ISSN:2394-3386

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 2, Special Issue
September 2015

Three-section Random Early Detection (TRED)

Keerthi M
PG Student
Federal Institute of Science and Technology, Angamaly, Kerala

Abstract

There are many Active Queue Management (AQM) mechanisms for Congestion Control in the network communication. Random Early Detection or RED is one of the effective AQM mechanisms. However, RED is particularly sensitive to the traffic load and parameters of the scheme itself. This paper presents Three-Section Random Early Detection or TRED, a minimal adjustment to RED. In TRED the packet dropping probability function is divided into three sections to distinguish between light, moderate and high loads in order to achieve a tradeoff in the delay and throughput. The NS2 simulation results show that TRED effectively improves the drawbacks of RED to achieve better throughput and decreased delay and hence to achieve better congestion control. Also, only very little work needs to be done to migrate from RED to TRED on Internet routers as only the packet dropping probability profile is adjusted.

Introduction

The quality of service of a network is affected by the congestion in the network. Network congestion occurs when the demand for a resource; say the link bandwidth, exceeds the capacity of the resource. Results of congestion include a high delay, wasted resources, and even global synchronization, i.e., the throughput drops to zero, and the response time tends to infinity. To ensure the quality of service of the network congestion has to be controlled or more ideally, avoided.

A couple of mechanisms have been implemented in order to prevent network congestion or reduce its effects. The Active Queue Management (AQM) mechanism is a congestion control mechanism that is implemented in the internet routers. In internet routers, active queue management (AQM) is responsible for the intelligent drop of network packets inside a buffer associated with a network interface controller (NIC), when that buffer becomes full or gets close to becoming full, with the larger goal of reducing network congestion. There are several active queue management mechanisms among which Random Early Detection (RED) , proposed by Floyd and Jacobson is a typical example. RED is a significant improvement over Drop Tail [1]. Drop Tail simply drops the incoming packets once the buffer is full. RED, on the other hand detects the incipient congestion and notifies the transmission control protocol (TCP) by probabilistically dropping packets before the queue when a router fills up. RED works by monitoring the average queue size. Also there are a few parameters, minimum threshold, maximum threshold, *P*max, etc that are set by RED. As the average queue size varies between the minimum and maximum thresholds, the packet dropping probability linearly changes between zero and maximum drop probability *P*max. Thus, the packet dropping probability function is linear to the change of the average queue size. If the average queue size exceeds the maximum threshold, all arriving packets are dropped. The advantage of RED over simple Drop Tail is that RED does not drop all the incoming packets unlike Drop Tail which drops all the incoming packets once the queue is full. But RED has a major disadvantage, RED is particularly sensitive to the traffic load and the parameters of the scheme itself.

Related Works

Adaptive RED (ARED), Gentle RED (GRED), HRED, PI, BLUE, etc are some of the schemes proposed over the past few years that are enhancements of RED which try to overcome the weakness of RED.

Floyed et al. proposed another AQM method called adaptive RED [2]. In this the average queue length is maintained within a target range. For this the parameter *maxp* is adapted using an additive increase multiplicative decrease policy. *maxp* is adapted not just to keep the average queue size between *minth* and

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 2, Special Issue
September 2015

maxth, but to keep the average queue size within a target range half way between *minth* and *maxth*. *maxp* is adapted slowly, over time scales greater than a typical round-trip time, and in small steps. By maintaining a predictable average queue size ARED reduces RED's parameter sensitivity. However, the ARED performance is lower than that of RED when facing a complex network environment.

Gentle RED or GRED replaces RED's discontinuity of the packet dropping probability from *P*max to 1 by a gentle slope. The drop probability increases linearly between *maxth* and the buffer size with a slope of (1 - *maxp*)/*maxth* . Obviously, RED as well as GRED has to drop an arriving packet if the instantaneous queue size equals the total buffer size. However, parameter tuning has been one of the main limitations of this scheme because many parameters must be manually tuned according to different network conditions.

Hyperbola RED (HRED) uses the hyperbola as the drop probability curve. It is a simple but efficient AQM algorithm. Although HRED shows low dependency on parameter setting, it suffers from slow responsive issue.

BLUE proposed by Feng *et al.* [3] uses packet loss and links idle events to manage congestion. Although BLUE is relatively simple to achieve and its packet loss is low, the long delay easily causes network congestion in certain situations.

