DSRED: An Active Queue Management Scheme for Next Generation Networks

Bing Zheng and Mohammed Atiquzzaman
Department of Electrical and Computer Engineering
The University of Dayton
Dayton, Ohio 45469-0226
zhengbin@flyernet.udayton.edu, atiq@ieee.org

Abstract

Random Early Detection (RED), an active queue management scheme, has been proposed by the Internet Engineering Task Force (IETF) to improve the throughput of TCP/IP based networks in the next generation routers. RED has a number of problems such as low throughput, large delay/jitter, and inducing instability in networks. Previous enhancements to RED attempted to improve the performance of RED by modifying the parameters of RED. However, such modifications result in limited improvement. In this paper, we propose a new active queue management scheme which achieves a higher throughput than RED and retains all the advantages of RED. Results show that our proposed scheme results in better performance than RED in terms of throughput and delay.

1 Introduction

Active queue management can improve the performance of TCP/IP, and is therefore recommended by IETF for use in the routers of Next Generation Internet [2, 9]. Random Early Detection (RED) [8], an active queue management scheme, was proposed to solve the global synchronization problem in TCP/IP based networks. Since its first proposal, the RED algorithm has been widely studied. RED uses a *single linear drop function* to calculate the drop probability of a packet, and uses four parameters and average queue size to regulate its performance. The parameters are Min_{th} and Max_{th} which represent buffer thresholds for packet drop at the gateway queue, Max_{drop} represents the maximum drop probability at Max_{th}, and w is weight parameter to calculate the average queue size from the instantaneous queue size.

The behavior of RED is still not fully understood [15]; especially the relationship between its parameters and performance in terms of throughput is still under research [19, 16, 5, 2]. Studies have shown that RED has several problems such as low throughput, unfairness to connections, large delay/jitter and instability as described below.

1.1 Related Work

The throughput of RED has been modeled by May et.al. [14]. They have shown that under heavy load, the throughput is inversely proportional to the load. Throughput can be increased by tracking the state of individual connections/links [19, 16, 11, 13] or by selecting appropriate parameters for RED [12, 6, 17].

Suter [19] studied the throughput of RED under per-flow queue management for several cases. It is found that with a large number of TCP connections, the throughput of RED is generally low. Moreover, with a mixture of bursty and greedy sources, RED suffers from unfairness and low throughput. When TCP has to compete with more aggressive sources, or is used in asymmetric networks with a perpetually congested reverse path, RED's throughput is very low.

To solve the problem of unfairness among links [1, 18], Kim et.al. studied the Fair Buffering RED (FB-RED) [11] for running TCP over ATM. Although FB-RED results in fairness among links, it however needs to track the information for all the links. This results in *scalability problems* which makes it impractical to be used in a large network.

Lin et.al. [13] proposed Fair RED (FRED) which relies on usage of buffer spaces by the different flows (per-active-flow accounting) to determine the drop rate of the flow. Although it achieves a fair drop rate for different flows, it needs to track the state of each flow which results in scalability problems similar to those in [11].

To solve the scalability problem of FB-RED [11] and FRED [13], Ott [16] proposed Stabilized Random Early Drop (SRED) which, like RED, preemptively discards packets with a load dependent drop probability when the buffer in a router gets congested. SRED can stabilize, over a wide range of load levels, the buffer occupancy at a level which is independent of the number of active connections.

0-7695-0912-6/00 $10.00 © 2000 IEEE

SRED therefore overcomes the scalability issues of [11, 13]. It however, suffers from *low throughput* as shown in their simulation results where the normalized throughput is very low even with small number of traffic flows.

Lakshman [12] carried out a simulation of TCP/IP over ATM to study the throughput of RED. It was found that an exponential drop function is better than the single linear drop function of RED. However, an exponential drop function requires more *computing power* and is not easily implemented in hardware.

Feng [6] showed that the effectiveness of RED depends, to a large extent, on the appropriate selection of the RED parameters. He also showed that there is no single set of RED parameters that work well under different congestion scenarios. He therefore, proposed an adaptive RED which self-parameterizes itself based on the traffic mix. However, adaptive determination of the RED parameters complicates buffer management of high speed routers/gateways.

Parris et.al. [17] pointed out that RED is not effective in the case of UDP sources. They proposed *Classed Based Threshold (CBT)*, which sets the buffer thresholds for packet dropping according to their traffic types (TCP vs. UDP) and priority classes. Their scheme tags UDP traffic which has its own drop thresholds that are different from the thresholds used for TCP traffic. The performance of TCP traffic is thus protected in the presence of UDP traffic.

Martin May [15] studied the *queuing delay and delay variance* (jitter) of RED. It is found that RED has a large delay variance which is also very sensitive to the weight parameter (w) of RED. The smaller the value of w, the larger the delay variance. Further, for real-time application, RED's delay is large.

To study the stability of the RED algorithm, Firoiu et.al. [7] modeled RED as a feedback control system, and pointed out that RED will induce network instability and major traffic disruption if not properly configured.

