LRED: A Robust and Responsive AQM Algorithm Using Packet Loss Ratio Measurement

Chonggang Wang, *Member*, *IEEE*, Jiangchuan Liu, *Member*, *IEEE*, Bo Li, *Senior Member*, *IEEE*, Kazem Sohraby, *Senior Member*, *IEEE*, and Y. Thomas Hou, *Senior Member*, *IEEE*

Abstract—Active queue management (AQM) is an effective means to enhance congestion control, and to achieve trade-off between link utilization and delay. The de facto standard, Random Early Detection (RED), and many of its variants employ queue length as a congestion indicator to trigger packet dropping. Despite their simplicity, these approaches often suffer from unstable behaviors in a dynamic network. Adaptive parameter settings, though might solve the problem, remain difficult in such a complex system. Recent proposals based on analytical TCP control and AQM models suggest the use of both queue length and traffic input rate as congestion indicators, which effectively enhances stability. Their response time generally increases however, leading to frequent buffer overflow and emptiness. In this paper, we propose a novel AQM algorithm that achieves fast response time and yet good robustness. The algorithm, called Loss Ratio-based RED (LRED), measures the latest packet loss ratio, and uses it as a complement to queue length for adaptively adjusting the packet drop probability. We develop an analytical model for LRED, which demonstrates that LRED is responsive even if the number of TCP flows and their persisting times vary significantly. It also provides a general guideline for the parameter settings in LRED. The performance of LRED is further examined under various simulated network environments, and compared to existing AQM algorithms. Our simulation results show that, with comparable complexities, LRED achieves shorter response time and higher robustness. More importantly, it trades off the goodput with queue length better than existing algorithms, enabling flexible system configurations.

Index Terms—Active queue management, congestion control, TCP, packet loss ratio.

1 Introduction

Buffer management for Internet routers plays an important role in congestion control [1], [2]. However, the two main objectives of buffer management, namely, high link utilization and low packet queuing delay, often conflict with each other. Specifically, given that most end-nodes employ the responsive Additive Increase and Multiplicative Decrease (AIMD) TCP congestion control, a small buffer generally achieves a low queuing delay, but suffers from excessive packet losses and low link utilization, and vice versa. In addition, a simple policy like the widely used First-In-First-Out (FIFO) Tail-Drop often causes strong correlations among packet losses, resulting in the well-known "TCP synchronization" problem [3].

To mitigate such problems, Active Queue Management (AQM) has been introduced in recent years [2], [3], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. The basic idea is to actively trigger packet dropping (or marking provided explicit congestion notification (ECN) [4] is enabled) before buffer overflow. Obviously, the drop probability should depend on the degree of congestion. The de facto AQM standard, Random Early Detection (RED) [5], and many of its variants employ queue length as a congestion indicator to trigger packet dropping. Despite their simplicity, these approaches often suffer from unstable behaviors in a dynamic network. Adaptive parameter settings, though they might solve the problem, remain difficult in such a complex system. Recent proposals based on analytical TCP control and AQM models suggest the use of both queue length and traffic input rate as congestion indicators, which effectively enhances stability. Their response time generally increases however, leading to frequent buffer overflow and emptiness.

In this paper, we argue that packet loss ratio, which has never been explored in previous AQM studies, is another important index. The packet loss ratio is measured as the fraction of the packets dropped by the router and is updated over time. Intuitively, for a well-designed AQM algorithm, the loss ratio should be close to the desired drop probability in a steady-state, and it deviates from the desired drop probability if the buffer is (or tends to) overflow or empty. In other words, an increasing packet loss ratio implies that congestion occurs, and a decreasing implies that the congestion is relieved.

Given the above observations, we propose a novel AQM algorithm, LRED, which incorporates the packet loss ratio as a complement to queue length for congestion estimation. We stress two salient features of this hybrid design: First, the calculation is simple and fast for both measures, which is desirable for high-throughput routers; second, it enables a multigranular update for the packet drop probability: upon

- *C. Wang is with the Electrical Engineering Department, University of Arkansas, Bell 3217, Fayetteville, AR 72701. E-mail: cgwang@ieee.org.*
- *J. Liu is with School of Computing Science, Simon Fraser University, TASC 9227, Burnaby, British Columbia, Canada V5A 1S6. E-mail: jcliu@cs.sfu.ca.*
- *B. Li is with the Computer Science Department, Hong Kong University of Science and Technology, Room 3505, Clear Water Bay, Hong Kong, China. E-mail: bli@cse.ust.hk.*
- *K. Sohraby is with the Electrical Engineering Department, University of Arkansas, Bell 3178, Fayetteville, AR 72701. E-mail: sohraby@uark.edu.*
- *Y.T. Hou is with The Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, 302 Whittemore Hall (0111), Blacksburg, VA 24061. E-mail: thou@vt.edu.*

Manuscript received 23 Feb. 2005; revised 20 Sept. 2005; accepted 6 Feb. 2006; published online 28 Nov. 2006.
Recommended for acceptance by Y.A. Oruc.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0179-0205.

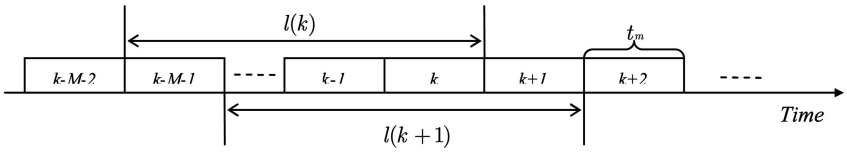


Fig. 1. Packet loss ratio measurement in LRED.

each packet arrival, we can use the instantaneous queue length mismatch to update the drop probability, so as to keep the queue length around an expected value; on a coarser grain, we can adjust the drop probability according to the packet loss ratio, so as to expedite control response. We have developed an analytical model for LRED, which suggests that this multigranular update improves not only the stability of the system, but also its responsiveness.

The performance of LRED is further examined under various simulated network environments, and compared to existing AQM algorithms, including Adaptive Virtual Queue (AVQ) [10], Proportional-Integra (PI) [12], and Random Exponentially Marking (REM) [13]. Our simulation results reveal that, with comparable complexities, LRED achieves shorter response time and better robustness. More importantly, it enables better trade-off between the goodput and queue length than existing algorithms, leading to flexible system configurations.

The remainder of this paper is organized as follows: Section 2 briefly introduces the existing AQM algorithms. In Section 3, we offer an overview of the LRED algorithm. Section 4 develops an analytical model for the combined TCP/AQM system and discusses its general properties. Based on this model, we analyze the stability and responsiveness of LRED and other AQM algorithms in Section 5. The simulation results for LRED are presented in Section 6. Finally, we conclude the paper and give out future works in Section 7.

2 Related Work

There have been numerous proposals on AQM in the past decade, and RED [5] is probably the most widely studied algorithm. RED uses the average queue length to calculate the packet drop probability and to regulate the queue length accordingly. Specifically, when the average queue length is greater than a preconfigured threshold (min_{th}), RED begins to drop newly arrived packets; the dropping probability increases linearly to the average queue length with a slope of max_p. Despite its simplicity, the optimal parameter configuration for RED remains a daunting task. Hence, several enhancements, like S-RED [7] and ARED [8], have been introduced to adaptively configure the parameters. Another variation is BLUE [9], which calculates the packet drop probability based only on two events: buffer overflow and emptiness. When the buffer overflows, it increases packet drop probability by δ_1 and, when empty, decreases by δ_2. Nevertheless, adaptive settings in such a complex system are still difficult. BLUE cannot well regulate the queue length to an expected value.

On the other hand, AVQ [10] uses input rate $x(t)$ to control packet drop and to achieve an expected link utility γ. Basically, the packet drop probability is proportional to the mismatch between $x(t)$ and γ. Through maintaining a virtual queue, AVQ deterministically drops packets upon each new packet arrival, realizing the same effect of probabilistic packet drop. AVQ achieves lower average queue length and higher link utility than RED and its variants [10].

Recently, some advanced algorithms use the queue length and input rate jointly to achieve better performance. One example is PI [12], which regulates the queue length to an expected value q_0 according to the queue mismatch and its integral. The latter is closely related to the input rate mismatch. If the network states are known a priori, optimal parameters of PI can be determined through a control-theoretical model. However, in dynamic networks, PI may have to use a conservative setting to ensure stability, yielding long response time. Another example is REM [13], which uses a linear combination of the queue mismatch and input rate mismatch to calculate the drop probability, and the input rate mismatch is equivalently simplified to the queue variance between two adjacent length samples.

LRED employs the packet loss ratio as a complement to the queue length for AQM. To the best of our knowledge, it has never been explored in the previous studies. The use of packet loss ratio enables LRED to catch network dynamics in time, thus achieving fast control response and better performance in terms of goodput, average queue length, and packet loss ratio.

3 Overview of LRED

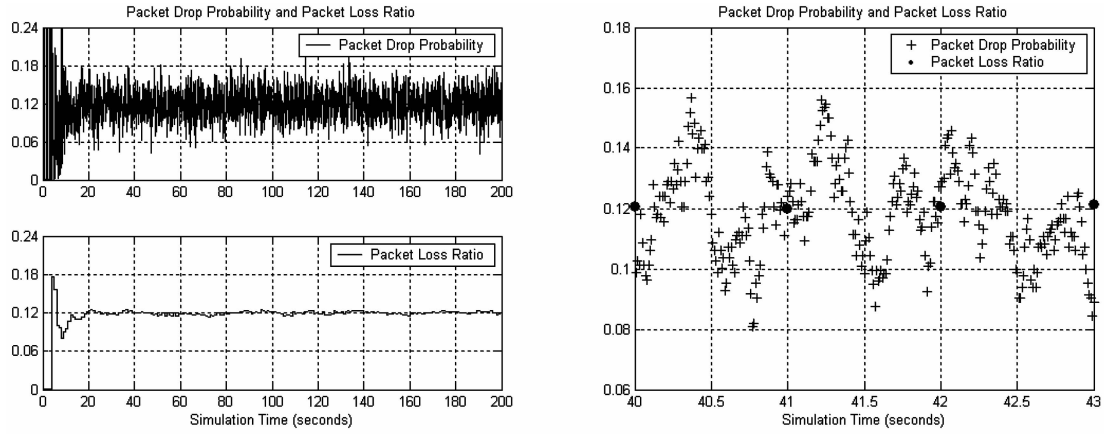
Similar to most existing AQM algorithms, LRED keeps a packet drop probability p, which is adaptively updated upon each packet arrival, and the incoming packets are dropped with probability p from the tail of the queue. The calculation of the drop probability is relatively simple, following two design rules: 1) when the queue length is close to an expected steady-state length q_0, the drop probability should be close to the packet loss ratio, and 2) when the queue length becomes larger (or smaller), the drop probability should be increased (or decreased) to regulate the queue length.

