SCITECH
RESEARCH ORGANISATION|

Volume 7, Issue 1
March 20, 2018

Journal of Information Sciences and Computing Technologies
www.scitecresearch.com/journals

Evaluation of Random Early Detection and Adaptive Random Early Detection in Benchmark Scenarios

Rohit P. Tahiliani, Sagar Sachdeva, Sachin Hadke, Shane Sheehan, Eamonn O Nuallain

School of Computer Science & Statistics
Trinity College Dublin, Ireland.

Abstract

In this paper, we evaluate Random Early Detection (RED) and Adaptive RED (ARED) in Benchmark Scenarios as detailed in RFC 7928. RED is one of the early proposed AQM mechanisms, which attains high throughput and keeps average delay low. Moreover, ARED is an extension to RED which eliminates the parameter sensitivity to improve the performance of RED.

The results indicate that RED outperforms ARED in scenarios with abrupt changes in traffic load. ARED is known to reduce the packet drops and therefore, in rest of the scenarios it can be observed that ARED outperforms RED.

Keywords: Active Queue Management; Random Early Detection; Buffer-bloat.

1. Introduction

Increase in the delay sensitive applications such as real-time multimedia (e.g., voice, gaming, etc.) have given riseto new challenges in queue management. Presence of large unmanaged buffers in the Internet results in a problem referred to as Buffer-bloat [3]. The demand for the interactive delay-sensitive applications on the web is increasing. With the reduced memory costs and in an effort to reduce the packet loss, the vendors have increased the network buffers. This addresses the problem of TCP throughput but gives rise to increased latency. Queuing delay problems occur when the traffic from the delay-sensitive application along with capacity seeking traffic co-exist in the same bottleneck queue. Managing buffers is indispensable as passive or unmanaged buffers lead to a number of problems such as: lockout [1], global synchronization [2] and Buffer-bloat.

Active Queue Management (AQM) mechanisms are being widely deployed at the bottlenecks to reduce the queuing delay and to improve the link utilization. These mechanisms avoid congestion by proactively signalling the sender about the level of congestion by either dropping the packets or by marking them.

Random Early Detection (RED) [2] is one of the most widely deployed AQM mechanism in routers. Moreover, Explicit Congestion Notification (ECN) [4] is the marking mechanism used in conjunction with RED. The performance of RED is highly dependent on the configuration of parameters.

Adaptive RED (ARED) [5] is an extension to RED, which removes the parameter sensitivity of RED and needs only one user configurable parameter - desired target average queue length.

AQM mechanisms are an active area of research. Researchers thus far, have tested the AQM mechanisms in the scenarios of their choice. However, the Active Queue Management and Packet Scheduling Working Group (AQM WG) standardizes the scenarios that can be used to assess the applicability, performance, and deploy ability of an AQM. RFC7928 [6] details the characterization guidelines to achieve a fair comparison between the AQM schemes by avoiding certain pitfalls. We compare RED and ARED in the benchmark scenarios provided in RFC 7928.

The rest of the paper is organized as follows: Section II provides a background of RED and ARED algorithms. Section III details the simulation scenarios and results. Finally, Section IV concludes the paper and discusses the scope for future work.

2. Background

2.1 Random Early Detection

RED overcomes the limitations of drop-tail queues and ensures fairness amongst the participating end hosts. The performance of RED is dependent on four parameters. These are minimum threshold min_{th}, maximum threshold max_{th}, queue weight factor w_q for exponential weighted movingaverage and maximum drop probability max_p. The operatorsmust set these parameters based on several factors such astopology and bandwidth.

On packet arrival, RED calculates the average queue sizeavg. The avg is calculated as shown in Equation (1). RED queues the packet if avg is less than $min_{th}$and drops the packetif avg is greater than max_{th}.

When avg is between$min_{th}$and max_{th}, RED drops the packet randomly with drop probability p_d as calculated in Equation (2).

$$avg = \left((1 - w_q) \times\ oldavg\right) + (w_q \times\ cur_q) \quad (1)$$

where, $oldavg$ is the average queue size during previous packet arrival; cur_q is the size of the current queue.

$$p_d = \begin{cases} 0 & avg < min_{th} \\ \frac{avg - min_{th}}{max_{th} - min_{th}} \times max_p & min_{th} \le avg < max_{th} \\ 1 & avg \ge max_{th} \end{cases} \quad (2)$$

If avg is greater than max_{th}then RED drops all the packets. This nature of RED makes it aggressive and results in significant drop in throughput. Therefore, a modified RED named Gentle RED (GRED) is proposed in [7]. With GRED, $p_d$is increased linearly frommax_{th} to $2 \times max_{th}$. As a result, the number of packet drops with GRED are reduced.

