IEICE TRANS. COMMUN., VOL.E89–B, NO.3 MARCH 2006

PAPER

# DSRED: A New Queue Management Scheme for the Next Generation Internet

**Bing ZHENG**[†] *and* **Mohammed ATIQUZZAMAN**[††a)], *Nonmembers*

**SUMMARY** Random Early Detection (RED), an active queue management scheme, has been recommended by the Internet Engineering Task Force (IETF) for the next generation routers. RED suffers from a number of performance problems, such as low throughput, large delay/jitter, and induces instability in networks. Many of the previous attempts to improve the performance of RED have been based on optimizing the values of the RED parameters. However, results have shown that such optimizations resulted in limited improvement in the performance. In this paper, we propose Double Slope RED (DSRED), a new active queue management scheme to improve the performance of RED. The proposed scheme is based on dynamically changing the slope of the packet drop probability curve as a function of the level of congestion in the buffer. Results show that our proposed scheme results in better performance than original RED.

***key words:*** *active queue management, routers, Internet, simulation, performance evaluation*

## 1. Introduction

Active Queue Management (AQM) can improve the performance of TCP, and has been recommended by the Internet Engineering Task Force (IETF) for use in the routers of the next generation Internet [1], [2]. The goals of AQM is three folds. First, to improve *throughput* by reducing the number of packets dropped. This is achieved by keeping the average queue length small in order to absorb naturally occurring bursts without dropping packets. Second, AQM provides low *delay* to interactive services by maintaining a small average queue length. Third, AQM avoids the *lock out* phenomenon arising from tail drop. Lock out phenomena results in a single connection (or a few flows) monopolizing the buffer space which, in turn, results in packets from other connections being dropped and causing unfairness in bandwidth sharing among connections.

Random Early Detection (RED) [3], an AQM scheme, was originally proposed to solve the global synchronization problem in TCP/IP based networks. It uses a *single linear drop function* to calculate the drop probability of a packet, and has four parameters and an average queue size to regulate its performance. The four parameters are $Min_{th}$ and $Max_{th}$ which represent buffer thresholds for packet drop at the queue, $Max_{drop}$ represents the maximum drop probability at $Max_{th}$, and $w$ is a weight parameter to calculate the average queue size as shown in Fig. 1.

Despite the fact that the performance of RED has been widely studied in the literature, its behavior is still not fully understood [4]–[6]; especially the relationship between its parameters and performance in terms of throughput is still under research [1], [7]–[9]. As will be described in detail in Sect. 2, studies have shown that RED suffers from low throughput, large delay/jitter, unfairness to connections, and induces instability in the network. The *objective* of this paper is to develop a new RED-based AQM scheme which will improve the throughput and delay characteristics of RED. This will also improve the performance of Differentiated Services [2], [10]–[12] which uses RIO (RED with In Out is a modified version of RED) in the core routers. In this paper, we propose a RED based AQM scheme, which is *based* on dynamically switching between two packet drop probability distributions, depending on the level of congestion. The main *contribution* of this paper is the development of a new AQM scheme, called DSRED, which improves the performance of RED in terms of throughout, packet loss, and queue size. DSRED can be used to increase the performance of Internet routers, and is shown to improve the *stability* of RED.

The rest of the paper is organized as follows. In Sect. 2, we describe in detail the various proposals and their effectiveness in improving the performance of RED. In Sect. 3, we describe and analyze our proposed *Double Slope Random Early Detection* (DSRED) scheme. The simulation configuration used to simulate RED and DSRED is given in Sect. 4, followed by results and discussion to compare the performance of RED and DSRED in Sect. 5. Concluding remarks are presented in Sect. 6.

## 2. Related Work

Previous studies on improving the performance of RED mainly fall into two categories: tracking the state of individual connections/links [7], [8], [13], [14], and selecting appropriate parameters for RED [15]–[19].

May et al. [20] have modelled the throughput of RED to show that under heavy load, the throughput is inversely proportional to the load factor. Suter [7] studied the throughput of RED under per-flow queue management for several cases. It was found that for a large number of TCP connections, the throughput of RED is generally low. Moreover, with a mixture of bursty and greedy sources, RED suffered from unfairness and low throughput. When TCP has to compete

---

Manuscript received February 18, 2005.
Manuscript revised July 27, 2005.

[†] The author is with New Focus Inc., San Jose, CA, 95138 USA.
[††] The author is with School of Computer Science, University of Oklahoma, Norman, OK 73019-6151, USA.
a) E-mail: atiq@ou.edu

DOI: 10.1093/ietcom/e89–b.3.764

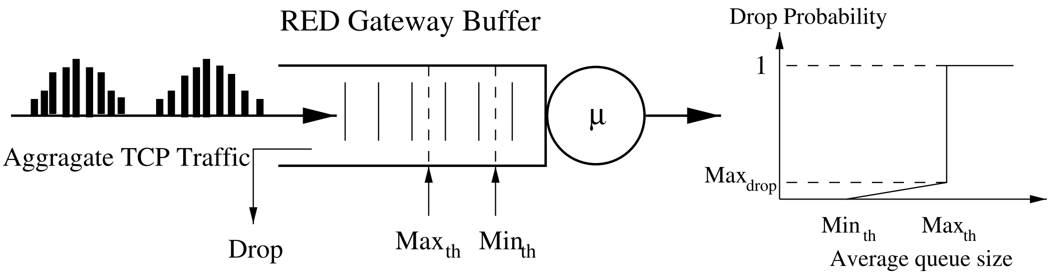

**Fig. 1** RED queue and its drop probability function.

**Table 1** Comparison of DSRED with other variants of RED. Note: (1) Good > Significant > Not significant > Poor. (ii) N/A: Not available.

