Available online at www.sciencedirect.com

ScienceDirect

Journal of Network and
Computer Applications 31 (2008) 429–445

Journal of NETWORK and COMPUTER APPLICATIONS

www.elsevier.com/locate/jnca

A framework to determine bounds of maximum loss rate parameter of RED queue for next generation routers☆

Bing Zheng[a], Mohammed Atiquzzaman[b,*]

[a]Bookham Technology, San Jose, CA 94124, USA
[b]School of Computer Science, University of Oklahoma, Norman, OK 73019-6151, USA

Received 6 September 2007; received in revised form 17 January 2008; accepted 9 February 2008

Abstract

Random early detection (RED) is expected to eliminate global synchronization by random active packet drop. Its packet drop probability is decided by the maximum packet drop probability in its drop function, buffer thresholds, and average queue length. It has been observed that for a large number of connections, a small value of the maximum packet drop probability may not eliminate global synchronization. Furthermore, since RED uses four parameters to regulate its performance, it is necessary to relate its maximum drop probability with those parameters. The *objective* of this paper is to develop a framework for the bounds of the maximum drop probability of RED, based on TCP channel model and traffic characteristics. The value of the maximum drop probability obtained by our model will make RED queue achieve its targeted goals.

Keywords: Random early detection; Congestion control; Performance evaluation; Routers

1. Introduction

The bandwidth of a TCP connection depends on the round trip time (RTT) and the packet drop probability of the connection. Random early detection (RED) gateways

---

☆This work was carried out while the first author was with University of Dayton.

*Corresponding author. Tel.: +1 405 325 8077; fax: +1 405 325 4044.
E-mail addresses: zhengbin@ieee.org (B. Zheng), atiq@ou.edu (M. Atiquzzaman).

1084-8045/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2008.02.003

provide bandwidth control since they use packet drop to imply congestion in the network. A RED gateway regulates its performance by four parameters and one control variable. Therefore, the packet drop probability depends not only on the congestion scenario, but also on the value of the RED parameters. As one of the key parameters in RED, the maximum packet drop probability, *Maxp*, is therefore related to the traffic pattern and other RED parameters. However, the value of *Maxp* suggested in Floyd and Jacobson (1993) and Floyd (1997) is independent of the traffic pattern and the values of the RED parameters. As a result, RED may not work optimally when a RED gateway supports a large number of connections. The *objective* of this paper *is to develop a model for the bounds of Maxp*, *based on TCP traffic characteristics* and RED parameters.

The effects of *Maxp* on queue management are two folds. The first is the value of *Maxp* itself, and the second is the relationship between the value of *Maxp* and the other parameters (Min_{th}, Max_{th}, and w) (Floyd and Jacobson, 1993) (see Fig. 2). In the first case, it has been shown that if *Maxp* is too small, RED is insufficient to notify senders, and tail drop will dominate the packet drop at the RED gateway (Feng et al., 1999a); too large a value of *Maxp* will lower link bandwidth. The second case is more complicated than first case, since RED queue regulates its performance by its four parameters.

The second problem mentioned above can be illustrated by the fact that RED uses average queue length as a control variable to calculate packet drop probability as given by

$$p_{RED} = Maxp\frac{avg - \mathrm{Min}_{th}}{\mathrm{Max}_{th} - \mathrm{Min}_{th}}.$$

Therefore, the actual packet drop probability p_{RED} is decided not only by the value of *Maxp* but also depends on the value of $\mathrm{Max}_{th} - \mathrm{Min}_{th}$. In other words, the actual packet drop probability is decided by the ratio of *Maxp* and $\mathrm{Max}_{th} - \mathrm{Min}_{th}$. This means that *Maxp* needs to be related to the link feature and the thresholds of a RED queue.

The contribution of this paper is the development of a modelling framework for the bounds of *Maxp* which are related to the TCP connection parameters and RED configuration parameters. The use of *Maxp* suggested by our model results in all packet drops being due to active drops (rather than the undesirable passive (tail) drops) while maintaining a high link utilization and avoiding any global synchronization. The value of *Maxp* suggested previously in the literature may eliminate passive drops, but *results in low utilization of the link bandwidth for TCP traffic*. Our model can be used by network engineers to determine an optimum value of *Maxp* based on traffic characteristics and values of RED parameters.

In Section 2 we describe the principles of RED, and previous work carried out in determining the parameters of RED (Zheng and Atiquzzaman, 2004). The assumptions we have made in developing our proposed model are outlined in Section 3, followed by detailed development of our modelling framework in Section 4. Simulation results are presented in Section 5, followed by conclusions in Section 6.

2. Random early detection

IETF recommends active queue management to reduce router queue length, reduce packet drops, and avoid lock out phenomena in Internet routers. RED (Braden et al., 1998) is the default active queue management scheme recommended by IETF for next generation Internet routers. As a result, various aspects of RED have been widely studied

```
For Each Packet Arrival
    Calculate the average queue size avg
    If Min_Threshold $\leqslant$ avg $<$ Max_Threshold
        Calculate probability p
            with probability p:
                Mark the arriving packet
    else if Max_Threshold $\leqslant$ avg
                Mark the arriving packet
```

Fig. 1. Algorithm of RED.

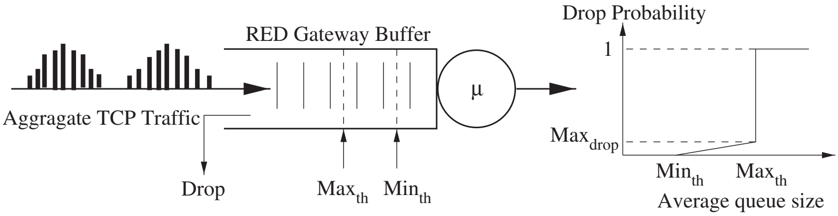

Fig. 2. Illustration of RED gateway queue.

in the literature (Lu and Wang, 2007; Wang et al., 2007; Guan et al., 2007; Kim and Lee, 2006). RED was proposed by Floyd and Jacobson (1993). Since then, a number of studies have been carried out to study its performance in terms of stability (La, 2004; Ranjan et al., 2004; Hollot et al., 2002), input traffic patterns (Feng et al., 2004), transport protocol (Wang et al., 2003), and application layer protocol (Le et al., 2003). Figs. 1 and 2 show the algorithm and drop function of RED. A router implementing RED accepts all packets until the queue reaches Min_{th} after which it drops packets with a linear probability distribution function. When the queue length reaches Max_{th}, all packets are dropped with a probability of one.