2.2 Adaptive Random Early Detection

Adaptive RED (ARED) is a minor change to the existing RED algorithm. ARED aims to eliminate the parameter sensitivity of RED. With ARED, only the desired target average queue length needs to be set. The rest of the parameters are automatically tuned. This auto-tuning of max_p makes ARED robust. ARED adopts to the traffic load and keeps avg between themin_{th} and$max_{th}$and updates max_paccordingly.

ARED uses the Additive Increase Multiplicative Decrease (AIMD) approach to updatemax_p. Therefore, under sudden changes of traffic load the performance of ARED is conservative. However, in order to minimize the impact on performance, ARED restricts$max_p$to stay within the range (0.01,0.5).

Detailed guidelines on the several parameter settings for the ARED algorithm can be found in [5].

3. Results

This section details the scenarios under which RED and ARED have been compared. RFC 7928 provides an exhaustive set of scenarios to compare AQM schemes. AQM evaluation suite in ns-3 [8] implements the scenarios as described in RFC 7928. Therefore, we use it to analyze the performance of RED and ARED. Moreover, the implementation of RED and ARED can be accessed in traffic control module in NS-3. Covariance between throughput and delay are used as performance metrics. We have performed our experiments on NS-3.26 version of Moreover, all the scenarios discussed in this paper make use of dumbbell topology. The following are the scenarios we consider for comparison of RED and ARED.

3.1 TCP Friendly Sender with the Same Initial Congestion Window

This scenario corresponds to Section 5.1.1 of RFC 7928. It helps to evaluate on how an AQM scheme performs against a TCP-friendly transport sender with same initial congestion window. Sender A transfers data to Receiver B with an initial congestion window set to 3 packets. The transport protocol used is TCP NewReno [9]. Figure 1. shows the covariance between throughput and delay measured for the protocol under this scenario. It can be observed that ARED provides better good put as compared to RED. However, the variations in the queuing delay for ARED are more due to its adaptive nature to update the maximum probability. RED in this scenario outperforms ARED and provides a good control on queuing delay.

3.2 TCP Friendly Sender with Different Initial Congestion Windows

This scenario corresponds to Section 5.1.2 of RFC 7928. It helps to evaluate on how an AQM scheme performs against a TCP-friendly transport sender with different initial congestion window. We consider two types of flows. One is a non-application limited TCP NewReno flow. The second is an application-limited TCP NewReno flow with initial congestion window set to 3 or 10 packets. Figure (b) shows the covariance between throughput and delay measured for the protocol under this scenario. It can be observed that RED attains high goodput but at the cost of high variations in queuing delay. On the other hand, ARED offers a decent goodput with fair amount of variations in queuing delay.

3.3 Aggressive Transport Sender

This scenario corresponds to Section 5.2 of RFC 7928. It helps to evaluate on how an AQM scheme performs against an aggressive transport sender. For instance, in congestion control mechanisms such as Additive Increase Multiplicative Decrease (AIMD), a larger AI and/or MD part would make the transport protocol aggressive. We consider a single, non-application-limited TCP NewReno flow. Figure (c) shows the covariance between throughput and delay measured for the protocol under this scenario. We can observe that, ARED has a tighter constraint on queue delay, with a slight compromise on the goodput. ARED outperforms RED in keeping the queue delay low.

3.4 Unresponsive Transport Sender

This scenario corresponds to Section 5.3 of RFC 7928. It helps to evaluate on how an AQM scheme performs against an unresponsive transport sender such as UDP. AQM guidelines propose two scenarios. First, to evaluate the queue build up. Second, to evaluate the responsiveness fraction. Section 5.3 in [6] provides comprehensive information on this scenario. Figure (d) shows the covariance between throughput and delay measured for the protocol under this scenario. In this scenario, RED keeps the queue delay low. ARED in this scenario will have no effect due to the presence of unresponsive traffic.