To achieve an adaptive yet stable estimation for the packet loss ratio, LRED estimates the ratio in every measurement period (t_m). Let $l(k)$ be the packet loss ratio of the kth measurement, which is calculated as the number of dropped packets over the total number of packets arrived during the latest M periods (see Fig. 1); that is:

$$l(k) = \frac{\left(\sum_{i=0}^{M-1} N_d(k-i)\right)}{\left(\sum_{i=0}^{M-1} N_a(k-i)\right)}, \quad (1)$$

where $N_d(k)$ is the number of packets dropped in the kth measurement period, and $N_a(k)$ is the number of packets arrived in the kth measurement period. The estimated packet loss ratio $\overline{l(k)}$ is calculated based on $l(k)$ using an exponential weighted moving average (EWMA), as follows:

$$\overline{l(k)} = \overline{l(k-1)} \cdot w_m + (1 - w_m) \cdot l(k), \quad (2)$$



(a) (b)

Fig. 2. Evolution of packet loss ratio $(\overline{l(k)})$ and packet drop probability (p) in a simulation. (a) Time duration from 0 to 200. (b) Time duration from 40 to 43.

where w_m is a weighting factor. In order to promptly catch loss changes, we set w_m to a small value.

Given the estimated packet loss ratio and the instantaneous queue length, a straightforward way to meet the design rules (to control the queue length around an expected steady-state value) is to use a linear function, $p = \overline{l(k)} + \alpha(q - q_0)$, where p is the packet drop probability and q is the instantaneous queue length. It is, however, often difficult to choose an optimal α. In particular, if $\overline{l(k)}$ is large and α is set too small, the packet drop probability for a small queue length q would still be quite high, resulting in low link utility. To avoid this, in LRED, we let p increase with the measured loss ratio; in other words, we have:

$$p = \overline{l(k)} + \beta\sqrt{\overline{l(k)}}(q - q_0), \tag{3}$$

where $\beta > 0$ is a preconfigured constant. Intuitively, the adjustment of the packet drop probability should be related to traffic rate change, in order to avoid buffer overflow or buffer emptiness. In (3), this adjustment is set to $\beta\sqrt{\overline{l(k)}}(q - q_0)$, which uses square-root function of $\overline{l(k)}$. This square-root function can guarantee that $\beta\sqrt{\overline{l(k)}}(q - q_0)$ is proportional to the percentage of traffic rate change, if we assume traffic flows are TCP. This is because the TCP throughput is reversely proportional to the square-root of the stable packet loss ratio [15]. Furthermore, given that $(q - q_0)$ depends on both the value of stable traffic rate and the percentage of traffic rate change, the product of $(q - q_0)$ and $\sqrt{\overline{l(k)}}$ depends on the percentage of traffic rate change only. Our analysis in Section 5 also shows that (3) ensures stable control. In the case of nonuniform packet sizes, different drop probabilities may have to be applied to packets with different sizes. Specifically, larger packets might have higher drop probabilities. In this case, we can maintain the expected queue length counted in bytes and, therefore, improve the fairness among TCP connections with different packet size.

It can be seen from (2) and (3) that LRED updates $\overline{l(k)}$ and p at different time scales. $\overline{l(k)}$ is updated every measurement period (see Fig. 1), while the packet drop probability p is recalculated each time a new packet comes. In a steady-state, $\overline{l(k)}$ could converge to a stable value. However, the AIMD mechanism in TCP implies that a TCP sender will always try to decrease (or increase) its sending rate if it detects packet loss (or not); therefore, the queue length in routers will unavoidably fluctuate and the packet drop probability p thus fluctuates around $\overline{l(k)}$. This is illustrated in Fig. 2, where we can see $\overline{l(40)} \approx \overline{l(41)} \approx \overline{l(42)} \approx \overline{l(43)} \approx 0.12$. When $t = 41.0$, $\overline{l(41)} \approx 0.12$. When $t < 41.0$, the packet drop probability fluctuates around 0.12.

We can observe from (3) and Fig. 2 that: 1) In between two adjacent instants k and $k+1$, when p is updated according to (3), $\overline{l(k)}$ remains constant and, therefore, p depends only on the control error $(q - q_0)$; or more precisely, it is proportional to $(q - q_0)$ only. 2) In a steady-state, $\overline{l(k)}$ will be close to the stable value and, therefore, p is still proportional to the control error $(q - q_0)$ in this case. As such, we believe that LRED is closer to a proportional controller.

Detailed operations for LRED can be found in Fig. 3 where q_0 is assumed to 100. We will further discuss its parameter settings and prove its stability with the square-root function in Section 5.

4 Model of Combined TCP/AQM Systems

In this section, we first review an analytical model for a combined TCP/AQM system [11], [14]. We then make several important observations on the model, which serve as the basis for designing a stable AQM algorithm, in particular, for LRED proposed in this paper.

4.1 The Combined TCP/AQM Models

We consider an abstract network of a single bottleneck with multiple TCP connections. Its status can be represented by a 3-tuple (N, C, R), where N is the number of TCP flows, C is the bottleneck link capacity, and R is the round-trip time (RTT). With the assumption that TCP flows are long-lived persistent and packet size is constant, the system equation

```
/* Parameters */
q_o : the expected value of queue length;                    q : the current instantaneous queue length;
p : the calculated packet drop probability;                  lossRatio : the estimated packet loss ratio;
β : weighting factor (=0.001 in this paper);                 t_m : the measurement period;
M : the number of the latest periods to measure lossRatio ;  w_m : the measurement weight;

/* Initialization */
arrPktNum=0; dropPktNum=0; allArrNum=0; allDropNum=0; index=0; q_o =100;
M=4; w_m =0.1; β =0.001; t_m =1.0; lossRatio=0.0;

LossRatioMeasure() /* Called every t_m seconds */
1   dropNum[index]=dropPktNum;
2   arrNum[index]=arrPktNum;
3   arrPktNum=0;
4   dropPktNum=0;
5   index++;
6   if (index==M)  index=0;
7   for (i=0; i<M; i++) { allDropNum+=dropNum[i]; }
8   for (i=0; i<M; i++) { allArrNum+=arrNum[i]; }
9   lossRatioTemp=allDropNum/allArrNum;
10  lossRatio=lossRatio*w_m+lossRatioTemp*(1- w_m);
11  allDropNum=0;
12  allArrNum=0;

Enqueue() /* Called upon each packet arrival*/
1   arrPktNum++;
2   p = lossRatio + β√lossRatio (q − q_o) ;
3   p = max(0, min(1, p)) ;
4   random=uniformRandom(0, 1);
5   if (buffer is full) {
6       Drop the packet;
7       dropPktNum++;
8   }
9   else if (random> p )  {Enqueue the packet; }
10  else {  Drop the packet;
11      dropPktNum++;
12  }
```

Fig. 3. Pseudocode of the LRED algorithm.

for the TCP congestion window (w) and the queue length (q) can be approximated as [11]:

$$\dot{w} = f(w,p) = \frac{1}{R} - \frac{w(t)w(t-R)}{\eta R} p(t-R), \tag{4}$$

$$\dot{q} = g(w,\, p) = \frac{N}{R} w(t) - C, \tag{5}$$

where p is the packet drop probability, and η is a parameter depending on TCP implementations. Let b be the number of packets acknowledged by a received acknowledgement (ACK), we have $\eta = 3/2b$ [15]. Equations (4) and (5) give a model of combined TCP/AQM system with long-lived flows and constant packet size. It is worth noting that the actual traffic mix in real Internet might be more complex; for example, flows can start and end all the time and the packet sizes may vary for different applications. Yet, existing studies have shown that this model offers a good approximation [11], [12], [14].

If we let $f(w,p) = 0$ and $g(w,p) = 0$, the TCP congestion window w_0 and packet drop probability p_0 in a steady-state can be calculated as:

$$w_0 = \frac{RC}{N}, \quad p_0 = \frac{\eta}{w_0^2} = \frac{\eta N^2}{R^2 C^2} < 1. \tag{6}$$

For ease of exposition, we assume each ACK acknowledges only one packet; therefore, $b = 1$ and $\eta = 1.5$. The steady-state throughput of a single TCP flow given by the above model becomes $\frac{C}{N} = \frac{\sqrt{3/2}}{R\sqrt{p_0}}$, which is consistent with the well-known TCP throughput equations [15].

From (4) and (5), it is clear that the combined TCP/AQM system is nonlinear. Let $\delta w = w - w_0$ and $\delta p = p - p_0$ be the mismatches (i.e., the deviations from the steady-state values) for the TCP congestion window and the packet drop probability, respectively. Equations (4) and (5) can be locally linearized around a stable point (w_0, p_0) by assuming $w(t) \approx w(t-R)$ as follows:

$$\delta\dot{w} = \frac{\partial f}{\partial w}\delta w + \frac{\partial f}{\partial p}\delta p = -[K_{11}\delta w + K_{12}\delta p(t-R)], \tag{7}$$

$$\delta\dot{q} = \frac{\partial g}{\partial w}\delta w + \frac{\partial g}{\partial p}\delta p = K_{21}\delta w, \tag{8}$$

where $K_{11} = \frac{2N}{R^2 C}$, $K_{12} = \frac{RC^2}{\eta N^2}$, and $K_{21} = \frac{N}{R}$ [11]. All of them are positive constants related to the network parameters.

Applying Laplace transform to (7) and (8), we have the following system equations:

$$sW(s) = -[K_{11}W(s) + K_{12}P(s)e^{-Rs}], \tag{9}$$

$$sQ(s) = K_{21}W(s). \tag{10}$$

In (9) and (10), there are three unknown variables: $W(s)$, $P(s)$, and $Q(s)$. Hence, it is necessary to find one more equation to solve the problem. This can be achieved by bridging $P(s)$ and $Q(s)$ through AQM. As discussed earlier, our proposed LRED is a proportional controller; we thus focus on the proportional AQM control in our analysis. Consider an AQM proportional control mechanism that determines p according to the instantaneous queue length q. Its general control equation can be formulated as:

$$\delta p = H_c \delta q. \tag{11}$$

The corresponding system equation can be written as:

$$P(s) = H_c * Q(s), \tag{12}$$

where $H_c > 0$ is a predefined parameter.

4.2 General Properties of Proportional AQM Control

From (9), (10), and (12), we can obtain the characteristic equation for such a system as:

$$s^2 + K_{11}s + K_cH_ce^{-Rs} = 0, \tag{13}$$

where $K_c = K_{12}K_{21} = \frac{C^2}{\eta N}$.

The stability of the above system depends on whether the root of (13), $s = \sigma + j\omega$, lies in the left half-complex plane. To facilitate its stability analysis, we have made the following observations on (13).

