International Journal on New Computer Architectures and Their Applications (IJNCAA) 2(2): 342-352
The Society of Digital Information and Wireless Communications, 2012 (ISSN: 2220-9085)

Dynamic Weight Parameter for the Random Early Detection (RED) in TCP Networks

Nabhan Hamadneh, David Murray, Michael Dixon, and Peter Cole
School of Information Technology, Murdoch University, WA, Australia
{n.hamadneh, d.murray, m.dixon, p.cole}@murdoch.edu.au

ABSTRACT

This paper presents the Weighted Random Early Detection (WTRED) strategy for congestion handling in TCP networks. WTRED provides an adjustable weight parameter w_q to increase the sensitivity of the average queue size in RED gateways to the changes in the actual queue size. This modification, over the original RED proposal, helps gateways minimize the mismatch between average and actual queue sizes in router buffers. WTRED is compared with RED and FRED strategies using the NS-2 simulator. The results suggest that WTRED outperforms RED and FRED. Network performance has been measured using throughput, link utilization, packet loss and delay.

KEYWORDS

TCP, RED, WTRED, TCP Performance, Queue Management.

1 INTRODUCTION

As the Internet has evolved and the number of users has dramatically increased, new techniques should be developed to grant fair resource allocation between users.

Congestion [1] is created when demand exceeds the available capacity. Due to uncoordinated resource sharing, the Internet has suffered from the problems of long delays in data delivery, wasted resources due to lost or dropped packets, and even possible congestion collapse [2]. Therefore, congestion control is a mandatory function that must be considered when designing a new TCP variant.

Queue management in intermediate routers has been the most efficient way to combat congestion over the last three decades. The traditional queue management technique, used for this purpose, is First In First Out (FIFO). Tail Drop (TD) was the earliest congestion control strategy to apply this technique. TD has numerous problems that are fixed in the Active Queue Management (AQM) approach.

Tail drop problems are caused by buffer overflows, and the delayed reaction to congestion. The AQM approach provides techniques to predict congestion and avoid buffer overflow by reducing the sending rate. Random Early Detection (RED) is the most popular strategy that adopts the AQM. However, RED has numerous problems. Many RED-based strategies have been proposed to rectify these problems [3-5]. This work discusses the problems associated with the RED-based strategies and presents new AQM technique for congestion handling.

The article is organized as follows: Section 2 describes the traditional congestion control strategies. RED and AQM are described in section 3. Section 4 describes the Refined Random Early Detection (FRED). Section 5 proposes the new Weighted RED (WTRED) strategy. Simulation and discussion is presented in section 6; and section 7 concludes the paper.

2 TRADITIONAL CONGESTION CONTROL STRATEGIES

TD drops packets from the tail of the queue. It has been shown that network performance under TD is degraded and causes four serious problems:

1. Full queue [1]: This problem occurs when a gateway continually sends full queue signals to sources for an extended period of time. In Fig. 1, a buffer of size 64 packets is full throughout the majority of the network operation time. In addition to the long delays associated with large queues, TD will penalize some connections by inequitably dropping packets. This will cause unfair resource allocation, which is illustrated in Fig. 2. In this figure, connection 2's window size is always lower than the other connections.
2. Lock out [1]: This problem occurs when TD allows a few connections to monopolize the whole buffer space. In Fig. 2, connection 4 receives more link bandwidth than the other connections in the network.

3. Global synchronization [6]: This problem occurs when all TCP senders reduce their sending rate simultaneously, reducing network throughput [6]. Fig. 3 shows TD algorithm causing global synchronization. Fig. 4 shows 5 seconds of global synchronization for the same scenario between time 20s to 25s.
4. Bias against bursty traffic [6]: Bursty connections receive more drops than other connections in the same network.

Early Random Drop (ERD) was designed to solve the problems of the TD strategy. When the actual queue size reaches a predefined threshold, ERD picks a random packet from the queue for drop. Table 1 illustrates the algorithm of ERD.