The basic idea behind RED is that a router detects congestion early by computing the average queue length (avg) and sets two buffer thresholds Max_{th} and Min_{th} for packet drops as shown in Fig. 2. The average queue length at time t, defined as $avg(t) = (1 - w)avg(t - 1) + wq(t)$, is used as a control variable to perform active packet drops. $avg(t)$ is the new value of the average queue length at time t, $q(t)$ is instantaneous queue length at time t, and w is a weight parameter for calculating $avg(t)$. Normally, w is much less than one. The packet drop probability p is calculated by

$$p = \mathrm{Max}_{drop} \frac{avg - \mathrm{Min}_{th}}{\mathrm{Max}_{th} - \mathrm{Min}_{th}}.$$

The RED algorithm therefore, includes two computational parts: computation of the average queue length and calculation of the drop probability.

The RED algorithm involves *four* parameters to regulate its performance. Despite lot of studies, the relationship between the four parameters and the behavior of RED is still not fully understood (May et al., 2000a; Verma et al., 2003; Low et al., 2003; Ranjan et al., 2004). Min_{th} and Max_{th} are the queue thresholds to perform packet drops, Max_{drop} is the packet drop probability at Max_{th}, and w is a weight parameter to calculate the average queue size from the instantaneous queue length. The average queue length follows a long-term increase of the instantaneous queue length. However, since w is much less than one, *avg* changes much slower than q. Therefore, *avg* follows the long-term changes of q, reflecting persistent congestion in the network when *avg* is high. By making the packet drop probability a function of the level of congestion, RED gateway can achieve a low packet drop probability during low congestion, while the drop probability increases as the congestion level increases.

The packet drop probability of RED is small in the interval Min_{th} and Max_{th}, and the packets to be dropped are chosen randomly from packets arriving from different hosts. As a result, packets coming from different hosts will not be dropped simultaneously. RED gateways, therefore, avoid global synchronization by randomly dropping packets.

2.1. *Selection of maximum packet drop probability,* Max_{drop}

It has been shown that there is no single set of RED parameters that work well under different congestion scenarios (Feng et al., 1999a). Based on the above observation, the authors proposed a more adaptive RED gateway, which self-parameterizes itself based on the traffic mix. Results show that traffic dependent parameterizations of a RED gateway can effectively reduce packet loss, while improving link utilization under a range of network loads. However, dynamically setting and configuring RED gateway parameters makes gateway management complicated.

The selection of the maximum drop probability (Max_{drop}) affects the performance of RED. If Max_{drop} is too small, then active packet drops will not be enough to prevent global synchronization. Too large a value of Max_{drop} will lower the throughput. Although Floyd (1997) suggests a Max_{drop} value of 0.1, selection of an optimal value of Max_{drop} according to the network and traffic situation is still an open issue (Feng et al., 1999b; May et al., 2000b).

In Feng et al. (1999a), it has been proved that the value of Max_{drop} depends not only on the bandwidth delay product, but also on the number of connections. It has been pointed out that the upper bound of Max_{drop} can be expressed as

$$\mathrm{Max}_{drop} \leqslant \frac{N * SS * C}{B\tau}, \tag{1}$$

where N is the number of connections, B is the total bandwidth, SS is the segment size, τ is the RTT, and C is a constant. From Eq. (1), it is seen that it is not possible to fix the value of Max_{drop} for a dynamically changing network environment, such as, varying number of connections, RTT, etc.

2.2. *Selection of buffer thresholds,* Min_{th} *and* Max_{th}

The buffer thresholds of RED can be determined as follows:

- It has been recommended that, for a RED gateway carrying only TCP traffic, Min_{th} should be around five packets, and Max_{th} should be at least three times of Min_{th} (Floyd, 1997).

- Non-TCP traffic do not employ the congestion control mechanisms of TCP. A different set of values are therefore, required for Min_{th} and Max_{th} in order to protect TCP traffic from non-TCP traffic (Anjum and Tassiulas, 1999).

2.3. Selection of weight parameter, w

RED uses the average queue length as a control variable to perform active packet drop. Calculation of the average queue length involves the previous average queue length and the instantaneous queue length modified by a weight parameter w. The average queue length therefore works as a low pass filter (LPF).

The average queue length is required to track persistent network congestion that occurs over a long time range while, at the same time, filtering out short time congestion. This requirement imposes limitations on the selection of w. If w is too small, the average queue length will not catch up with the long range congestion which may result in the failure of active queue management. If w is too large, the average queue length will track the instantaneous queue, which will also degrade the performance of active queue management. Therefore, the value of w should be related to the traffic flowing in the queue.

A simple model to calculate w was developed in Floyd and Jacobson (1993) and Floyd (1997). However, the assumptions in developing the model of w were too simple to reflect real TCP traffic. Therefore, in certain situations, the values given in Floyd and Jacobson (1993) and Floyd (1997) may result in non-optimal performance of the RED queue (Zheng and Atiquzzaman, 2000).

A more realistic model for determining w has been proposed in Zheng and Atiquzzaman (2000), where the aggregate TCP traffic has been taken into consideration. Results have shown that the value obtained from the model in Zheng and Atiquzzaman (2000) gives better performance than the values in Floyd and Jacobson (1993) and Floyd (1997) under certain cases.