Another set of algorithms apply control theory to model TCP with RED dynamics. These algorithms develop feedback mechanisms and control laws to achieve an optimal and stable operating point. Control theory is successfully introduced into AQM, and PI [4], PID, PD algorithms are proposed, which uses instantaneous queue length as congestion measurement index. PI (Proportional Integral algorithm) regards queue reference value and instantaneous queue length as controller inputs. In [7], a PD controller is added to RED. The PD controller adapted the RED parameter *P*max that would more systematically improve the queue length stabilization attempted by ARED. To provide a faster response than a PI controller, a PID controller is proposed in [8] as another viable AQM.

Methodology

RED, a queue-based AQM mechanism and designed heuristically, is the first developed AQM scheme to be deployed in TCP/Internet Protocol networks for the replacement of Drop Tail. The initial objectives of RED is to detect incipient congestion, to achieve fairness among flows with differing levels of burstiness, to control the queue lengths to low values to minimize queuing delay, to prevent the correlation of packet drops and global synchronization, to minimize packet loss, and to provide a high link utilization [9].

The RED scheme drops packets with a certain probability by computing the average queue length (ave) to notify traffic sources about the early stages of network congestion. The average queue length is calculated as the result of the exponentially weighted moving average (EWMA) which is expressed as follows

$$\text{ave} = (1 - wq)\,\text{ave} + wq\,q. \qquad (1)$$

In the aforementioned formula, q is the instantaneous queue length, and $wq \in [0, 1]$ is the weighting factor.

In addition to EWMA weight *wq*, RED has three more parameters, i.e., minimum threshold minth, maximum threshold maxth, and the maximum dropping probability *P*max. RED does not drop any incoming packets if the average queue length is below minth. However, if the average queue length increases above minth but is below maxth, RED drops incoming packets with a probability proportional to the average queue length linearly. When the average queue length exceeds maxth, all the arriving packets are dropped. Fig. 1 depicts the RED control function curve, which is the packet dropping probability as a function of the average queue length.

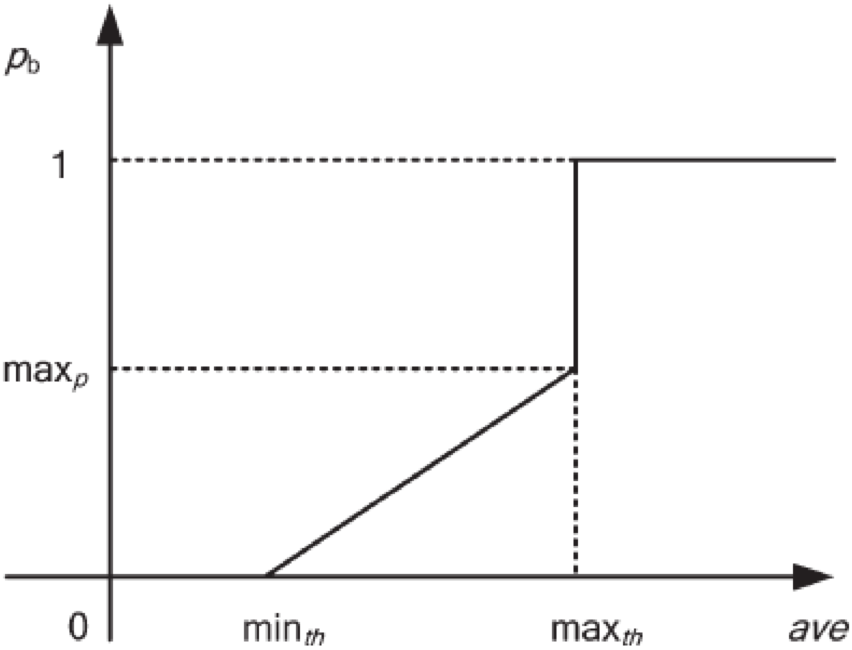

Fig.1. RED's packet dropping probability curve

The packet dropping probability being the function of the average queue length is calculated by the following equations

$$Pb = \begin{cases} 0, & ave \in [0, minth] \\ maxp \left(\frac{ave - minth}{maxth - minth}\right), & ave \in [minth, maxth) \\ 1, & ave \in ave \in [maxth, +\infty) \end{cases} \quad (2)$$

RED eliminates Drop Tail defects such as deadlock, full queues, and global synchronization, but there are still many deficiencies in RED. Forced drops or link underutilization will occur without reasonably set RED parameters, meaning RED is very sensitive to parameter setting. Multiple TCP global synchronization will also occur in certain traffic load environments, leading to queue oscillation, throughput reduction, and an intensified delay jitter.