3.5 Less than Best-Effort Transport Sender

This scenario corresponds to Section 5.3 of RFC 7928. It helps to evaluate on how an AQM scheme performs against a Less-than-Best-Effort (LBE) congestion control scheme such as LEDBAT [10]. LBE transport protocols are designed to have a smaller bandwidth and/or delay impact on standard TCP than standard TCP itself when they share a bottleneck with it [6]. Figure (e) shows the covariance between throughput and delay measured for the protocol under this scenario. It can be observed that, ARED provides a tighter control on the queue delay as compared to RED.

3.6 Mild Congestion

This scenario corresponds to Section 8.2.2. of RFC 7928. It helps to evaluate on how an AQM scheme performs under Mild Congestion. Section 8.2.1 in [6] details the guidelines to define the different levels of congestion. This scenario generates light load of incoming traffic. The transport protocol used is TCP NewReno. Figure (f) shows the covariance between throughput and delay measured for the protocol under this scenario. It can be observed that, in terms of Goodput both ARED and RED have a similar performance.

However, under light load the control on queuing delay with RED is better as compared to ARED.

3.7 Medium Congestion

This scenario corresponds to Section 8.2.3. of RFC 7928. It helps to evaluate on how an AQM scheme performs under Medium Congestion. This scenario generates medium load of incoming traffic. The transport protocol used is TCP NewReno. Figure (g) shows the covariance between throughput and delay measured for the protocol under this scenario. We can observe that ARED offers a good control on the queue length and a reasonably fair goodput.

3.8 Heavy Congestion

This scenario corresponds to Section 8.2.4. of RFC 7928. It helps to evaluate on how an AQM scheme performs under Heavy Congestion. This scenario generates heavy load of incoming traffic. The transport protocol used is TCP NewReno. Figure (h) shows the covariance between throughput and delay measured for the protocol under this scenario. The performance of ARED is similar to that in Scenario 5. In general, the performance of RED is good under mild congestion. Whereas, ARED performs good under medium and heavy congestion loads.

3.9 Varying Congestion Level

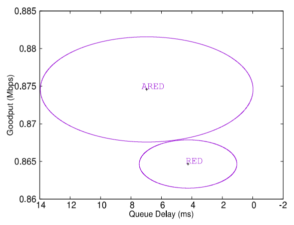
This scenario corresponds to Section 8.2.5. of RFC 7928. It helps to evaluate on how an AQM scheme performs when the congestion level is varied. For the timescale 0-20s the congestion is kept mild; 20-40s the congestion is medium and for 40-60s the congestion is heavy. Figure (i) shows the covariance between throughput and delay measured for the protocol under this scenario. We can infer that, ARED like in Scenario 5 & 6 provides a tighter control on the queue length.

3.10 Varying Available Capacity

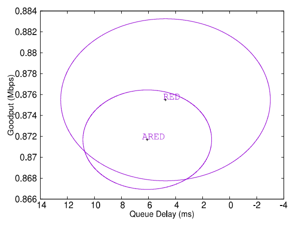
This scenario corresponds to Section 8.2.6. of RFC 7928. It helps to evaluate on how an AQM scheme performs under varying bandwidths. In this scenario, we vary the bandwidth between the bottlenecks routers. We consider two experiments. One, where the capacity varies within a large timescale.

Capacity is set to 100 Mbps for 0-20s for phase 1. For phase 2 i.e, 20-40s, the capacity is set to 10Mbps and so on. Figure (j) shows the covariance between throughput and delay measured for the protocol under this experiment. Second, where the capacity varies within a short timescale. Capacity is set to 100 Mbps for 0-100ms for phase 1. For phase 2 i.e, 100-200ms, the capacity is set to 10Mbps and so on. Figure (k) shows the covariance between throughput and delay measured for the protocol under this experiment. We know that ARED fails to perform well under sudden changes in traffic load. Therefore, in both the experiments we can observe that in comparison to ARED, RED offers a tight control on the Queue delay.

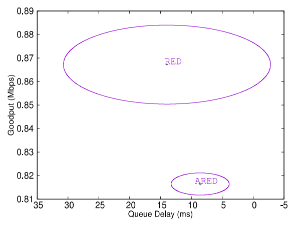
Figure 1. Comparison of RED and ARED under scenarios with varying traffic load and varying traffic type.