First, if the root of (13) strictly lies in the left half-complex plane, the combined system, defined by network parameters (N, C, R) and AQM control parameter H_c, is stable. Hence, given (N, C, R), we can choose H_c to satisfy this condition. On the other hand, given control parameter H_c, only some of the system (N, C, R) can be stably controlled. Specifically, when $R = 0$, the root of (13) is:

$$s = s_0 = \frac{-K_{11} \pm \sqrt{K_{11}^2 - 4K_cH_c}}{2}, \tag{14}$$

which strictly lies in the left half-complex plane irrespective of $K_{11}^2 \geq 4K_cH_c$ or $K_{11}^2 < 4K_cH_c$. Therefore, the system with zero delay is stable.

When $R > 0$ and letting $s = \sigma + j\omega$, the root of (13) can be calculated as follows:

$$\sigma^2 - \omega^2 + K_{11}\sigma + K_cH_ce^{-R\sigma}\cos(R\omega) = 0, \tag{15}$$

$$2\sigma\omega + K_{11}\omega - K_cH_ce^{-R\sigma}\sin(R\omega) = 0. \tag{16}$$

Clearly, the imaginary part (ω) of root s is nonzero; otherwise, (15) will be invalid. Also note that s changes continuously in the complex plane when N, C, R, or H_c changes continuously. Assume the first time that s meets the imaginary axis is at $R = R^+$. When $0 \leq R < R^+$, s should strictly lie in the left half-complex plane, and the system is thus stable. The remaining problem therefore is to find the value of R^+.

We first consider the absolute imaginary root $(s = j\omega)$ with $\omega > 0$ (the case of $\omega < 0$ is symmetric). When $R > 0$, (13) can be rewritten as:

$$T(s) = \frac{e^{-Rs}K_cH_c}{s(s + K_{11})} = -1. \tag{17}$$

Given (17), the following conditions on magnitude and angles must be met,

$$|T(j\omega)| = 1, \angle T(j\omega) = (2k+1)\pi, \quad k = 0, \pm1, \pm2,$$

and ω can be calculated as:

$$\omega(N, C, R, H_c) = \omega = \sqrt{y(N, C, R, H_c)/2}, \tag{18}$$

$$y(N, C, R, H_c) = \sqrt{K_{11}^4 + 4K_c^2H_c^2} - K_{11}^2, \tag{19}$$

$$R\omega + \arctan\left(\frac{\omega}{K_{11}}\right) + \frac{\pi}{2} = (2k+1)\pi, \quad k = 0, 1, 2. \tag{20}$$

Since K_{11} is a decreasing function of R, and K_c is independent of R, we have that $\omega(N, C, Rq, H_c)$ is an increasing function of R, if H_c is independent of or an increasing function of R. Therefore, R^+ corresponds to the smallest R satisfying (13) and (17). Now, the problem is to determine the value of k that yields the smallest R, which can be solved through the following lemma.

Lemma 1. *When $k = 0$, (18), (19), and (20) yield the smallest value of R and ω, if H_c is independent of or an increasing function of R. Also, (20) can be simplified to:*

$$R\omega + \arctan\left(\frac{\omega}{K_{11}}\right) = \frac{\pi}{2}. \tag{21}$$

Proof. We prove this by contradiction. Assume that $k \to (R_k, \omega_k)$, $k > 0$, $R_k > 0$, and $\omega_k > 0$. According to (20), we have:

$$\begin{aligned} R_k\omega_k - R_0\omega_0 &= 2k\pi + \left[\arctan\left(\frac{\omega_0}{K_{11}}\right) - \arctan\left(\frac{\omega_k}{K_{11}}\right)\right] \\ &> 2k\pi - \pi/2 > 0. \end{aligned}$$

Assume $R_k \leq R_0$, we have $\omega_k \leq \omega_0$ according to (18) and (19), if H_c is independent of or an increasing function of R. Since $R_k \leq R_0$ and $\omega_k \leq \omega_0$, we have $R_k\omega_k \leq R_0\omega_0$, which contradicts that $R_k\omega_k - R_0\omega_0 > 0$. Hence, $R_k > R_0$, or equivalently, (18), (19), and (20) yield the smallest value of R and ω when $k = 0$. □

Lemma 2. *Given network parameters(N, C) and AQM control parameter H_c. Assume that R^+ satisfies:*

$$R^+\omega + \arctan\left(\frac{\omega}{K_{11}}\right) = \frac{\pi}{2}, R^+ > 0, \tag{22}$$

where ω is the solution to (18) and (19). If function $y(N, C, R, H_c)$ in (19) is an increasing function of R, then the system is stable for all $R < R^+$, and R^+ is unique.

Proof. Since $y(N, C, R, H_c)$ is an increasing function of R, according to Lemma 1, R^+ is the smallest R that satisfies (13) and (14), and is unique. It also means that the first time the root meets the imaginary axis is at $R = R^+$. Therefore, the system is stable for all $R < R^+$. □

Lemma 3. *Given network parameters (C, R) and AQM control parameter H_c. Assume that N^- satisfies:*

$$R\omega + \arctan\left(\frac{\omega}{K_{11}}\right) = \frac{\pi}{2}, N^- > 0, \tag{23}$$

where ω is the solution to (18) and (19). If function $y(N, C, R, H_c)$ in (18) is an increasing function of R and a decreasing function of N, then the system is stable for $N > N^-$.

Proof. Let $R(N)$ be the solution to (13) with $N > N^-$. If $R < R(N)$, from Lemma 2, the system is stable for all $N > N^-$. Since $y(N, C, R, H_c)$ in (18) is a decreasing function of N, we have:

$$\omega(N) = \omega(N, C, R, H_c) < \omega(N^-, C, R, H_c) = \omega(N^-).$$

Moreover, K_{11} is an increasing function of N, which implies:

$$\begin{aligned} R\omega(N) + \arctan\left(\frac{\omega(N)}{K_{11}}\right) &< R\omega(N^-) \\ + \arctan\left(\frac{\omega(N^-)}{K_{11}}\right) &= \frac{\pi}{2}. \end{aligned}$$

Since $R(N)\omega(N) + \arctan[\omega(N)/K_{11}] = \frac{\pi}{2}$, we have $R(N) > R$ and, hence, for all $N > N^-$, the system is stable. □

Lemma 4. *Given network parameters (N, C, R), and assume that H_c^* satisfies:*

$$R\omega + \arctan\left(\frac{\omega}{K_{11}}\right) = \frac{\pi}{2}, H_c^* > 0, \tag{24}$$

where ω is given by (18) and (19). If function $y(N, C, R, H_c)$ in (19) is an increasing function of both R and H_c, then the system is stable for all $H_c < H_c^*$.

Proof. Similar to that of Lemma 3. □

Theorem 1. *Let the network parameters be (N^-, C, R^+), and assume that H_c^+ satisfies:*

$$R^+\omega + \arctan\left(\frac{\omega}{K_{11}}\right) = \frac{\pi}{2}, H_c^+ > 0, \tag{25}$$

where ω is given in (18) and (19). If function $y(N, C, R, H_c)$ in (19) is an increasing function of R, a decreasing function of N, and an increasing function of H_c, then the system is stable for $H_c < H_c^+$, $N > N^-$, and $R < R^+$.

Proof. Directly follows Lemmas 2 through 4. □

5 Analysis of LRED and Parameter Settings

5.1 Stability Analysis of LRED

We now investigate the stability of LRED and discuss the settings of several important parameters, in particular, β. Since the packet loss ratio is close to the stable packet drop probability p_0 in LRED, i.e., $\overline{l(k)} \approx p_0$, we can approximately rewrite (3) as:

$$p = p_0 + \beta\sqrt{p_0}(q - q_0), \tag{26}$$

where, according to (6), $p_0 = \eta N^2/(R^2C^2)$. It follows that:

$$\delta p = \beta\sqrt{p_0}\delta q, \tag{27}$$

or

$$P(s) = \beta\sqrt{p_0}Q(s), \tag{28}$$

and the system transfer function of LRED (see (12)) is thus given by:

$$H_c = P(s)/Q(s) = \beta\sqrt{p_0}. \tag{29}$$

Substituting (29) for H_c in (18) and (19), we have the following lemma.

Lemma 5. *For LRED, function $y(N, C, R, H_c)$ in (19) is a decreasing function of N, and an increasing function of β. Moreover, if $\beta < \frac{\sqrt{2\eta(2N)^2}}{(R^3C^3)}$, it is an increasing function of R.*

Proof. For LRED, $y(N, C, R, H_c)$ in (18) can be calculated as:

$$y(N, C, R, H_c) = \sqrt{\left(\frac{2N}{R^2C}\right)^4 + \frac{4\beta^2C^2}{\eta R^2}} - \left(\frac{2N}{R^2C}\right)^2. \tag{30}$$

It can be easily observed that $y(N, C, R, H_c)$ is a decreasing function of N and an increasing function of β.

We now consider its relation with R. The derivative of function y with respect to R is:

$$\frac{\partial y}{\partial R} = \frac{-\frac{8(2N)^4}{R^9C^4} - \frac{8\beta^2C^2}{\eta R^3}}{2\sqrt{(\frac{2N}{R^2C})^4 + \frac{4\beta^2C^2}{\eta R^2}}} + \frac{4(2N)^2}{R^5C^2} = \frac{-f_1(R)}{f_2(R)} + f_3(R).$$

Note that y is an increasing function of R if $\frac{\partial y}{\partial R} > 0$, which is equivalent to:

$$[f_2(R) \cdot f_3(R)]^2 - f_1^2(R) = \frac{64\beta^2[2\eta(2N)^4 - \beta^2R^6C^6]}{\eta^2R^{12}C^2} > 0.$$

It follows that $\beta < \sqrt{2\eta}(2N)^2/(R^3C^3)$. □

Theorem 2. *Given network parameters (N^-, C, R^+), and assume that β_0 satisfies:*

$$R^+\omega + \arctan\left(\frac{\omega}{K_{11}}\right) = \frac{\pi}{2}, \beta_0 > 0, \tag{31}$$

where ω is defined in (18) and (19), and $H_c = \beta\sqrt{p_0}$ in LRED. If

$$\beta < \beta^+ = \min\left(\beta_0, \frac{\sqrt{2\eta}(2N^-)^4}{(R^+)^3C^3}\right),$$

the system is stable for any $N > N^-$ and $R < R^+$.

Proof. Directly follows Lemma 5 and Theorem 1. □

Given Theorem 2, it is easy to find β that guarantees the stability of the system. For example, consider a network in which the $\text{mean packet size} = 500$ bytes, $C = 2,500$ packets/sec (or equivalently, 10 Mbps), $N^- = 300$, and $R^+ = 0.35$ seconds. The AQM parameter β^+ is thus 0.001, and, for any $\beta < \beta^+$, the system is stable with all $N > N^-$ and $R < R^+$.