| RED Variant | Throughput Increase | Decrease in Delay/Jitter | Fairness Improvement | Shortcoming |
|---|---|---|---|---|
| FRED | Not significant | N/A | Good for TCP/TCP | Scalability |
| FBRED | Not significant | N/A | Good for TCP/TCP | Scalability |
| SRED | Poor | Good | N/A | Low Throughput |
| CBT-RED | Significant | Good | N/A | Scalability |
| XRED | Significant | N/A | N/A | Extension Header in UDP. Additional list files |
| BRED | Significant | N/A | Good for TCP/UDP | Scalability |
| DSRED | Good | Good | N/A | |
| BLUE | Significant | Good | N/A | |
| REM | Significant | Good | N/A | Implementation complexity |
| SFB | Significant | N/A | Good | Complexity Misclassification |

with aggressive sources, or is used in asymmetric networks with a perpetually congested reverse path, TCP's throughput is very low when RED is used. Feng et al. [21] have proposed a modified RED (MRED) gateway for providing better control over the burstiness level while retaining the advantages of RED.

It is known that RED is not fair in distributing bandwidth among the different links [22], [23] it serves. To solve the problem of unfairness among links, Kim et al. [13] studied the Fair Buffering RED (FB-RED) for running TCP over ATM. The main idea of FB-RED is to use the bandwidth delay product of a link to calculate the drop probability of the RED queue associated with the link. They implemented two cases: (i) the first one uses the inverse of the bandwidth delay product to calculate the maximum drop probability; (ii) the second one uses the inverse of the square root of the bandwidth delay product to calculate the maximum drop probability. Although FB-RED increases the fairness among links, it however needs to track the information for all the links, resulting in *scalability problems* for large networks.

Lin et al. [14] proposed Fair RED (FRED) which relies on the usage of buffer space by the various flows (per-active-flow accounting) to determine the drop rate of the flows. Although it achieves a fair drop rate for different flows, it needs to track the state of each flow which may result in scalability problems as in [13].

To solve the scalability problem of FB-RED [13] and FRED [14], Ott [8] proposed Stabilized Random Early Drop (SRED). Like RED, it preemptively discards packets with a load dependent drop probability when the buffer in a router gets congested, but has an additional feature which estimates the number of active connections or flows; the estimate is obtained without collecting or analyzing state information of individual flows. Instead of calculating the average queue size, it uses the number of active flows and instantaneous queue sizes to determine the packet drop probability of the flows. Over a wide range of load, SRED can stabilize the buffer occupancy at a level which is independent of the number of active connections. Although SRED overcomes the scalability issues of [13], [14], it however, suffers from *low throughput*, even with a small number of traffic flows, as shown in their simulation results. Other research on improving the stability of RED can be found in [24]–[26].

Feng [16] showed that the effectiveness of RED depends, to a large extent, on the appropriate selection of the *RED parameters*. He showed that there is no single set of RED parameters that work well under all congestion scenarios. He therefore, proposed adaptive RED which self-parameterizes itself based on the traffic mix. Results show that such traffic-dependent parameterization of RED can effectively reduce packet losses, while improving the link utilization under a range of TCP loads. Adaptive determination of the RED parameters, however, requires additional computing power in the routers.

Parris et al. [17] pointed out that because RED relies on the congestion control mechanisms of TCP, it is not effective for UDP sources, which do not use congestion control. In order to solve the above problem, they proposed an extension of RED, called *Class Based Threshold (CBT)*, which sets the buffer thresholds for packet dropping on the basis of traffic type (TCP vs. UDP) and priority classes. The performance of TCP traffic is thus protected in the presence of UDP traffic. Wang et al. [27] have proposed an adaptive fuzzy-based queue management algorithm which obtains higher goodput and stable queue length than RED even under UDP flows.

May et al. [4] studied the *queuing delay* and *delay variance* of RED. It was 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. In [28], [29], the authors studied the performance of RED for Web applications. Christiansen et al. [28], [29] found that the RED parameters that result in the best link utilization, unfortunately, resulted in poor delay performance.

To study the *stability* of the RED algorithm, Firoiu et al. [30] modelled RED as a feedback control system. They pointed out that RED will induce network instability and major traffic disruption if not properly configured. RED becomes unstable when delay increases, or more strikingly, when link capacity increases [6].

Table 1 summarizes the features of DSRED against other variants of RED. It shows that DSRED improves throughput and also decreases the delay/jitter as compared to other RED variants.

## 3. Double Slope Random Early Detection (DSRED) Scheme

As discussed in the previous section, although a number of variants of RED have been proposed, the improvements have been limited due to the fact that many of them are based on modifying the parameters of RED. In this section, *we propose a new active queue management scheme*, called the Double Slope Random Early Detection (DSRED), to improve the performance of RED.

### 3.1 Notations

We define the following variables which are used to describe and evaluate the average packet drop probability of DSRED in Sects. 3.2 and 3.3, respectively.