Table 1, Early Random Drop (ERD)'s Algorithm, see [7] for more details.

```
if(queue_length > drop_level)
then
if (get.random( ) <
drop_probablity)
then
drop(packet)
```

Early Random Drop solved the problems of the TD. Despite these advantages; some performance problems persist under ERD. ERD motivated the creation of the Random Early Detection (RED); which represents a new generation of active queue management techniques.

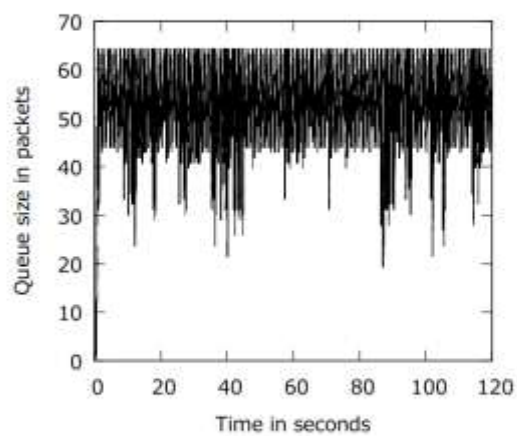


Figure 1, Full queue problem of TD strategy.

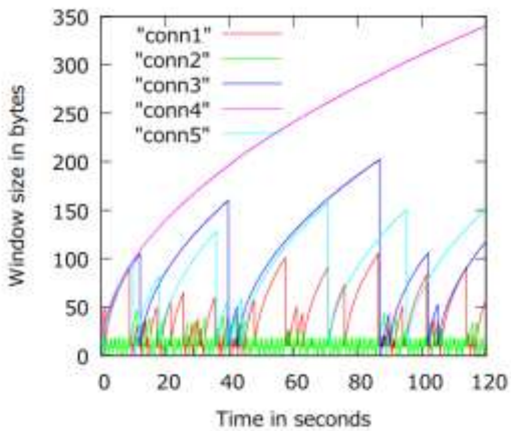


Figure 2, Lock out problem of TD strategy.

3 AQM and RED

Active Queue Management (AQM) is an approach to preempt congestion. RED is a specific AQM implementation. It maintains an Exponentially Weighted Moving Average (EWMA) and two predefined thresholds; the maximum threshold (max_{th}) and minimum threshold (min_{th}). RED monitors the average queue size instead of monitoring the actual queue size.

3.1 RED Algorithm

RED divides the router queue into three areas. If the average queue size (avg), Eq. (1), is less than the minimum threshold; then the network is steady and no action is necessary. When avg exceeds the minimum threshold but is less than the maximum threshold, then RED drops packets with probability (p_a) Eq. (2). If the average exceeds the maximum threshold, then RED drops every arriving packet.
RED is supposed to solve the traditional problems of queue management techniques. The success of RED as a queue management strategy is debated [8-11]. This article highlights two main problems of RED; which are the parameter configurations and the mismatch between the average and actual queue sizes.

$$avg = \left(1 - w_q\right) * avg + w_q * q \quad (1)$$

$$p_a = p_b(1/(1 - count * p_b) \quad (2)$$

$$p_b = max_p \frac{avg - min_{th}}{max_{th} - min_{th}} \quad (3)$$

Where:
avg: Average queue size
w_q: Weight parameter, $0 < w_q <= 1$
q: Current queue size
p_b : Immediately marking probability
max_p: Maximum value of p_b.
min_{th}: Minimum threshold.
max_{th}: Maximum threshold.
p_a: Accumulative drop probability
count: Number of arrived packets since the last dropped one

3.2 Disadvantages of RED

3.2.1 Parameter Configuration

The parameters of RED are very sensitive and erroneous configuration will degrade RED performance to the level of TD.

Threshold: The values of the minimum and maximum thresholds are assigned depending on the desirable actual average of the queue size. The maimum threshold represents the maximum average queue size that is allowed in the queue. Hence, higher maximum thresholds will increase delays. On the other hand, a lower maximum threshold will decrease throughput. In addition, higher minimum thresholds increase throughput and link utilization [6]. WTRED, uses different criteria to configure these parameters. This is further detailed in Sec. 5.2.