5.2 Response Time Analysis and Comparison

Since enhancing stability and minimizing response time often conflict with each other, existing algorithms such as PI [12] and REM [13] have tried to find a trade-off between them. If network parameters, in particular, N and R, are known a priori, these algorithms can achieve a stable control with minimized response time. In a dynamic network, however, it is difficult to accesses these parameters precisely. Hence, they generally resort to a conservative design that guarantees stability, but may sacrifice the corresponding response time. For example, the default parameter for PI in NS2 Simulator [16] is set based on a small N and large R. When N increases or R decreases, it will yield a long response time, though the system remains stable.

In this section, we provide a simple analysis on the response times of the typical AQM schemes. We focus on a highly dynamic scenario: at time $t = 0$, N TCP flows become active simultaneously, where N is large enough such that the buffer is fully filled before the system converges to a steady state with packet drop probability p_0 and expected queue length q_0.

We first consider PI [12], which periodically updates its packet drop probability with a sampling frequency f_{PI} (Hz). Each update is as follows:

$$p(k) = p(k-1) + a[q(k) - q_0] - b[q(k-1) - q_0], \tag{32}$$

where $a > 0$ and $b > 0$ are two constants [12]. We can assume that $p(0) = 0$ and $q(k) \approx Q$ before reaching a steady-state, where Q is the maximal buffer size. It follows that:

$$p(k) \approx k(a-b)(Q-q_0). \tag{33}$$

Denote $p(k_0) = p_0$. The lower bound of the response time of PI (RT^-_{PI}) is thus:

$$RT^-_{PI} \geq \frac{k_0}{f_{PI}} = \frac{p_0}{((Q-q_0)(a-b)f_{PI})}. \tag{34}$$

In REM [13], packet drop probability is also periodically updated with a sampling frequency f_{REM} (Hz), as follows [13]:

$$p(k) = 1 - \phi^{-u(k)}, \tag{35}$$

$$u(k) = u(k-1) + \gamma[q(k) - (1-\alpha)q(k-1) - \alpha q_0)], \tag{36}$$

$$u(k) = \max(0,\ u(k)), \tag{37}$$

where $\phi > 1$, $\gamma > 0$, and $\alpha > 0$ are three constants, and their optimal values are derived in [13]. Similarly, we can also assume that $u(0) = 0$, $u(k) \ll 1$, and $q(k) \approx Q$ before reaching the steady-state, which follows that:

$$u(k) \approx k\gamma\alpha(Q-q_0), \tag{38}$$

$$\begin{aligned} p(k) = 1 - \phi^{-u(k)} &= 1 - \sum_{i=0}^{+\infty} \frac{[-u(k)\ln\phi]^i}{(i!)} \\ &\approx u(k)\ln\phi = k\gamma\alpha(Q-q_0)\ln\phi, \end{aligned} \tag{39}$$

and the lower bound of the response time for REM (RT^-_{REM}) is thus:

$$RT^-_{REM} \geq \frac{k_0}{f_{REM}} = \frac{p_0}{f_{REM}\gamma\alpha(Q-q_0)\ln\phi}. \tag{40}$$

From (34) and (40), we can see that the response times of PI or REM mainly depend on the following parameters: buffer size Q, expected queue length q_0, and packet drop probability p_0 (which is an increasing function of TCP flow number N, and a decreasing function of round-trip time R and link capacity C). Consequently, under heavy congestion or with a high drop probability, PI and REM suffer from long response times. When buffer size Q is small, the responsiveness of PI and REM becomes worse as well.

In LRED, the response time is mainly influenced by the period (t_m) to measure the packet loss ratio. In particular, under heavy traffic (e.g., when N is large and/or R is small), packets will be dropped frequently and the packet loss ratio can be accurately estimated within a couple of measurements. As a result, LRED is very responsive in this case, while PI and REM perform poorly given that p_0 is high. More importantly, when network conditions change dramatically, LRED can quickly converge to new steady states. When the traffic load is light (e.g., when N is small and/or R is large), there are few packet losses, and more rounds of measurement are thus needed for accurately estimating the stable packet loss ratio. In this case, the response time of LRED would slightly increase, but remain comparable to that of PI and REM, as will be shown in our simulations. In summary, LRED decouples the response time and packet drop probability, making its response time almost independent of the congestion levels.

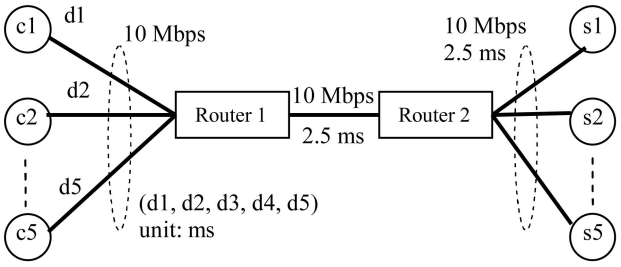


Fig. 4. Network topology for simulations.

6 Simulation Results

In this section, we examine the performance of LRED through NS2 [16] simulations. We also compare it with existing AQM schemes, in particular, PI [12] and REM [13]. For loss ratio measurement in LRED, we set $w_m = 0.1$, $t_m = 1.0$ second, and $M = 4$; β is set to 0.001 according to Theorem 2. The network topology for the simulation is the commonly used dumb-bell topology (see Fig. 4). In this topology, five clients are linked to router 1, and five servers are behind router 2. The capacity of each link is 10 Mbps, and the link between router 1 and router 2 thus becomes a bottleneck. The link delay between router 2 and any server is 2.5 ms, and the delays between the clients (c_1 through c_5) and router 1 are heterogeneous, denoted by a 5-tuple $(d_1, d_2, d_3, d_4, d_5)$. All the flows in the network are uniformly distributed among the pairs of client c_i and server s_i. The default packet size is 500 bytes. The buffer size of each router is 200 packets unless another value is explicitly configured such as in Experiment 6. We run each simulation for 200 seconds, which is long enough to observe both transient and steady-state behaviors of an AQM scheme.

The following parameter settings are used in our simulations. 1) For REM [13], $\phi = 1.001$, $\alpha = 0.1$, $\gamma = 0.001$, and the sampling interval is 2 ms. These values are adapted from [13] and [16]. 2) For PI [12], we use two settings: *default* and *optimal*, respectively, denoted by PI and PI*. The default values for PI are $a = 0.00001822$, $b = 0.00001816$, and the sampling frequency is 170 Hz, which are adapted from [16]; the optimal values for PI* are derived according to the design rules in [12], which depend on network parameters (N^-, C, R^+). Here, N^- is the minimum number of the TCP flows and R^+ is the maximal round-trip time. Hence, the optimal values for PI* are not fixed for the experiments.

In our study, we focus on the following key performance metrics: goodput, average queue length, average queue deviation, and packet loss ratio. The goodput is the overall throughput of the system excluding retransmissions, the average queue length is calculated as the arithmetic mean of the instantaneous queue lengths, and the average queue deviation is the average of the absolute deviations of the instantaneous queue lengths from the mean.

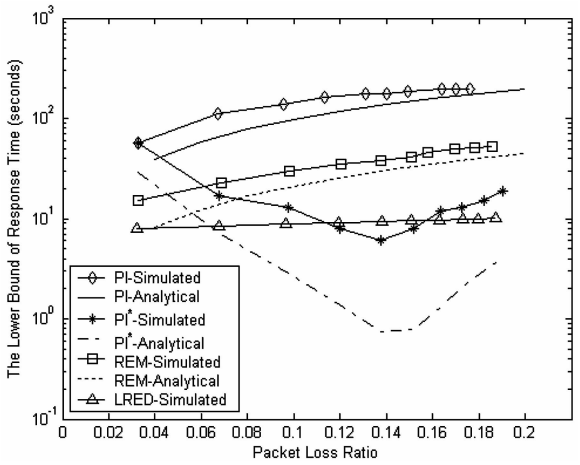


Fig. 5. Experiment 1: Response times for the AQM schemes.

6.1 Homogenous Traffic: Long-Lived TCP Flows Only

In this set of experiments, we focus on a network with homogeneous traffic, i.e., all are persistent TCP flows.

Experiment 1: Response time under various congestion degrees. We first investigate the response times for the AQM schemes to reach a steady state. The expected queue length q_0 is set to 100 packets, and the vector for link delays between Clients and Router 1, $(d_1, d_2, d_3, d_4, d_5)$, is (10, 50, 100, 150, 200). The total number of TCP flows, N, varies from 100 to 1,000, which leads to 10 scenarios with various degrees of congestion. For PI*, we set N^- from 100 to 1,000 for the 10 scenarios, R^+ is set to 450 ms for $N^- \leq 500$, and 500 ms to 900 ms for N^- from 600 to 1,000, which guarantee that $N^- < (R^+ * C)/2$ according to the design rule of PI in [12].

The response times as a function of the loss ratio are presented in Fig. 5. It can be seen that when the packet loss ratio increases (which is a result of an increase of the number of TCP flows), the response times of PI or REM increase, for both simulated and analytical results. The mismatches between the simulated and analytical results for PI and REM are relatively small, especially with high loss ratios. PI*, however, has a big mismatch between its simulated and analytical results. The main reason is that we assume the buffer is always full before reaching a steady-state when analyzing the lower bound of the response time in the Section 5. Since PI* uses the optimal parameter according to the design rules in [12], it has a faster response time than PI using the default parameters. Nonetheless, the response time of LRED is generally lower than other schemes, and it is nearly independent of the packet loss ratio. As a result, the gaps between LRED and other schemes under a high loss ratio are pretty significant.

Experiment 2: Stability under extreme conditions. In this experiment, we examine the stability of the AQM schemes under two extreme cases: 1) light congestion with a small number of TCP flows N and large round-trip time R, and 2) heavy congestion with a large N and small R. Fig. 6 presents the instantaneous queue lengths under such two cases: 1) In the first, we set $N = 80$ and $d_1 = d_2 = d_3 = d_4 = d_5 = 250\ ms$. Therefore, we have $N^- = 80$ and $R^+ = 550$ ms for PI*. 2) In the second, $N = 800$ and $d_1 = d_2 = d_3 = d_4 = d_5 = 10\ ms$ and, accordingly, we have $N^- = 800$ and $R^+ = 100$ ms for PI*. Other parameters are the same as those in Experiment 1. We can see that LRED and PI* have similar response times, but LRED has less overshoots and smaller queue deviations. It can still be seen that, under heavy congestion, LRED achieves a shorter response time and better stability than PI and REM. Under light congestion, the queue length of REM drastically oscillates and almost out of control. On the contrary, LRED and PI can still stably regulate the queue, and LRED is relatively better. Note that the response time of LRED slightly increases in the light congestion case because it needs more rounds to have a stable loss ratio estimate. Nevertheless, its response time is still comparable to that to PI or PI*.