- *N:* Buffer size at the gateway in units of packets.
- $K_l$: Threshold (in packet) for average queue length to start active packet dropping at the buffer; this is the same as in [3]
- $K_h$: Threshold (in packet) for average queue length to start active packet dropping at the buffer with probability of one; this is the same as in [3]
- $K_m$: Threshold (in packet) for average queue length to change the drop function slope;
- $\alpha$: Slope of the drop function for the first linear segment between $K_l$ and $K_m$. Since the congestion is not severe in this segment, we need to drop as few packets as possible. $\alpha$ is, therefore, chosen to have a small value;
- $\beta$: Slope of the drop function for the second linear segment between $K_m$ and $K_h$. In this segment, the packet drop probability should be high to warn senders of congestion. Therefore, $\beta$ should be larger than $\alpha$;
- $\gamma$: Mode selector for adjusting the slopes of the drop functions;
- $\lambda$: Rate of traffic arrival at the buffer in packets/second;
- $\mu$: Packet processing rate at the buffer in packets/second;
- $\rho$: Offered load factor defined as $\frac{\lambda}{\mu}$;
- *q:* Instantaneous queue length in packets;
- *avg:* Average queue length as defined in [3];
- *w:* Weight parameter as defined in [3] to calculate *avg*;
- $V(i)$: Steady state probability of $avg = i$;
- $p_a(i)$: Packet acceptance probability at State $i$;
- $p_d(i)$: Packet dropping probability at State $i$; this is probability that the buffer drops an arriving packet when $avg = i$. Note that $p_d(i) = 1 - p_a(i)$;
- $\lambda_i$: Effective arrival rate at the buffer at 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 at the buffer over all possible state;
- *D:* Average queuing delay at the buffer; this can be calculated by Little's formula;
- $Max_{drop}$: Maximum drop probability of RED;
- *Normalized throughput:* defined during a time period by $\frac{\text{total packets received at destination}}{\text{total packets sent by sources}}$.

### 3.2 The Principle of DSRED

As shown in Figs. 2 and 3, DSRED is based on dividing the buffer segment between $K_l$ and $K_h$ into two sub-segments separated at $K_m$. The drop function from $K_l$ to $K_h$ is described by two linear segments with slopes $\alpha$ and $\beta$ respectively. The slopes of these two linear segments are complementary, and are adjusted by a mode selector, $\gamma$. Although, in real world, $K_m$ will be configured by the gateway administrator, in this paper we set $K_m$ to $0.5(K_l + K_h)$ for illustration purposes. The drop function, $p_d(avg)$, of DSRED can then 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, $\alpha, \beta$ and *avg* are given by:

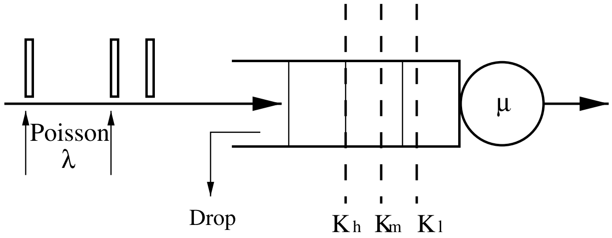

**Fig. 2** Model for DSRED buffer at gateway.

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

$$\beta = \frac{2\gamma}{K_h - K_l} \tag{3}$$

$$avg = (1-w)avg + wq \tag{4}$$

The above equations, which govern packet drops in DSRED, translate to the following rules:

- No packet is dropped when the average queue length, *avg*, is less than $K_l$;
- When the average queue length, *avg*, is between $K_l$ and $K_m$, packets are dropped according to the drop function with slope $\alpha$;
- When the average queue length, *avg*, is between $K_m$ and $K_h$, packets are dropped according to the drop function with slope $\beta$;
- When the average queue length, *avg*, is $K_h$ or higher, packets are dropped with a probability of one.

### 3.3 Packet Drop Probability of DSRED

We described our proposed DSRED scheme in Sect. 3.2. In this section, we develop an analytical model to study the average packet drop probability ($P_D$) and queueing delay ($D$) as a function of the offered load factor ($\rho$). The normalized throughput can then be obtained from $1 - P_D$.

We model packet arrivals to the buffer by a Poisson process with rate $\lambda$. Although Poisson traffic may not accurately reflect the arrival process in real networks [31], [32] having long range dependent traffic, it can provide a tractable model to understand the behavior of TCP. The validity of this assumption has been proved in [4], [20], [33].

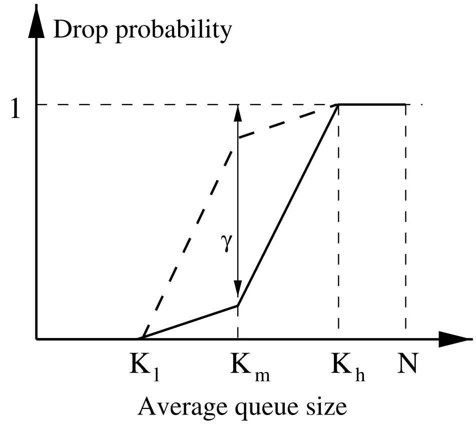

**Fig. 3** Drop function of DSRED.

As shown in Fig. 2, the buffer is modelled by a FIFO queue with a processing rate of $\mu$. An arriving packet is dropped with a probability based on the 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 the offered load factor. The state diagram is shown in Fig. 4, where $\lambda_i$ is the *effective arrival rate* for State $i$ (defined as $avg = i$), and $\mu$ is the constant departure rate.

$V(i)$, the steady state probability of the average queue length being in State $i$, can be expressed as [34]:

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

where $V(0)$ is given by:

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

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

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

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

The average packet drop probability $P_D$ is given by:

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

From Little's formula, the average queuing delay, $D$, is given by [34]:

