# Reduction of queue oscillation in the next generation Internet routers

By: Shan Suthaharan

S. Suthaharan (2007), "Reduction of queue oscillation in the next generation Internet routers," Computer Communications Journal, Elsevier, vol. 30, no. 18, 10 Dec. 2007, pp. 3881-3891.



**Abstract:**
The Internet routers employing the random early detection (RED) algorithm for congestion control suffer from the problem of chaotic queue oscillation. It is well known that the slowly varying nature of the average queue size computed using an exponentially weighted moving average (EWMA) used in the RED scheme causes this chaotic behavior. This paper presents a new mathematical function to model the weighting parameter used in the EWMA. The proposed weighting function incorporates the knowledge of the dynamic changes in the congestion characteristics, traffic characteristics and queue normalization. Using this pragmatic information eliminates the slowly varying nature of the average queue size. It is evident from our simulations that the proposed approach not only reduces the chaotic queue oscillation significantly but also provides predictable low delay and low delay jitter with high throughput gain and reduced packet loss rate even under heavy load of traffic conditions.
**Keywords:** Random early detection; Queue oscillation; Congestion characteristics; Traffic characteristics; Queue normalization

**Article:**
## *1. Introduction*
At this time the drop-tail mechanism is the only practicable solution in the Internet routers to control network congestion when the queue buffer becomes full. When the Internet routers start to drop significant amount of packets, in practice the network operators add more memory (buffer) to the routers. This practicable solution leads to high delay and delay jitter at the Internet routers causing poor Quality of Service (QoS) to the Internet users. In generalized term the schemes called the Active Queue Management (AQM) techniques have been proposed to alleviate this problem. The most commonly referred AQM technique is the random early detection (RED) and it was introduced by Floyd and Jacobson [1] in 1993.

The RED scheme uses a number of processes and control parameters. The processes include calculation of average queue size using EWMA and calculation of probabilities for making decisions to drop packets.
The control parameters include a weight parameter $w_q$ (used to calculate the average queue size in order to achieve a stable operating point for the queue size), two thresholds parameters $q_{min}$ and $q_{max}$ (used to control the average queue size) and the maximum packet drop probability $p_{max}$ (used to make packet drop decisions). The control parameters can be tuned to achieve low delay and low delay jitter with high throughput gain and reduced packet loss rate in a congested network. Network operators are naturally interested in the predictable low delay and low delay jitter, which are the major components of the QoS requirements. However, the guideline associated with the RED scheme suggests choosing a small constant value for $w_q$. It makes the average queue size sensitive to different congestion characteristics, traffic characteristics and queue normalization resulting in an instantaneous queue oscillation.

The instantaneous queue oscillation leads the system to a chaotic state. The chaotic behavior of the instantaneous queue oscillation has been studied by Ranjan, Abed and La [2] using the *bifurcation* behavior. They stated that the bifurcations occur as the average queue size slowly varied due to small and inaccurate value of $w_q$ and it moves the queue dynamics from a stable fixed point to an oscillatory behavior and finally to a chaotic state. Thus the network operators cannot provide their customers a QoS delivery in congested routers. Therefore, continuous tuning of control parameters for different traffic conditions becomes necessary to adapt to the average queue size and to keep the system from reaching to a chaotic state. In spite of its drawbacks, the RED algorithm has been recommended by the Internet Engineering Task Force (IETF) with the expectation that the users of the RED in the Internet routers make their own effort to select suitable control parameters for their network [3]. Therefore it becomes necessary to find an automatic tuning mechanism for the RED.

Since the introduction of the RED algorithm several AQM techniques [4–14,17,18] have been proposed including the adaptive RED (denoted by ARED) [11], the refined RED (we denote it by R-RED) [12], the double slope RED (DS-RED) [13] and the modified RED (M-RED) [14]. The ARED scheme was proposed in 2001 as an improvement to the RED scheme. The ARED, while satisfying all the AQM requirements [1], increases throughput when the network is not congested and decreases the congestion otherwise. It also offers *predictable* average delay at the router which satisfies the QoS requirement. The ARED scheme incorporates the link speed to set $w_q$ (which is a constant) automatically to achieve its goals. The ARED keeps the average queue size within a target range halfway between the thresholds $q_{min}$ and $q_{max}$ modifying $p_{max}$ slowly and infrequently using the AIMD (additive-increase–multiplicative-decrease) policy. This process gives a better result than the RED in general for throughput gain with the added advantage of a predictable queuing delay. However, the chaotic queue oscillation is still a problem in ARED especially in a heavily congested network.

M-RED technique [14] was proposed by Feng et al. mainly for the purpose of handling bursty traffic conditions in RED gateways. The authors suggested a simple modification to the conditions used by the RED to make packet marking/dropping decisions. Although this approach pro-vides higher throughput than RED it cannot reduce the chaotic queue oscillation because it does not incorporate a mechanism for that. Similarly the DS-RED technique [13] was also developed focusing on improving the throughput compared to the RED scheme. In this technique the fundamental way of calculating the drop function in RED has been changed to achieve better stability in the changes in average queue size. The same as the M-RED this technique also does not incorporate any mechanism to control the queue oscillation.

The queue oscillation problem that is caused by the slowly varying nature of the average queue size in RED was addressed by Wang and Shin in 1999 in their R-RED technique [12]. In the R-RED technique the dynamic adjustments of the queue weight $w_q$ and maximum drop probability $p_{max}$ were incorporated. The R-RED sets different values for the weight within six different windows that are defined with respect to the surplus over buffer size. The main purpose of doing this is to detect the transient congestion quickly when the buffer is nearly full and to sense the congestion resulting from bursty traffic. At the same time they also tuned the setting of maximum drop probability so that the burstiness of incoming traffic is controlled to avoid buffer overflow. The parameters $w_q$ and $p_{max}$ are adjusted discretely (not continuously) at different levels between the minimum and maximum threshold within the queue buffer.

This paper presents a technique called “AutoRED” that adopts the desired properties of the RED or the ARED scheme, when it is used with the RED-based or the ARED-based schemes, respectively. In contrast to R-RED the proposed approach adjusts the parameter $w_q$ continuously and uses the same approach as the RED to modify $p_{max}$. The proposed technique modifies the fundamental way of calculating the average queue size using EWMA in the RED scheme. It models the weight used in the EWMA model by using the information of the network characteristics, namely congestion characteristics, traffic characteristics and queue normalization. This approach eliminates the slowly varying nature of the average queue size while preserving RED’s original goal of filtering the changes in the instantaneous queue size resulted from the burst of traffic or transient congestion so that these instantaneous changes do not give significant increases in the average queue size. Thus AutoRED reduces the instantaneous queue oscillation to avoid the networks going into a chaotic state even in a highly congested network. It can also provide predictable low delay and low delay jitter with high throughput gain and reduced packet loss rate. The remainder of this paper is organized as follows: the core structure of the RED scheme is presented in Section 2. In Section 3 the proposed method is discussed. Section 4 presents simulation results. Section 5 presents the final conclusion.