2.4. Calculation of average queue length

RED uses four parameters and one state variable to regulate its performance. The state variable is the average queue length, which is defined as $avg = (1 - w)avg + wq$ and works as an LPF (Floyd and Jacobson, 1993). In the above expression w is a weight parameter and q is the instantaneous queue size of gateway buffer. The average queue length controls the active packet drop in the RED queue. The advantages of using average queue length to control active packet drop are (1) accumulating short-term congestion and (2) tracing long-term congestion. However, the LPF characteristic of average queue will be also featured with *slow time response* to the changes of long-term congestion in network. This will be harmful to the throughput and delay performance of RED gateway. For example, after a long-term congestion, the average queue length will stay high even if the instantaneous queue is back to normal or low; RED will therefore continue dropping packets even after the end of congestion (May et al., 2000b) resulting in low throughput. The slow response of the average queue length will result in the throughput restoring slowly after heavy congestion (Christiansen et al., 2000). A larger values of w can improve the response time, but will result in the RED queue tracing short-term congestion; this violate the rules for AQM required by IETF (Braden et al., 1998).

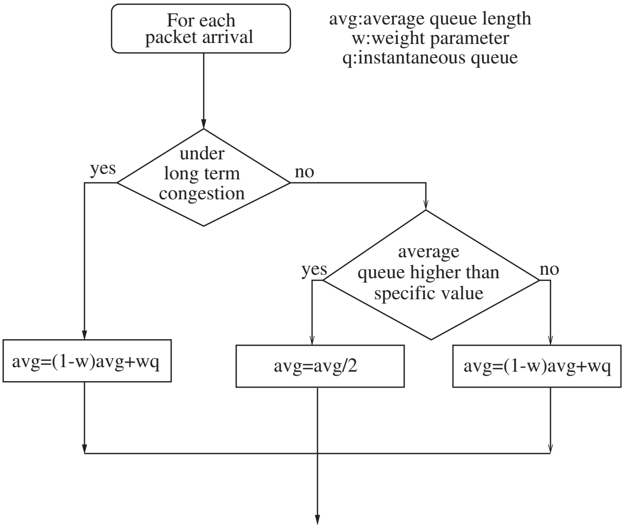

Fig. 3. LPF/ODA algorithm (Zheng and Atiquzzaman, 2002).

In Zheng and Atiquzzaman (2002), a more effective definition and algorithm for calculating *avg* is proposed. The new algorithm is called *low pass filter/over drop avoidance* (*LPF/ODA*) which calculates the average queue length as follows:

- During long-term congestion, calculate the average queue length with an LPF as given in Floyd and Jacobson (1993). During this period, the RED queue is in the *active drop phase*.
- If average queue length is high at the end of long-term congestion, halve the average queue length. During this period, the RED queue is in the *ODA phase*.
- If the average queue length is below a specific threshold after the end of long-term congestion, renew the value of average queue length using the LPF model.

The LPF/ODA algorithm is shown in Fig. 3. Results have shown that the LPF/ODA algorithm improves the response time, throughput and delay of RED queues.

3. Modelling assumptions and notations

To facilitate further discussion, we present the assumptions and notation used in our model.

3.1. Assumptions

We make the following assumptions, which will be used to develop the model for determining *Maxp* in Section 4. *Note that some of the assumptions have been used in previous work* as mentioned below:

- The RED gateway queue is initially empty (also assumed in Floyd and Jacobson, 1993).
- The average queue length is initially zero (also assumed in Floyd and Jacobson, 1993).
- When RED performs well, the average queue length will vary within a small range.

- In the long term, the active packet drop always works.
- RTT, τ, for a connection is constant (used in Padhye et al., 2000).
- A TCP source's congestion window at time t is determined by the packet drops at time $t-\tau$ (as in Padhye et al., 2000).
- Long-term congestion, which varies slowly, is described by a slow function $g(i)$, where i corresponds to the ith calculation of the average queue length.
- Short-term congestion, which varies fast, is described by a fast function $f(i)$ corresponding to the ith calculation of the average queue length.

$$f(i) = q_0\left(\frac{1-(-1)^i}{2} + b\frac{1-(-1)^{i-1}}{2}\right). \tag{2}$$

- The instantaneous queue length, $q(i)$, is described by the modulation of the fast function ($f(i)$) by a slow function ($g(i)$), i.e., $q(i) = g(i)f(i)$.

3.2. Notations

We define the following variables, which are used in our model in Section 4.

- w: weight parameter for calculation of average queue length at a RED gateway;
- $q(m)$: instantaneous queue size of the RED gateway during the mth calculation of the average queue length. From our assumptions in Section 3.1, $q(0) = 0$;
- μ: bottleneck link rate;
- $avg(m)$: average queue length of the RED gateway at the mth calculation (see Fig. 4). It is defined as

$$avg(m) = (1-w) * avg(m-1) + w * q(m), \tag{3}$$

  where $avg(0) = 0$ from our assumptions in Section 3.1;
- τ: RTT in terms of calculation interval of average queue length;
- $Maxp$: maximum packet drop probability for RED queue;
- K_l: minimum buffer threshold for RED gateway to perform active packet drop;
- K_h: maximum buffer threshold for RED gateway to perform packet drop with probability of one;
- $W_i(m)$: TCP congestion window size for the ith connection at time m;
- N: total number of connections;
- $a(m)$: normalized instantaneous queue defined by $q(m)/K_l$;
- $b(m)$: normalized average queue length defined by $avg(m)/K_l$.

In the next section, we develop the model of $Maxp$ based on the above assumptions.

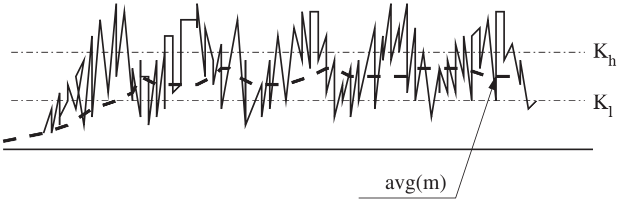

Fig. 4. Long-term performance of average queue length versus instantaneous queue.

