regular Paper



An Improvement over Random Early Detection Algorithm: A Self-Tuning Approach

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

[1]Computer Engineering Department, University of Mohaghegh Ardabili, Ardabil, Iran

[2]Sama technical and vocational training college, Islamic Azad University, Ardabil Branch, Ardabil, Iran

[3]Young Researchers and Elite Club, Ardabil Branch, Islamic Azad University, Ardabil, Iran

[*]Corresponding Author' Information: Tel.: +98-45-33361902, E-mail address: alipasandi@yahoo.com

ARTICLE INFO

ARTICLE HISTORY:
Received 15 December 2013
Revised 1 February 2014
Accepted 15 February 2014

KEYWORDS:
Internet
Congestion control
Active Queue Management (AQM)
Random Early Detection (RED)
Self-Tuning

ABSTRACT

Random Early Detection (RED) is one of the most commonly used Active Queue Management (AQM) algorithms that is recommended by IETF for deployment in the network. Although RED provides low average queuing delay and high throughput at the same time, but effectiveness of RED is highly sensitive to the RED parameters setting. As network condition varies largely, setting RED's parameters with fixed values is not an efficient solution. We propose a new method to dynamically tuning RED's parameters. For this purpose, we compute the rate of which the queue is occupied and consider it as a congestion metric that will be forecasted when the queue is overloaded. This meter is used to dynamically setting RED parameters. The simulation results show the effectiveness of the proposed method. According to the results, we achieve a significantly higher utilization and less packet loss comparing to original RED algorithm in dynamic conditions of the network.

1. INTRODUCTION

Congestion control of internet plays an important role in availability and stability of internet network. Nowadays, Transmission Control Protocol (TCP) is the dominant protocol among the Internet protocols. A major reason is that congestion control and avoidance mechanism employed in TCP contributes to internet congestion control. TCP flows at the end host use packet loss or Round-Trip Time (RTT) difference to detect and react to congestion by decreasing the transmission rate. Many researches [1]-[3] demonstrate that end-to-end congestion control mechanisms (such as one employed in TCP) are obligatory, but they are not adequate, so congestion control should be assisted by routers; otherwise the danger of congestion collapse still exists, because the end host cannot singly dominate the unresponsive and non-TCP-compatible flows. The routers chip in congestion control by two different ways [1]: scheduling and queue management. Scheduling is not in this scope. Queue management algorithms manage the queue occupation by dropping packets when necessary or appropriate [1]. Queue management algorithms, from the point of dropping packets, can be categorized into two classes: The first class is Passive Queue Management (PQM), which does not apply any preventative packet drop before the router buffer becomes full or attains a predefined value. The second class is Active Queue Management (AQM), which apply preventative packet drop before the router buffer becomes full.

Drop-tail is one of the traditional PQM algorithms that drops incoming packets when the queue is full. The drawbacks of using PQM algorithms in routers are that, first, there will be high queuing delays at congested routers [4], because queue is often full; this happening is called full-queue phenomenon, second, a

few number of flows can monopolize the queue whereas the other flows perpetually get fined; this event is called lock-out phenomenon and caused unfairness [1], [5], [6].

Random Early Detection [7] (RED) is one of the most prominent AQM algorithms as a solution for the full-queue and lock-out phenomenon. RED is suggested by Internet Engineering Task Force (IETF) for deployment in the network [1]. Although, RED provides simultaneously high throughput and low average queuing delay [4], but effectiveness of RED is highly sensitive to the RED parameters setting [8]-[10]. However original RED uses fixed values for its parameters, and as regards network condition varies largely, fixed parameters setting of original RED algorithm is not efficient. Many researchers [11], [12] investigated the behavior of the RED in various conditions or proposed modification in [13], [14] over RED algorithm.

In this paper, to rectify the shortcoming of the original RED algorithm, we propose a new method to dynamic tuning of RED's parameters based on network condition. For this purpose, we introduce a new measure that foresights network future load and then RED's parameters are adjusted based on it. The proposed algorithm has been implemented in NS2 [15] simulator. Simulation results show the effectiveness of the proposed algorithm which solves the parameters setting problem and enhances RED's performance, remarkably.

The rest of this paper is organized as follows. Section 2 briefly describes RED algorithm. In Section 3, we introduce our new measure which called QRT, and then describe our improved RED algorithm which called QRTRED. Section 4 presents the simulation results, and finally section 5 brings concluding remarks.