Experiment 3: Varying the expected queue length. This experiment is used to study the performance of AQM schemes when the expected queue length varies. We fix the number of persistent TCP flows to $N = 400$. The expected queue length q_0 varies from 20 to 40, 60, 80, 100, 120, 140, and 160. The other parameters are the same as that in the Experiment 1. Accordingly, for PI*, we set $N^- = 400$ and $R^+ = 450$ ms. We can see from Fig. 7 that PI achieves higher throughput and lower loss ratio. The reason is that its slow

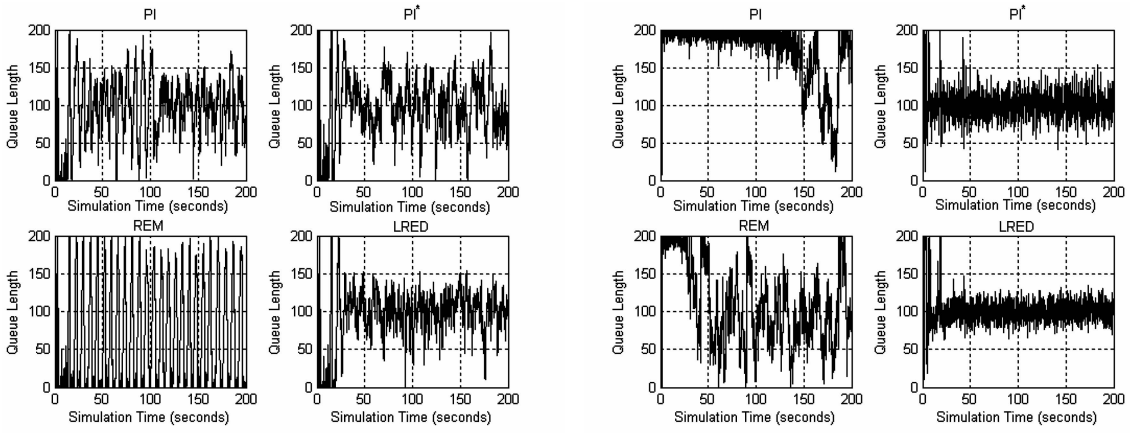


Fig. 6. Experiment 2: Queue length under two extreme cases. (a) Light congestion $(p_0 = 0.0025)$. (b) Heavy congestion $(p_0 = 0.165)$.

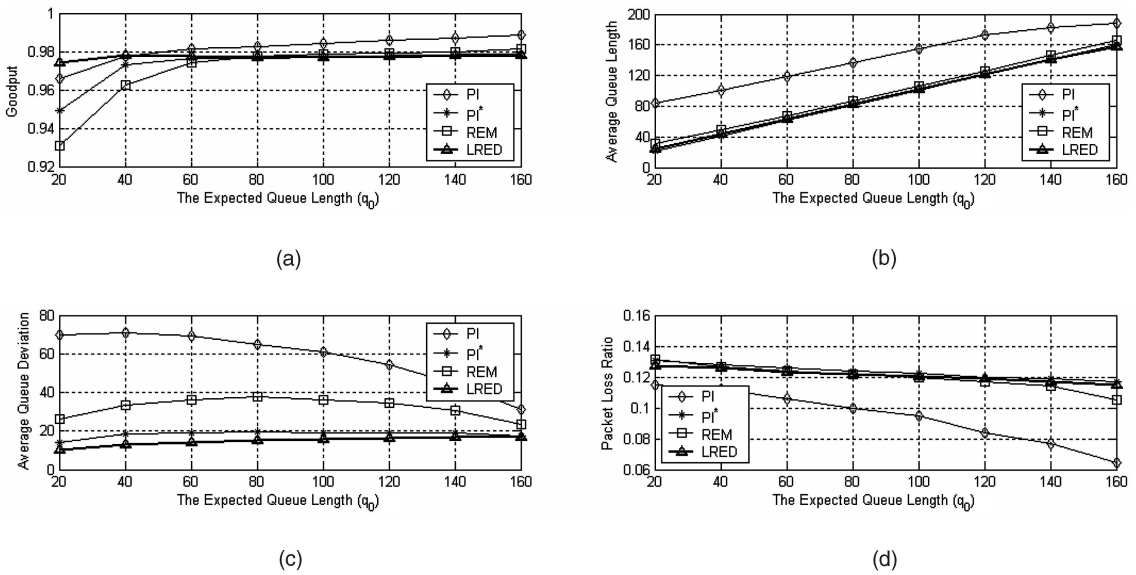


Fig. 7. Experiment 3: Comparisons of (a) goodput, (b) average queue length, (c) average queue deviation, and (d) packet loss ratio.

response leads to longer queue length than the other three schemes (as shown in Fig. 8), which in turn reduces the loss ratio and increases the goodput. On the contrary, LRED, PI*, and REM effectively realize the expected average queue length and, thus, have almost the same packet loss ratio. Compared to PI* and REM, LRED has a higher goodput when q_0 is smaller than 60, as shown in Fig. 7a. Moreover, LRED has the smallest queue deviation. Fig. 8 presents the instantaneous queue lengths for the AQM schemes when q_0 equals 20 and 160. It can be seen that LRED achieves better trade-off between the average queue length and other QoS performance measures, including goodput and loss ratio.

6.2 Nonhomogenous Traffic: Hybrid Flows

Experiment 4: Adding unresponsive UDP flows. In this experiment, we investigate the performance of the AQM schemes with the existence of unresponsive UDP flows. In addition to the 100 persistent TCP flows, we introduce 100 UDP flows arriving in interval [50 sec, 150 sec]. Each is an ON/OFF flow, where the durations of the ON and OFF states are exponentially distributed with a mean of 1.0 second. The density of the UDP traffic over the total traffic, ρ, ranges from 0.1 to 0.9, and the rate of each UDP flow is $r = \rho \times 10\ \mathrm{Mbps}/100$ during the ON period. Other settings are the same as those in Experiment 1. Accordingly, we choose $N^- = 100$ and $R^+ = 450\ \mathrm{ms}$ for PI*.

Fig. 9 presents results of the goodput, average queue length, average queue deviation, and packet loss ratio, as functions of the UDP traffic density. It can be seen that LRED generally outperforms PI, PI*, and REM in all these performance measures, especially when the UDP traffic density is high. Note that REM achieves better performance than PI in this experiment; however, it is stable with quite restricted network conditions only, as shown in Experiment 2.

We also present the instantaneous queue lengths for the AQM schemes in Fig. 10. There are two interesting observations. First, when ρ increases, LRED can still

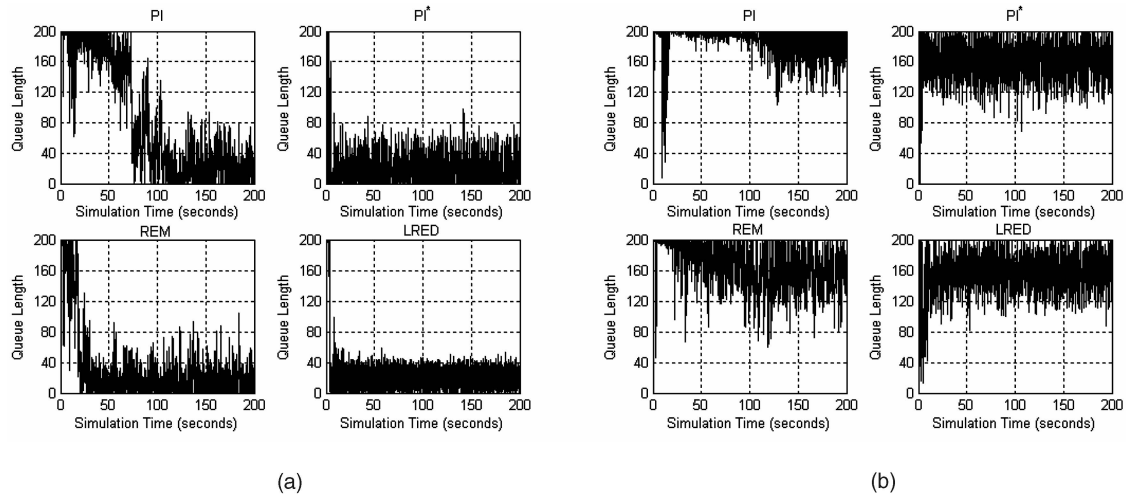


Fig. 8. Experiment 3: Queue length for AQM schemes. (a) $q_0 = 20$. (b) $q_0 = 160$.

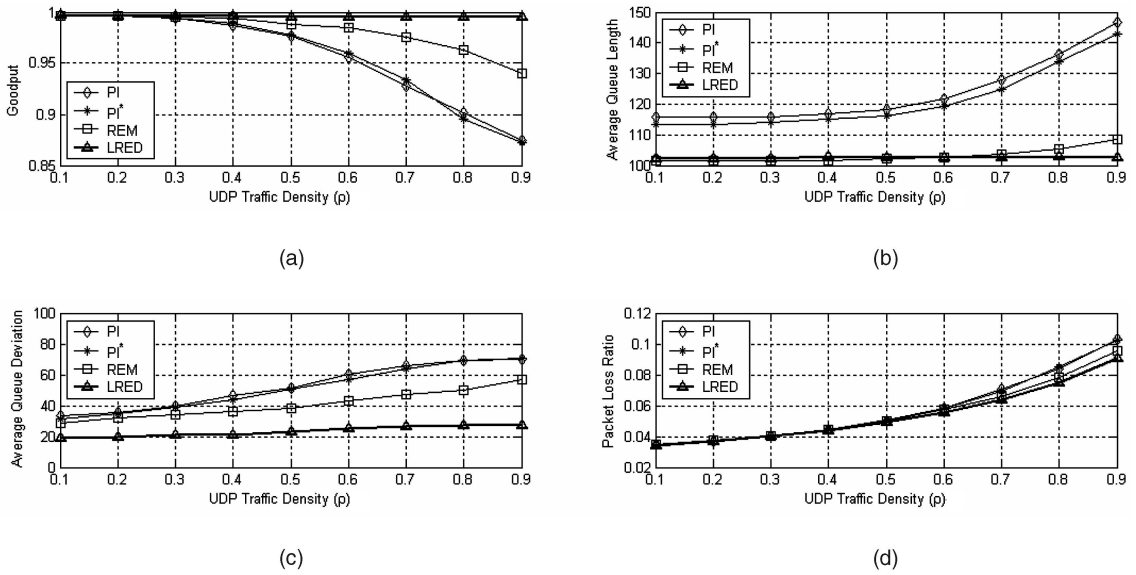


Fig. 9. Experiment 4: Comparisons of (a) goodput, (b) average queue length, (c) average queue deviation, and (d) packet loss ratio.

regulate the queue length to the expected value with much smaller under or overshoots than PI, PI*, and REM. Second, the buffers of PI, PI*, and REM are overflowed (or empty) for a long time when the UDP flows start arriving from 50 seconds (or stops after 150 seconds), especially if ρ is large, i.e., 0.9. This is due to the slow responsiveness of PI, PI*, and REM, which result in low goodput and high loss ratio. On the contrary, there is only a short-term increase (or decrease) at time 50 seconds (or 150 seconds), which implies that LRED has a much shorter response time.