1.2 Objectives of this Paper

From the above discussion, we observe the RED has problems such as *low throughput, large delay/jitter, unfairness to connections, and inducing instability in the network.* Of these problem, *low throughput* is the most important problem which needs to be carefully addressed. Since IETF recommends RED (and its variant RED with In Out (RIO)) in Differentiated Services [4, 9, 10, 3] as the active queue management in next generation networks, it is important to develop mechanisms to remove the above problems associated with RED. The *objective* of this paper is to develop an active queue management scheme which will improve the throughput and delay characteristics of RED. This will in turn improve the QoS of Internet gateways. In this paper, we *propose and evaluate the performance of a new active queue management scheme* which has a higher performance than RED. Our proposed scheme uses a combination of two different drop probability distributions to achieve a higher performance than RED.

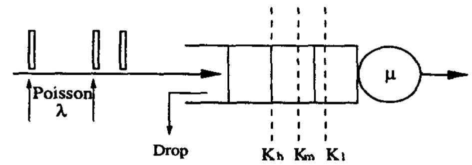

Figure 1. Model for DSRED buffer at gateway.

The rest of the paper is organized as follows. In Section 2, we propose and analyze the *Double Slope Random Early Detection* (DSRED) scheme. Section 3 gives simulation results and discussion on the performance of the proposed scheme, followed by conclusions in Section 4.

2 Double Slope Random Early Detection (DSRED) Scheme

As stated above, previous modifications to RED were based on the linear drop function of RED. The modifications attempted to modify the parameters of RED but resulted in limited improvement in throughput. In this section, *we propose a new active queue management scheme* called the Double Slope Random Early Detection (DSRED) scheme which has a higher performance than RED.

2.1 The Principle of DSRED

The proposed scheme is called *Double Slope Random Early Detection* (DSRED). As shown in Figures 1 and 2, the idea is that the gateway buffer segment between K_l and K_h is divided into two sub-segments separated by K_m as shown in Figure 1. The overall drop function from K_l to K_h are described by two linear segments with slope α and β respectively. The slopes for these two linear segments are complementary and are adjusted by the mode selector γ. Here, the K_m is set as $0.5(K_l + K_h)$, which can be configured by gateway administrator. The drop function, $p_d(avg)$, of DSRED can be expressed as:

$$p_d(avg) = \begin{cases} 0 & avg < K_l \\ \alpha(avg - K_l) & K_l \leq avg < K_m \\ 1 - \gamma + \beta(avg - K_m) & K_m \leq avg < K_h \\ 1 & K_h \leq avg \leq N \end{cases} \quad (1)$$

where, α, β and avg are given by:

$$\alpha = \frac{2(1-\gamma)}{K_h - K_l} \quad (2)$$

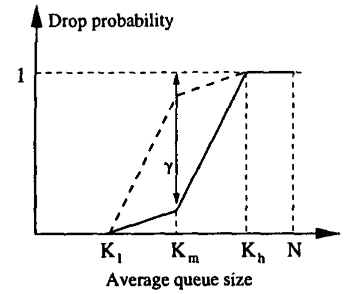

Figure 2. Drop function of DSRED.

$$\beta = \frac{2\gamma}{K_h - K_l} \quad (3)$$

$$avg = (1 - w)avg + wq \quad (4)$$

The above equations governing packet dropping in DSRED translate to the following rules:

- When the average queue length, avg, is less than K_l, no packet is dropped;
- When the average queue length, avg, is between K_l and K_m, packets are dropped according to the drop function with slope α;
- When the average queue length, avg, is between K_m and K_h, packets are dropped according to the drop function with slope β;
- When the average queue length, avg, reaches K_h or higher value, packets are dropped with a probability of one.

The algorithm of DSRED is shown in Figure 3.

2.2 Notations

To facilitate further discussion, we define the following variables which will be used to describe and evaluate the performance of our proposed algorithm.

- N: Buffer size at the gateway in packets;
- K_l: Threshold for average queue length (defined in [8]) to start packet dropping at the buffer;
- K_h: Threshold for average queue length (defined in [8]) to start packet dropping at the buffer with a probability of 1;
- K_m: Threshold for average queue length to change the drop function slope;
- α: Drop function slope for the first linear segment between K_l and K_m;

```
For each packet arrival
    Calculate average queue length avg
    If avg < Kl
        No drop
    elseif Kl <= avg < Km
        Calculate drop probability based on slope α
        Drop the packet
    elseif Km <= avg < Kh
        Calculate drop probability based on slope β
        Drop packet
    else
        Drop packet
```

Figure 3. Algorithm of DSRED.

- β: Drop function slope for second linear segment between K_m and K_h;
- γ: Mode selector for adjusting drop function slopes;
- λ: Rate of traffic arrival at the buffer in packets/second;
- μ: Packet processing rate at the gateway buffer in packet/second;
- ρ: Offered load factor defined as $\frac{\lambda}{\mu}$;
- q: Instantaneous gateway queue length in packet;
- avg: Average queue length as defined in [8];
- w: Weight parameter as defined in [8] to calculate avg;
- $V(i)$: Steady state probability for queue state being in State i; this is defined as the probability that the $avg = i$;
- $p_a(i)$: Packet acceptance probability at state i; this is probability that the buffer accepts an arriving packet given that the buffer is in State i.
- $p_d(i)$: Packet dropping probability at state i; this is probability that the buffer drops an arriving packet given that the buffer is in State i. Note that $p_d(i) = 1 - p_a(i)$;

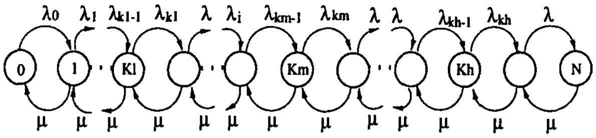

Figure 4. Steady state diagram for analysis of DSRED.