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

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

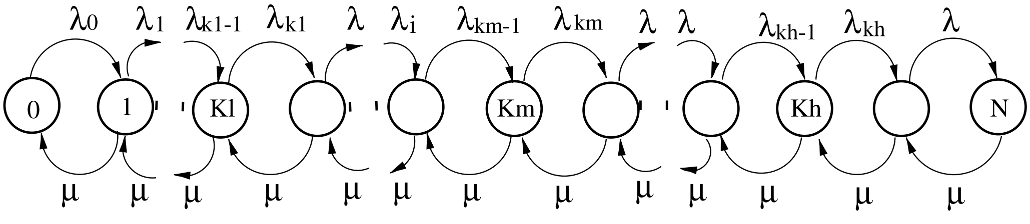

**Fig. 4** Steady state diagram for the buffer using DSRED.

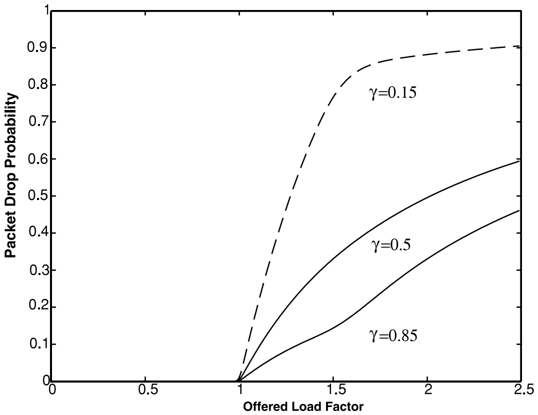

**Fig. 5** Theoretical curve of packet drop probability versus offered load factor for different $\gamma$ in DSRED.

### 3.4 Comparison between DSRED and RED

DSRED is *similar* to RED in two respects. First, both of them use linear drop functions to provide smoothly increasing drop action based on the average queue length. Secondly, they use the same function to calculate the average queue length. As a result of the above two similarities, 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 uses the average queue length, which is related to the long term congestion level. With an increase in congestion, the packet drop probability will increase at a higher rate, rather than at a constant rate (as in RED). This will give an early warning to hosts to backoff, preventing congestion from getting worse.
- The operating mode of DSRED can be easily adjusted by changing the slopes of the DSRED drop function using a single parameter, $\gamma$. By adjusting only $\gamma$, DSRED can initially achieve a high drop rate, followed by a low drop rate, or vice versa. This provides a flexible AQM scheme to handle complicated network congestion situations.

## 4. Simulation Configuration

OPNET 5.1, an event driven network simulation tool which is widely used in academia and industry, was used to carry out simulations to compare the performance of RED and DSRED. This section describes the simulation configuration, and presents a performance comparison between RED and DSRED.

The simulation configuration is shown in Fig. 6. To provide a fair comparison with RED, we used the same network topology and configuration as in [3], [14], [16]. 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 two cases: (i) *Heavy Load* and (ii) *Low Load*. The network simulation configuration parameters are as follows:

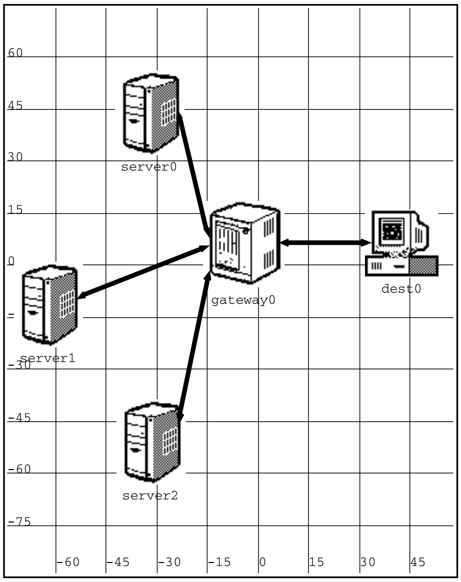

**Fig. 6** Network configuration for simulation.

- *Server0 to Gateway0:* Propagation delay 1 ms, link rate 100 Mbps.
- *Server1 to Gateway0:* Propagation delay 5 ms, link rate 100 Mbps.
- *Server2 to Gateway0:* Propagation delay 3 ms, link rate 100 Mbps.
- *Destination0 to Gateway0:* Propagation delay 5 ms, bottleneck link rate 10 Mbps. The sum of the link rates from the three TCP servers to the gateway is 300 Mbps, which is much higher than the bottleneck link rate of 10 Mbps, resulting in congestion at the bottleneck link.
- $\mu$*=1 ms/packet*.
- *Buffer Size*: 200 packets, as in [3].
- $K_l = 6$, $K_h = 20$: The values were chosen such that $K_h \geq 2K_l$, as suggested in [3].
- $w = 0.07$.
- $Max_{drop} = 0.1$ and 0.2 for $\gamma = 0.96$ and 0.91 respectively. We chose these parameter sets to make the studies more general.

To enable a fair comparison, we have used the same value of the propagation delay as in [3], [4]. The following measurements were made at the gateway, the server, and the destination.

- *TCP traffic received at the destination:* the number of packets received by the destination during the simulation period;
- *Server TCP load:* the number of packet sent by the TCP sources during the simulation period; we use

Heavy and Low for our simulation runs.

- *Queuing Delay:* queuing delay at the buffer;
- *Queue Size:* queue size of the buffer;
- *Packet Drop:* the number of packets dropped at the gateway.

## 5. Results

In this section, we present simulation results to compare the performance of DSRED and RED in terms of queuing *delay*, gateway *queue size*, and packet *drop* under low and heavy loads for various values of RED parameters. We also consider the *stability* and *performance* of DSRED for different values of *Round Trip Time* (RTT), which corresponded to different network sizes.

### 5.1 Delay, Queue Size and Packet Drop at Heavy Load

Figures 7, 8 and 9 compare the time averages of the queuing delay, queue size and packets dropped, respectively, for DSRED and RED under *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$. It is seen that the performance of *DSRED is better than RED for the entire simulation period under heavy load.* For example, at time=100 seconds, the queue size for DSRED is less than half of the value for RED (see Fig. 8), and the packet drop for DSRED is only one-fifth as compared to that of RED (Fig. 9).

To compare the performance for a different set of parameters ($\gamma = 0.91$ and $Max_{drop} = 0.2$), Figures 10, 11 and 12 compare the queuing delay, queue size and packet loss respectively between DSRED and RED. It is found that DSRED *continues to perform better* than RED.