## 2. Core structure of RED

In the RED scheme [1] the average queue size denoted by $q_{avg,t}$ is calculated using the following EWMA equation:

$$q_{avg,t} = (1 - w_q) * q_{avg,t\text{-}1} + w_q q_t \qquad (1)$$

In this equation qt represents the instantaneous queue size at time t and wq is a small constant which is chosen according to the guideline of the RED scheme to allow network operators to tune the performance for different network conditions. This formula is used to compute the average queue size when the instantaneous queue qt is not empty. However, when the queue is empty the RED scheme [1] suggests the following formula to calculate the average queue size:

$$q_{avg,t} = (1 - w_q)^m * q_{avg,t\text{-}1}\ ;\ m = f(t - T) \qquad (2)$$

In this equation $t$ is the current time, $T$ is the start of the queue idle time and $f(\cdot)$ is a linear function of the time $t$. In order to interpret the effect of the weight parameter $w_q$ and the network characteristics on the average queue size, Eq. (1) is rewritten as follows:

$$q_{avg,t} - q_{avg,t\text{-}1} = w_q * (q_t - q_{avg,t\text{-}1}) \qquad (3)$$

In this equation, the term $q_{avg,\,t}$ — $q_{avg,t\text{-}1}$ on the left hand side defines the changes in the average queue size over a unit time. Similarly the term $q_t$ — $q_{avg,t\text{-}1}$ on the right hand side describes the changes in the instantaneous queue size with respect to the average queue size over a unit time. The above equation reflects *two* types of network characteristics namely the congestion characteristics (which include transient congestion) and the traffic characteristics (which include burst of traffic). The purpose of the weight $w_q$ on the right hand side is to filter the short-term changes in the queue size resulted from the burst of traffic or transient congestion. Thus the short-term change in the queue size does not significantly affect the average queue size increase, because the weight $w_q$ is set to a small constant value as recommended by the RED scheme.

Alternatively, the ARED recommends the weighting parameter $w_q$ to be calculated using a time constant —1/ ln(1 — $w_q$) and the channel capacity [11]. This alternative way of obtaining a value for $w_q$ is now used in both the RED scheme and the ARED scheme. From the Eq. (3), we can see that the average queue size increase is small and it only reflects the instantaneous network characteristics, not its dynamic changes in the network characteristics.

The goal of this paper is to model the weight $w_q$ as a time dependent parameter reflecting the dynamics of three types of network characteristics: (i) dynamics of congestion characteristics which include changes in transient congestion, changes between congestion and no congestion status, (ii) dynamics of traffic characteristics which include changes in burst of traffic and steady state and (iii) queue normalization. Incorporating the stated network characteristics into the calculation of $w_q$ allows the proposed scheme to derive a meaningful increase in the average queue size and eliminates the slowly varying nature of the average queue size that leads to chaos. As a result the chaotic queue oscillation is avoided or reduced significantly.

## 3. Proposed method

The proposed scheme redefines the EWMA model used in the RED scheme as follows:

$$q_{\mathrm{avg},t} - q_{\mathrm{avg},t-1} = w_{q,t}(q_t - q_{\mathrm{avg},t-1}) \qquad (4)$$

and

$$w_{q,t} = \frac{2p_t(1-p_t)(5.923 + |q_t - q_{\mathrm{avg},t-1}|)}{bs * \ln(5.923 + |q_t - q_{\mathrm{avg},t-1}|)} \qquad (5)$$

where $q_t$ is the instantaneous queue size at time $t$, $q_{avg,t-1}$ is the average queue size at time $t$ - 1, $q_{avg,t}$ is the average queue size at time $t$, $w_{q,t}$ is the newly defined time dependent weighting function, $p_t$ is the probability that the net-work can lead to congestion at each step according to the information available at current time $t$, ln() is the natural logarithm, and $bs$ is the buffer size. The weight $w_{q,t}$ is modeled as a combination of three network characteristics: dynamics of congestion characteristics, dynamics of traffic characteristics and queue normalization. To further clarify the proposed model, the definition of the weighting parameter $w_{q,t}$ is rewritten as a product of three functions as follows:

$$w_{q,t} = p_t(1 - p_t) * \frac{2(5.923 + |q_t - q_{\mathrm{avg},t-1}|)}{\ln(5.923 + |q_t - q_{\mathrm{avg},t-1}|)} * \frac{1}{bs} \tag{6}$$

The first, the second and the third functions are denoted by $\mathcal{I}_t$, $\mathcal{J}_t$ and $\mathcal{K}_{bs}$, respectively, as follows:

$$\mathcal{I}_t = p_t(1 - p_t) \tag{7}$$

$$\mathcal{J}_t = \frac{2(5.923 + |q_t - q_{\mathrm{avg},t-1}|)}{\ln(5.923 + |q_t - q_{\mathrm{avg},t-1}|)} \tag{8}$$

$$\mathcal{K}_{bs} = \frac{1}{bs} \tag{9}$$

### 3.1. Congestion characteristics

The first function $I_t$ is time dependent and it describes the dynamics of congestion characteristics of the network. It is used in the calculation of the average queue size so that the average queue size can be changed with respect to the dynamics of congestion characteristics of the network. It represents the probability that the system can lead to congestion in two steps (two consecutive times) according to the information available at time $t$. In order to describe this model mathematically, a new random variable $\mathcal{Y}_t$ as a function of $q_t — q_{avg,t-1}$ (which is an indicator for congestion) is defined. It describes the dynamic changes in congestion characteristics of the network as follows:

$$\mathcal{Y}_t = \begin{cases} 1 & q_t - q_{\mathrm{avg},t-1} > 0 \\ 0 & q_t - q_{\mathrm{avg},t-1} \leqslant 0 \end{cases} \tag{10}$$

where 1 indicates that the network is leading to congestion and 0 indicates that the network is leading to no congestion.

The variable $\mathcal{Y}_t$ forms a sequence of independent Bernoulli trials [15] between two possible outcomes, congestion (1) or no congestion (0). If $p_t$ represents the probability that the network can lead to congestion at time $t$+1 based on the queue status at time $t$ then the number of steps $\mathcal{Z}_t$ required to obtain the first indication of congestion obey the probability law of Geometric distribution [15]:

$$\mathcal{P}[\mathcal{Y}_t = n] = p_t(1 - p_t)^{n-1} \tag{11}$$

Therefore, $P[\mathcal{Y}_t = 2] = p_t\,(1 — p_t)$ represents the probability that the network is leading to congestion in two steps according to the information available at time $t$. In other words, it defines the probability that the system alternates between congestion and no congestion in two consecutive steps. In the proposed model this probability is used to alter $q_t — q_{avg,t-1}$ in order to incorporate the dynamics of congestion characteristics in the calculation of the average queue size. Fig. 1 shows the parameter $I_t$ as a function of $p_t$ and we can see that the dynamic range of this component is [0,0.25].