Current drop probability (p_b): This probability is a function of the maximum drop probability max_p, Eq. (3). A higher max_p parameter will result in a higher drop rate. Section 6.3 shows how this can cause problems with FRED's implementation.

Weight parameter: This parameter is used to calculate the average queue size. It reflects the sensitivity of the average to the actual changes in the queue size. It takes values from zero to one. Setting the weight parameter to larger values, means that fewer packets are required to increase the average from *A* to *B*. For instance: if a RED gateway with a weight parameter of 0.001 needs 60 packet arrivals to increase the average queue size from 6 to 10, then the same gateway with 0.003 weight parameter will need fewer packets (40) to increase the average from 6 to 10.

3.2.2 The mismatch between the average and actual queue sizes

The macroscopic behavior of a router reffers to the stable dynamics of the *average queue size*. These stable dynamics reflect the long-term behavior of a router. In contrast, the short-term dynamics of the *actual queue size* are called the microscopic behavior of a router.

There are some studies that showed variant dynamics between the average and actual queue sizes [3-5]. For instance, when a traffic burst arrives at the RED gateway, the actual queue size is rapidly increased, resulting in buffer overflow. If the queue weight parameter is too small then the average queue size will be increased slowly, despite accumulating congestion. Sources will reduce their sending rate after a congestion signal is triggered due to packet drop at the gateway. After the congestion is recovered and the actual queue size is reduced, the average queue size will be high due to previous peaks in the actual queue size. Therefore, RED will continue packet dropping even after the congestion has been recovered which unfairly penalizes new packets.

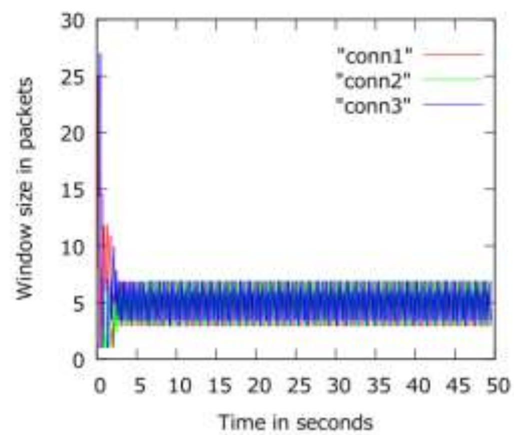


Figure 3, TD global synchronization.

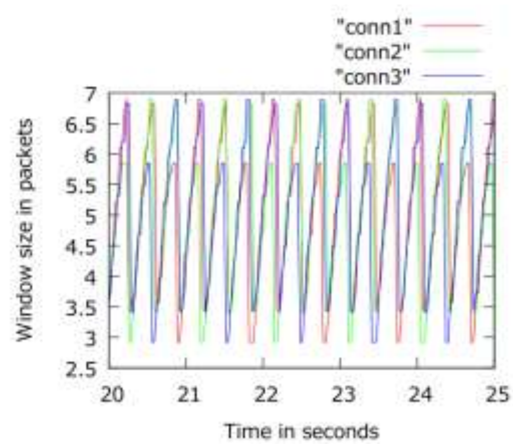


Figure 4, TD global synchronization (x 50).

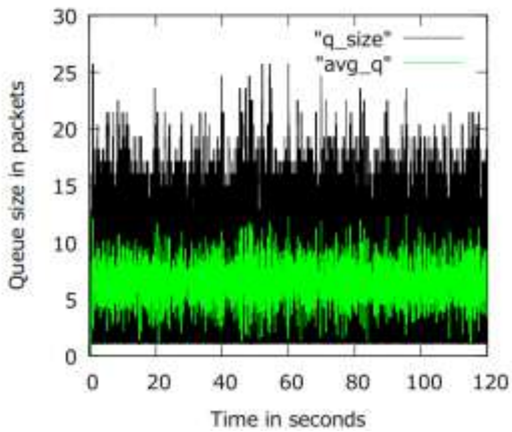


Figure 5, Average and actual queue sizes on a RED gateway.