This work divides the average queue length ave $\in$ (min_{th},max_{th}) depicted in figure 1 into three equal sections with the interval size of $\Delta = (max_{th}-min_{th})/3$.

In the interval ave $\in$ (min_{th}, $min_{th} +\Delta$) the load is considered to be low and so is the congestion in the network. Thus, a smaller packet dropping probability than RED is required in this interval.

In the interval ave $\in$ [$min_{th} +\Delta$, $min_{th} + 2\Delta$) the load and the network congestion are considered to be moderate; thus, the original RED linear packet dropping probability can be maintained as such.

With a high load and heavy congestion in the interval ave $\in$ [min_{th}, $+2\Delta$, max_{th}) a larger packet dropping probability than RED should be used.

Therefore, the expected TRED packet dropping probability function is depicted in Fig. 2 based on the assumptions made above.

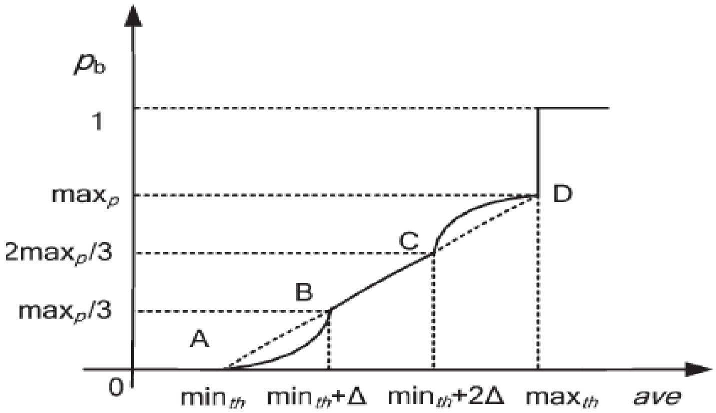

Fig.2. TRED's packet dropping probability curve

Set *A*, *B*, *C*, and *D* as the four points in TRED's packet dropping probability curve, wherein the coordinate of point *A* is (minth, 0), that of point *B* is (min_{th} +Δ, max_p /3), that of point *C* is (min_{th} +2Δ, 2max_p /3), and that of point *D* is (max_{th}, max_p).

Define the expressions of curve *AB* and curve *CD* as

$$p1=a1(ave-b1)3 + c1, \ ave \in (min_{th}, min_{th}+\Delta) \qquad (3)$$

$$p2=a2(ave-b2)3 + c2, \ ave \in [min_{th}+2\Delta, max_{th}) \qquad (4)$$

Here, setting point *A* as the vertex of curve *AB*, then the derivative of curve *AB* at point *A* is 0, and $a1 \neq 0$; then, we have

$$b1 = min_{th} \qquad (5)$$

The coordinates of points *A* and *B* are substituted into (3), respectively, and combined with formula (5) to give

$$P1 = 9 \, maxp \left(\frac{ave-minth}{maxth-minth}\right)^3, \quad ave \in (\, min_{th}, min_{th}+\Delta) \qquad (6)$$

Setting point *D* as the vertex of curve *CD*, then the derivative of curve *CD* at point *D* is 0, and $a2 \neq 0$; then, we have

$$b2 = max_{th} \qquad (7)$$

The coordinates of points *C* and *D* are substituted into (4), respectively, and combined with (7) to give

$$P2 = 9 \, maxp \left(\frac{ave-minth}{maxth-minth}\right)^3 + maxp, \quad ave \in [minth+ 2\Delta, maxth) \qquad (8)$$

The packet dropping function of TRED can be summarized as follows

$$Pb = \begin{cases} 0, & ave \in [0, min_{th}] \\ 9 \, max_p \left(\frac{ave-minth}{maxth-minth}\right)^3, & ave \in (\, min_{th}, min_{th}+\Delta) \\ max_p \frac{ave-minth}{maxth-minth}, & ave \in [min_{th}, max_{th}) \\ 9 \, max_p \left(\frac{ave-minth}{maxth-minth}\right)^3 + max_p, & ave \in [min_{th}+ 2\Delta, max_{th}) \\ 1, & ave \in ave \in [max_{th}, +\infty) \end{cases}$$

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 2, Special Issue
September 2015

Analysis by Simulation

For simulations with the ns-2 simulator [10] the topology shown in Fig.3 was used.