- λ_i: Effective arrival rate at the buffer when it is in state i. This is defined as the rate at which packets are queued and is given by $\lambda p_a(i)$;
- P_D: Average packet drop probability for gateway buffer overall possible state;
- D: Average queuing delay at the buffer which is calculated by Little's formula;
- Max_{drop}: Maximum drop probability of RED;
- *Normalized throughput:* defined during a time period by $\frac{\text{total packets received at destination}}{\text{total packets sent by sources}}$.

2.3 Analysis of Packet Drop vs. Offered Load Factor

In the previous section, we have described our proposed DSRED scheme. In this section, we develop an analytical model of DSRED to study its normalized throughput as a function of different offered load. Since, the normalized throughput can be obtained from $1 - P_D$, we therefore develop a model to determine the average packet drop probability P_D.

We model packet arrivals to the buffer by a Poisson process with rate λ. Although Poisson traffic may not reflect the the arrival process in real networks, it can provide a tractable model to understand the behavior of DSRED. The validity of this assumption has been proved in [14, 15].

The model is shown in Figure 1 where the gateway buffer is modeled by a FIFO queue with a processing rate of μ. An arriving packet is dropped with a probability which is based on average queue length. We use the Markov chain to analyze the steady queue distribution from which we derive the average packet drop probability (P_D) versus offered load factor. The state diagram is shown in Figure 4 where λ_i is the *effective arrival rate* for State i and μ is the constant departure rate. Let $V(i)$ denote the steady state probability of the average queue length being in State i. It can be expressed as [20]:

$$V(i) = V(0)\prod_{j=0}^{i-1}\frac{\lambda_j}{\mu_j} \qquad (5)$$

where $V(0)$ is given by:

$$V(0) = \frac{1}{\sum_{i=0}^{N} V(i)} \qquad (6)$$

From Equation (1), the packet acceptance probability $p_a(i) = 1 - p_d(i)$ and the effective arriving rate λ_i can be expressed as:

$$p_a(i) = \begin{cases} 1 & i < K_l \\ 1-\alpha(i-K_l) & K_l \le i < K_m \\ \gamma-\beta(i-K_m) & K_m \le i < K_h \\ 0 & K_h \le i \le N \end{cases} \qquad (7)$$

$$\lambda_i = \begin{cases} \lambda & i < K_l \\ \lambda(1-\alpha(i-K_l)) & K_l \le i < K_m \\ \lambda(\gamma-\beta(i-K_m)) & K_m \le i < K_h \\ 0 & K_h \le i \le N \end{cases} \qquad (8)$$

The average packet drop probability P_D is given by:

$$P_D = 1 - \sum_{i=0}^{N} V(i)P_a(i) \qquad (9)$$

From Little's formula, average queuing delay D is given by [20]:

$$D = \sum_{i=0}^{N}\frac{(i+1)V(i)}{\mu}P_a(i) \qquad (10)$$

Figure 5 shows packet drop probability versus offered load factor ($\frac{\lambda}{\mu}$) for different values of γ. It is seen that for a given γ, a high offered load factor results in high packet drop which will result in a decrease in throughput. For the same offered load factor, drop performance can be varied by changing γ.

2.4 Comparison between DSRED and RED

DSRED is similar to RED in two respects. First, both of them use linear drop functions to give smoothly increasing drop action based on average queue length. Secondly, they calculate the average queue length using the same definition to account for the effect of long term congestion. Therefore, DSRED inherits the advantages of RED. However, DSRED's two-segment drop function provides much more flexible drop operation than RED.

The most important differences and advantages of DSRED as compared to RED are as follows:

- The two segment drop function of DSRED use the average queue length which is related to long term congestion level. When congestion increases, drop will increase with higher rate instead of constant rate. This will give an early warning to hosts to backoff, preventing congestion from getting worse. As the consequence, congestion will be relieved and throughput will increase.

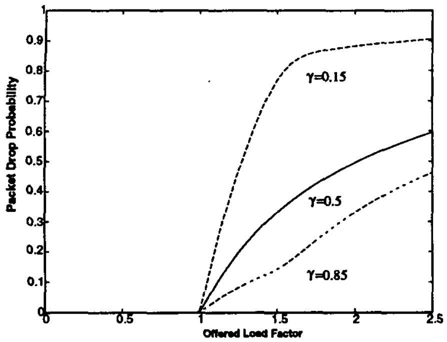

Figure 5. Theoretical curve of packet drop probability versus offered load factor for different γ in DSRED.

- The two segments of the drop function can be adjusted flexible by the parameter γ. Therefore, the operating mode of DSRED can be easily adjusted by a single parameter γ. i.e., by adjusting γ, one can get high drop rate first followed by a low drop rate, or vise verse. This will provide a more effective way than RED to handle complicated network congestion situations.