#### 3. 1. 1. Calculating the probability $p_t$

The probability ($p_t$) that the network can lead to congestion at time $t$ + 1 based on the queue status at time $t$ is calculated using the following equation:

$$p_t = n_t / (n_t + n'_t).$$

In this equation $n_t$ and $n'_t$ stand for the number of times $q_i \geq q_{avg,i\text{-}1}$ and $q_i < q_{avg,i\text{-}1}$, respectively, within the time duration $i = 0$ ..., $t$. Hence the probability that the network can lead to no congestion at time $t + 1$ based on the queue status information at time $t$ would be

$$1 - p_t = n'_t / (n_t + n'_t)$$

This process is used in the proposed approach to calculate the congestion characteristics function $I_t = p_t\,(1 - p_t)$ dynamically.

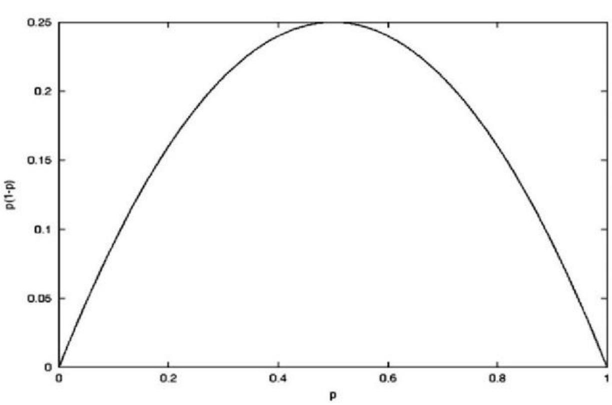


Fig. 1. A mathematical function that describes the dynamics of congestion characteristics.

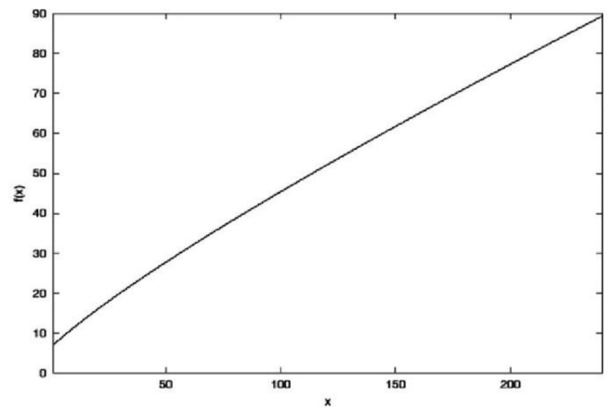


Fig. 2. A mathematical function that describes the dynamics of traffic characteristics.

### *3.2. Traffic characteristics*

The second function $\mathcal{J}$ is also time dependent but it describes the dynamics of the traffic characteristics of the network. It is used in the calculation of the average queue size so that the average queue size can be changed with respect to the dynamics of the traffic characteristics of the network.

In order to model this parameter, the effect of 1-unit packet increase in the queue size over two consecutive steps $t$ and $t + 1$ is considered. That is, we call "1-unit packet increase" when $q_t - q_{avg,t-1} = 1$ or when $q_t - q_{avg,t-1} = x_t$ and $q_{t+1} - q_{avg,t} = x_t + 1$. Suppose there was a 1-unit packet increase in the queue size at time $t$ (i.e. $q_t - q_{avg,t-1} = x_t = 1$) then the effect on the traffic characteristics by a 1-unit packet increase at time $t + 1$ (i.e. $q_{t+1} - q_{avg,t} = 1 + 1$) is 50%. Similarly if there was a 2-unit packet increase at time $t$ (i.e. $q_t - q_{avg,t-1} = x_t = 2$) then the effect on the traffic characteristics by a 1-unit packet increase at time $t + 1$ (i.e. $q_{t+1} - q_{avg,t} = 2 + 1$) is 33.3% which is less than 50%. Using this trend the effect of 1-unit packet increase on the traffic characteristics over two consecutive steps can be defined as the following function $f(x_t)$ with respect to increasing queue size $x_t$:

$$|f(x_t + 1) - f(x_t)| \leq 0.5 \qquad (12)$$

where the queue size increase $x_t$ satisfies $1 \leq x_t \leq bs$. In the proposed model the following function is chosen:

$$\mathcal{J}_t = f(x_t) = 2.0 * \frac{(5.923 + x_t)}{\ln(5.923 + x_t)} \qquad (13)$$

The reason for choosing this function is that it satisfies the modeling requirement and the condition in Eq. (12) as follows:

$$\left| 2.0 * \frac{(5.923 + x_t + 1)}{\ln(5.923 + x_t + 1)} - 2.0 * \frac{(5.923 + x_t)}{\ln(5.923 + x_t)} \right| \leqslant 0.5 \qquad (14)$$

The Fig. 2 shows that the dynamics of the traffic characteristics follow the linear property with respect to increasing queue size. However Fig. 3 confirms that an effect of 1-unit packet increase on the dynamics of the traffic characteristics is nonlinear with respect to network's increasing queue size.

### *3.3. Justification for the constant 5.923*

Let us first generalize the Eq. (14) as follows to obtain a value for the parameter "$a$":

$$\left| 2.0 * \frac{(a + x_t + 1)}{\ln(a + x_t + 1)} - 2.0 * \frac{(a + x_t)}{\ln(a + x_t)} \right| \leqslant 0.5 \qquad (15)$$

In the proposed model the effect of 1-unit packet increase on the dynamics of traffic characteristics, when 1-unit packet increase was in the queue, is 50%. That is the following condition is satisfied when $x_t = 1$:

$$\left| 2.0 * \frac{(a+2)}{\ln(a+2)} - 2.0 * \frac{(a+1)}{\ln(a+1)} \right| \leqslant 0.5 \qquad (16)$$

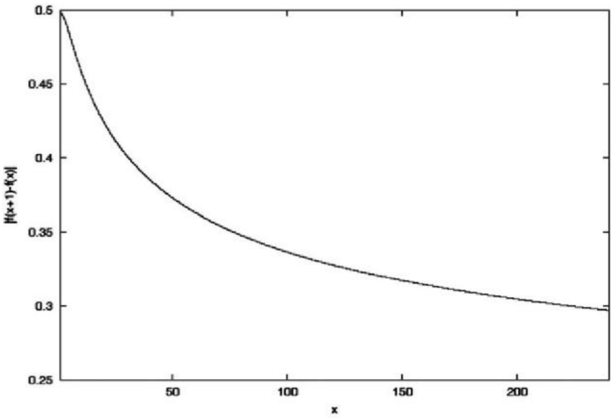


Fig. 3. Function: the effect of 1-unit packet increase.