FTP traffic with n TCP flows is simulated in the topology. There are S1-Sn hosts on the left side as sources and D1-Dn hosts that act as sinks in the right side. Each host has a link-capacity of 5 Mb/s to the routers. The propagation delay to their connecter router R1 to R2 is 20 ms.. The two routers R1 and R2 are connected via a bottleneck link with capacity 5Mb and propagation delay 10 ms where RED and TRED are performed.

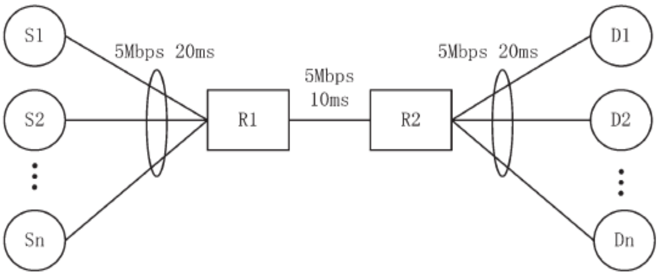

Fig.3. Simulation Topology

RED parameters tend to be set according to the industry [11]. The thresholds for the linear packet dropping probability function are min_{th} = 24 packets and max_{th} = 72 packets. The buffer size is 120 packets. The simulation time is 200 s, with the maximum packet dropping probability set to $max_p = 0.1$ and $w_q = 0.002$.

To validate the improvement achieved by TRED, this paper compares the performance of TRED with that of RED in low, moderate, and high-load network environments. The router buffer size was varied from 110 to 200 with a granularity of 10 packets.

Simulation 1: Low load scenario

There are 6 TCP connections in this scenario. Table 1 makes a comparison of the throughput obtained by TRED and RED. Table 2 makes a comparison of delay exhibited by both the schemes.

Table 1. Throughput comparison in a low load scenario

Buffer Size	RED	TRED
110	13.7464	13.6051
120	13.8539	13.6699
130	13.8813	13.6087
140	13.8945	13.5203
150	13.8734	13.6634
160	13.7898	13.6113
170	13.8485	13.5799
180	13.783	13.6032
190	13.7848	13.6658
200	13.7283	13.5926

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 2, Special Issue
September 2015

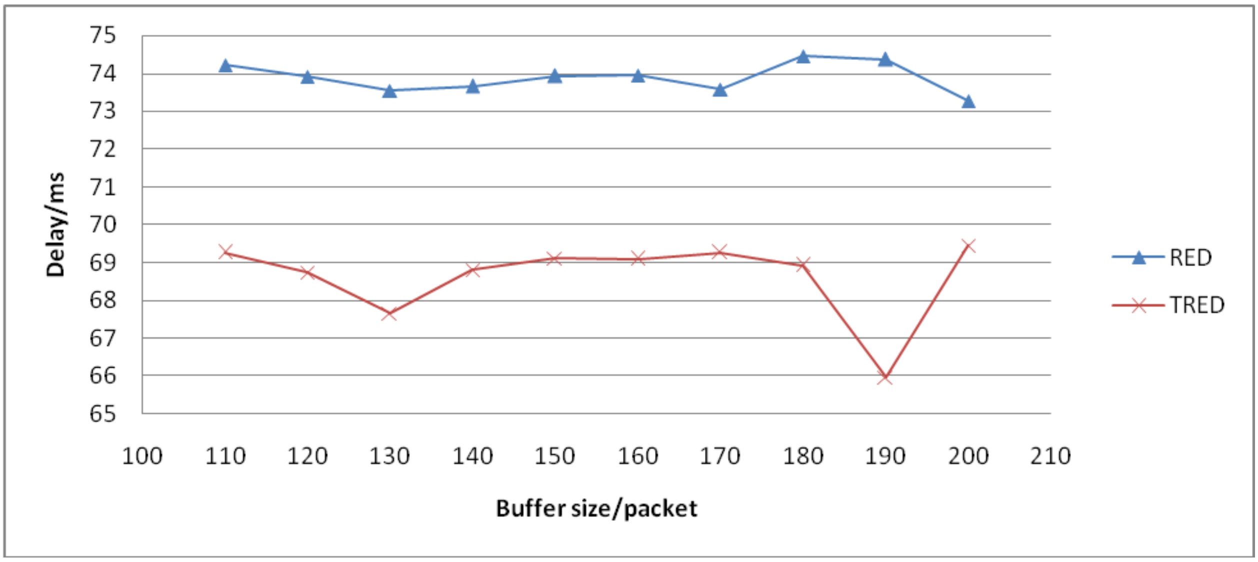

Fig.4. Throughput changes with buffer size in a moderate load scenario

Table 2. Delay comparison in a low load scenario

Buffer Size	RED	TRED
110	67.8689	69.7795
120	67.7220	69.7315
130	67.7872	69.9577
140	67.3836	70.0269
150	67.6302	69.6738
160	67.9130	69.7985
170	67.8581	69.9583
180	67.8787	69.7970
190	67.9591	69.6464
200	68.0325	69.7889