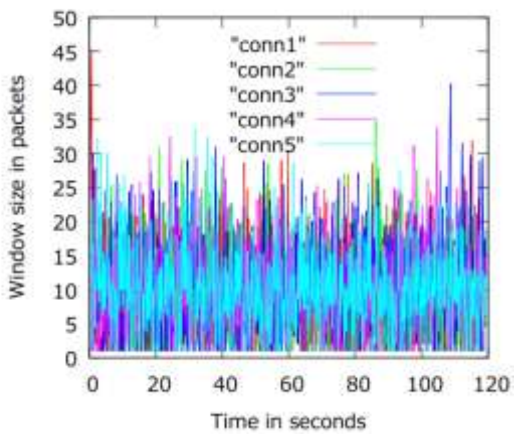


Figure 6, Congestion window size on a RED gateway.

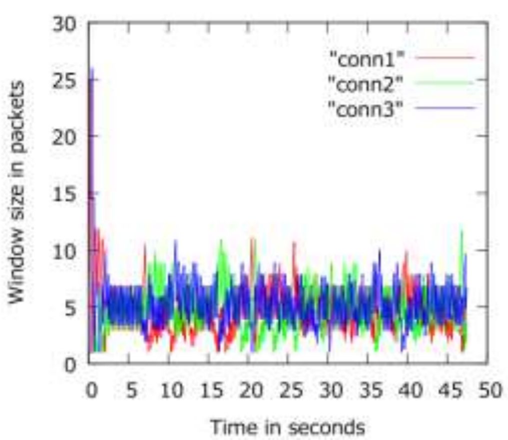


Figure 7, Congestion window size on a RED gateway without global synchronization.

4 FRED

Refined Random Early Detection (FRED) [12] is a RED modification that uses a dynamic weight w_q and max_p to control congestion. It divides the router's buffer space between the maximum threshold and minimum threshold into six sub-phases. As the average increases from the minimum threshold to the maximum threshold, the max_p is increased. A different value is assigned for the different sub-phases illustrated in Table 2.

Table 2, The maximum drop probability for FRED's sub-phases.

Subphase	**1**	**2**	**3**	**4**	**5**	**6**
$\boldsymbol{max_p}$	max_p	$2max_p$	$4max_p$	$6max_p$	$8max_p$	$10max_p$

In addition to a dynamic max_p parameter, FRED maintains a dynamic weight parameter w_q. It also makes use of a third threshold called the warn threshold. The weight parameter is adjusted whenever the actual queue size exceeds the warn threshold. FRED normally assigns half of the buffer size to this value. After the actual queue size exceeds the warn threshold, it has to go through another six sub-phases, but this time, with different weight parameters for each sub-phase. The weight values for these sub-phases are illustrated in Table 3. In order to apply the weight parameter sub-phases, the actual queue size must be greater than the average queue size.

Table 3, The weight parameter for FRED's sub-phases.

Subphase	1	2	3	4	5	6
w_q	w_q	$4w_q$	$8w_q$	$12w_q$	$16w_q$	$20w_q$

RED calculates the average queue size for every packet arrival. FRED extends RED by additionally calculating the average queue size for every packet departure.

5 WTRED

The optimal value for the weight parameter, in RED, is still an open issue. This parameter can take values from 0 to 1. However, the recommended value for this parameter is 0.002 [6]. It is the same for the maximum and minimum thresholds. The recommended values are 5 and 15 packets for the minimum and maximum thresholds respectively [6].
Section 5.1 investigates the weight parameter through simulation. The target is the optimal range of the weight parameter that generates the best network performance. Throughput, link utilization, delays and loss parameters have been evaluated to determine this range. Accordingly, a new parameter configuration for RED is deduced and used to motivate the design of our proposal scheme, WTRED.