Table 1
Calculating the constant 5.923

| *a* | *f*(*a*) | *a* | *f*(*a*) | *a* | *f*(*a*) |
|---|---|---|---|---|---|
| 5.905 | 0.499809 | 5.917 | 0.49981 | 5.929 | 0.49981 |
| 5.906 | 0.49981 | 5.918 | 0.49981 | 5.930 | 0.49981 |
| 5.907 | 0.49981 | 5.919 | 0.49981 | 5.931 | 0.49981 |
| 5.908 | 0.49981 | 5.920 | 0.49981 | 5.932 | 0.49981 |
| 5.909 | 0.49981 | 5.921 | 0.49981 | 5.933 | 0.49981 |
| 5.910 | 0.49981 | 5.922 | 0.49981 | 5.934 | 0.49981 |
| 5.911 | 0.49981 | 5.923 | 0.49981 | 5.935 | 0.49981 |
| 5.912 | 0.49981 | 5.924 | 0.49981 | 5.936 | 0.49981 |
| 5.913 | 0.49981 | 5.925 | 0.49981 | 5.937 | 0.49981 |
| 5.914 | 0.49981 | 5.926 | 0.49981 | 5.938 | 0.49981 |
| 5.915 | 0.49981 | 5.927 | 0.49981 | 5.939 | 0.49981 |
| 5.916 | 0.49981 | 5.928 | 0.49981 | 5.940 | 0.499809 |

To find a numeric value for the parameter "*a*" the mathematical expression on the left hand side is graphed for a range of values of the parameter "*a*". This graph is presented in Fig. 4. From this figure we can see that the function is closer to the maximum value of 0.5 within the range $5.90 \leq a \leq 5.95$. Therefore we extracted the data in this range and presented in Table 1 and then plotted them in Fig. 5. Using the data in this table the average of the parameter "*a*" is calculated to 5.9225 (rounded to 5.923) and used in the model.

### *3.4. Queue normalization*

The third function $\mathcal{K}_{bs}$ is a time independent parameter and it allows normalization of instantaneous queue size changes with respect to the buffer size. Therefore using these three parameters $I_t$, $\mathcal{J}_t$, and $\mathcal{K}_{bs}$ the weight $w_{q,t}$ can be obtained dynamically to incorporate the dynamic changes in the congestion characteristics and traffic characteristics in the calculation of average queue size. This new way of calculating the weight parameter eliminates the chaotic queue oscillation. Hence the weight $w_q$ is adjusted according to the 1-unit packet increase characteristics using this mathematical expression. It eliminates the slowly varying nature of the average queue size and provides continuous adjustment to the average queue size with respect to the traffic and congestion characteristics resulted from 1-unit packet increase in the instantaneous queue.

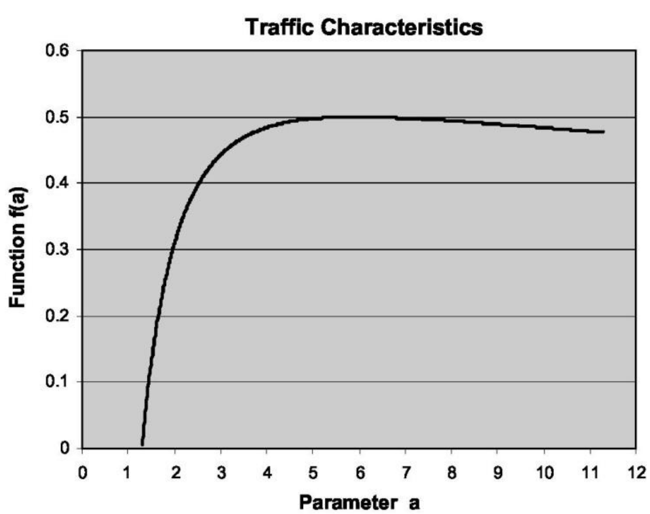


Fig. 4. Function: finding the constant 5.923.

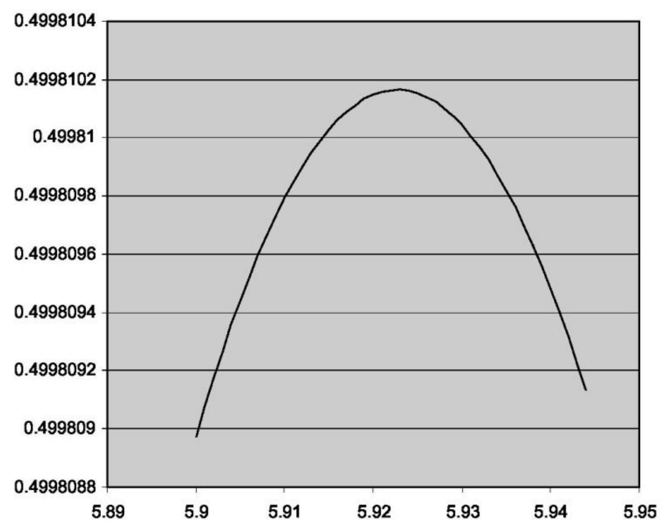


Fig. 5. Function: finding the constant 5.923.

## 4. Simulation results

The experimental set up for the simulations use a simple dumbbell topology, shown in Fig. 6, with a congested link of 10 Mbps bandwidth and 20 ms delay between the routers R1 and R2 in the bottleneck link. NS-2 is used in the simulations [16].

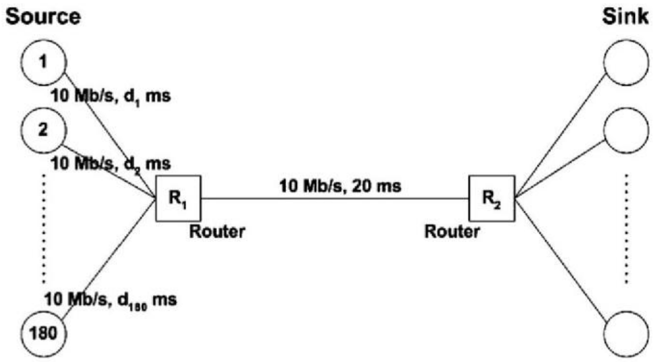


Fig. 6. Network configuration for the simulations.

It has 180 source (flows) links with bandwidth of 10 Mbps each and random delay ($d_1$, $d_2$,...,$d_{180}$) between 0 and 270 ms each. It also has 180 sink links. The simulation uses 180 FTP connections using TCP Reno with the maximum window sizes of 8000 and with ECN (Early Congestion Notification) mechanism turned on. The buffer size of 240 packets at bottleneck router is assigned and it is defined as a function 2 * $q_{max}$ of the maximum threshold $q_{max}$ in RED. The duration of the simulation is 60 s. TCP packet size is set to 1500 bytes.