4. Modelling *Maxp*: lower and upper bounds

According to the definition of packet drop probability in RED queue, the packet drop probability $p(m)$ is given by

$$p(m) = \frac{avg(m) - K_l}{K_h - K_l} Maxp. \tag{4}$$

By defining $\alpha = Maxp/(K_h - K_l)$, Eq. (4) can be rewritten as

$$p(m) = \alpha K_l(b(m) - 1). \tag{5}$$

The difference equation for the instantaneous queue $q(m)$ is

$$q(m) - q(m-1) = \sum_{i=1}^{N} \frac{W_i(m)}{\tau_i} - \mu, \tag{6}$$

where the left-hand side is the net change of instantaneous queue size, the right-hand side is the difference between the incoming and outgoing data. To simplify the discussion, we consider N *iid* TCP connections with the same RTT. In this case, Eq. (6) becomes

$$q(m) - q(m-1) = \frac{NW(m)}{\tau} - \mu. \tag{7}$$

Expressing Eq. (7) in normalized form, we have

$$a(m) - a(m-1) = \frac{NW(m)}{\tau K_l} - \frac{\mu}{K_l}. \tag{8}$$

It has been proved in Padhye et al. (2000) that $W(m)$ can be expressed as

$$W(m) = \frac{C}{\sqrt{p(m-\tau)}}, \tag{9}$$

where C is a constant. By substituting Eq. (5) into (9), we have

$$W(m) = \frac{C}{\sqrt{\alpha K_l}\sqrt{b(m-\tau)}\sqrt{1 - \frac{1}{b(m-\tau)}}}, \tag{10}$$

where $b(m)$ is the normalized average queue length. For active queue management to work, it must have $b(m) \geqslant 1$. Therefore, $1/b(m-\tau) \leqslant 1$ always holds. By using $1/\sqrt{1-x} \simeq (1 + x/2)$, we have

$$W(m) \simeq \frac{C}{\sqrt{\alpha K_l}\sqrt{b(m-\tau)}} \left(1 + \frac{1}{2b(m-\tau)}\right). \tag{11}$$

From the definition, the normalized average queue is expressed as

$$b(m) = (1 - w)b(m-1) + wa(m). \tag{12}$$

We have

$$a(m) = \frac{b(m) - (1-w)b(m-1)}{w}. \tag{13}$$

By substituting $W(m)$, $a(m)$ and $a(m-1)$ into Eq. (8) (Zheng and Atiquzzaman, 20003):

$$\frac{b(m)-(1+(1-w))b(m-1)+(1-w)b(m-2)}{w}+\frac{\mu}{K_l} = \frac{N}{\tau K_l}\frac{C}{\sqrt{\alpha K_l}\sqrt{b(m-\tau)}}\left(1+\frac{1}{2b(m-\tau)}\right). \tag{14}$$

Finally, from Eq. (4), we have

$$Maxp = \frac{(CN)^2}{(\mu\tau)^2}\frac{(K_h-K_l)\left(1+\dfrac{K_l}{2\,avg(m)}\right)^2}{avg(m-\tau)} \times \frac{1}{\left(1+\dfrac{avg(m)-(1+(1-w))avg(m-1)+(1-w)avg(m-2)}{w\mu}\right)^2}. \tag{15}$$

From the definition of *avg* and our assumptions, we have (Zheng and Atiquzzaman, 20003):

$$avg(m)-(1+(1-w))avg(m-1)+(1-w)\times avg(m-2) = (1-w)^{m-1}wg(\overline{m})(f(m)-f(m-1)). \tag{16}$$

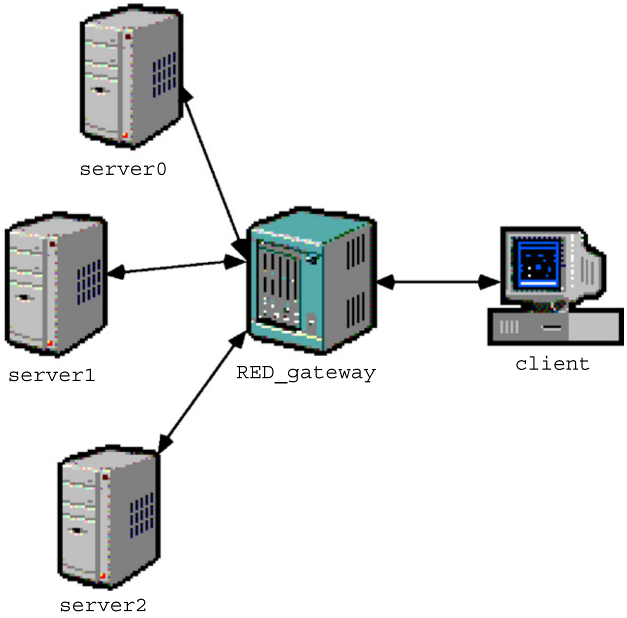

Fig. 5. Network configuration for simulation.

From our assumption, we have

$$f(m)-f(m-1)=q_0\left(\frac{-(-1)^m+(-1)^{m-1}}{2}+b\frac{-(-1)^{m-1}+(-1)^{m-2}}{2}\right)$$
$$=\begin{cases} q_0(b-1) & \text{if } m \text{ is even,}\\ -q_0(b-1) & \text{if } m \text{ is odd.}\end{cases} \tag{17}$$

Therefore, we have

$$\left(1+\frac{avg(m)-(1+(1-w))avg(m-1)+(1-w)avg(m-2)}{w\mu}\right)$$
$$=\pm\frac{(1-w)^{m-1}g(\overline{m})q_0(b-1)}{\mu}. \tag{18}$$