Experiment 5: Adding short-lived tcp flows. Besides unresponsive UDP flows, short-lived TCP flows can also affect the performance of an AQM scheme. In this set of experiments, we introduce short-lived TCP flows, which arrive in intervals [50 sec, 150 sec] following to a Poisson process. The mean arrival rate λ varies from 10 flows/second to 100 flows/second, and the length of each short-lived TCP flow is uniformly distributed in [1.0 sec, 2.0 sec]. Other parameters are the same as those in Experiment 4 and $N^- = 100$ and $R^+ = 450$ ms are still set for PI*.

The average queue lengths, average absolute queue deviations, goodputs, and packet loss ratios for the three AQM schemes in this experiment are compared in Fig. 11. Clearly, LRED outperforms PI, PI*, and REM in all these measures. In addition, Fig. 12 shows the instantaneous queue length for PI, PI*, REM, and LRED for $\lambda = 30$ and 100. LRED is again more stable due to its good responsiveness.

We also compare LRED with AVQ [10] in such a heterogeneous traffic environment. AVQ is known to be effective in regulating the queue length and achieving high link utilization [10]. In the design rules of AVQ in [10], $\alpha < 0.2$ is required to guarantee stability for this scenario ($N^- = 100$, $R^+ = 450$ ms, and $C = 2,500$ packets/seconds). Hence, we let parameter α vary from 0.01 to 0.15 and set the expected utility of AVQ, γ, to 0.98. To make a fair comparison, we also set q_0 in LRED to small values (10 and 20), which match the average queue length in AVQ.

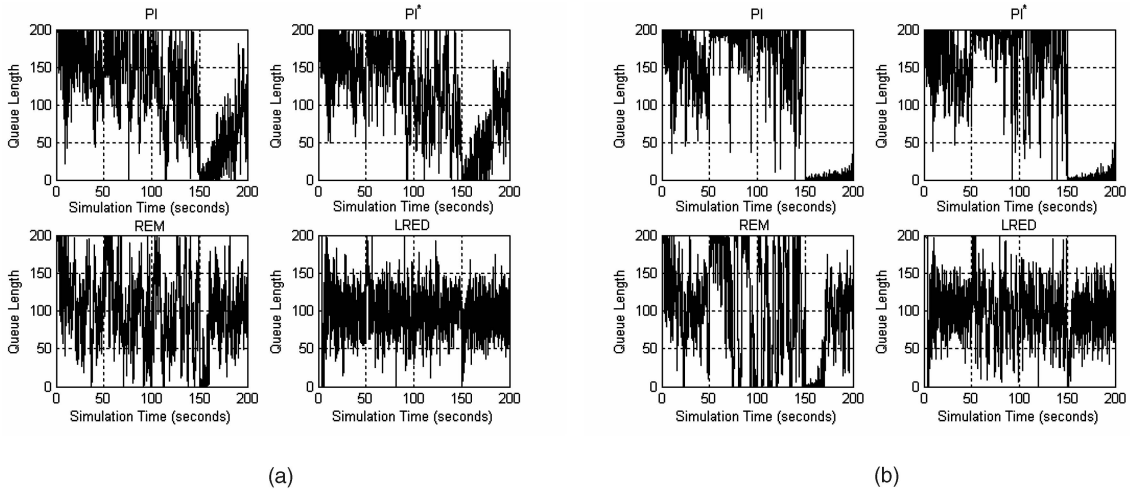


Fig. 10. Experiment 4: Queue lengths for AQM schemes, where ρ is UDP traffic density. (a) UDP traffic density (ρ) is 0.5. (b) UDP traffic density (ρ) is 0.9.

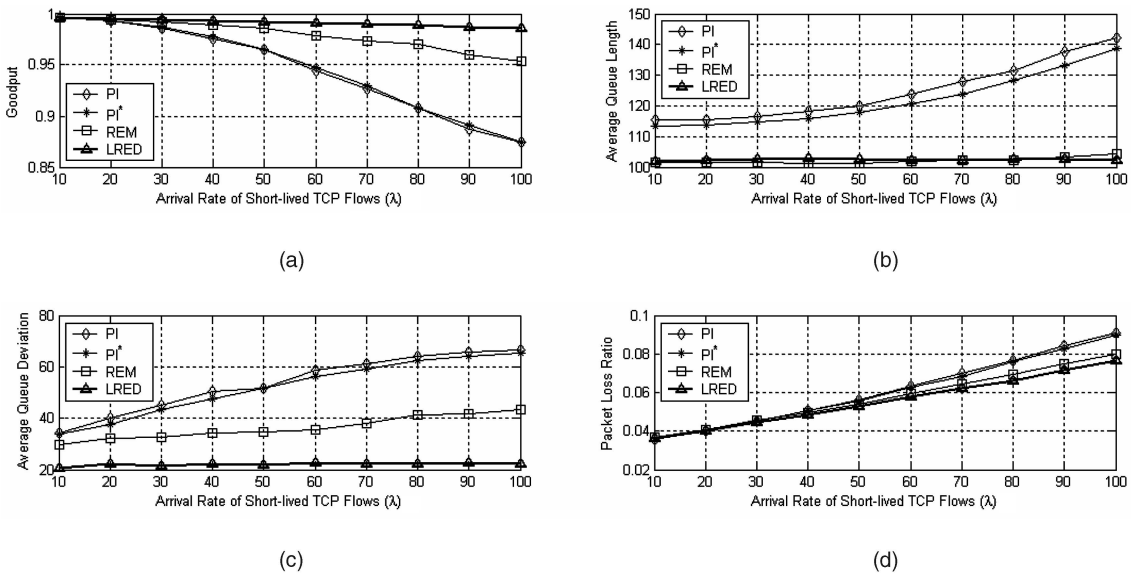


Fig. 11. Experiment 5: Comparisons for PI, REM, and LRED. (a) Goodput, (b) average queue lengths, (c) average queue deviations, and (d) packet loss ratios.

The performance measures are presented in Fig. 13. When α decreases, AVQ achieves higher goodput and lower loss ratio, but larger average queue length as well as queue deviation. When $\lambda \leq 60$, LRED with $q_0 = 20$ is better than AVQ. When $\lambda > 60$, LRED with $q_0 = 10$ outperforms AVQ. It implies that LRED achieves better performance than AVQ, if assigned with a small q_0.

According to [17] and [18], the Pareto distributions of file sizes and durations could contribute to the self-similar characteristics of the Internet traffic. Hence, we have also conducted experiments with the packet interarrival time and flow's duration being set to Pareto distributions, with the same means in the previous experiment. The results are presented in Fig. 14, for PI, PI*, REM, and LRED, respectively. It shows that LRED still has faster response and better performance, and the difference between the results of Poisson and Pareto distributions is generally insignificant. However, it is worth noting that the Pareto distribution is not necessary the best Internet traffic model, particularly considering that the Internet traffic has always been changing with such emerging new technologies and applications as peer-to-peer communications. We expect that, in our future study, more results can be obtained using up-to-date Internet traces or advanced traffic models, e.g., the Fractional Gaussian Noise (FGN) distribution [19].

6.3 Two-Way Traffic: Forward and Reverse Direction

Experiment 6: Two-way traffic and two-way congestion. In this experiment, we introduce two-way traffic: 1) In the reverse direction, namely, from the clients to the servers (see Fig. 4), only short-lived TCP flows with a Poisson arrival process of a 200 flows/second arrival rate are configured. The other parameters related to short-lived TCP flows are the same as that in the Experiment 5. 2) In the forward direction, or from the servers to the clients, there

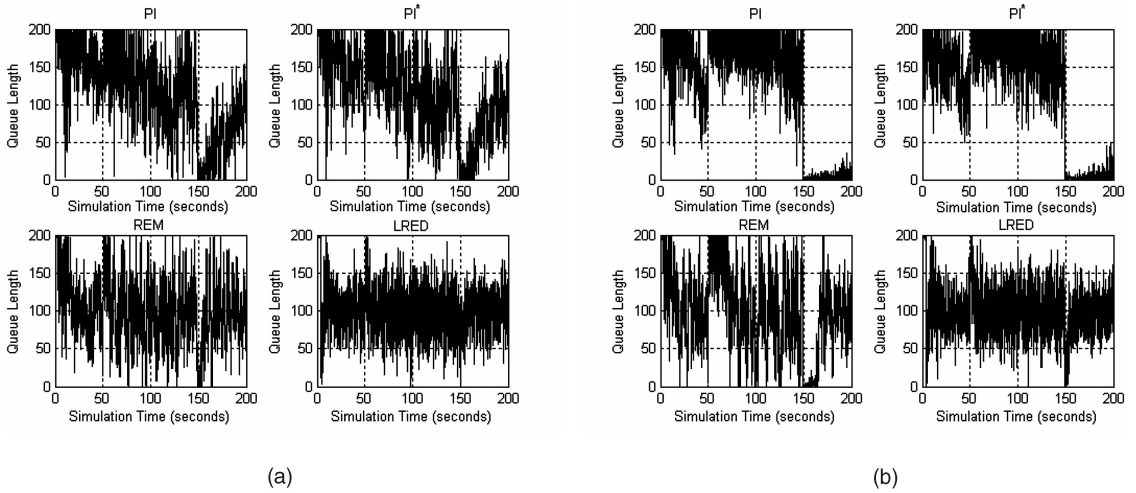


Fig. 12. Experiment 5: Queue lengths for the AQM schemes, where λ is the short-lived TCP arrival rate. (a) $\lambda = 30$ flows/second. (b) $\lambda = 100$ flows/second.