5.1 WTRED Motivations

The maximum and minimum thresholds in RED divide the buffer into three main areas. The first area lies between empty queue and the minimum threshold. The second area is the area between the minimum threshold and maximum threshold. The third area is the area between the maximum threshold and buffer limit. These areas are referred to as the green, yellow and red areas respectively.

In RED, the maximum and minimum thresholds are preset independently. The available buffer space is not taken into account when setting these parameters. The weight parameter is also a prefixed parameter. Prior work has suggested that the maximum threshold should be set to twice the minimum threshold [6].
Figures 8 to 11 illustrate the network performance for the topology depicted in Fig. 12. The NS-2 simulator is used to examine the four network performance parameters with weight parameters in the range 0.001 to 0.1. The results suggest that RED works most efficiently when the weight parameter is between 0.001 and 0.003. To be more speciffic, two performance parameters are improved: throughput and link utilization. Loss rate will be at an acceptable level but the delays will be increased.

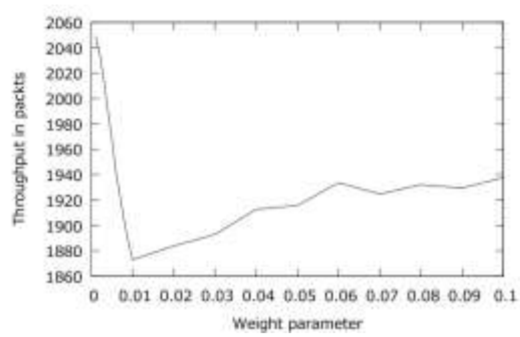


Figure 8, Throughput for a range of weight parameters from 0.001 - 0.1.

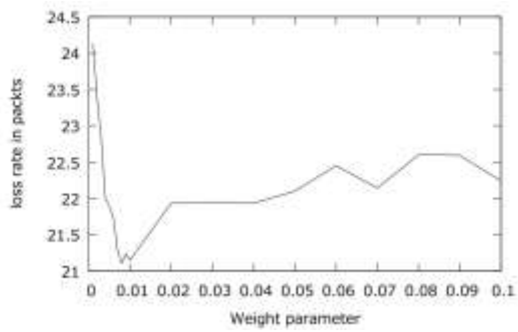


Figure 9, Loss rate for a range of weight parameters from 0.001 - 0.1.

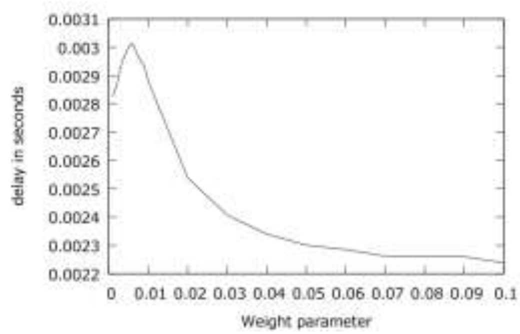


Figure 10, Delays for a range of weight parameters from 0.001 - 0.1.

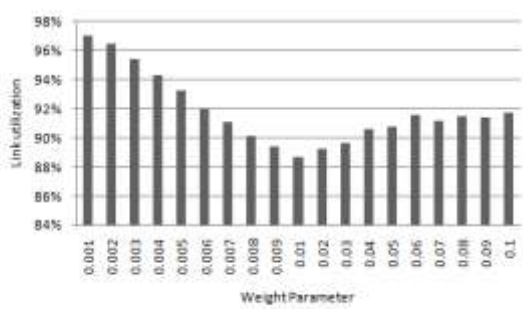


Figure 11, Link utilization for a range of weight parameters from 0.001-0.1.