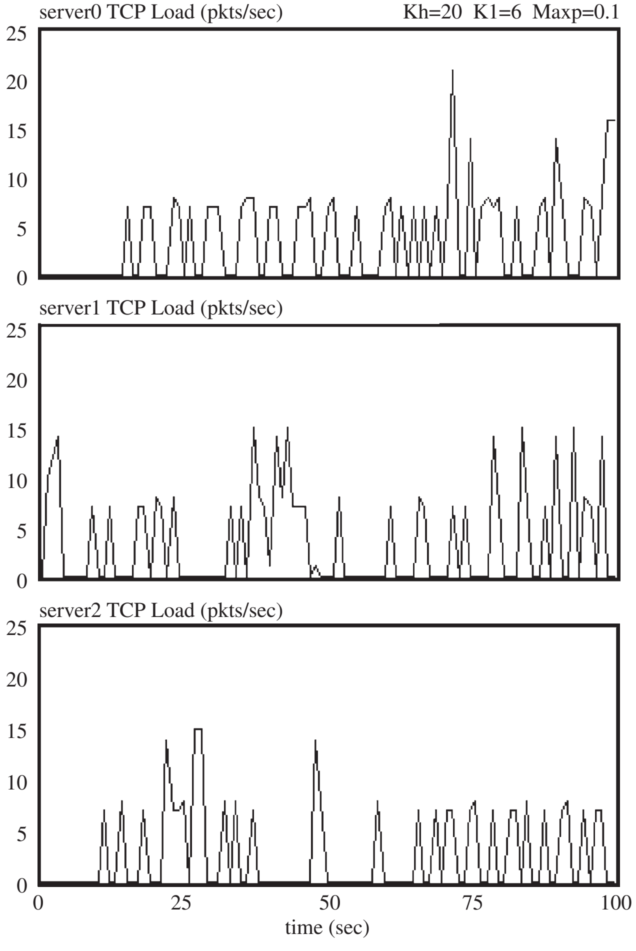

Fig. 6. TCP load for $Maxp^0 = 0.1$ and $K_h^0 - K_l^0 = 14$.

Since w is very small for a RED queue, for long-term performance, m is very large, resulting in a very small value of $(1-w)^{m-1}$. Moreover, comparing with the link rate μ, $(1-w)^{m-1}g(\overline{m})q_0(b-1)$ is so small that the term $(1-w)^{m-1}g(\overline{m})q_0(b-1)/\mu$ can be approximated by 0. Then the expression for *Maxp* can be expressed as

$$Maxp = \frac{(CN)^2}{(\mu\tau)^2}\frac{(K_h-K_l)\left(1+\dfrac{K_l}{2\,avg(m)}\right)^2}{avg(m-\tau)} \tag{19}$$

$avg(m-\tau)$ is called the *target average queue length* that RED tries to achieve in the long term. For RED to perform satisfactorily, the relationship $avg(m)\in[K_l,K_h]$ must be satisfied. From the drop function of the RED gateway queue, if the long term $avg(m)<K_l$, there will be no active packet drop. In this case, active queue management will not work; packet drops will be due queue overflow. On other hand, if $avg(m)>K_h$, each arriving packet will be dropped, giving rise to the problem of *global synchronization*.

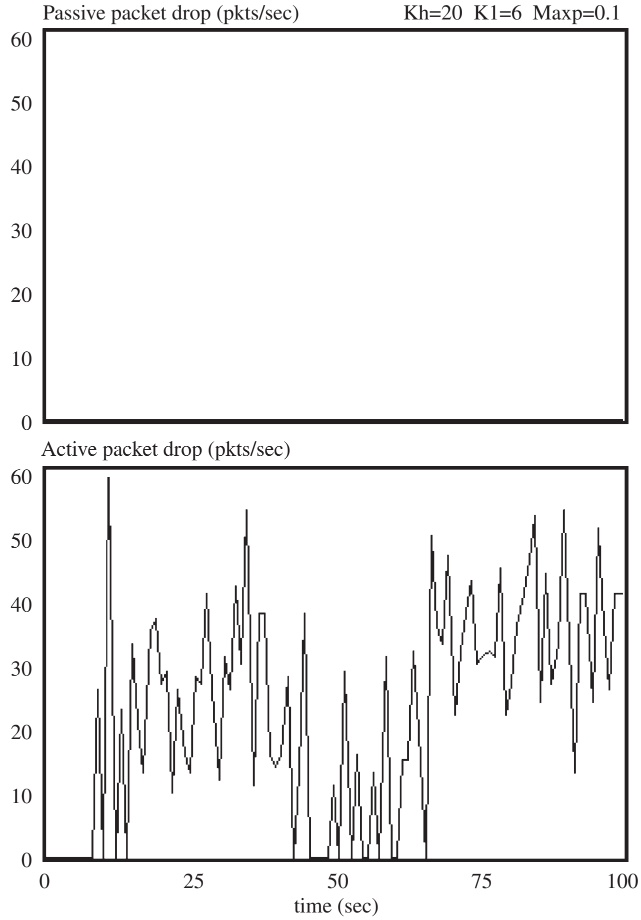

Fig. 7. Packet drop for $Maxp^0 = 0.1$ and $K_h^0 - K_l^0 = 14$.

From this discussion, we conclude that *avg*(*m*) will control *Maxp* between its *Upper* and *Lower* bounds, $Maxp^U$ and $Maxp^L$, respectively, as given below:

$$Maxp^U \leqslant \frac{(NC)^2}{(\mu\tau)^2}\frac{(K_h - K_l)\left(1+\dfrac{K_l}{2K_l}\right)^2}{K_l}, \tag{20}$$

$$Maxp^L \geqslant \frac{(NC)^2}{(\mu\tau)^2}\frac{(K_h - K_l)\left(1+\dfrac{K_l}{2K_h}\right)^2}{K_h}. \tag{21}$$

5. Performance evaluation

To test the model developed in the above section, we carried out simulations using the OPNET 5.1D network simulation tool. Before describing our results, we describe the network topology and simulation configuration.