3 Simulation Configurations and Results

Simulations were carried out with the OPNET5.1 event driven network simulation tool which is widely used in academic research and industry. This section describes the simulation configuration and the performance of DSRED from simulation.

3.1 Simulation Configurations

The simulation configuration is shown in Figure 6. To provide a fair comparison with RED, we use same network topology and similar network configuration as in [8]. In our simulation, three ftp sources send packets to the same destination via the gateway. To make the comparison more general, we investigated the performance of DSRED and RED with TCP sources under *Heavy Load* and *Low Load*. The network simulation configuration parameters are as follows:

- *Server0 to Gateway0:* Propagation delay $1ms$, link rate $100Mbps$.
- *Server1 to Gateway0:* Propagation delay $5ms$, link rate $100Mbps$.

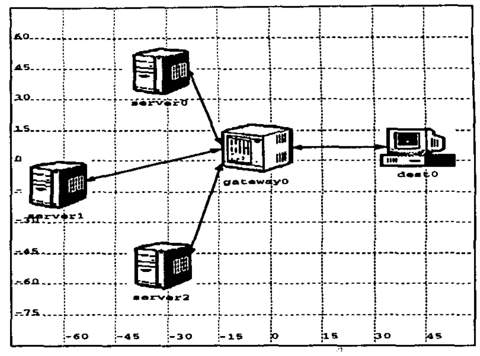

Figure 6. Network configuration for simulation.

- *Server2 to Gateway0:* propagation delay $3ms$, link rate $100Mbps$.
- *Destination0 to Gateway0:* Propagation delay $5ms$, bottleneck link rate $10Mbps$. The sum of link rate from three TCP servers to gateway is $300Mbps$, which is much higher than the bottleneck link rate $10Mbps$. Therefore, the aggregated traffic streams from these three TCP servers will make the gateway congested because of the low bottleneck link rate.
- $\mu = 1ms/packet$.
- *Gateway Buffer Size:* 200 packets, similar as in [8].
- $K_l = 6$, $K_h = 20$: The values were chosen such that $K_h \geq 2K_l$ as suggested in [8].
- $w = 0.07$.
- $Max_{drop} = 0.1$ *and* 0.2, $\gamma = 0.96$ *and* 0.91 respectively, this is to make the studies more general

To enable a fair comparison, we have used the same values of the propagation delay as used by [8, 15]. The following quantities are collected at the gateway, the server, and the destination.

- *Destination TCP traffic received:* the number of packets received by destination during the simulation period;
- *Server TCP load:* the number of packet sent by TCP sources during simulation period;
- *Gateway Queuing Delay (QD):* queuing delay at the gateway;
- *Gateway Queue Size (QS):* queue size of the buffer at the gateway;

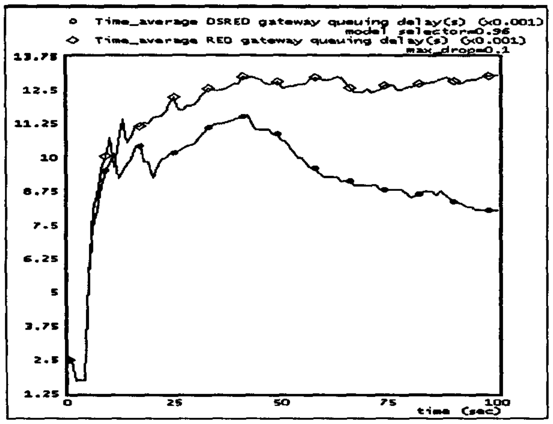

Figure 7. Time average of gateway queuing delay for DSRED and RED with *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

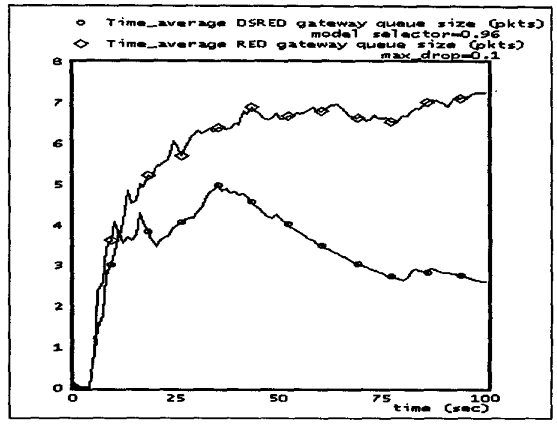

Figure 8. Time average of gateway queue size for DSRED and RED with *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

- *Packet Drop (PD) at Gateway:* the number of packets dropped at the gateway.

3.2 Simulation Results

In this section, we present results obtained from our simulation. For the purpose of comparison, we compared the normalized throughput, gateway queue size, queuing delay and gateway packet drop performance for DSRED and RED.