### 5.2 Delay, Queue Size and Packet Drop at Low Load

Figures 13, 14 and 15 compare the queuing delay, queue size and packets dropped, respectively, for DSRED and RED under *Low Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$. *It is seen that DSRED performs better than RED for low loads also.*

To compare the performance for a different set of pa-

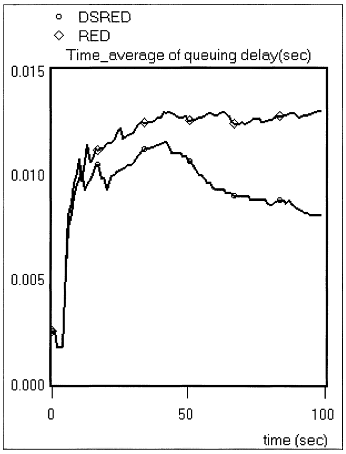

**Fig. 7** Time average of queuing delay for DSRED and RED under *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

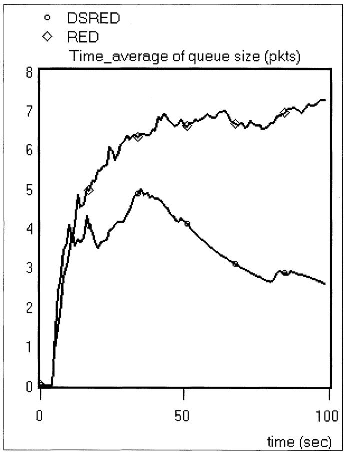

**Fig. 8** Time average of queue size for DSRED and RED under *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

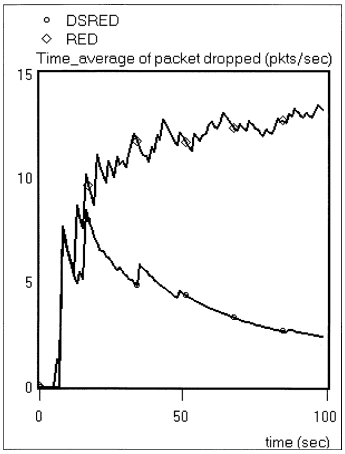

**Fig. 9** Time average of packets dropped for DSRED and RED under *Heavy Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

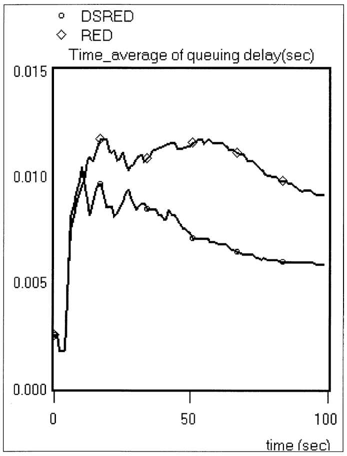

**Fig. 10** Time average of queuing delay for DSRED and RED under *Heavy Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

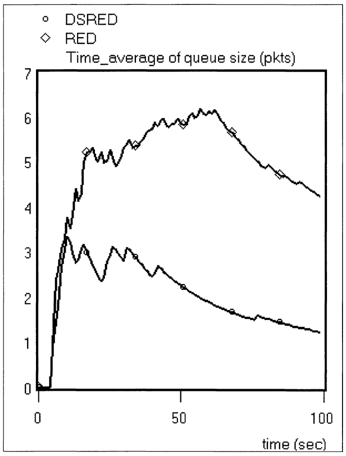

**Fig. 11** Time average of queue size for DSRED and RED under *Heavy Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

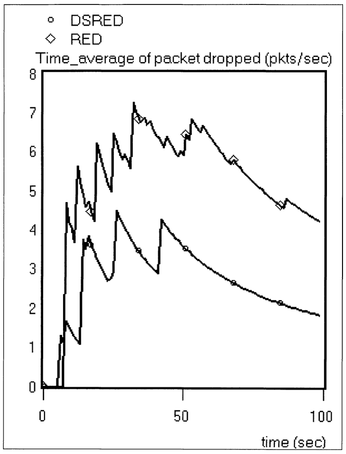

**Fig. 12** Time average of packets dropped for DSRED and RED under *Heavy Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

rameters ($\gamma = 0.91$ and $Max_{drop} = 0.2$), Figs. 16, 17 and 18 compare the delay, queue size and packet loss respectively between DSRED and RED. It is found that DSRED continues to perform better than RED.

### 5.3 Stability of DSRED

In this section, we compare the stability of DSRED and RED. Comparison of the delay, queue size and packet drop for heavy (Figs. 7 to 12) and low (Figs. 13 to 18) loads show that the change in delay, queue size and packet drop due to the load changing from low to heavy is less for DSRED as compared to RED. For example, at simulation time of 88 seconds, the queueing delay for DSRED is 8.75 msec for both light and heavy loads; on the contrary the delays are 12 and 13 msec at low and heavy loads, respectively, for RED. These figures also shows that *DSRED maintains a steady performance* at both heavy and low loads.

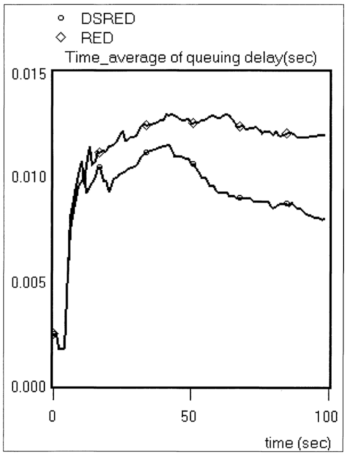

**Fig. 13** Time average of gateway queuing delay for DSRED and RED under *Low Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

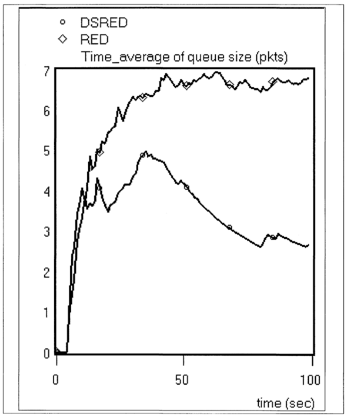

**Fig. 14** Time average of gateway queue size for DSRED and RED under *Low Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