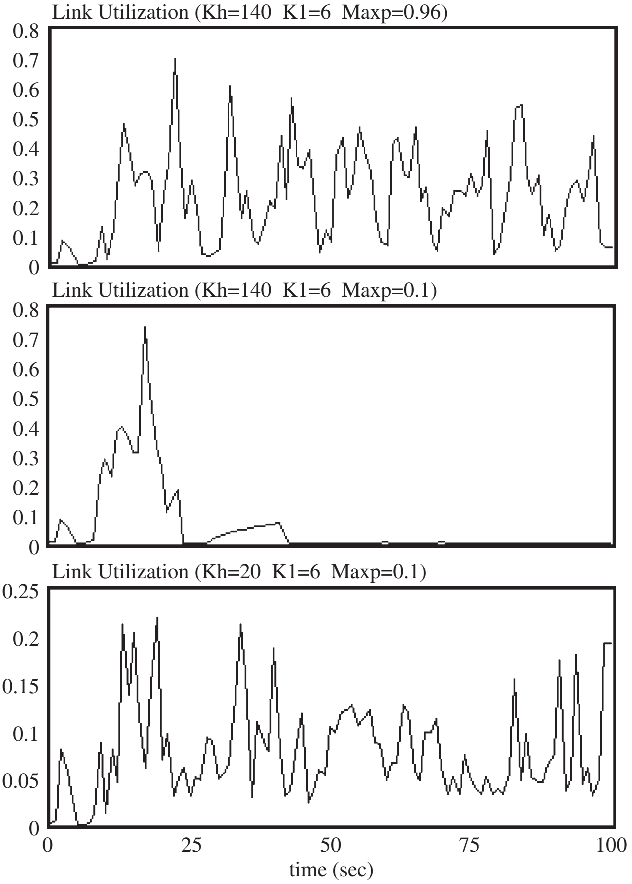

Fig. 8. Link utilization for $Maxp^1 = 0.96$ and $K_h^1 - K_l^1 = 134$, $Maxp^0 = 0.1$ and $K_h^1 - K_l^1 = 134$, $Maxp^0 = 0.1$ and $K_h^0 - K_l^0 = 14$.

5.1. Simulation configurations

The network topology is shown in Fig. 5. Three TCP sources send ftp traffic to a client via a RED gateway. To ensure a fair comparison with the value of *Maxp* in the original RED, the values of the configuration parameters were the same as those suggested in Floyd (1997).

- *Server*0 *to RED gateway link*: Propagation delay 1 ms, link rate 100 Mbps.
- *Server*1 *to RED gateway link*: Propagation delay 5 ms, link rate 100 Mbps.
- *Server*2 *to RED gateway link*: Propagation delay 3 ms, link rate 100 Mbps.
- *Client to RED gateway link*: Propagation delay 5 ms, bottleneck link rate 10 Mbps. To induce congestion at the RED queue, the bottleneck link rate has been chosen to be 30 times smaller than the sum of link rates feeding the bottleneck link.
- *Gateway processing speed*: 1 ms per packet.
- *Gateway queue size*: 200 packets.
- $w = 0.07$, $K_l^0 = 6$, $K_h^0 = 20$ and $K_l^1 = 6$, $K_h^1 = 140$.
- $Maxp^0 = 0.1$ (Floyd, 1997) and $Maxp^1 = Maxp^0(K_h^1 - K_l^1)/(K_h^0 - K_l^0)$.

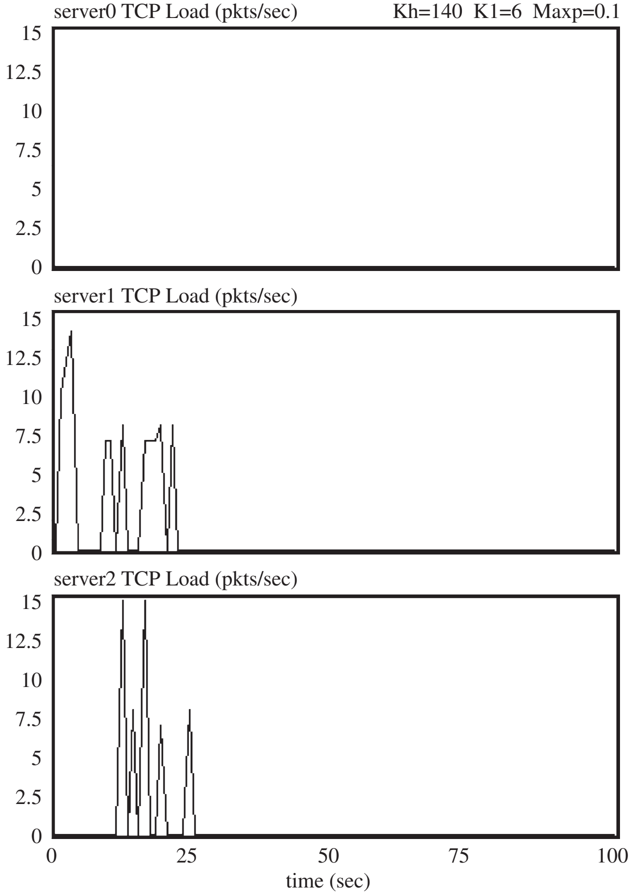

Fig. 9. TCP load for $Maxp^0 = 0.1$ and $K_h^1 - K_l^1 = 134$.

5.2. Results and discussion

Figs. 6 and 7 show the TCP traffic load at the three servers, and active and passive packet drops (tail drops) for $Maxp^0 = 0.1$ and $K_h^0 - K_l^0 = 14$. It can be seen that all the packets are due to active drops with zero passive drops, i.e., the RED queue works in active drop. *The TCP senders did not suffer from global synchronization. The bottleneck link has a reasonable utilization as shown in Fig. 8.*

Figs. 9 and 10 show the simulation results for $Maxp^0 = 0.1$ and changing $K_h^0 - K_l^0 = 134$ and 14, respectively. In the above analysis, the ratio $Maxp^0/(K_h^1 - K_l^1)$ is so small (less than 0.0008) that the active packet drop is insufficient to eliminate passive (tail) drops. Therefore, all packet drops are due to tail drops, resulting in global synchronization of TCP senders. TCP senders stop sending after time 25 s. Therefore, *the bottleneck link bandwidth is wasted* after time 25 s, as shown in Fig. 8.