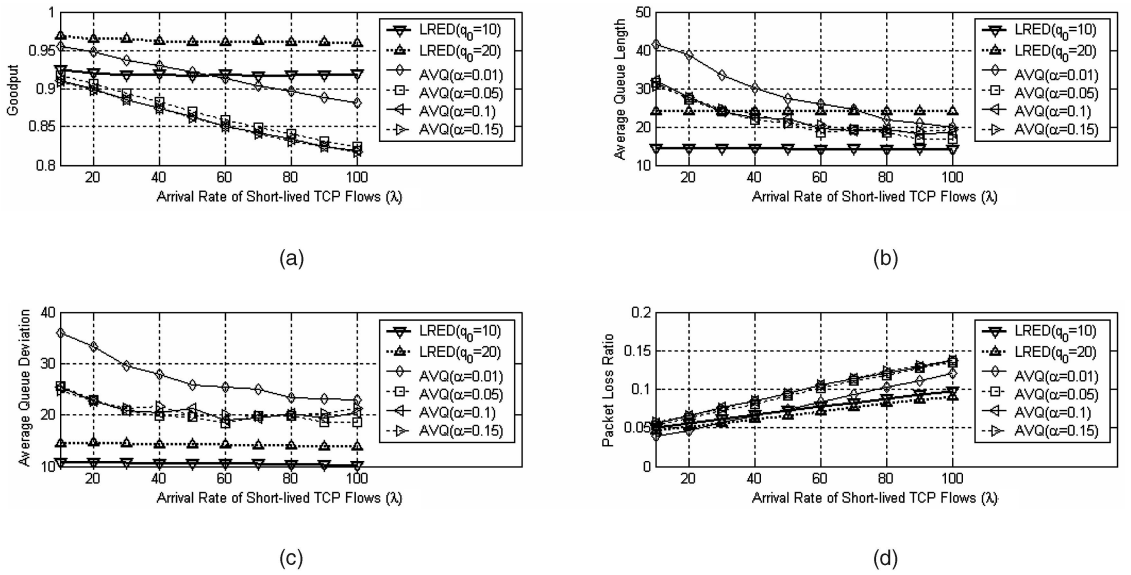


Fig. 13. Experiment 5: Comparisons between LRED and AVQ. (a) Goodput, (b) average queue lengths, (c) average queue deviations, and (d) packet loss ratios.

are 300 persistent TCP flows. We set $N^- = 200$ and $R^+ = 600$ ms for PI*. Note that congestion will occur in both directions of the link connecting Routers 1 and 2 in Fig. 4. We set the simulation time to 100 seconds, which enables 20,000 short-lived TCP flows, and is long enough to discover the performance difference among the AQM schemes (see Figs. 15 and 16).

We collect the queue length for the AQM schemes in both directions of the congested link. In Fig. 15 ($q_0 = 100$ and $Q = 200$), we can see that, all the AQM schemes have noticeable overshoots in this case with bidirectional congestion on the same link; yet, LRED controls the queue length better than others. The forward traffic that consists of persistent TCP flows is influenced more noticeably than the short-lived TCP flows in the reverse direction (Fig. 15b). To avoid buffer overflow (see in Fig. 15b), we increase the buffer size from $Q = 200$ to $Q = 400$. The results are presented in Fig. 16, where both PI and PI* still show slower response and bigger overshoot. Although REM responds more quickly, its queue length is often below the expected value (100), leading to lower goodput. On the contrary, LRED still regulates queue length around the expected value (100) and achieves better control effect than other schemes.

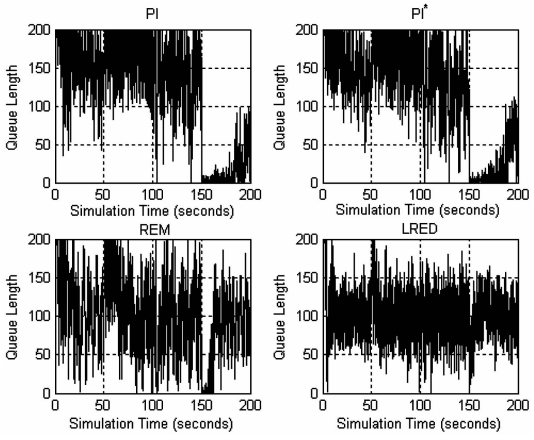


Fig. 14. Experiment 5: Queue lengths for the AQM schemes when the short-lived TCP flows follow a Pareto process ($\lambda = 100$ flows/second).

6.4 Summary

Our simulation results suggest that LRED achieves fast response even under heavy congestion and its performance is quite good in terms of goodput, queue length and queue deviation, and packet loss ratio. PI and REM, however, have slower response, which leads to performance degradation under dynamic network environments. Specifically, their packet dropping probability is iteratively computed (see (32) and (35)); if p_0 is high (e.g., when the RTT R is low and the number of TCP flows N is large), PI and REM need a long time for the drop probability p to converge to p_0. The convergence rate and response time of LRED are almost independent of p_0, as shown in Fig. 5, but mainly influenced by the measurement period. More importantly, when the network is highly dynamic, LRED can quickly converge to a new stable state through its multigranular update.

LRED relies on the measured packet loss ratio and, therefore, the parameters involved in measurement should be carefully configured. Specifically, the measurement weight (w_m) in (2) should be set to a small value in order to capture the latest packet loss. Regarding measurement period (t_m), if it is too small, the measurement could be inaccurate; but if it is too big, the response time can be longer. Our experience shows that $w_m = 0.1$ and $t_m = 1.0$ are reasonable choices in most scenarios.

Another parameter in LRED is β, whose guideline is given by Theorem 2. Note that, if β is too big, the packet drop probability calculated by (3) will be either bigger than 1 (when $q > q_0$) or smaller than 0 (when $q < q_0$). In this case, LRED behaves like a virtual Tail-Drop with a virtual buffer size of q_0. Fig. 17 presents the simulated results for such a scenario, where $\beta = 10$. It can be seen that the packet drop probability almost equals 1 or 0, and the queue length stays below 100, like in a traditional Tail-Drop buffer.

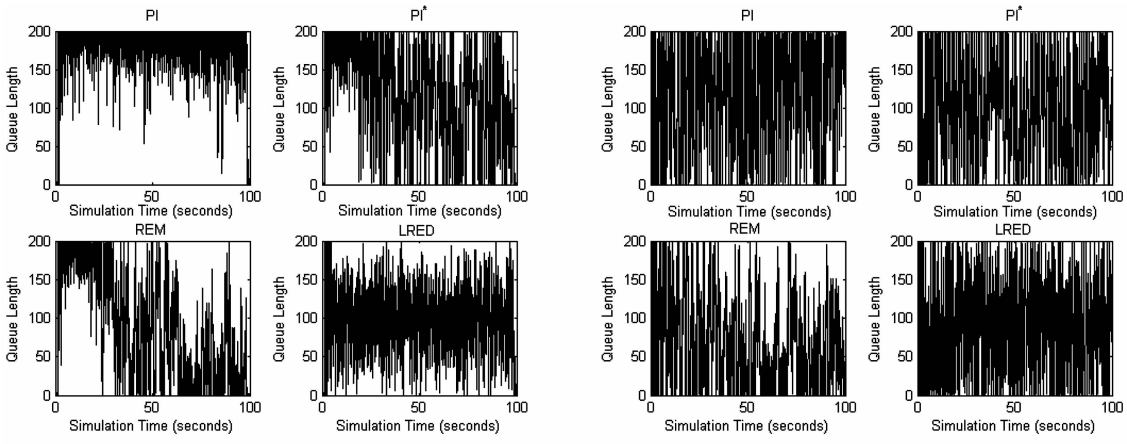


Fig. 15. Experiment 5: Queue lengths for the AQM schemes ($q_0 = 100$, $Q = 200$). (a) The reverse direction (Short-lived TCPs). (b) The forward direction (Long-lived TCPs).

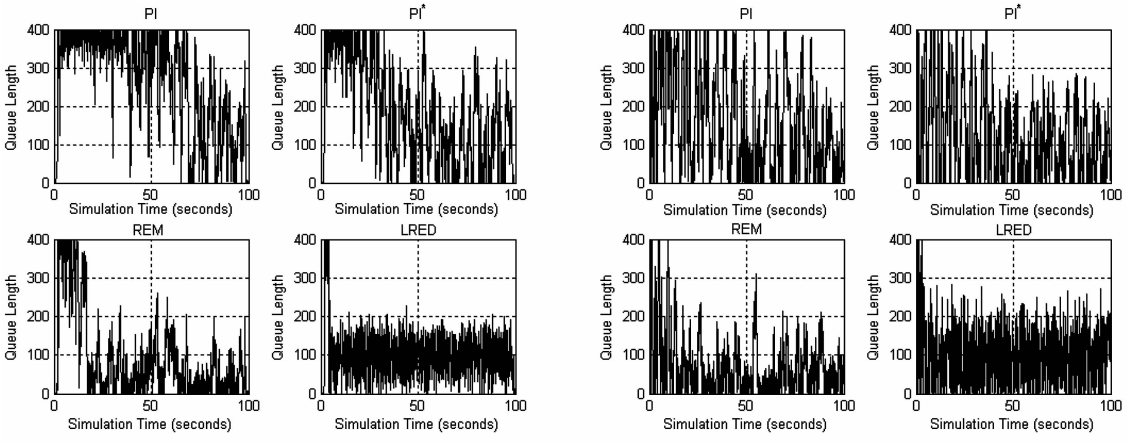


Fig. 16. Experiment 5: Queue lengths for the AQM schemes ($q_0 = 100$, $Q = 400$). (a) The reverse direction (Short-lived TCPs). (b) The forward direction (Long-lived TCPs).

7 Conclusions and Future Works

In this paper, we have proposed a novel AQM algorithm, LRED, which incorporates packet loss ratio as a complement to queue length for congestion estimation. In LRED, the packet drop probability is updated over multiple grains: on a fine grain, LRED uses the instantaneous queue length mismatch to update the drop probability upon each packet arrival; on a coarse grain, LRED adjusts the drop probability according to the packet loss ratio, which has never been considered in existing AQM algorithms. We have developed an analytical model for LRED, which suggests that this multigranular update improves not only the stability of the system, but also its responsiveness. Such observations have been validated by our simulation results under various configurations. We have also compared LRED with existing AQM algorithms, including PI, REM, and AVQ. Our results have showed that LRED remains quite stable when the number of TCP flows and round-trip times vary significantly, or when many short-lived flows or unresponsive UDP flows exist in the network. Moreover, it can effectively control the queue to the expected length, and achieves a better trade-off between the goodput and queue length. Finally, LRED achieves reasonably good performance under two-way traffic.

There are many possible future works for enhancing the LRED algorithm. We are particularly interested in extending

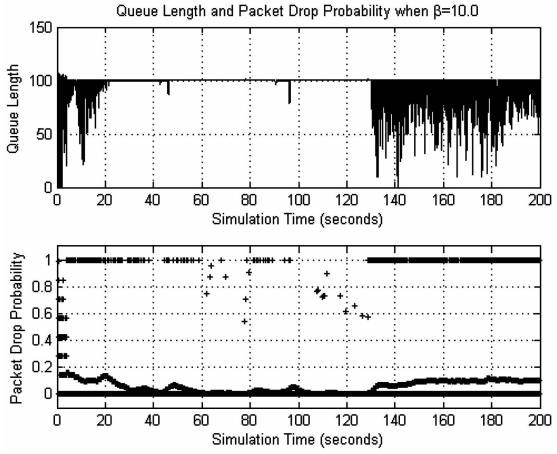


Fig. 17. LRED dynamics with a large β ($N = 400$, $q_0 = 100$).