The RED control parameters used in this simulation are (i) $w_q$ = 0.0004; (ii) $q_{min}$ = 40; (iii) $q_{max}$ = 120; (iii) $q_{max}$ = 3$q_{min}$; (iv) $p_{max}$ = 0.02; (v) ECN bits turned on; and (vi) gentle-mode turned on. The value of $p_{max}$ is set to 0.02 (1/50) to have roughly 1 out of 50 of the arriving packets to be dropped when the average queue size is at halfway between $q_{min}$ and $q_{max}$. The value 0.0004 of $w_q$ is calculated based on the guideline [11] suggested in ARED scheme.

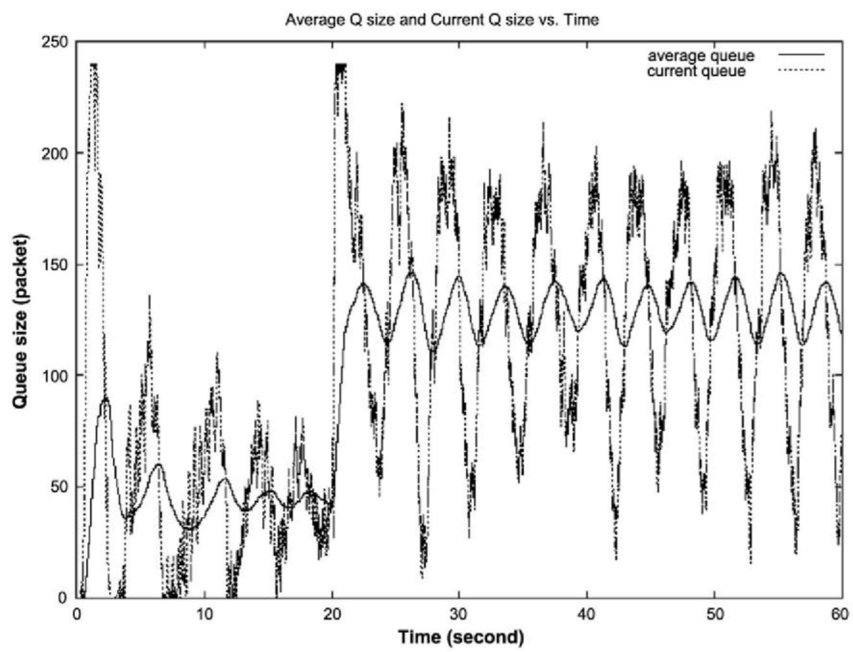


Fig. 7. Current and average queue size evolution (RED runs for the entire duration of 60 s).

### 4.1. Comparison of queue oscillation

To show the chaotic queue oscillation behavior in RED and ARED implementations, a scenario of sudden traffic increase in the network by increasing the number of flows is chosen. In this scenario 10 flows ran for entire 60 seconds and suddenly 170 new flows joined the traffic at the 20th second at the same time, made 180 flows ran for remaining 40 seconds. It can be clearly seen, in Figs. 7 and 8, respectively, that the RED and the ARED cannot tolerate such a sudden heavy load of traffic. Both the RED and the ARED scheme suffer from the chaotic problem. However the ARED tries to bring the average queue size down according to its goal of yielding low delay. With the same topology and traffic conditions, the RED or the ARED were used for the first 11 seconds and then ran AutoRED for the next 49 seconds to compare how effectively AutoRED reduces the queue oscillation resulting from the RED or the ARED. Therefore the time axis should be considered as three segments: (i) RED or ARED running alone from 0 to 11th seconds; (ii) AutoRED running from 11 to 20th seconds with same density of traffic and (iii) AutoRED running with the increase traffic density from 20 to 60th seconds. These segments are referred to as S1, S2 and S3, respectively. The results of this simulation are presented in Figs. 9 and 10, respectively.

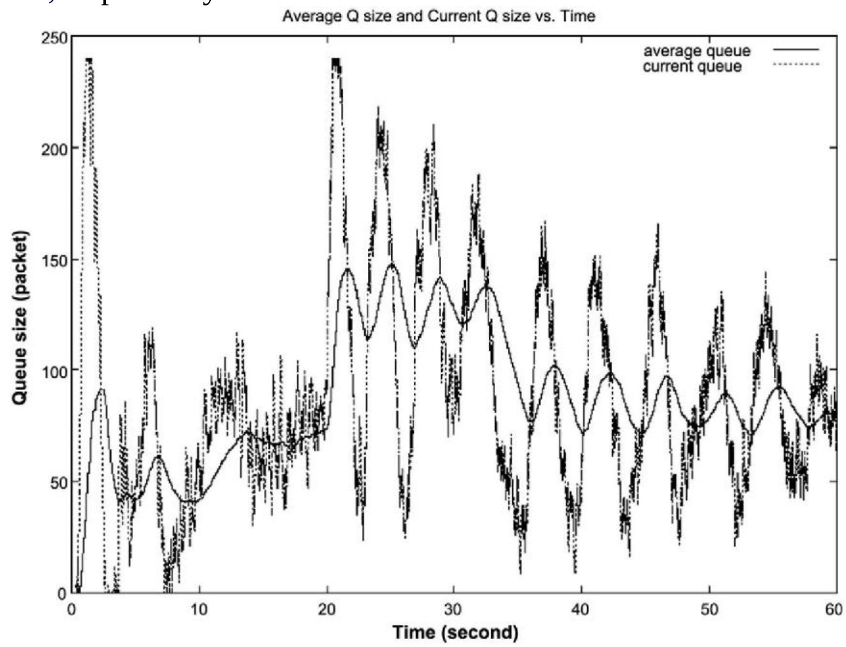


Fig. 8. Current and average queue size evolution (ARED runs for the entire duration of 60 s).

Comparing Figs. 7 and 9 and Figs. 8 and 10, it can be clearly seen that the AutoRED significantly reduces the instantaneous queue oscillation than the RED and the ARED, respectively. For example in the segment S2, where the moderate traffic condition is present, the AutoRED started to control the queue oscillation. Similarly in the segment S3, where the heavy traffic condition is present, the AuotRED have the full control over the queue oscillation. Also the rapid but small fluctuations in the average queue size show predictable low delay and low delay jitter under heavy load of traffic conditions.

Now comparing the results of AutoRED with the results of R-RED (Fig. 11) we can see similar behavior in the control of queue oscillation. The queue oscillation behavior of R-RED supports the results in Fig. 7(d) of paper [12]. Due to the discrete adjustment in $w_q$ and $p_{max}$ of R-RED algorithm the fluctuation is not fast enough to give high throughput and low packet loss rate (see the results in Table 2). We also noticed, in R-RED, that the length of the average queue size and the instantaneous queue length are closer to each other and it is also an indication of the higher packet loss rate and the low throughput conditions.