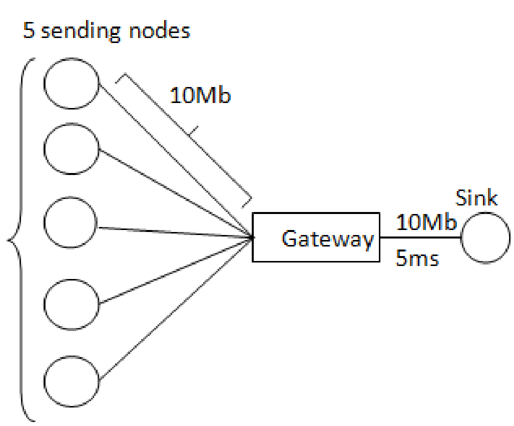


Figure 12, Simulation network topology.

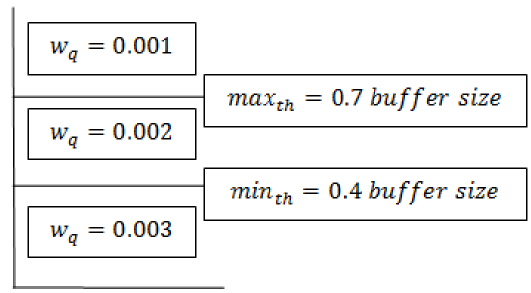


Figure 13, WTRED algorithm.

These results agree with the original parameter recomendations for RED [6] that suggests a weight parameter of 0.002. When the average reaches the yellow area, RED starts dropping packets. In this area there is no need to have higher values for the weight parameter. Hence, WTRED assigns the value 0.002 to the weight parameter. Another reason to use a lower weight parameter in this area is to maintain reasonable time before the average hits the maximum threshold; because when the maximum threshold is reached, the drop probability will be 1.0 and every arriving packet will be dropped. Setting the weight parameter to 0.002 in this area is better than using high values for the maximum drop probability.

High max_p values lead to shorter time to reach the maximum threshold which will reduce the link utilization. When the average queue size exceeds the maximum threshold and enters the red area, RED will drop every packet arriving at the gateway. FRED uses higher weight parameters to increase the sensitivity to the changes in the actual queue size. In this case, the actual queue size and the average queue size will closely follow each other. This mismatch makes FRED's behavior approach the TD strategy, droping packets based on the actual queue size.

5.2 WTRED Proposal

WTRED is proposed based on the simulation results in Sec. 5.1. The mechanism in this study uses different weight parameters for each area in RED. WTRED adjusts the maximum and minimum thresholds based on the actual buffer size.
The new parameter configuration in WTRED is to assign the weights 0.003, 0.002 and 0.001 to the green, yellow and red areas respectively. Also, the minimum threshold will be set at 40% of the buffer size and the maximum threshold will be 70% of the buffer size. This high minimum threshold is to grant higher network throughput and link utilization. The weight parameter is assigned the value 0.003 for the average to respond quickly to the changes in the green area. In the case where persistent traffic bursts accumulate the queue size, the average will be increased faster to hit the minimum threshold and initiate congestion recovery. Fig. 13 illustrates the WTRED algorithm.

6 SIMULATIONS

WTRED, RED and FRED are compared against the four network performance parameters which are: throughput, link utilization, average delay and packet loss. The network topology in Fig. 12 is used to run 11 different scenarios with different weight parameters and buffer sizes. Table 4, illustrates the weight and buffer size for each scenario used in this simulator.

Table 4, Buffer sizes for FRED's sub-phases.

w_a	Buffer
0.001	40
0.0015	50
0.002	60
0.0025	70
0.003	80
0.0035	90
0.004	100
0.0045	110
0.005	120
0.0055	130
0.006	140

6.1 Network Topology

A NS-2 simulation script is used to define five FTP sources. A full duplex link with 10Mb/s bandwidth connects each source with the gateway. Connection delays are uniformly distributed between 1*ms* and 5*ms*. Another full duplex link with 10Mb/s bandwidth and 5*ms* delay connects the gateway with a TCP sink. The packet size is 552 bytes and the TCP variant is Reno. Fig. 12 illustrates the simulation network topology.