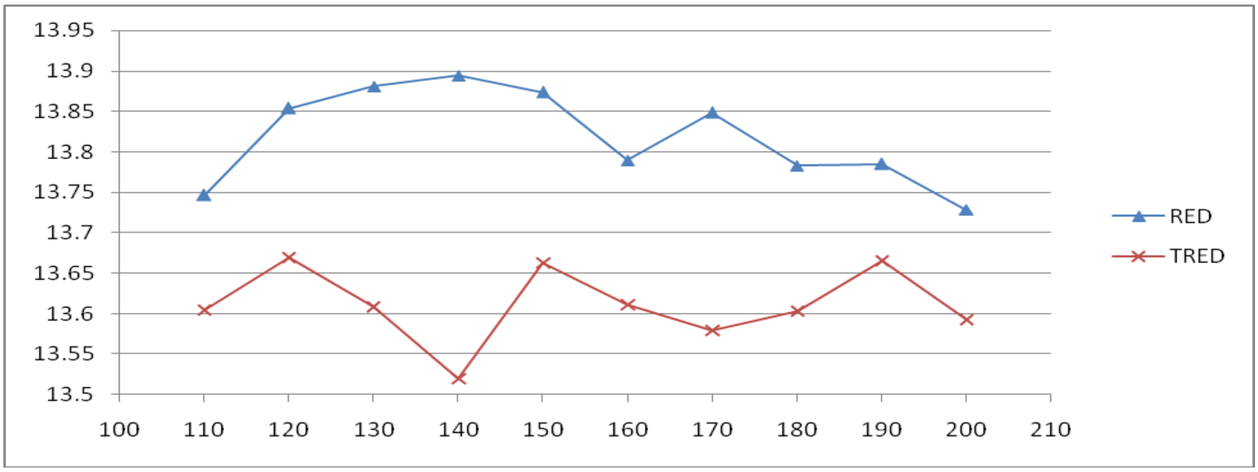

Fig.5. Delay changes with buffer size in moderate load scenario

The delay of TRED is larger than that of RED with the buffer size changing throughout in the range 110 to 200. As for the throughput, it increases in case of RED. This is because the TCP source will send more packets with the packet dropping probability reducing. The average queue length of TRED will then increase while the router receives more packets, leading to larger delay and throughput than RED.

Simulation 2: Moderate load scenario

There are 40 TCP connections in this scenario. The results of the simulation are shown in tables 3 and 4.

Table.3. Throughput comparison in a moderate load scenario

Buffer Size	RED	TRED
110	15.8945	15.9793
120	15.9196	15.9925
130	15.9287	15.9926
140	15.9215	15.9590
150	15.9332	15.9611
160	15.9351	15.9828
170	15.9154	15.9782
180	15.9125	16.0125
190	15.9125	15.9807
200	15.9191	15.9931