2. ORIGINAL RED ALGORITHM AND ITS PROBLEM

The original idea of RED algorithm, which was first proposed by Floyd and Jacobson [7], is preventing buffer overflow by early detection of full queue. For this purpose, RED uses four control parameters which are minimum threshold min_{th}, maximum threshold max_{th}, maximum mark (drop) probability Max_p, queue weight w_q. RED detects incipient congestion status by calculating weighted average of the queue length which hereinafter is denoted by *avg*, and alerts TCP senders that congestion has happened by dropping packets deliberately at a certain probability which hereinafter is denoted by *p(avg)*. *avg* and *p(avg)* are defined as Equation (1) and (2), respectively. In original RED algorithm the value of min_{th} and max_{th} are 5 and 15, respectively. If *avg* was lower than min_{th}, then drop probability of packets will be zero. If *avg* was higher than max_{th}, then drop probability of packets will be one. If *avg* is higher than min_{th} but still below max_{th}, drop probability of packets will be *p(avg)*.

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

$$p(avg) = \frac{avg - max_{th}}{max_{th} - min_{th}} max_p \tag{2}$$

Lock-out and full-queue phenomenon that lead to global synchronization can be prevented by RED gateways. Also, RED gateways can reduce packet loss rates, and minimize biases against burst sources [16]. All of these, as regards that RED has low-overhead and simple algorithm have led the IETF to recommend the use of RED as a default active queue management in Internet routers. Since RED was introduced in 1993, many researches have been done to study its performance [8]-[12], [17] and many enhancements or variants were proposed [4], [16], [18]-[27]. These studies shown that although RED can surpass traditional drop-tail queues and can improve TCP performance, but the performance of RED considerably depends on the values of its control parameters.

In heavily congested networks, RED's congestion notification must be more aggressive, in order to avoid packet loss due to buffer overflow [16]. On the other hand, in lightly congested networks, RED's congestion notification must be less aggressive, in order to both preventing excessive packet drop and to maintain link utilization at an acceptably high level. But, the aggressiveness level of RED significantly depends on setting of its control parameters. Nonetheless the value of these control parameters in original RED algorithm is invariable and hence as regards network condition varies largely, the invariant aggressiveness of RED is not efficient in dynamic condition of network.

In order to RED router to be efficient, its control parameters should be set to proper values according to the condition of the network (congestion level, the number of active connections, link bandwidth, and so on). However, as noted by other researchers [9] , [10] it is infeasible to gain one parameter set to make the RED algorithm work effectively in dynamic condition of network. If the control parameters of RED are not configured correctly, then its throughput is often under that of drop-tail routers.

3. PROPOSED SOLUTION

To overcome the limitations of RED algorithm, we introduce a new meter which indicates network condition, and then, RED's control parameters are configured dynamically based on it. Our new metric, that called *QRT*, infers network conditions from rapidity of the buffer occupancy in the router. To do this we measure speed of change in the quantity of occupied cells of buffer. If the quantity of occupied

cells is increased, then the *QRT* will have a positive value proportional to the network congestion level, meaning that offered load to the network is increasing. If the quantity of occupied cells is decreased, scilicet the quantity of free cells is increased, then the *QRT* will have a negative value, meaning that offered load to the network is decreasing. So, with *QRT* metric that indicates network dynamic conditions, we can dynamically adjust the RED's control parameters proportional to network dynamic conditions. In our algorithm, the aggressiveness of RED is tuned by updating of only max_{th} and min_{th} based on *QRT* value.

The Fig. 1 shows the proposed algorithm to dynamically RED's parameter setting, where *T* is the total buffer size and E_i is the number of occupied cell of buffer in interval *i*. In QRTRED algorithm, the value of min_{th} is 5 plus Φ so that Φ is proportional to *QRT* value. If *QRT* is zero or less than zero, then Φ is a 15% total buffer size and else if *QRT* is higher than zero, Φ varies from 0% to 14% of total buffer size according to *QRT* value.

1. in each interval *i* separated by fluctuation of τ =2 cells in the number of occupied cell of buffer
2. Calculate QRT_i
3. if $QRT_i \le 0$
4. $\Phi_i = 0.15 * (T\_E_i)$;
5. else
6. $\Phi_i = 1/QRT_i * (T - E_i)$;
7. $min_{th} = 5 + \Phi_i$;
8. $max_{th} = 3 * min_{th}$;

Figure 1: The proposed algorithm to dynamically RED's parameter setting

4. Simulation Results

In order to evaluate the proposed approach, we implement it by some modifications on RED module of NS2 simulator. Then, it is run over the network shown in Fig. 2 and is compared with original RED algorithm in dynamic conditions of network.