6.2 Simulation Results

RED [6] suggests that, in order to filter out transient congestion at the gateway, the weight parameter must be assigned small values. Low weights mean that the EWMA will respond slowly to actual queue size changes. This reduces the gateway's ability to detect the initial stages of congestion. The maximum and minimum threshold values are restricted by the desired average queue size. Also, the difference between the maximum threshold and minimum threshold must be sufficiently large to avoid global synchronization. Small differences between maximum threshold and minimum threshold allow the average queue size to oscillate up to the maximum threshold.

Figures 14 to 17 depict the throughput, packet losses, average delay and link utilization respectively. Figure 14 shows that WTRED achieved the highest throughput among the three strategies. FRED generates very poor throughput due to the high weight parameters and maximum drop probability. As shown in Fig. 15, FRED increases the loss rate. FRED also has the lowest delays among the three strategies as illustrated in Fig. 16. This comes at the price of very poor throughput, Fig. 14 and link utilization, Fig. 17.

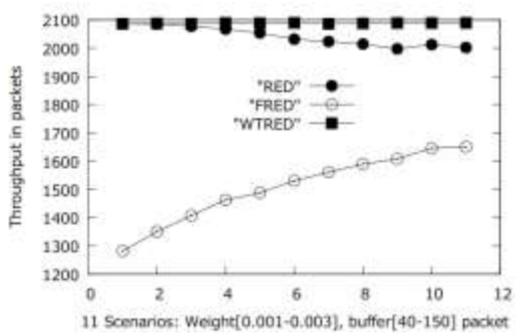


Figure 14, Network throughput for RED, FRED and WTRED.

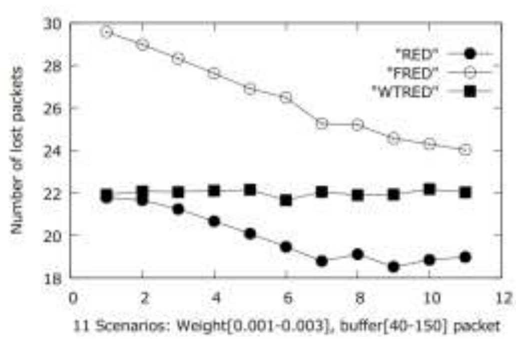


Figure 15, Packet loss rate for RED, FRED and WTRED.

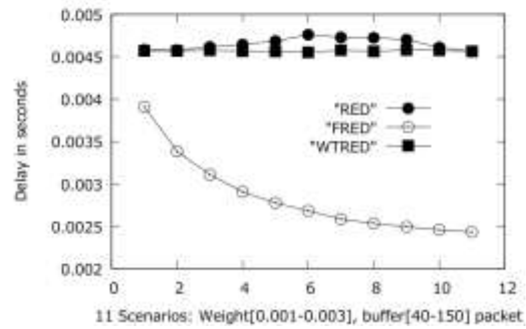
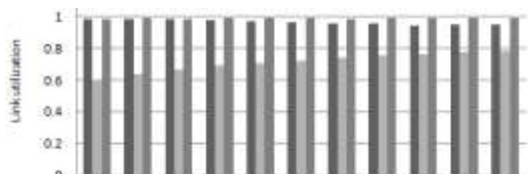


Figure 16, Average delay for RED, FRED and WTRED.



	1	2	3	4	5	6	7	8	9	10	11
RED	0.98	0.98	0.98	0.97	0.97	0.96	0.95	0.95	0.94	0.95	0.94
FRED	0.60	0.64	0.66	0.69	0.70	0.72	0.73	0.75	0.76	0.77	0.78
WTRED	0.98	0.98	0.98	0.98	0.98	0.99	0.98	0.98	0.98	0.98	0.99

Figure 17, Link utilization for RED, FRED and WTRED.

The figures demonstrate that WTRED outperforms RED and FRED. WTRED improves throughput and link utilization while maintaining acceptable delays and loss rates.

6.3 Issues with RED's and FRED's implementations

Research has shown that parameter setting in RED is sensitive and problematic [3-5]. FRED proposed a new parameter configuration for RED in an effort to increase network performance. Unfortunately, FRED has numerous drawbacks.