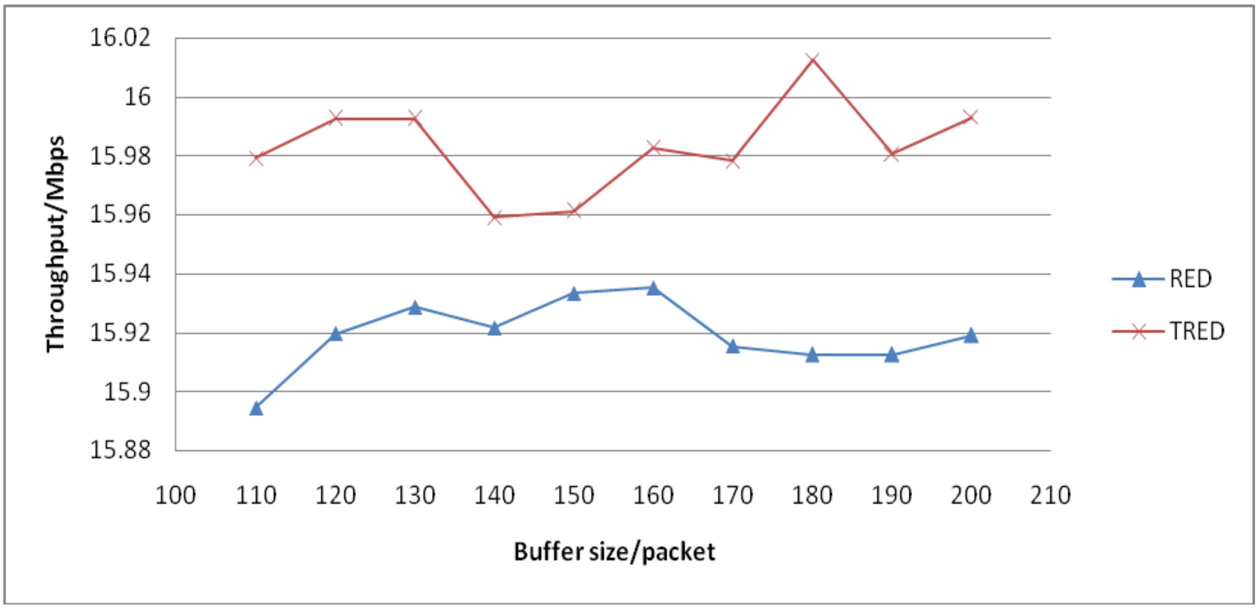

Fig.6. Throughput changes with buffer size in a moderate load scenario

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 2, Special Issue
September 2015

Table.4. Delay comparison in a moderate load scenario

Buffer Size	RED	TRED
110	73.7211	70.0719
120	73.7976	70.1697
130	74.0758	70.3123
140	73.7627	70.3698
150	73.7172	70.401
160	73.9717	69.8397
170	73.7671	70.1048
180	74.0585	70.8553
190	73.7421	69.8391
200	73.8403	70.2689