Figure 7 shows the time average of the gateway queuing delay for DSRED and RED with *Heavy Load* when $\gamma = 0.96$ and $Max_{drop} = 0.1$. Figures 8 and 9 show the time average of queue size and packet dropped. From these figures, it is seen that the DSRED gateway always has lower queuing delay, smaller queue size and lower packet drop than the RED gateway queue during the entire simulation period. For example, the time average of packet drop for DSRED gateway is only $\frac{1}{5}$ of the value for RED at simulation time of 100 second. Therefore, DSRED gateway has higher normalized throughput than RED gateway. As seen from Figure 8, the average gateway queue size for DSRED gateway is less than $\frac{1}{2}$ of the value for RED. A smaller queue size also results in a lower queuing delay.

Figures 10, 11 and 12 show the gateway queuing delay, queue size and packet drop respectively for *Heavy Load* with $\gamma = 0.91$ and $Max_{drop} = 0.2$. In this case, since the packet drop probability is a little higher than the previous case for same queue situation, we get lower queuing delay and queue size. Although the queue performance of RED gateway improves slightly for $Max_{drop} = 0.2$ compared with $Max_{drop} = 0.1$, DSRED is still much better than RED in terms of normalized throughput, gateway queuing delay, queue size, and packet drop performance.

Figures 13, 14 and 15 show a comparison of the time average of gateway queuing delay, queue size and packet drop respectively for DSRED and RED for *Low Load* with $\gamma = 0.96$ and $Max_{drop} = 0.1$. Similar as in *Heavy Load* case, our DSRED outperforms the RED gateway in normalized throughput, queuing delay, queue size, and packet dropped. These figures also shows that our DSRED keeps a steady performance at both *Heavy Load* and *Low Load.* For example, both at *Heavy Load* and *Low Load*, DSRED gateway has a time average of queuing delay 8.75ms at simulation time 88 second. On the contrary, RED gateway has a time average of queuing delay 13ms at *Heavy Load* and a time average of queuing delay 12ms at *Low Load* at simulation time 88 second. For the packet drop, we also get similar performance. Therefore, we conclude that DSRED will have more steady performance than RED and is more suitable for networks where the network load varies from time to time.

Figures 16, 17 and 18 show a comparison of the time average of gateway queuing delay, queue size and packet drop respectively for DSRED and RED for *Low Load* with $\gamma = 0.91$ and $Max_{drop} = 0.2$. In this case, since the packet drop probability is a little higher than the case of $\gamma = 0.96$ and $Max_{drop} = 0.1$ for same queue situation, both DSRED and RED gateway will keep a smaller queue size and smaller queuing delay. However, our proposed DSRED gateway has a more steady performance than RED. For example, at simulation time 100 seconds, the time average of queue size for DSRED is 2.6 packets for $\gamma = 0.96$ and 1.3 packets for $\gamma = 0.91$, the change is about 1.3

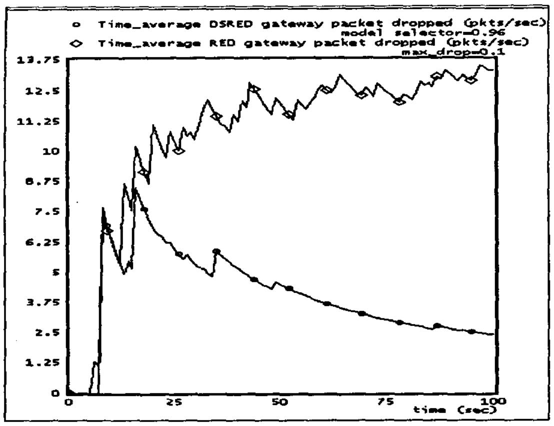

Figure 9. Time average of gateway packet dropped for DSRED and RED with *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

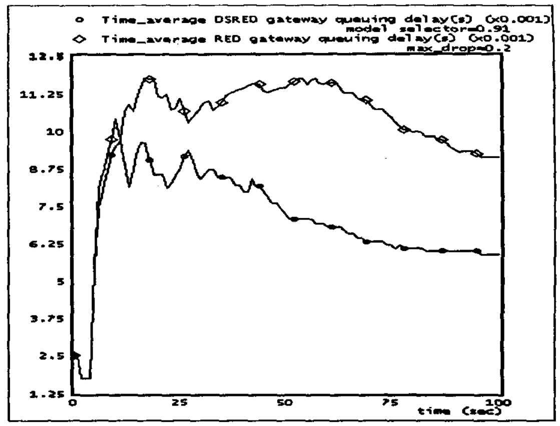

Figure 10. Time average of gateway queuing delay for DSRED and RED with *Heavy Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

packets. On the contrary, for RED, the time average of queue size is 6.8 packets for $Max_{drop} = 0.1$ at simulation time 100 seconds and 4.2 packets for $Max_{drop} = 0.2$, the change is about 2.6, two times larger than that of DSRED. Therefore, we conclude that our DSRED is more stable than RED for different configuration parameter. To give a brief comparison, the simulated results are summarized in Tables 1 and 2. We conclude, in performance of normalized throughput, gateway queuing delay, queue size, and packet drop, DSRED outperforms RED.

Figures 19 and 20 show the time average of the server TCP load for DSRED and RED in *Heavy Load* case. Figure 21 shows the time average of the TCP traffic received at the destination for DSRED with $\gamma = 0.96$ and RED with $Max_{drop} = 0.1$. The normalized throughput can be calculated from the server TCP load and the TCP traffic received at the destination. In addition to calculating the normalized throughput (using Figures 19, 20 and 21), we have also calculated (using Fgiures 7 to 18) and compiled a summary of the queuing delay, gateway queue size and packet drop probability for DSRED and RED in Tables 1 and 2. As seen from the tables, DSRED gateway has a much higher normalized throughput than RED.