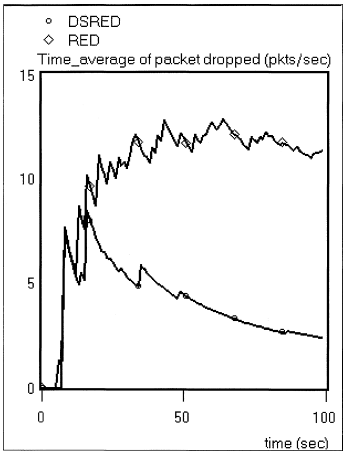

**Fig. 15** Time average of gateway packet dropped for DSRED and RED under *Low Load* for $\gamma = 0.96$ and $Max_{drop} = 0.1$.

### 5.4 Effect of Round Trip Time (RTT)

It has been shown [16] that the performance of RED depends on $Max_{drop}$, which is related to the network size (re-

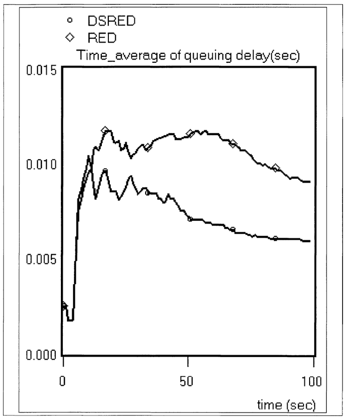

**Fig. 16** Time average of gateway queuing delay for DSRED and RED under *Low Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

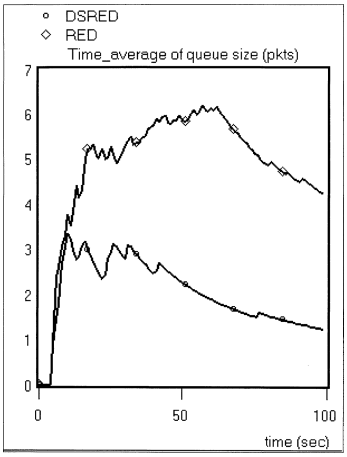

**Fig. 17** Time average of gateway queue size for DSRED and RED under *Low Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

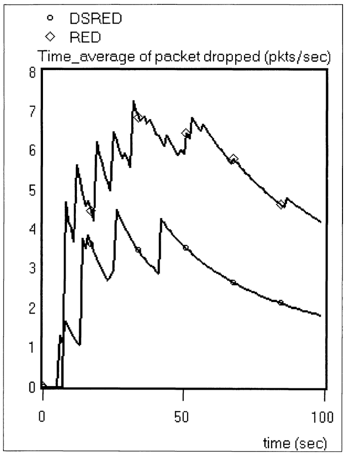

**Fig. 18** Time average of gateway packet dropped for DSRED and RED under *Low Load* for $\gamma = 0.91$ and $Max_{drop} = 0.2$.

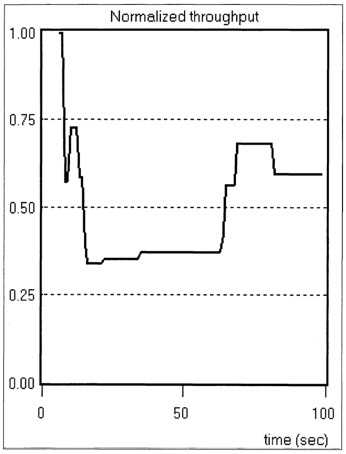

**Fig. 19** Normalized throughput of DSRED under *Heavy Load* for $\gamma = 0.96$.

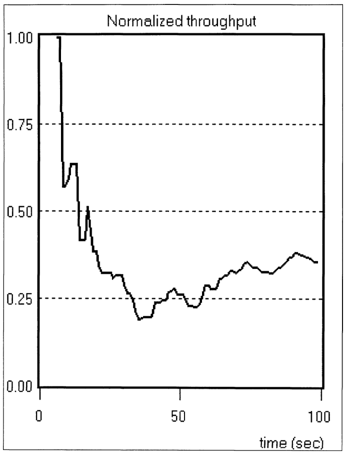

**Fig. 20** Normalized throughput of RED under *Heavy Load* for $Max_{drop} = 0.1$.

flected by RTT). In this subsection, we present the performance of DSRED and RED under the same parameter values as in Sect. 5, but with a different RTT. The network topology is the same as in Fig. 6, and in order to change the RTT, the propagation delays are set as follows:

- Propagation delays from *Server0*, *Server1*, *Server2* to *Gateway0* were set to 3, 15 and 9 ms, respectively.
- Propagation delay from *Destination0* to *Gateway0* was 5 ms.

The normalized throughput of DSRED and RED with $\gamma = 0.96$ and $Max_{drop} = 0.1$ are shown in Figs. 19 and 20, respectively, for heavy loads. At the end of the simulation period (100 secs), the throughput of DSRED and RED are about 0.6 and 0.36 respectively.

The above figures are repeated for the parameter set $\gamma = 0.91$, $Max_{drop} = 0.2$ in Figs. 21 and 22. Here also, we see that at the end of the simulation, the throughput of DSRED and RED are 0.5 and 0.35 respectively. It is seen that DSRED has higher normalized throughput during the entire simulation time for both the cases.

### 5.5 Summary of Performance

To provide a quick comparison between the performance of DSRED and RED, the results discussed above are summarized in Tables 2 and 3. In addition to the results presented in the previous sections, the tables also show the performance (along with their throughput) of DSRED and RED under medium load. The tables demonstrate the higher performance of DSRED in terms of queuing delay, queue size, packet drop, and throughput.