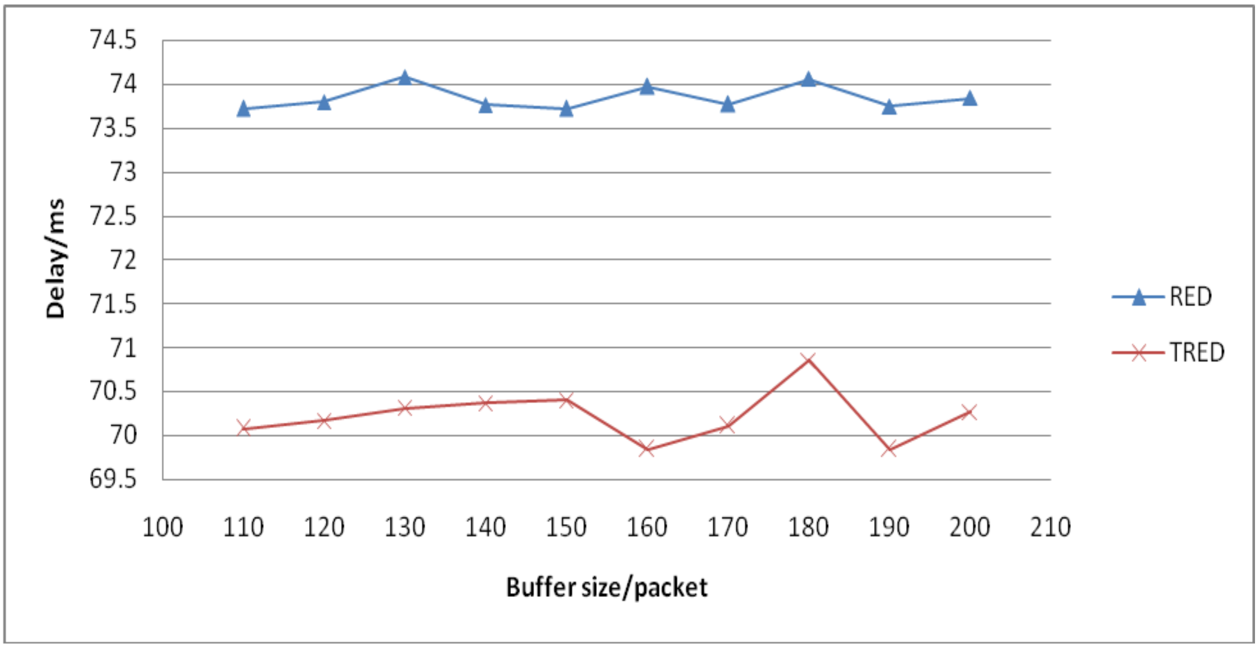

Fig.7. Delay changes with buffer size in moderate load scenario

It can be observed from tables 3 and 4 that TRED gives better throughput in most cases of varying buffer size when compared to RED. The throughput is almost the same for both TRED and RED in some cases. The delay exhibited by TRED is considerably less and better than what is shown by RED. That is, when the traffic load is moderate, the performance of TRED and RED are consistent in some cases or TRED performs better than RED.

Simulation 3: High load scenario

There are 80 TCP connections in this scenario. The results of the simulations are shown in tables 5 and 6.

Table.5. Throughput comparison in a high load scenario

Buffer Size	RED	TRED
110	16.3461	16.4823
120	16.3622	16.4659
130	16.3424	16.4543
140	16.3487	16.4415
150	16.3711	16.4699
160	16.3768	16.4492
170	16.3846	16.5042
180	16.3337	16.4622
190	16.3452	16.4822
200	16.3636	16.4448

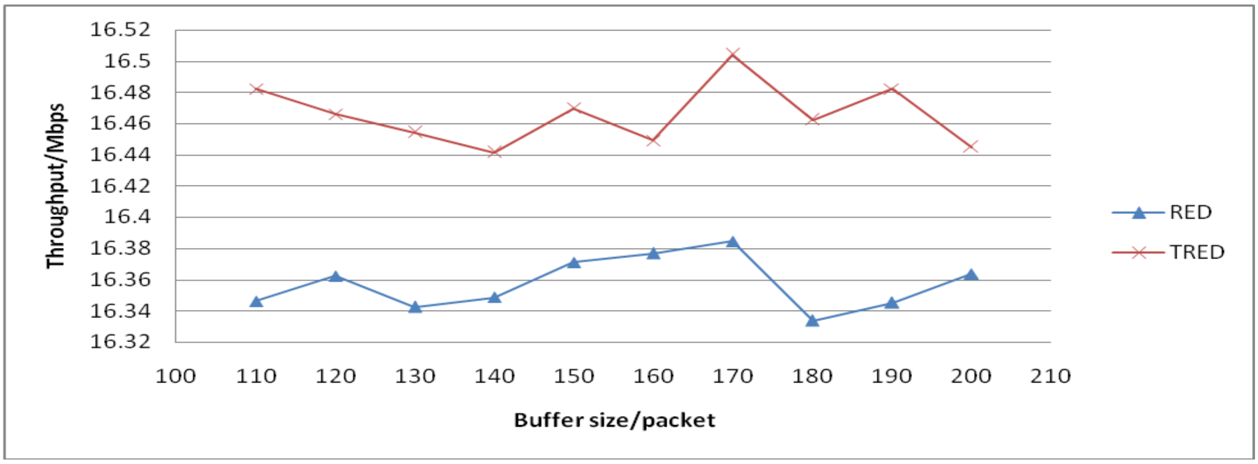

Fig.8. Throughput changes through buffer size in high load scenario

Table.6. Delay comparison in a high load scenario

Buffer Size	RED	TRED
110	74.2256	69.265
120	73.9178	68.7432
130	73.5475	67.6628
140	73.6541	68.7994
150	73.9255	69.1131
160	73.9403	69.096
170	73.5754	69.2562
180	74.4538	68.9219
190	74.3663	65.9581
200	73.2652	69.4423

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 2, Special Issue
September 2015