In Fig. 2, S_i, D_i and R_i represent sender hosts, destinations and routers, respectively. Sender *i* communicates with destination *i* (*i*=1, 2... 20) by TCP Reno. The bandwidth and propagation delay of each link are labeled on it. The buffer size of each queue used in RED routers or proposed QRTRED routers is 100 packets, the size of data packet is 1 kbytes, and the size of ACK is 40 bytes for both original RED and proposed RED algorithm. In our scenario, one FTP source starts at time *t*=0 sec and then two another FTP source would start respectively at *t*=20 sec and *t*=40 sec and these two sources stop at *t*=50. Then, at *t*=60 and *t*=80 sec five new and fourteen new FTP sources join to the network, respectively. These dynamic sources are used to simulate the dynamic condition of network. The simulation time is 100 sec. In our simulation, we compare tree router discipline. The first discipline is the RED with default value, second is the RED with min_{th}=10 and max_{th}=40, and third is the QRTRED.

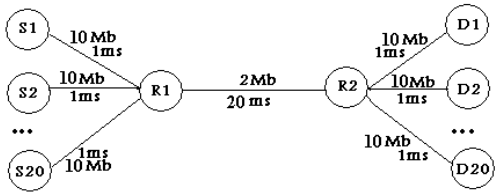


Figure 2: Network simulation topology

The simulation results in view point of bottleneck utilization, number of dropped packets, and average Queue size are shown in Fig.3-5. Fig. 3 shows the number of dropped packets in QRTRED, RED with default values and RED with min_{th}=5 and max_{th}=40. As shown in Fig. 3 and Table 1, the number of dropped packets in QRTRED is less than the others. The reason of high packet loss in original RED algorithm is that the default value of min_{th} is five packet and default value of max_{th} is three times min_{th}, these values make RED very aggressive, but this level of aggressiveness happens only in heavily congested networks. Thus, this level of aggressiveness of RED in lightly congested networks leads to unnecessary packet drops which alert TCP senders to decrease sending rate and also causes link under-utilization [28]. Moreover, this unnecessary packet drops affects on all of the resources even they are used in transit, and makes them wasted. It is due to this fact that the packets are dropped before they reach their destination [16], [29].

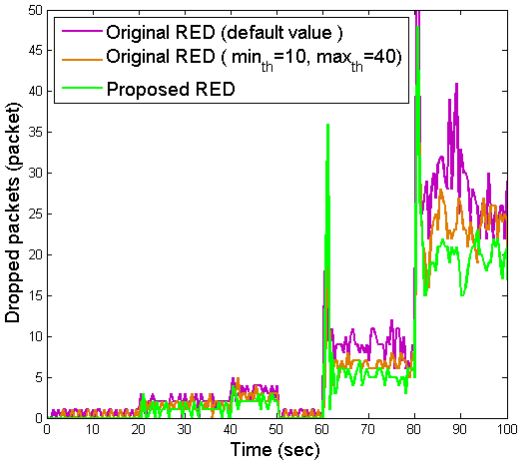


Figure 3: Comparison of dropped packets

Utilization of bottleneck link is compared in Fig. 4. It is shown that QRTRED has higher utilization than

others. Also, it is important to note that QRTRED has noticeable high utilization than two others at time of t=0 to t=20 sec and t=50 to t=60 sec. Because it acts proportional to network condition and as regards in these timeframes only one flow is active, so network traffic is light and thus, there is no need for aggressive packet drop policy that used in original RED algorithm. Therefore, in these timeframes QRTRED drops none of packets and as a result its utilization is more than the RED. Also, Fig. 3 and Fig. 4 demonstrate that QRTRED has better performance because it acts proportional to network dynamic conditions. Fig. 5 compares the average queue size of these three algorithms. Although the average queue size of RED is appear slightly lower than the others, but this is due to RED's min_{th} and max_{th} values that make it very aggressive in all conditions of network. As demonstrated in Fig. 5, the average queue size of RED with min_{th}=10 and max_{th}=40 is higher than RED with default value. However, despite all these slightly increasing of queue, size in contrast with increasing of utilization and decreasing of dropped packets is negligible.