## *4.2. Dynamics of the AutoRED*

### 4.2.1. The weighting parameter $w_{q,t}$

The weighting parameters $w_q$ and $w_{q,t}$ are compared in Fig. 12. In this figure the $w_q$ values are recorder for the first 11 seconds and then the changes in $w_{q,t}$ are recorded for the remaining 49 seconds. It can be clearly seen that the RED's weighting parameter $w_q$ is constant in the first 11 seconds and then it is upgraded to $w_{q,t}$ to an adaptive status. It also indicates that the AutoRED makes rapid changes to the weighting parameter to reduce the oscillation behavior of the queue size.

### 4.2.2. Congestion characteristics

Fig. 13 shows the congestion characteristics presented in Eq. (7) and Fig. 1. This graph shows that the probabilistic expression $I_t = p_t(1 — p_t)$ represents the true congestion characteristics resulted in our simulation.

That is, the level of congestion increases with respect to the increase in the actual congestion density (note that more traffic joined the queue at time 20 s). Also it indicates that the dynamics of the congestion (congestion to no congestion and no congestion to congestion in two steps) is lower and rapid when the density of the traffic is high.

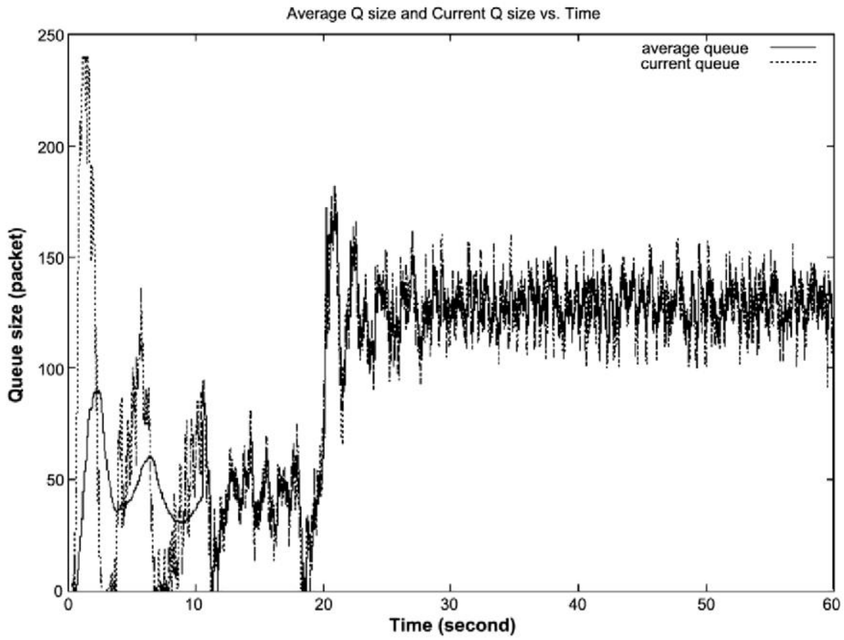


Fig. 9. Current and average queue size evolution (RED runs for the first 11 seconds and AutoRED runs for the next 49 seconds).

### 4.2.3. Traffic characteristics

Fig. 14 shows the traffic characteristics presented in Eq. (8) and Fig. 2. The similarity in the rapid changes in the segments S2 and S3 indicates that the effect of 1-unit packet increase on the *dynamics* of the traffic is consistent over the different density of the network traffic.

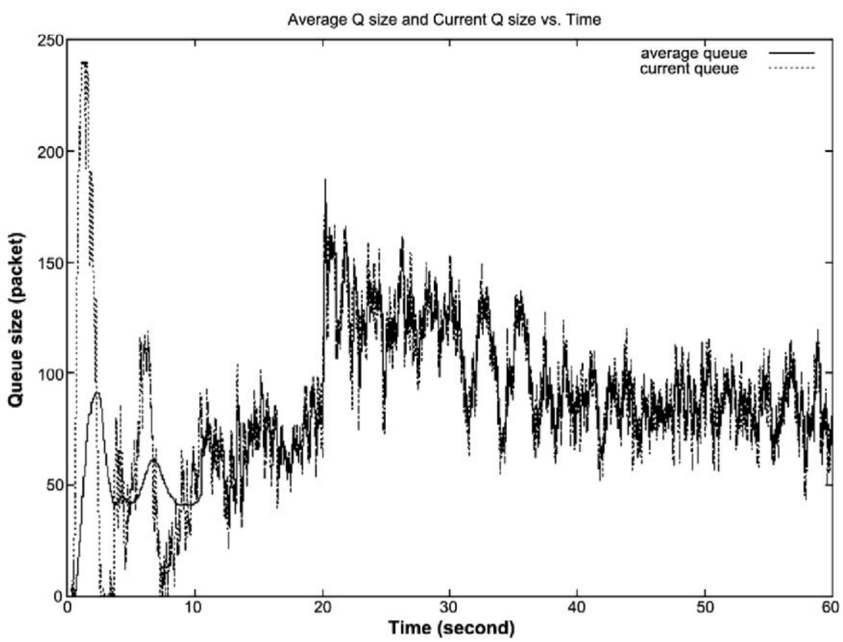


Fig. 10. Current and average queue size evolution (ARED runs for the first 11 seconds and AutoRED runs for the next 49 seconds).

### *4.3. Performance analysis*

Table 2 provides the results of number of packets dropped, number of packets sent, throughput, throughput gain, packet loss rate when RED, ARED, R-RED, Auto-RED with RED and AutoRED with ARED are implemented. It can be seen in this table that the proposed AutoRED with RED scheme performs better than the RED scheme and similarly the proposed AutoRED with ARED scheme performs better than the ARED scheme. It can also be noticed that the R-RED has low number of packets dropped as well as low number of packets sent, which results in a low throughput, throughput gain and high packet loss rate for the R-RED scheme. The overall comparison of the results in the table indicates that the AutoRED with RED algorithm provides higher throughput gain (92.14%) and reduced packet loss rate (7.86%).