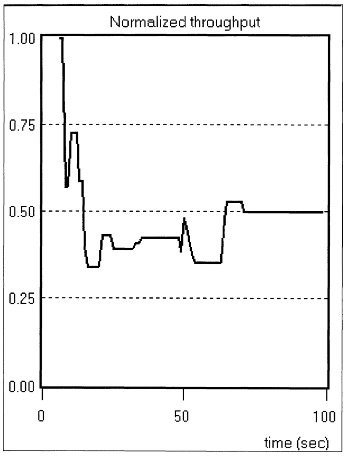

**Fig. 21** Normalized throughput of DSRED under *Heavy Load* for $\gamma = 0.91$.

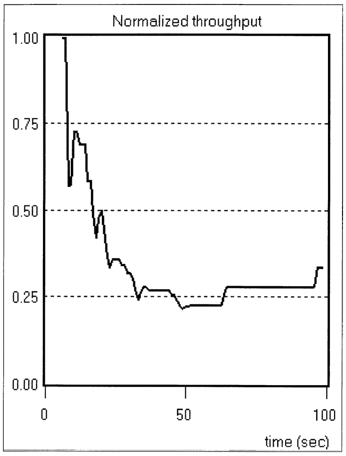

**Fig. 22** Normalized throughput of RED under *Heavy Load* for $Max_{drop} = 0.2$.

DSRED achieves lower *queuing delay* and *queue size* than RED under both low and heavy traffic loads. This is because, DSRED changes the slope of its drop function based on the level of congestion in the network. When the length of the queue exceeds a certain threshold, DSRED drops packets much more aggressively than RED.

## 6. Conclusions

In this paper, we proposed DSRED, an active queue management scheme to improve the performance of Internet gateways. DSRED exploits a two-segment drop function to achieve the improved performance.

DSRED achieves lower *queuing delay* and *queue size* than RED under both low and heavy traffic loads. This is because, DSRED changes the slope of its drop function based on the level of congestion in the network. When the length of the queue exceeds a certain threshold, DSRED drops packets much more aggressively than RED. It has been shown that DSRED also results in a *lower packet drop rate* (which results in a higher throughput) than RED.

Simulation results for different RTT values (representing different network sizes) have shown that DSRED exhibits more robustness than RED; the throughput and queuing delay of DSRED have also been shown to be steadier than RED.

In conclusion, our proposed active queue management scheme performs better than RED. It can be easily implemented in Internet routers with about 50 lines of C code. Future work consists of exhaustive study of the fairness of DSRED.

**Table 2** Performance comparison of DSRED ($\gamma = 0.96$) and RED ($Max_{drop} = 0.1$).

| | DSRED | | | RED | | |
|---|---|---|---|---|---|---|
| Measured Parameters | H Load | M Load | L Load | H Load | M Load | L Load |
| Normalized Throughput | 0.525 | 0.808 | 0.372 | 0.445 | 0.49 | 0.355 |
| Average Queuing Delay (ms) | 8.13 | 6.25 | 8.13 | 13.1 | 11.2 | 11.8 |
| Average Queue Size (Pkt) | 2.6 | 1.5 | 2.7 | 7.3 | 6.3 | 6.8 |
| Average Packet Drop (Pkt/s) | 2.5 | 1.87 | 2.5 | 13.1 | 10 | 11.25 |

**Table 3** Simulated gateway performance for DSRED ($\gamma = 0.91$) and RED ($Max_{drop} = 0.2$).

| | DSRED | | | RED | | |
|---|---|---|---|---|---|---|
| Measured Parameters | H Load | M Load | L Load | H Load | M Load | L Load |
| Normalized Throughput | 0.682 | 0.689 | 0.64 | 0.482 | 0.487 | 0.47 |
| Average Queuing Delay (ms) | 5.9 | 5.9 | 5.9 | 9.06 | 9.06 | 9.06 |
| Average Queue Size (Pkt) | 1.3 | 1.3 | 1.3 | 4.3 | 4.3 | 4.8 |
| Average Packet Drop (Pkt/s) | 1.9 | 1.9 | 1.9 | 4.2 | 4.2 | 4.2 |

## References

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

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

[3] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Trans. Netw., vol.1, no.4, pp.397–413, Aug. 1993.

[4] M. May, T. Bonald, and J.-C. Bolot, "Analytic evaluation of RED performance," IEEE INFOCOM, pp.1415–1424, Tel-Aviv, Israel, March 2000.

[5] R. Verma, A. Iyer, and A. Karandikar, "Towards an adaptive RED algorithm for achieving delay-loss performance," IEE Proc. Commun., vol.150, no.3, pp.163–168, June 2003.

[6] S.H. Low, F. Paganini, J. Wang, and J.C. Doyle, "Linear stability of TCP/RED and a scalable control," Comput. Netw., vol.43, pp.633–647, 2003.

[7] 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 J. Sel. Areas Commun., vol.17, no.6, pp.1159–1169, June 1999.

[8] T.J. Ott, T.V. Lakshman, and L. Wong, "SRED: Stabilized RED," IEEE INFOCOM99, pp.1346–1355, New York, March 1999.

[9] W.C. Feng, D.D. Kandlur, and D. Saha, "Adaptive packet marking for maintaining end to end throughput in a differentiated service Internet," IEEE/ACM Trans. Netw., vol.7, no.5, pp.685–697, Oct. 1999.

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

[11] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," RFC 2598, June 1999.

[12] Y. Bernet, A. Smith, S. Blake, and D. Grossman, "A conceptual model for diffserv routers," draft-ietf-diffserv-model-04.txt, March 2000.

[13] W.-J. Kim and B.G. Lee, "The FB-RED algorithm for TCP over ATM," IEEE GLOBECOM, pp.551–555, Sydney, Australia, Nov. 1998.