TABLE 1
COMPARISON OF THE ALGORITHMS

Algorithm	Average Utilization	Total Drop
RED with default value	91.14	1720
RED with min_{th}=10, max_{th}=40	93.36	1367
Proposed RED	94.51	1136

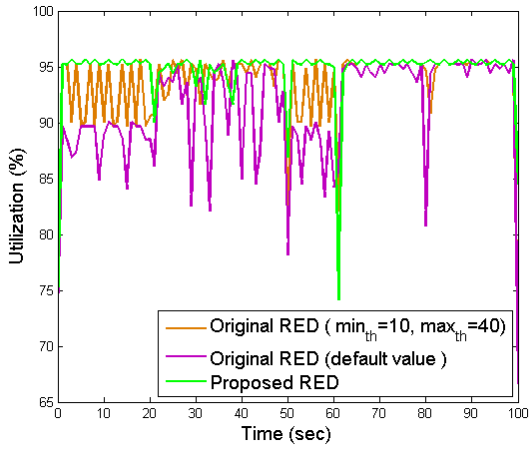


Figure 4: Comparison of the utilization

5. CONCLUSIONS

This paper proposed a new method to improve the RED algorithm performance. We introduced a new metric called QRT to detect any change in the network congestion status. Then, we configured RED's control parameters (max_{th} and min_{th}) based on the QRT. The simulation results show the effectiveness of the proposed method. According to the results, we achieved a significantly higher utilization and less packet loss comparing to the original RED algorithm in dynamic conditions of the network. It is important to note that the proposed algorithm solves the parameters setting problem without making additional overhead to the original algorithm. It is notable that QRT is reliable to apply in other problems that need predicting network load status without making additional overhead.

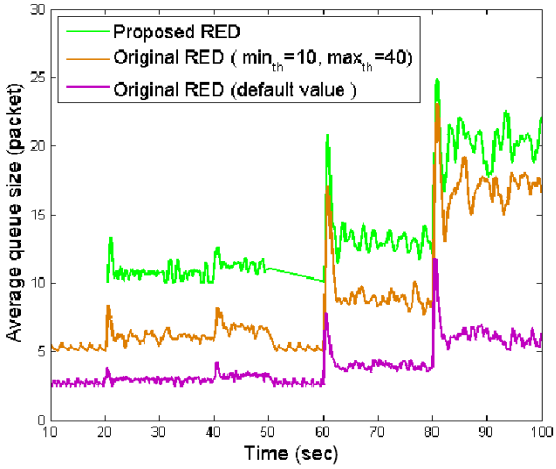


Figure 5: Comparison of average queue size

REFERENCES

[1] B, Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, and D. Estrin, "Recommendations on queue management and congestion avoidance in the Internet", *Network Working Group RFC 2309*, 1998.

[2] J. Nagle,"Congestion control in IP/TCP internetworks,"*ACM SIGCOMM Computer Communication Review*, vol. 25, no. 1, pp. 61-65, 1995.

[3] S. Floyd, K. Fall, "Router mechanisms to support end-to-end congestion control," *Technical report*, February 1997, Available from http://wwwnrg. ee. lbl. gov/floyd/end2end-paper. html.

[4] S. Floyd, R. Gummadi, S. Shenker,"Adaptive RED: An algorithm for increasing the robustness of RED's active queue management,"2001, Preprint, available from http://www. icir. org/floyd/papers. html.

[5] S. Floyd, V. Jacobson, "On traffic phase effects in packet-switched gateways," Internetworking: Research and Experience, vol. 3, no. 3, pp, 115-156, 1992.

[6] S. Floyd, "Connections with multiple congested gateways in packet-switched networks part 1: one-way traffic," *ACM SIGCOMM Computer Communication Review*, 1991; vol. 21, no. 5, pp. 30-47, 1991.

[7] S. Floyd, V. Jacobson,"Random early detection gateways for congestion avoidance,"*IEEE/ACM Transactions on Networking*, vol. 1, no.4, pp. 397-413, 1993.

[8] T. Bonald, M. May, JC. Bolot,"Analytic evaluation of RED performance,"*in Proc. 2000 IEEE 9 th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM2000*,vol. 3, pp. 1415-1424.

[9] M. May, J. Bolot, C. Diot, B. Lyles, "Reasons not to deploy RED,"*In the 7th International Workshop on Quality of Service IWQoS'99 IEEE*, pp. 260-262, 1999.

[10] M. Christiansen, K. Jeffay, D. Ott, S. Donelson,"Tuning RED for web traffic,"*IEEE/ACM Transactions on Networking*,vol. 9, no. 3, pp. 249-64, 2001.