4 Conclusion

In this paper, we proposed a new active queue management scheme for Internet gateway to improve the gateway performance in terms of normalized throughput, queuing delay, queue size, and packet drop. Similar to RED, our proposed scheme drops packet based on long term congestion with random choice of packets to drop. Therefore, our

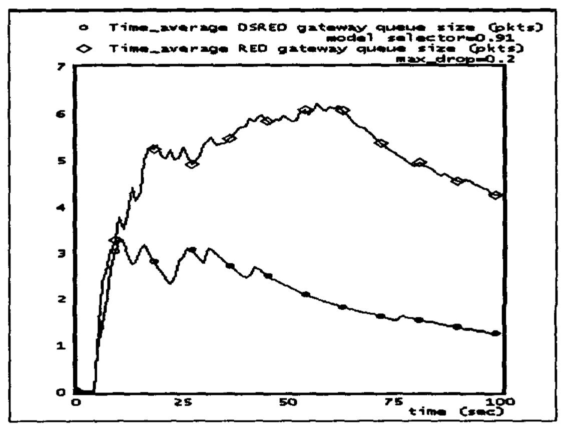

Figure 11. Time average of gateway queue size for DSRED and RED with *Heavy Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

Table 1. Simulated Gateway Performance for DSRED ($\gamma = 0.96$) and RED ($Max_{drop} = 0.1$).

	DSRED		RED	
Parameters	H Load	L Load	H Load	L Load
Norm Thrupt	0.525	0.372	0.445	0.355
Avg QD(s)	0.0081	0.0081	0.013	0.0118
Avg QS(Pkt)	2.6	2.7	7.3	6.8
Avg PD(Pkt/s)	2.5	2.5	13.1	11.25

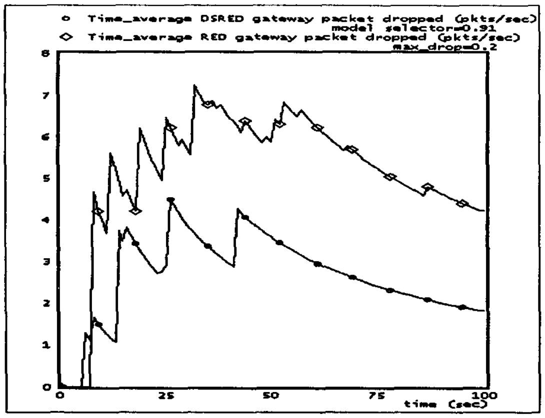

Figure 12. Time average of gateway packet dropped for DSRED and RED with *Heavy Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

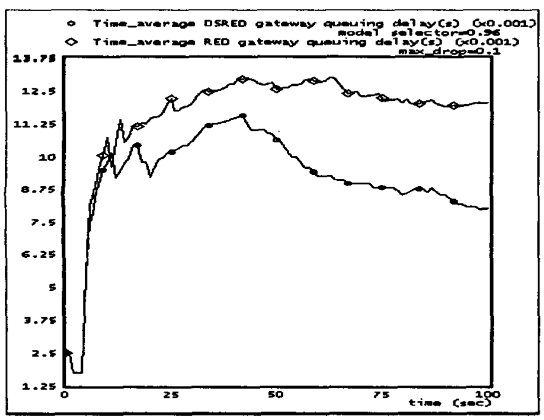

Figure 13. Time average of gateway queuing delay for DSRED and RED with *Low Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

Table 2. Simulated Gateway Performance for DSRED ($\gamma = 0.91$) and RED ($Max_{drop} = 0.2$).

	DSRED		RED	
Parameters	H Load	L Load	H Load	L Load
Norm Thrupt	0.682	0.64	0.482	0.47
Avg QD(s)	0.0059	0.0059	0.0091	0.0091
Avg QS(Pkt)	1.3	1.3	4.3	4.8
Avg PD(Pkt/s)	1.9	1.9	4.2	4.2

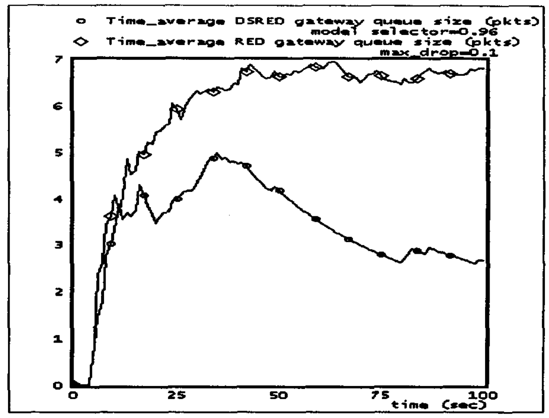

Figure 14. Time average of gateway queue size for DSRED and RED with *Low Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

scheme inherits the advantages of RED. Unlike RED, our proposed DSRED scheme uses a two segment drop function instead of a single one in RED. This results in a low packet drop probability at a low congestion level and gives early warning for long term congestion.