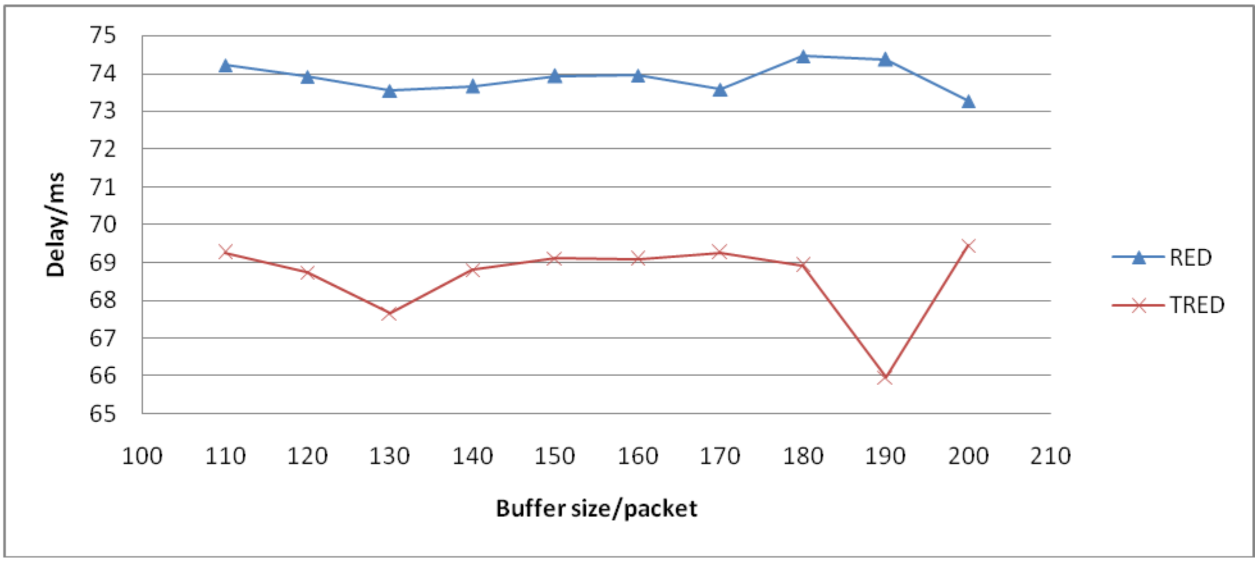

Fig. Delay changes through buffer size in high load scenario

TRED exhibits a higher throughput in all the cases of varying buffer size. The delay of TRED reduces because the TCP sources will send fewer packets with the packet dropping probability increasing. Thus, it can be concluded that TRED performs better compared to RED in a high load scenario.

Conclusion

This paper presents TRED which aims at improving the performance of RED in terms of throughput and delay in low, moderate and high load scenarios. TRED divides RED's average queue length between the two thresholds in three different intervals. The packet dropping probability in theses three intervals are also different that helps in adapting to different traffic loads. The NS2 simulations conducted efficiently proves the performance of TRED better than that of RED in terms of throughput and delay. TRED enhances the ability to regulate network congestion, improving network resource utilization and the scheme's stability.

In order to improve the result even more, the performance of TRED with ECN could be studied as future work. A lot of research has proved that AQM with ECN performs more efficiently than without ECN.

References

[1] R. Shorten, C. King, F. Wirth, and D. Leith, "Modeling TCP congestion control dynamics in drop-tail environments," *Automatica*, vol. 43, no. 3, pp. 441–449, Mar. 2007.

[2] J. Wang, L. Rong, and Y. Liu, "A robust proportional controller for AQM based on optimized second-order system model," *Comput. Commun.*, vol. 31, no. 10, pp. 2468–2477, Jun. 2008.

[3] W. Feng, K. G. Shin, D. D. Kandlur, and D. Saha, "The BLUE active queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 10, no.4, pp. 513–528, Aug. 2002.

[4] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. 20th Annu. Joint Conf. IEEE INFOCOM*, 2001, vol. 3, pp. 1726–1734.

[5] Chen-Wei Feng, Lian-Fen Huang, Cheng Xu, and Yao-Chung Chang,"Congestion control scheme performance analysis based on non-linear RED" IEEE Systems,

[6] R. Adams, "Active queue management: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1425–1476, 2013.

[7] J. Sun, K. T. Ko, G. Chen, S. Chan, and M. Zukerman, "PD-RED: To improve the performance of RED," *IEEE Commun. Lett.*, vol. 7, no. 8, pp. 406–408, Aug. 2003.

[8] F. Yanfie, R. Fengyuan, and L. Chuang, "Design a PID controller for active queue management," in *Proc. 8th IEEE ISCC*, 2003, pp. 985–990.

[9] S. Ryu, C. Rump, and C. Qiao, "Advances in Active Queue Management (AQM) based TCP congestion control," *Telecommun. Syst.*, vol.25, no. 3/4, pp. 317–351, Mar. 2004.

[10] Network Simulator ns-2, see http: / /www. isi .edu/nsnam/ns /.

[11] L. Hu and A. D. Kshemkalyani, "HRED: A simple and efficient active queue management algorithm," in *Proc. IEEE 13th ICCCN*, 2004, pp. 387–393.