[11] G. Patil,S. McClean, G. Raina, "Drop tail and RED queue management with small buffers: stability and Hopf bifurcation," *ICTACT Journal on Communication Technology:*

Special Issue on Next Generation Wireless Networks and Applications, vol. 2, no. 2, pp. 339-344, 2011.
[12] D. Amol, P. Rajesh, "A Review on Active Queue Management Techniques of Congestion Control," *Proceedings of International Conference on Electronic Systems, Signal Processing and Computing Technologies ICESC2014 IEEE*, pp. 166-169, 2014.
[13] MK. Dadhania, KV. Kumar, "Modified RED Algorithm to Improve the Performance of Web Traffic," *Proceedings of 3thInternational Conference on Advanced Computing and Communication Technologies ACCT2013 IEEE*, pp. 187-194, 2013.
[14] AH. Ismail, A. El-Sayed, Z. Elsaghir, IZ. Morsi ,"Enhanced Random Early Detection (ENRED)," *International Journal of Computer Applications*, vol. 92, no. 9, pp. 25-88, 2014.
[15] DARPA/VINT, LBNL, XEROX, UCB AND USC/ISI (2001), "Network simulator ns- 2", Information available at http://www.isi.edu/nsnam/ns/. Code available at http://www.isi.edu/nsnam/dist/
[16] WC. Feng, DD. Kandlur, D. Saha, KG. Shin, "A self-configuring RED gateway," *Proceedings of 8th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM'99 IEEE*, pp. 1320-1328, 1999.
[17] TA. Trinh, S. Molnár,"A comprehensive performance analysis of random early detection mechanism,"*Telecommunication Systems*,vol. 25, no. 1-2, pp. 9-31, 2004.
[18] TJ. Ott, TV. Lakshman, LH. Wong, "Sred: stabilized red," *Proceedings of 8th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM'99 IEEE*, pp. 1346-1355, 1999.
[19] J. Sun, KT. Ko, G. Chen, S. Chan, M. Zukerman, "PD-RED: to improve the performance of RED," *IEEE Communications letters*, vol. 7, no. 8, pp. 406-408, 2003.
[20] J. Aweya, M. Ouellette, DY. Montuno, "A control theoretic approach to active queue management," *Computer Networks*, vol. 36, no. 2, pp. 203-235, 2001.
[21] D. Lin, R. Morris, "Dynamics of random early detection," *ACM SIGCOMM Computer Communication Revie*w, vol. 27, no. 4, pp. 127-137, 1997.
[22] C. Wang, J. Liu, B. Li, K. Sohraby, YT. Hou, "LRED: a robust and responsive AQM algorithm using packet loss ratio measurement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 1, pp. 29-43, 2007.
[23] W. Chen,SH. Yang, "The mechanism of adapting RED parameters to TCP traffic, "*Computer Communications*, vol. 32, no. 13, pp. 1525-1530, 2009.
[24] B. Hariri, N. Sadati, "NN-RED: an AQM mechanism based on neural networks," *Electronics Letters*, vol. 43, no. 19, pp. 1053-1055, 2007.
[25] B. Abbasov, S. Korukoglu, "Effective RED: An algorithm to improve RED's performance by reducing packet loss rate," *Journal of Network and Computer Applications*, vol. 32, no. 3, pp. 703-709, 2009.
[26] C. Zhang, J. Yin, Z. Cai, W. Chen, "RRED: robust RED algorithm to counter low-rate denial-of-service attacks," *Communications Letters, IEEE*, vol. 14, no. 5, pp. 489-491, 2010.
[27] B. Zheng, M. Atiquzzaman, "A framework to determine bounds of maximum loss rate parameter of RED queue for next generation routers," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 429-445, 2008.
[28] GR. Arce, KE. Barner, L. Ma, "Median RED algorithm for congestion control," *in Proc. ICASSP '04 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp.341-344, 2004.
[29] W. Feng, DD. Kandlur, D. Saha, KG. Shin, "Techniques for eliminating packet loss in congested TCP/IP networks," University of Michigan, Tech. Rep. CSE-TR-349-97. 1997.

BIOGRAPHIES

Shahram Jamali is an associate professor leading the Computer Networking Group at the Department of Engineering, University of Mohaghegh Ardabili. He teaches on computer networks, network security, computer architecture and computer systems performance evaluation. Dr. Jamali received his M.Sc. and Ph.D. degrees from the Dept. of Computer Engineering, Iran University of Science and Technology in 2001 and 2007, respectively. Since 2008, he is with Department of Computer Engineering, University of Mohaghegh Ardabili and has published more than 80 conference and journal papers. He also serves as editor member for some international journals.

Neda Alipasandi received her B.Sc. and M.Sc. degrees in Computer Engineering in 2007 and 2011, respectively from the Islamic Azad University-Zanjan Branch. Her research interests are mainly network control and management, TCP/IP and revolutionary algorithms (GA, PSO).

Bita Alipasandi received her B.Sc. and M.Sc. degrees in Computer Engineering in 2008 and 2012, respectively from the Islamic Azad University-Zanjan Branch. Her research interests are mainly network control.