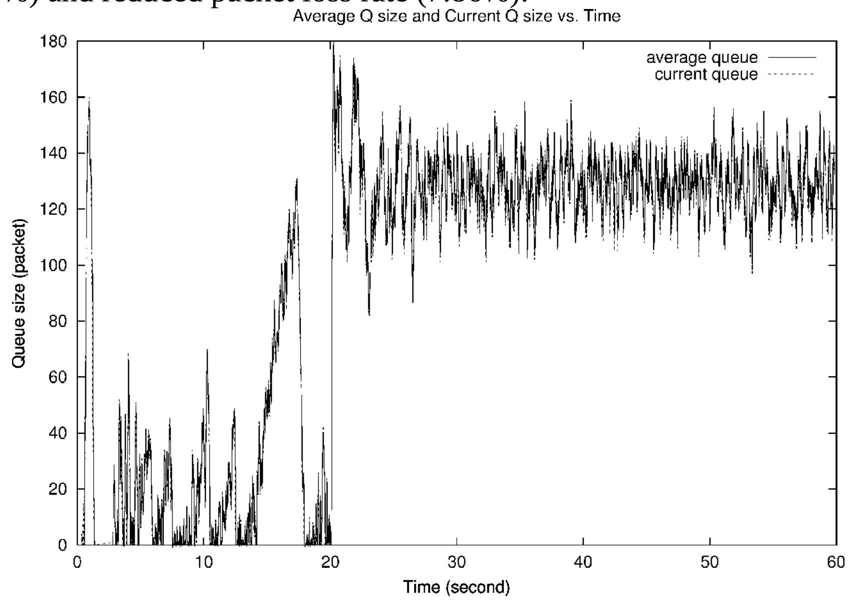


Fig. 11. Current and average queue size evolution of R-RED scheme.

### 4.4. Discussion

This section answers the question of why the equation in (5) reduces the queue oscillation appropriately in the RED-based algorithms. In the original RED scheme the weight $w_q$ is chosen as a small constant to ignore the short-lived bursty traffic when the average queue size is calculated to make the packet drop decision. The choice of small value for the weight leads to slowly varying effect on the average queue size resulting in slow response to transient congestion as well as to high continuous bursty traffic. This slow response characteristic is causing instantaneous queue oscillation. That is the fluctuation between the buffer over-flow and buffer under-utilization.

Table 2
Performance analysis

| | RED | ARED | R-RED | AutoRED with RED | AutoRED with ARED |
|---|---|---|---|---|---|
| Packet dropped | 4297 | 4927 | 4379 | 4069 | 4516 |
| Packet sent | 47740 | 48025 | 46257 | 47714 | 48025 |
| Throughput (kbps) | 795 | 800 | 770 | 795 | 800 |
| Throughput gain (%) | 91.74 | 90.70 | 91.35 | 92.14 | 91.40 |
| Packet loss rate (%) | 8.26 | 9.30 | 8.65 | 7.86 | 8.60 |

The R-RED adjusts the weight $w_q$ and $p_{max}$ using six thresholds as a function of minimum and maximum thresholds used in the RED scheme. The main idea is to incorporate some strength of the burstiness or transient congestion into the calculation of $w_q$ in addition to incorporating the presence of bursty traffic as carried out in the RED scheme. This technique reduces the effect of slowly varying nature of the average queue size and as a result reduces the queue oscillation. In contrast the AutoRED focuses on reducing this slowly varying effect by incorporating congestion characteristics, traffic characteristics and buffer normalization. It adjusts the weight $w_q$ continuously thus provides meaningful rapid changes to the weight $w_q$.

The AutoRED uses the strength and the effect of both the burstiness and the transient congestion. The strength quantifies the amount of the burstiness and the transient congestion. Similarly the effect quantifies the amount of threat of the burstiness and the transient congestion. The strength of the burstiness and the transient congestion is determined by the parameter $\mathcal{J}_t$ in Eq. (8) and the buffer normalization in Eq. (9). This strength is obtained using the meaningful traffic scenario (i.e. the strength of the 1- unit packet increase) explained in Eqs. (12)–(16). The effect of the burstiness and the transient congestion is determined by the parameter $I_t$ in Eq. (7). If the RED gateway is less likely to have transition between congestion state to no congestion state or vice versa (that is $I_t$ is closer to a 0) then the congestion history states that the current burstiness or the transient congestion is insignificant so we can use this small value to ignore the burstiness as used in the RED scheme. If it is high likely to have transition between congestion to no congestion state or vice versa (that is $I_t$ is closer to a 0.25) then the congestion history states that the current burstiness or transient congestion is significant so we incorporate the amount of burstiness or transient congestion in the calculation of the average queue size.

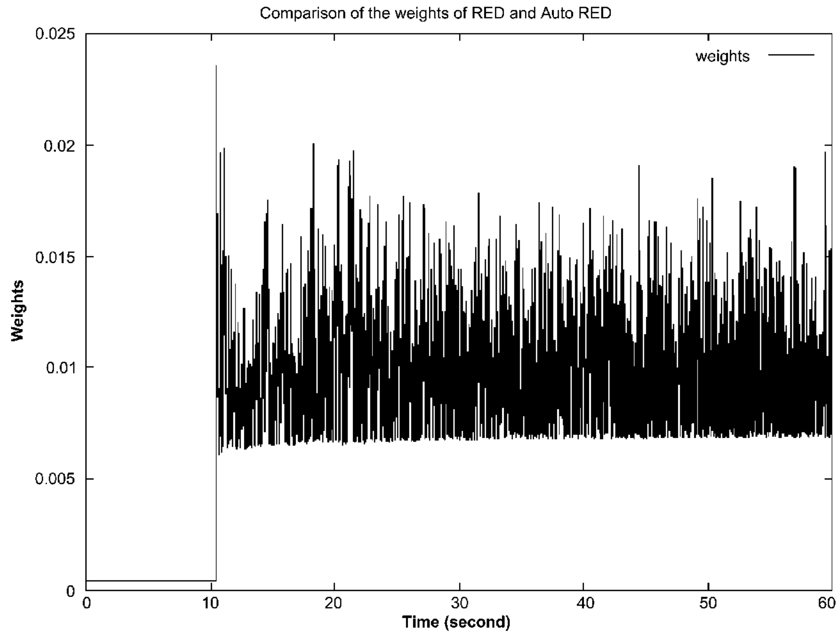


Fig. 12. Monitoring the weighting parameters: $w_q$ is recorded in the first 11 seconds (RED is running) and $w_{q,t}$ is recorded in the next 49 seconds (AutoRED is running).

The AutoRED, by incorporating the traffic characteristics, congestion characteristics and queue normalization makes rapid meaningful changes to the weight. This rapid and meaningful change in the weight $w_q$ can be clearly seen in Fig. 12. This characteristic eliminates the slowly varying nature of the average queue size and in turn allows the RED mechanism to make rapid and meaningful packet drop decisions.