Figs. 11 and 12 show the simulation results for $Maxp^1 = Maxp^0(K_h^1 - K_l^1)/(K_h^0 - K_l^0)$. As indicated by our theoretical model, the ratio $Maxp^1/(K_h^1 - K_l^1)$ is restored to the proper

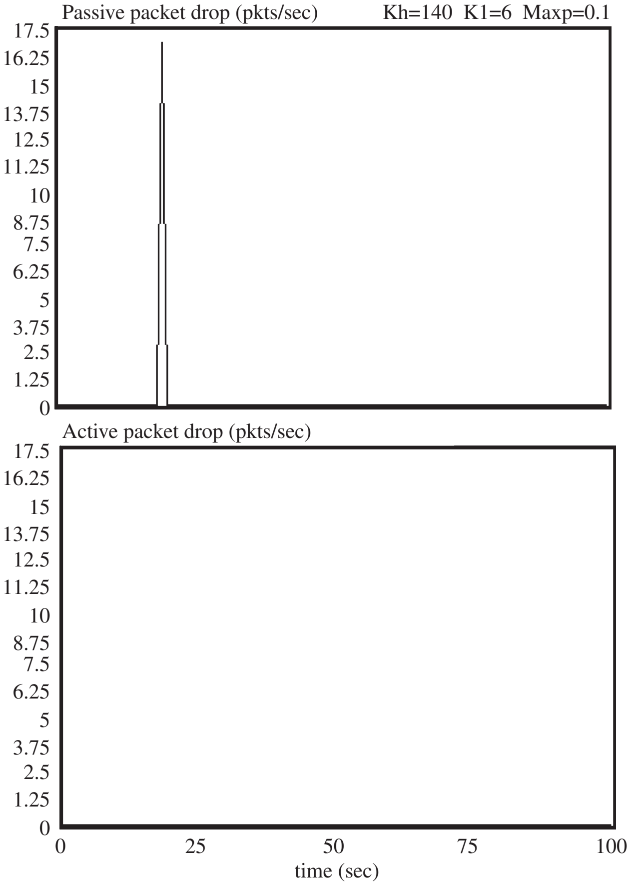

Fig. 10. Packet drop for $Maxp^0 = 0.1$ and $K_h^1 - K_l^1 = 134$.

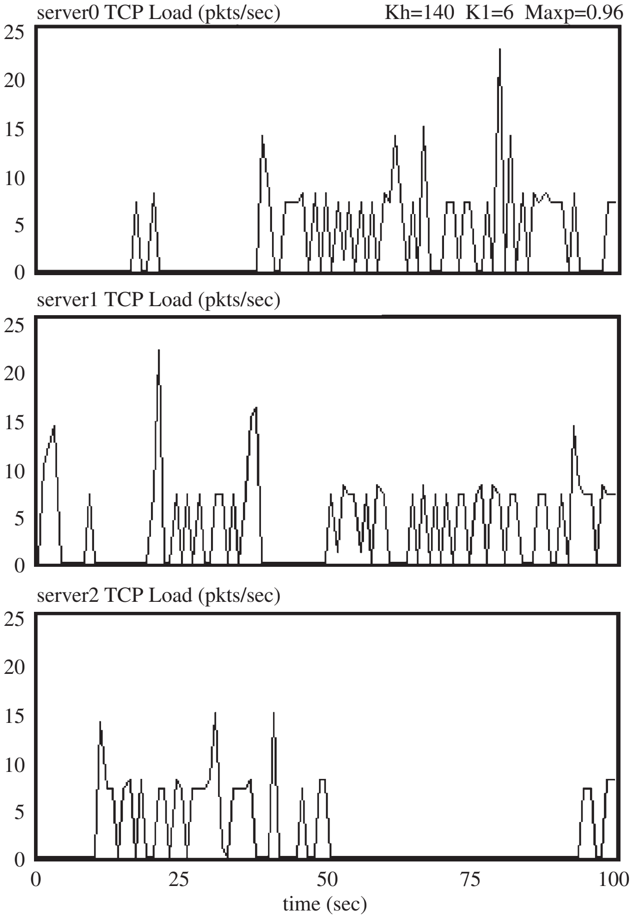

Fig. 11. TCP load for $Maxp^1 = 0.96$ and $K_h^1 - K_l^1 = 134$.

value so that the *active packet drops are sufficient to eliminate passive drops. Therefore, all packet drops are active drops, resulting in elimination of global synchronization. The congestion is relieved, resulting in a higher bottleneck link utilization as shown in Fig.* 8.

6. Conclusions

In this paper, a framework to determine *Maxp* of RED gateways have been proposed and developed. Simulation results have shown that the model properly relates *Maxp* to the buffer threshold and RED parameters. For a given TCP link, a large buffer threshold results in a large *Maxp*. Since, a fixed value of *Maxp* is not suitable for all configurations, inappropriate combinations of *Maxp* with other configuration parameters will prevent RED from achieving its desired goals. Our model can be used by network engineers to determine an optimum value of *Maxp* based on traffic characteristics and values of RED parameters.