We have shown that *DSRED has a much higher normalized throughput* than RED in both the heavy load and low load cases. For the same network load and configuration, our proposed DSRED algorithm results in *lower average queuing delay and queue size* than RED. It is also found that DSRED has *better packet drop performance* than RED, resulting in a higher normalized throughput than RED. Our proposed scheme is easily implemented with about 50 lines of C code.

References

[1] Y. Atsami, E. Kondoh, O. Altintas, and T. Yoshida. Improving fairness and stability in best effort service: A new congestion control algorithm for SACK-TCP. *IEICE Transaction on Communication*, E81-B(11):2023–2033, 1998.

[2] B.Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. M. C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommedations on queue management and congestion avoidance in the Internet. RFC 2309, April, 1998.

[3] Y. Bernet, A. Smith, S. Blake, and D. Grossman. A conceptual model for diffserv routers. http://www.ietf.org/internet-drafts/draft-ietf-diffserv-model-04.txt, March, 2000.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service. RFC 2475, Dec, 1998.

[5] W. chang Feng, D. D. Kandlur, and D. Saha. Adaptive packet marking for maintaining end to end throughput in a

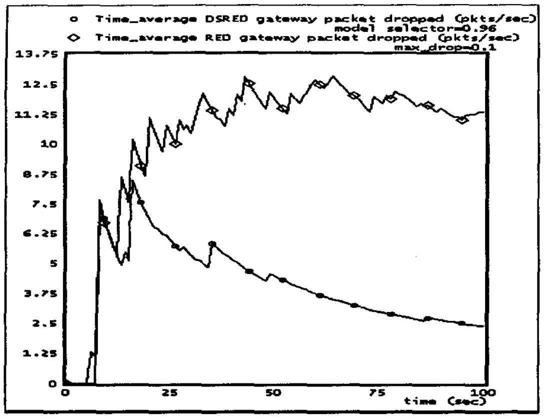

Figure 15. Time average of gateway packet dropped for DSRED and RED with *Low Load* **for** $\gamma = 0.96$ **and** $Max_{drop} = 0.1$.

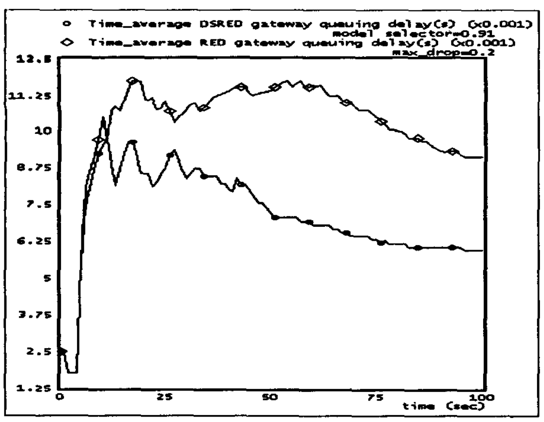

Figure 16. Time average of gateway queuing delay for DSRED and RED with *Low Load* **for** $\gamma = 0.91$ **and** $Max_{drop} = 0.2$.

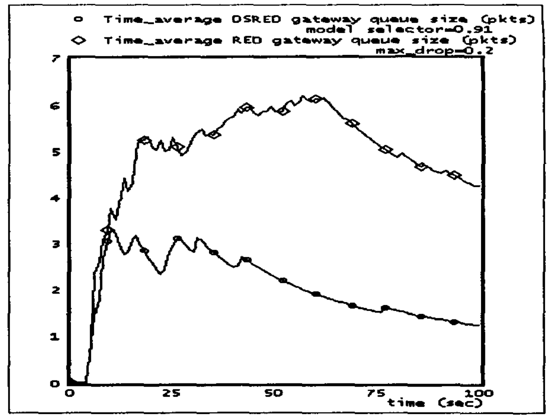

Figure 17. Time average of gateway queue size for DSRED and RED with *Low Load* **for** $\gamma = 0.91$ **and** $Max_{drop} = 0.2$.

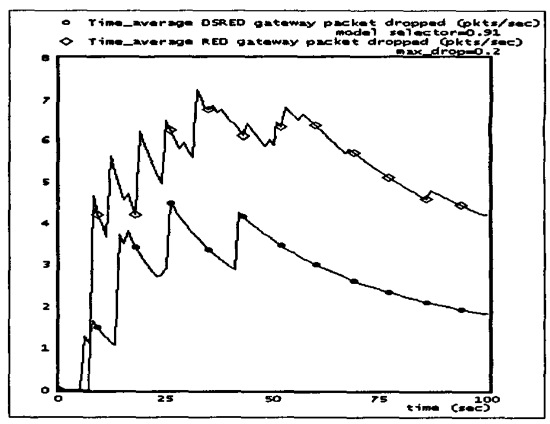

Figure 18. Time average of gateway packet dropped for DSRED and RED with *Low Load* **for** $\gamma = 0.91$ **and** $Max_{drop} = 0.2$.