[14] D. Lin and R. Morris, "Dynamics of random early detection," ACM SIGCOMM, pp.127–137, Cannes, France, Sept. 1997.

[15] T.V. Lakshman, A. Neidhardt, and T.J. Ott, "The drop from front strategy in TCP and in TCP over ATM," IEEE INFOCOM, pp.1242–1250, San Francisco, CA, March 1996.

[16] W.C. Feng, D.D. Kandlur, D. Saha, and G. Kang, "Self-configuring RED gateway," IEEE INFOCOM, pp.1320–1328, New York, March 1999.

[17] M. Parris, K. Jeffay, and F. Donelson Smith, "Lightweight active router queue management for multimedia networking," Proc. SPIE, pp.162–174, San Jose, CA, USA, Jan. 1999.

[18] Y. Sakai, Y. Matsushita, T. Matsuda, and M. Yamamoto, "Mismatch of packet error recovery mechanisms for bit error and handover in wireless TCP," IEICE Trans. Commun., vol.E87-B, no.9, pp.2626–2633, Sept. 2004.

[19] S. Floyd, "Recommendation on using the gentle variant of red," www.icir.org/floyd/red/gentle.html, March 2000.

[20] M. May, J.-C. Bolot, A. Jean-Marie, and C. Diot, "Simple performance models of differentiated services schemes for the Internet," IEEE INFOCOM99, pp.1385–1394, New York, March 1999.

[21] G. Feng, A.K. Agarwal, A. Jayaraman, and C.K. Siew, "Modified RED gateways under bursty traffic," IEEE Commun. Lett., vol.8, no.5, pp.323–325, May 2004.

[22] Y. Atsumi, E. Kondoh, O. Altintas, and T. Yoshida, "Improving fairness and stability in best effort service: A new congestion control algorithm for SACK-TCP," IEICE Trans. Commun., vol.E81-B, no.11, pp.2023–2033, Nov. 1998.

[23] C.M. Pazos, S.A. Juan C, and M. Gerla, "Using back pressure to improve TCP performance with many flows," Proc. INFOCOM99, pp.431–438, New York, March 1999.

[24] R.J. La, "Instability of a tandem network and its propagation under RED," IEEE Trans. Autom. Control, vol.49, no.6, pp.1006–1011, June 2004.

[25] T. Ziegler, S. Fdida, C. Brandauer, and B. Hechenleitner, "Stability of RED with two-way tcp traffic," Ninth International Conference on Computer Communications and Networks, pp.214–219, Las Vegas, Oct. 2000.

[26] C.V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," IEEE Trans. Autom. Control, vol.47, no.6, pp.945–959, June 2002.

[27] C. Wang, B. Li, K. Sohraby, and Y. Peng, "AFRED: An adaptive fuzzy-based control algorithm for active queue management," 28th Annual IEEE International Conference on Local Computer Networks, 2003, pp.12–20, Oct. 2003.

[28] M. Christiansen, K. Jeffay, D. Ott, and F. Smith, "Tuning RED for Web traffic," IEEE/ACM Trans. Netw., vol.9, no.3, pp.249–264, June 2001.

[29] L. Le, J. Aikat, K. Jeffay, and F.D. Smith, "The effects of active queue management on web performance," SIGCOMM: 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.265–276, 2003.

[30] V. Firoiu and M. Borden, "A study of active queue management for congestion control," IEEE INFOCOM, pp.1435–1444, Tel-Aviv, Israel, March 2000.

[31] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self similar nature of ethernet traffic (extended version)," IEEE/ACM Trans. Netw., vol.2, no.1, pp.1–15, Feb. 1994.

[32] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," IEEE/ACM Trans. Netw., vol.3, no.3, pp.226–244, June 1995.

[33] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," ACM SIGCOMM2000, pp.151–160, Stockholm, Sweden, Oct. 2000.

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

**Bing Zheng** received his Ph.D. degree in electrical engineering in August 2001, from University of Dayton. He received a B.Sc. in physics in 1984 and an M.Sc. in electro-optics in 1990. Before joining the Department of Electrical and Computer Engineering of University of Dayton in August 1997, he was an associate professor working in tunable WDM technology, optical fiber technology, integrated optoelectronic technology, and flat panel display technology with the University of Electronic Science and Technology of China (UESTC). His working areas include TCP, IP, ATM, multimedia transmission, VHDL/FPGA/Digital system design, tunable WDM technology, fiber optical technology, and photonic/optoelectronic technology etc.

**Mohammed Atiquzzaman** earned his M.Sc. and Ph.D. degrees in Electrical Engineering and Electronics from the University of Manchester Institute of Science and Technology, U.K. Currently he is a faculty in the School of Computer Science at the University of Oklahoma, USA. Dr. Atiquzzaman has over 180 publications in leading journals and conferences in the areas of communications & networking, parallel/distributed computing and image processing. His research has been funded through state and federal agencies, such as National Aeronautics and Space Administration (NASA), National Science Foundation (NSF), and the United States Air Force. Mohammed is the co-editor-in-chief of the *Computer Communications* journal, and serves on the editorial boards of the *IEEE Communications Magazine*, *Journal of Real-Time Imaging*, *Telecommunication Systems* journal, International Journal on *Wireless and Optical Communications*, and *International Journal of Sensor Networks*. He was the technical program co-chair of the *2003 Workshop on High Performance Switching and Routing (HPSR2003)*, and *SPIE Quality of Service over Data Networks* conference. He has been involved in the program committees of many leading international conferences such as INFOCOM, Globecom, ICCCN, LCN, etc. He has delivered keynote speeches and invited talks at various conferences, and serves as panelist in conferences and funding organizations. He is a senior member of the IEEE. His homepage is at www.cs.ou.edu/˜atiq