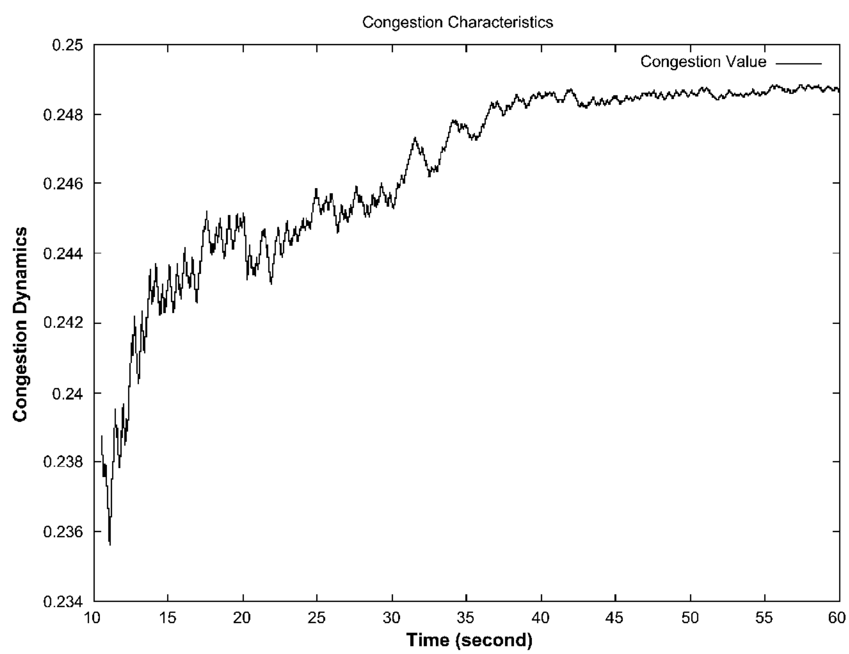


Fig. 13. Monitoring the congestion characteristics.

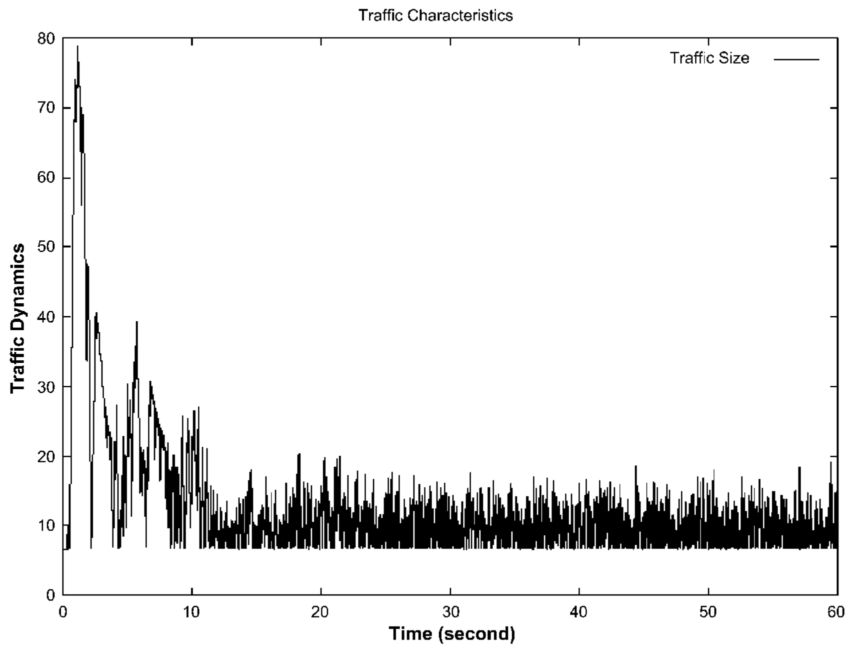


Fig. 14. Monitoring the traffic characteristics.

## 5. Conclusion

This paper presents a RED-based AQM technique that significantly reduces the chaotic queue oscillation problem in RED-based algorithms while providing predictable low delay and delay jitter with high throughput gain and reduced packet loss rate. The use of network characteristics (congestion characteristics, traffic characteristics and queue normalization) as defined in this paper to modify the weights in RED-based algorithm leads to improved Internet performance.

## References

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

[2] P. Ranjan, E.H. Abed, R.J. La, Nonlinear instabilities in TCP-RED, in: Proc. IEEE INFOCOM, 2002.

[3] IETF RFC 2309. Available from: <http://www.ietf.org/rfc/ rfc2309.txt/>.

[4] S. Suthaharan, Markov Model Based Congestion Control for TCP, in: Proc. of the IEEE Computer Society Press, the 37th Annual Simulation Symposium (ANSS-37 2004), 2004, 285–292, Arlington, Virginia (April 18–22).

[5] W. Feng, D. Kandlur, D. Saha, K Shin, A self-configuring RED gateway, in: Proc. IEEE INFOCOM, 1999, pp. 1320–1328.

[6] J. Aweya, M. Ouellette, D.Y. Montuno, A control theoretic approach to active queue management, Computer Networks 36 (2001) 203–235.

[7]W.C. Feng, K.G. Shin, D.D. Kandlur, D. Saha, The blue active queue management algorithms, IEEE/ACM Transactions on Net-working 10 (4) (2002) 513–528.

[8]T. Eguchi, H. Ohsaki and M. Murata, On control parameters tuning for active queue management mechanisms using multivariate analysis, in: Proc. of the 2003 symposium on applications and the Internet (SAINT03), Springer, pp. 8.

[9]S. Athuraliya, S.H. Low, V.H. Li, Q. Yin, REM: active queue management, IEEE Network (2001) 48–53.

[10] C. Wang, B. Li, Y.T. Hou, K. Sohraby, Y. Lin, LRED: a robust and responsive AQM algorithm using packet loss ratio measurement, IEEE Transaction on Parallel and Distributed Systems 18 (1) (2007) 29–43.

[11] S. Floyd, R. Gummadi, S. Shenker, Adaptive RED: an algorithm for increasing the robustness of RED's active queue management, Available from: <http://www.icir.org/floyed/>.

[12] H. Wang, K.G. Shin, Refined design of random early detection gateways, Proceedings of IEEE Global Telecommunications Conference (1999) 769–775.

[13] B. Zheng, M. Atiquzzaman, DSRED: an active queue management scheme for next generation networks, Proceedings of annual IEEE conference on local computer networks (2000) 242–251.

[14] G. Feng, A.K. Agarwal, A. Jayaraman, C.K. Siew, Modified RED gateways under bursty traffic, IEEE Communications Letters 8 (5) (2004) 323–325.
[15] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, Inc., 1965.
[16] The Network Simulator: ns-2 [online]. Available from: <http:// www.isi.edu/nsnam/ns/>.
[17] J. Koo, B. Song, K. Chung, H. Lee, H. Kahng, MRED: a new approach to random early detection, Proceedings of International Conference on Information Networking (2001) 347–352.
[18] C.H. Chien, W. Liao, A self-configuring RED gateway for quality of service (QoS) networks, Proceedings of IEEE International Conference of Multimedia and Expo (2003) 793–796.