LRED to support differentiated QoS, where packets with different priority may have nonuniform dropping probabilities. Meanwhile a quantitative analysis of LRED's response time is very important to more precisely evaluate LRED's performance. The effect of substituting packet dropping with packet marking is also worth investigating. Another challenging work is to model LRED's performance for short-lived TCP flows, which will complement our existing analytical results with long-lived flows. Finally, we are interested in examining the performance of LRED under more realistic Internet traffic traces, or more sophisticated traffic models that reflects the recent Internet development, e.g., Fractional Gaussian Noise (FGN) distribution [19].

ACKNOWLEDGMENTS

The research was supported in part by grants from RGC under the contracts HKUST6104/04E, HKUST6165/05E, and HKUST6164/06E, and by grants from NSF China under the contracts 60429202 and 60573115. J. Liu's work was supported in part by a Canadian NSERC Discovery Grant 288325, an NSERC Research Tools and Instruments Grant, a Canada Foundation for Innovation (CFI) New Opportunities Grant, and an SFU President's Research Grant.

REFERENCES

[1] R. Gurin and V. Peris, "Quality-of-Service in Packet Networks: Basic Mechanisms and Directions," *Computer Networks,* vol. 31, no. 3, pp. 169-179, Feb. 1999.
[2] Y.-T. Hou, D. Wu, B. Li, T. Hamada, I. Ahmad, and H. J. Chao, "A Differentiated Services Architecture for Multimedia Streaming in Next Generation Internet," *Computer Networks,* vol. 32, no. 2, pp. 185-209, Feb. 2000.
[3] B. Braden et al., "Recommendations on Queue Management and Congestion Avoidance in the Internet," *IETF RFC2309,* Apr. 1998.
[4] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Comm. Rev.,* vol. 24, pp. 10-23, Oct. 1994.
[5] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Networking,* vol. 1, no. 4, pp. 397-413, Aug. 1993.
[6] S. Floyd, "Recommendation on Using the "Gentle_" Variant of RED," http://www.icir.org/floyd/red/gentle.html, Mar. 2000.
[7] W. Feng, D.D. Kandlur, D. Saha, and K.G. Shin, "A Self-Configuring RED Gateway," *Proc. IEEE INFOCOM,* vol. 3, pp. 1320-1328, Mar. 1999.
[8] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," http://www.icir.org/floyd/papers/adaptiveRed.pdf, Aug. 2001.
[9] W. Feng, D.D. Kandlur, D. Saha, and K.G. Shin, "The Blue Active Queue Management Algorithms," *IEEE/ACM Trans. Networking,* vol. 10, no. 4, pp. 513-528, Aug. 2002.
[10] S. Kunniyur and R. Srikant, "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," *Proc. ACM SIGCOMM,* pp. 123-134, Aug. 2001.
[11] C. Hollot, V. Misra, D. Towsley, and W. Gong, "A Control Theoretic Analysis of RED," *Proc. IEEE INFOCOM,* vol. 3, pp. 1510-1519, Apr. 2001.
[12] C. Hollot, V. Misra, D. Towsley, and W. Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows," *Proc. IEEE INFOCOM,* vol. 3, pp. 1726-1734, Apr. 2001.
[13] S. Athuraliya, S. Low, V. Li, and Q. Yin, "REM: Active Queue Management," *IEEE Network Magazine,* vol. 15, pp. 48-53, May 2001.
[14] Y. Gao and J.C. Hou, "A State Feedback Control Approach to Stabilizing Queues for ECN-Enabled TCP Flows," *Proc. IEEE INFOCOM,* vol. 3, pp. 2301-2311, Mar. 2003.
[15] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and Its Empirical Validation," *Proc. ACM SIGCOMM,* pp. 303-314, Aug. 1998.
[16] *Network Simulator-NS2,* http://www.isi.edu/nsnam/ns/, 2005.
[17] M.E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Trans. Networking,* vol. 5, no. 6, pp. 835-846, Dec. 1997.
[18] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," *Proc. ACM SIGMETRICS,* pp. 151-160, June 1998.
[19] T. Karagiannis, M. Molle, and M. Faloutsos, "Long-Range Dependence Ten Years of Internet Traffic Modeling," *IEEE Internet Computing,* vol. 8, no. 5, pp. 57-64, Sep.-Oct. 2004.

Chonggang Wang (S'00-M'04) received the BEng degree (cum laude) from Northwestern Polytechnic University (NPU), Xi'an, China, in 1996, and the MS and PhD degrees in communication and information systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, and Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999 and 2002, respectively. From September 2002 to November 2003, he was a research associate with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, P.R. China. He has been a postdoctoral research fellow with the University of Arkansas, Fayetteville, since July 2004. He is a corecipient of 2004 National Awards for Science and Technology Advancement in Telecommunications. His current research interests include wireless sensor networks, wireless mesh networks, wireless/mobile networks, and congestion and flow control. He is a member of the IEEE.

Jiangchuan Liu (S'01-M'03) received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is currently an assistant professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada, and was an assistant professor at The Chinese University of Hong Kong from 2003 to 2004. He was a recipient of Microsoft Research Fellowship (2000), a recipient of the Hong Kong Young Scientist Award (2003), and a coinventor of one European patent and two US patents. He won first-class honors in several regional and national programming contests. His research interests include Internet architecture and protocols, media streaming, wireless ad hoc networks, and service overlay networks. He serves as TPC member for various networking conferences, including IEEE INFOCOM, IEEE MASS, and IWQoS. He was TPC cochair for The First IEEE International Workshop on Multimedia Systems and Networking (WMSN '05), Information System cochair for IEEE INFOCOM '04, and a guest-editor for *ACM/Kluwer Journal of Mobile Networks and Applications* (*MONET*), special issue on energy constraints and lifetime performance in wireless sensor networks. He is an editor of *IEEE Communications Surveys and Tutorials*. He is a member of the IEEE and the IEEE Communications Society, and an elected member of Sigma Xi.

Bo Li (S'89-M'92-SM'99) received the BEng (summa cum laude) and MEng degrees in computer science from Tsinghua University, Beijing, in 1987 and 1989, respectively, and the PhD degree in electrical and computer engineering from the University of Massachusetts at Amherst in 1993. Between 1993 and 1996, he worked on high-performance routers and ATM switches at IBM Networking System Division, Research Triangle Park, North Carolina. Since 1996, he has been with the Department of Computer Science, Hong Kong University of Science and Technology. Since 1999, he has also held an adjunct researcher position at the Microsoft Research Asia (MSRA), Beijing, China. His current research interests are on adaptive video multicast, peer-to-peer streaming, and resource management in mobile wireless systems, across layer design in multihop wireless networks, content distribution and replication. He has published 80 journal papers and held several patents in above areas. He received the Outstanding Young Investigator Award by the National Natural Science Foundation of China (NSFC) in 2004. He has been on editorial boards of the *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Vehicular Technology*, *ACM/Kluwer Journal of Wireless Networks (WINET)*, *IEEE Journal of Selected Areas in Communications (JSAC)*-wireless communications series, *ACM Mobile Computing and Communications Review (MC2R)*, *Elsevier Ad Hoc Networks*, *SPIE/Kluwer Optical Networking Magazine (ONM)*, and *KICS/IEEE Journal of Communications and Networks (JCN)*. He served as a guest editor for *IEEE Communications Magazine* special issue on active, programmable, and mobile code networking (April 2000), *ACM Performance Evaluation Review* special issue on mobile computing (December 2000), and *SPIE/Kluwer Optical Networks Magazine* special issue on wavelength routed networks: architecture, protocols, and experiments (January/February 2002), *IEEE Journal on Selected Areas in Communications* special issue on protocols for next generation optical WDM networks (October 2000), special issue on recent advances in service-overlay network (January 2004), and special issue on quality of service delivery in variable topology networks (September 2004), and *ACM/Kluwer Mobile Networks and Applications (MONET)* special issue on energy constraints and lifetime performance in wireless sensor networks (second quarter of 2005). In addition, he has been involved in organizing more than 40 conferences, especially IEEE Infocom since 1996. He was the co-TPC Chair for IEEE Infocom 2004. He is a senior member of the IEEE.

Kazem Sohraby received the BS (highest distinction), MS, and PhD degrees, all in electrical engineering, took graduate courses in computer science at the George Washington University, and received the MBA degree from the Wharton School, University of Pennsylvania. He is a professor and head of the Department of Computer Science and Computer Engineering at the University of Arkansas at Fayetteville. He also serves as a principal consultant on contracts with the Defense Information Systems Agency (DISA). His areas of interest include computer networking, signaling, switching, performance analysis, and traffic theory. Prior to joining Bell Labs, Dr. Sohraby served as the head of Network Design and Planning Department at Computer Sciences Corporation, Systems Division (1978-1983). He has a total of more than 20 patent applications (14 granted) on computer protocols, wireless and optical systems, circuit and packet switching, and optical Internet. He has several publications, including a book on the performance and control of computer communications networks. He is a distinguished lecturer of the IEEE Communications Society, and director of IEEE Communications Society (online content), and served as its president's representative on the Committee on Communications and Information Policy (CCIP). He served as chair of several conferences in both the IEEE and ACM, ACM Mobicom special interest group, and is vice chair of the IEEE Infocom executive committee. He also served on the education committee of the IEEE Communications Society, is on the editorial boards of several publications, and panelist and reviewer with the US National Science Foundation. He served as grant referee and reviewer with the US Army, and the Natural Sciences and Engineering Research Council of Canada. He is a senior member of the IEEE.

Y. Thomas Hou (S'91-M'98-SM'04) received the BE degree from the City College of New York in 1991, the MS degree from Columbia University in 1993, and the PhD degree from Polytechnic University, Brooklyn, New York, in 1998, all in electrical engineering. From 1997 to 2002, he was a researcher at Fujitsu Laboratories of America, IP Networking Research Department, Sunnyvale, California. Since the Fall of 2002, he has been an assistant professor at Virginia Tech, the Bradley Department of Electrical and Computer Engineering, Blacksburg, Virginia. His current research interests include resource (spectrum) management and networking issues for SDR-enabled wireless networks, optimization and algorithm design for wireless ad hoc and sensor networks, and video communications over dynamic ad hoc networks. In the recent past, he also worked on scalable architectures, protocols, and implementations for differentiated services Internet, service overlay networking, video streaming, and network bandwidth allocation policies and distributed flow control algorithms. He has published more than 100 journal and conference papers in the above areas and is a recipient of the 2004 IEEE Communications Society Multimedia Communications Best Paper Award, the 2002 IEEE International Conference on Network Protocols (ICNP) Best Paper Award, and the 2001 *IEEE Transactions on Circuits and Systems for Video Technology (CSVT)* Best Paper Award. He is a member of the ACM and a senior member of the IEEE.