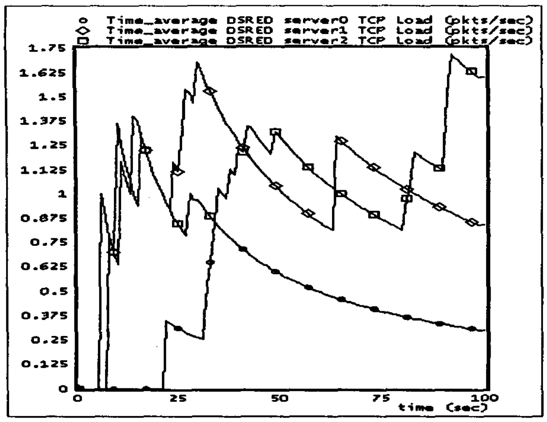

Figure 19. Time average of TCP load for DSRED at *Heavy Load* with $\gamma = 0.96$.

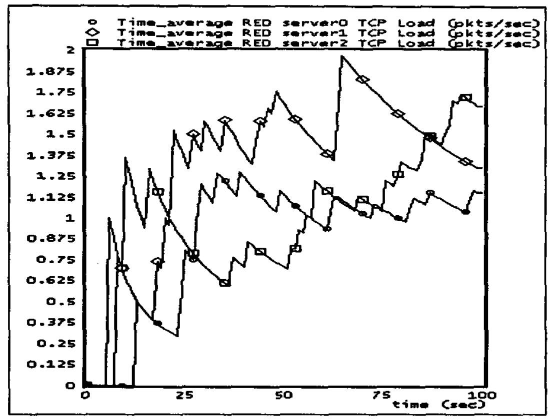

Figure 20. Time average of TCP load for RED at *Heavy Load* with $Max_{drop} = 0.1$.

differentiated service internet. *IEEE/ACM Transactions on Networking*, 7(5):685–697, Oct, 1999.

[6] W. chang Feng, D. D. Kandlur, D. Saha, and G. Kang. Self-configuring RED gateway. In *IEEE INFOCOM99*, pages 1320–1328, New York, March, 1999.

[7] V. Firoiu and M. Borden. A study of active queue management for congestion control. In *IEEE INFOCOM2000*, Tel-Aviv, Israel, March 26-30, 2000.

[8] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transanctions on Networking*, 1(4):397–413, August, 1993.

[9] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding PHB. RFC 2597, June, 1999.

[10] V. Jacobson, K. Michols, and K. Poduri. An expedited forwarding PHB. RFC 2598, June, 1999.

[11] W.-J. Kim and B. G. Lee. The FB-RED algorithm for TCP over ATM. In *IEEE GLOBECOM98*, pages 551–555, Sydney, Australia, Nov 8-12, 1998.

[12] T. V. Lakshman, A. Neidhardt, and T. J. Ott. The drop from front strategy in TCP and in TCP over ATM. In *IEEE INFOCOM96*, San Francisco, CA, March 26-28, 1996.

[13] D. Lin and R. Morris. Dynamics of random early detection. In *Proceedings of ACM SIGCOMM97*, pages 127–137, Cannes, France, Sept 14-18, 1997.

[14] M. May, J.-C. Bolot, A. Jean-Marie, and C. Diot. Simple performance models of differentiated services schemes for the internet. In *IEEE INFOCOM99*, New York, March, 1999.

[15] M. May, T. Bonald, and J.-C. Bolot. Analytic evaluation of RED performance. In *IEEE INFOCOM2000*, Tel-Aviv, Israel, March 26-30, 2000.

[16] T. J. Ott, T. V. Lakshman, and L. Wong. SRED: Stabilized RED. In *IEEE INFOCOM99*, pages 1346–1355, New York, March, 1999.

[17] M. Parris, K. Jeffay, and F. D. Smith. Lightweight active router queue management for multimedia networking. In *Proceedings of SPIE*, pages 162–174, San Jose, CA, USA, Jan 25-27, 1999.

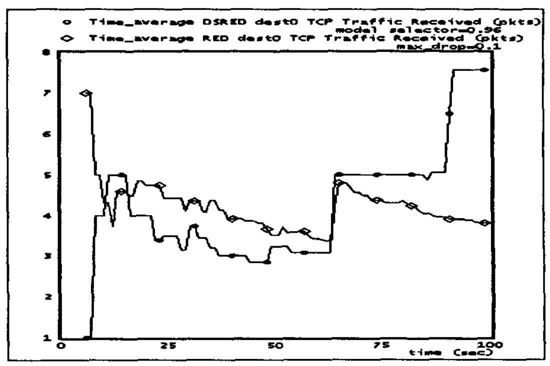

Figure 21. Time average of destination traffic received for DSRED and RED at *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

[18] C. M. Pazos, S. A. J. C, and M. Gerla. Using back pressure to improve TCP performance with many flows. In *Proceedings INFOCOM99*, pages 431–438, New York, March 1999.

[19] B. Suter, T. V. Lakshman, D. Stiliadis, and A. K. Choudhury. Buffer management schemes for supporting TCP in gigabit routers with per-flow queuing. *IEEE Journal on Selected Areas in Communications*, 17(6):1159–1169, June, 1999.

[20] K. S. Trivedi. *Probability and statistics with reliability, queuing and computer science application.* Prentice-Hall, Englewood Cliffs, 1982.