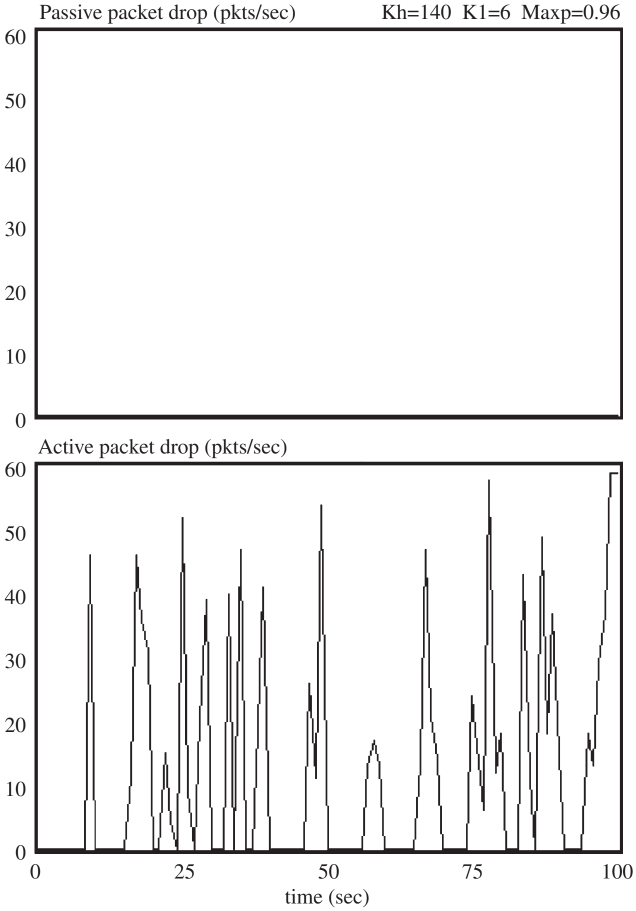

Fig. 12. Packet drop for $Maxp^1 = 0.96$ and $K_h^1 - K_l^1 = 134$.

References

Anjum FM, Tassiulas L. Balanced RED: an algorithm to achieve fairness in the Internet. In: IEEE INFOCOM, New York, March 1999.

Braden B, Clark D, Crowcroft J, Davie B, Deering S, Estrin D, et al. Recommendations on queue management and congestion avoidance in the Internet. In: RFC 2309, April 1998.

Christiansen M, Jeffay K, Ott D, Smith FD. Tuning RED for web traffic. In: ACM SIGCOMM, Stockholum, Sweden, August 2000.

Feng G, Agarwal AK, Jayaraman A, Siew CK. Modified RED gateways under bursty traffic. IEEE Commun Lett 2004;8(5):323–5.

Feng WC, Kandlur DD, Saha D, Kang G. Self-configuring RED gateway. In: IEEE INFOCOM, New York, March 1999a. p. 1320–8.

Feng WC, Kandlur DD, Saha D. Adaptive packet marking for maintaining end to end throughput in a differentiated service Internet. IEEE/ACM Trans Networking 1999b;7(5):685–97.

Floyd S. RED: discussions of setting parameters ⟨www.aciri.org/floyd/REDparameters.txt⟩, November 1997.

Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEEE/ACM Trans Networking 1993;1(4):397–413.

Guan L, Awan I, Woodward M, Wang X. Discrete-time performance analysis of a congestion control mechanism based on RED under multi-class bursty and correlated traffic. J Syst Software 2007;80(10):1716–25.

Hollot CV, Misra V, Towsley D, Gong W. Analysis and design of controllers for AQM routers supporting TCP flows. IEEE Trans Autom Control 2002;47(6):945–59.

Kim T, Lee K. Refined adaptive RED in TCP/IP networks. In: SICE-ICASE international joint conference, Busan, South Korea, October 18–21, 2006.

La RJ. Instability of a tandem network and its propagation under RED. IEEE Trans Autom Control 2004; 49(6):1006–11.

Le L, Aikat J, Jeffay K, Smith FD. The effects of active queue management on web performance. In: SIGCOMM: 2003 conference on applications, technologies, architectures, and protocols for computer communications, 2003. p. 265–76.

Low SH, Paganini F, Wang J, Doyle JC. Linear stability of TCP/RED and a scalable control. Comput Networks 2003;43(5):633–47.

Lu J-J, Wang Z-Q. Analysis of RED based on control theory. Control and Decision 2007;22(7):745–8.

May M, Bonald T, Bolot J-C. Analytic evaluation of RED performance. In: IEEE INFOCOM, Tel-Aviv, Israel, March 26–30, 2000a.

May M, Diot C, Lyles B, Bolot J. Influence of active queue management parameters on aggregate traffic performance. Research Report, Institut National de Recherche en Informatique et en Automatique, April 2000b.

Padhye J, Firoiu V, Towsley DF, Kurose JF. Modeling TCP Reno performance: a simple model and its empirical validation. IEEE/ACM Trans Networking 2000;8(2):133–45.

Ranjan P, Abed EH, La RJ. Nonlinear instabilities in TCP-RED. IEEE/ACM Trans Networking 2004;12(6): 1079–92.

Verma R, Iyer A, Karandikar A. Towards an adaptive RED algorithm for achieving delay-loss performance. In: IEE proceedings communications, vol. 150, no. 3, June 2003. p. 163–8.

Wang C, Li B, Sohraby K, Peng Y. AFRED: an adaptive fuzzy-based control algorithm for active queue management. In: 28th annual IEEE international conference on local computer networks, October 2003. p. 12–20.

Wang L, Cai L, Liu X, Shen X. Stability and TCP-friendliness of AIMD/RED systems with feedback delays. Comput Networks 2007;51(15):4475–91.

Zheng B, Atiquzzaman M. A framework to determine the optimal weight parameter of RED in next generation Internet routers. Technical Report, The University of Dayton, Department of Electrical and Computer Engineering, July 2000.

Zheng B, Atiquzzaman M. Low pass filter/over drop avoidance (LPF/ODA): an algorithm to improve the response time of RED gateways. Int J Commun Syst 2002;15:899–906.

Zheng B, Atiquzzaman M. Active queue management in TCP/IP networks. In: Performance of TCP over satellite networks. Englewood Cliffs, NJ: Prentice-Hall; 20003.

Zheng B, Atiquzzaman M. Active queue management. In: Hassan M, Jain R, editors. High performance TCP/IP networking. Englewood Cliffs, NJ: Prentice-Hall; 2004. p. 281–307 [chapter 12].