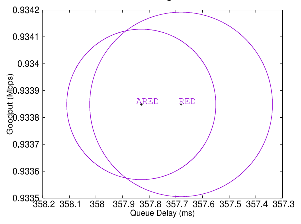
(a) Same Initial Congestion Window



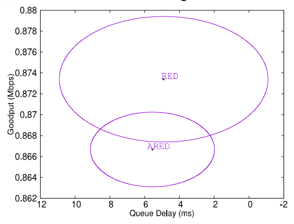
(b) Different Initial Congestion Window



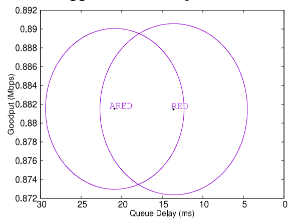
(c) Aggressive Transport Sender



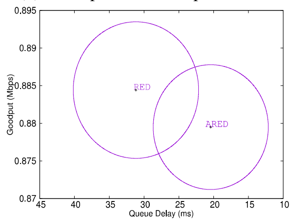
(d) Unresponsive Transport Sender



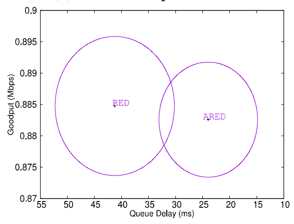
(e) LBE Transport Sender



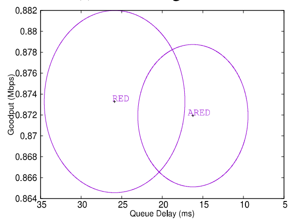
(f) Mild Congestion



(g) Medium Congestion



(h) Heavy Congestion



(i) Varying Congestion Level

4. Discussion and Conclusion

In this paper, we compare RED and ARED in benchmark scenarios as detailed in RFC 7928. It standardizes the scenarios that can be used to assess the applicability, performance and deployability of an AQM. We cover a wide range of scenarios to exhaustively test the performance of RED and ARED. We cover all the scenarios that are described in Section 5 and Section 8.2 of RFC 7928 [6]. We observe that the performance of RED is better than ARED in Scenario 1.1, Scenario 5 and Scenario 8. Performance of ARED degrades in these scenarios due to it's inability to adapt to the abrupt changes in traffic load.
However, ARED outperforms RED in the rest of the scenarios. This is attributed to ARED auto-tuning several parameters which makes it robust. We believe that the results presented in this paper can help the network operators and researchers to gain insights about the performance of RED and ARED under different scenarios. This can help them make decisions on the AQM to be deployed in their network devices.
As an extension to this work, we look forward to cover scenarios as proposed in Section 9 and Section 10 of RFC 7928 [6]. Furthermore, we plan to compare ARED with one of the recently proposed variant of RED, Curvy RED [11] in the benchmark scenarios.

References

[1] M. Hassan and R. Jain. High Performance TCP/IP Networking, volume 29. Prentice Hall, 2003.

[2] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. Networking, IEEE/ACM Transactions on, 1:397–413, 1993.

[3] R. Pan, P. Natarajan, C. Piglione, M. S. Prabhu, V. Subramanian, F. Baker, and B. VerSteeg. PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem. In High Performance Switching and Routing (HPSR), 2013 IEEE 14th International Conference on, pages148–155. IEEE, 2013.

[4] K. Ramakrishnan, S. Floyd, D. Black, et al. The addition of Explicit Congestion Notification (ECN) to IP, 2001.

[5] S. Floyd, R. Gummadi, S. Shenker, et al. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management, 2001.

[6] N Kuhn, P Natarajan, N Khademi, and D Ros. Characterization guidelines for active queue management (aqm). Technical report, 2016.

[7] Sally Floyd. Recommendation on using the gentle variant of red, march 2000. See http://www. icir. org/floyd/red/gentle. html, 2000.

[8] Network Simulator 3. https://www.nsnam.org, 2011.

[9] Tom Henderson, Sally Floyd, Andrei Gurtov, and Yoshifumi Nishida. The NewReno Modification to TCP's Fast Recovery Algorithm. Technical report, 2012.

[10] Sea Shalunov, Greg Hazel, Janardhan Iyengar, and Mirja Kuehlewind. Low extra delay background transport (ledbat). Technical report, 2012.

[11] Bob Briscoe. Insights from Curvy RED (Random Early Detection), July 2015.