FRED uses a very high max_p value. In some phases this value is ten times the initial value in RED. Given the same queue conditions, sometimes FRED will drop ten times as many packets as RED. The maximum threshold in this case is actually reduced, resulting in a lower average queue size and lower average delay. Although FRED lowers delays, its overall performance is poor due to lower throughput, link utilization and higher loss rates, as demonstrated in section 6.2.

The suggested exponent value of the weight parameter using the normalized notation is -3. For example, the default value for w_q in RED is 0.002. When the actual queue size exceeds the warn threshold, FRED starts to increase the weight parameter. In Table 3, sub-phase 6, FRED multiplies the weight parameter by 20. In this case, w_q is not just doubled, it is also shifted one decimal point.

TD does not maintain a EWMA. RED maintains the average between the minimum and maximum thresholds while allowing transient bursts of traffic [6].

Higher weights means that the average queue size will closely follow the actual queue size. In case of bursty traffic overwhelming the gateway for an extended period, FRED will behave like a TD algorithm.

7 CONCLUSION

Traditional queue management strategies, such as TD and ERD, control congestion in TCP networks by monitoring the actual queue size in routers. These strategies suffer from problems such as: full queue, lock out, global synchronization and bias against bursty traffic.

The new generation of active queue management techniques, such as RED, are believed to solve these problems by monitoring the average queue size. However, RED variants generate two serious problems which are: complex parameter configuration and mismatch between the average and actual queue sizes.

This article presents the Weighted Random Early Detection (WTRED); which is a RED-based strategy to reduce the mismatch between the average and actual queue sizes. WTRED also provides a new parameter configuration, specifically the weight parameter, to enhance network performance. WTRED is compared with RED and FRED strategies using the NS-2 simulator. The results show that WTRED offers superior network performance.

8 REFERENCES

1. Ryu, S., Rump, C., Qiao, C.: Advances in Internet Congestion Control. IEEE Communications Surveys and Tutorials. vol. 5, (2003).
2. Jacobson, V.: Congestion Avoidance and Control. Proc. ACM SIGCOMM’88. vol. 18. no. 4. pp. 314-329. (1988).
3. Christiansen, M et al.: Tuning RED for Web Traffic, IEEE/ACM Trans. Net., vol. 9, no. 3, pp. 249 –64, June (2001).
4. Firoiu, V. and Borden, M.: A Study of Active Queue Manage-ment for Congestion Control, Proc. INFOCOM 2000, pp. 1435–44. (2000).
5. May, M. et al.: Influence of Active Queue Parameters on Aggregate Traffic Performance, Technical Report no. 3995, INRIA, Sophia Antipolis,France, http://www.inria.fr/RRRT/RR-3995.html, (2000).
6. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans Netw. vol. 1, pp. 397-413. (1993).
7. Hashem, E. S.: Analysis of Random Drop for Gateway Congestion Control. Massachusetts Institute of Technology (1989).
8. May, M., Bolot, J., Diot, C., Lyles, B.: Reasons Not to Deploy RED. In IEEE/IFIP IWQoS pp. 260-262. London, UK (1999).
9. Wu, C., Yang, S.H.: The mechanism of adapting RED parameters to TCP traffic, Computer Communications, vol 32, no. 13-14, pp. 1525-1530, (2009).

10. Liu., Feng , Guan, Z.H., Wang., H.O.: Controlling bifurcations and chaos in TCPUDPRED, Real World Applications, vol. 11, no. 3, pp. 1491-1501, (2010).
11. Pei, L.J., Mu, X.W., Wang,R.M., Yang, J.P.: Dynamics of the Internet TCPRED congestion control system, Real World Applications, vol. 12, no. 2, pp. 947-955, (2011).
12. Wang, H., Shin, K.: Refined Design of Random Early Detection Gatewways. Presented at the Global Telecommunication Conference- Globcom'99. (1999).