Performance Evaluation of Active Queue Management Using a Hybrid Approach

Chin-Ling Chen* Chia-Chun Yu
Department of Information Management, National Pingtung Institute of Commerce, Taiwan
E-mail: clchen@mail.npic.edu.tw

Abstract— **A hybrid approach, namely, Proportional Rate-based Control (PRC), is proposed to maintain the queue length around the target level by 1) dropping the packet whose arrival rate exceeding maximum desired rate; 2) using virtual queue to control the packet whose arrival rate above minimum desired rate. We compare the performance of PRC through simulation with several well-known AQM schemes such as RED, DRED, SRED, AVQ and VRC. The study shows that the proposed scheme is more effective at stabilizing the queue size and has lower loss rate than others. Another appealing feature of the proposed scheme is that the system responds quickly and accurately to changing network conditions.**

Index Terms—**active queue management, rate-based control and queue regulation**

I. INTRODUCTION

In recent years, much effort has been concentrated on Active Queue Management (AQM) research. AQM has been proposed as a solution for detecting incoming congestion early and delivering the congestion notification to the source, so that the source can reduce the sending rate to avoid possible buffer overflow. Many methods were proposed to achieve an efficient and stable situation. We may categorize them into two main approaches: queue-based [1-7] and rate-based control [8-14]. Random Early Detection (RED) [1], the emerging concept of queue-based control, has been proposed as simple solution to the AQM problem. RED maintains an exponentially weighted moving average of queue length to detect the incoming possible congestions. If the average queue size is larger than a set maximum threshold, the incoming packets are marked with an Explicit Congestion Notification (ECN) bit to notify the sources that the router is experiencing congestion. Traffic sources then respond to the ECN by adjusting their sending rate. RED is known to suffer from problems like parameter sensitivity and inability to marking packets as far apart as possible, thus resulting in instability and unfairness. Furthermore, ECN-delivering requires packet-discarding in backlog queue. The heavier the load becomes, the more feedback signals have sent. The load-dependent property incurs higher delay.

Many approaches have been proposed to help to improve the efficiency of RED-like algorithms. Like RED, both SRED [2] and DRED [3] preemptively discard packets with a load-dependent probability. SRED calculates the number of active flows to help to stabilize the queue length. However, the equilibrium queue length strongly depends on the number of active TCP connections. DRED, on the other hand, uses a simple control-theoretic approach to stabilize queue occupancy at a level independent of the number of active TCP connections. DRED also allows transient traffic bursts to be queued without unnecessary packet drops. However, the control gain selection is based on empirical investigation and simulation analysis. No efficient method for dynamically and optimally adapting control gains has been proposed. Dynamic-RED [4] investigate the issue of stabilizing the queue length at a given target from a model-based and control theoretic standpoint. This study evaluates the stability range of the gain, and provides guidelines for control gain selection to stabilize queue occupancy, and thus improves the performance of DRED. Different from the previous studies whose focus on designing another queuing mechanism or enhancing RED core, KRED [5] uses a Kohonen neural network to compute the optimal parameters to achieve stable queue length. Hybrid random early detection (HRED) [6] combines the more effective elements of recent algorithms with a RED core. HRED maps instantaneous queue length to a drop probability, automatically adjusting the slope and intercept of the mapping function to account for changes in traffic load and to keep queue length within the desired operating range.

StoRED [7], based on RED scheme and inspired by the well known stochastic fair queuing, is especially designed for a variety of applications which need to prevent unresponsive flows from overwhelming others. StoRED tunes the packet dropping probability of RED for all the flows by considering the bandwidth share obtained by the flow. The dropping probability is adjusted such that the packets of the flow with higher transmission rate will more likely be dropped than flows with lower rate. StoRED applies time-varying hashing function to map flows to different counting bins whose dropping probabilities are proportional to their traffic loads. Misbehaving flows tend to be assigned heavily loaded bins. StoRED is able to discipline misbehaving flows and achieve adjustable fairness.

RED and its variants are known to suffer from problems like parameter sensitivity and inability to capture input traffic load fluctuations accurately, thereby resulting in instability. Actually, queue length is

---

Manuscript received January 5, 2011; accepted May 6, 2011.
This paper is supported by National Science Council, Taiwan. The project number is NSC 94-2213-E-251-006-

doi:10.4304/jcp.7.5.1196-1203

insensitivity to traffic load variation. Rate-based control, on the other hand, estimate congestion status by monitoring the link load directly. We may further classify rate-based control into three approaches: Virtual Rate Control (VRC) [8], Random Exponential Marking (REM) [9-10] and Adaptive Virtual Queue (AVQ) [11-14]. VRC algorithm [8] has been proposed to tightly regulate the queue and to achieve high utilization with small packet loss, under various network conditions. VRC attempts to maintain the equilibrium input rate near the link capacity and to stabilize the queue length. VRC is a proportional-integral-derivative (PID) controller. It has been shown that rate-based control exhibits faster response than queue-length-based control. REM [9] maintains a price associated with a link that is updated periodically. The control parameters of price include queue length, target queue length, aggregate input rate and available bandwidth. The marking/dropping probability is a function of price. With ECN, the receiver responds the marks to its source and the source is expected to reduce its sending rate accordingly. REM is demonstrated to outperform RED and DropTail [9]. However, REM suffers problems such as sluggish response time and high queue length jitter. AREM [10], thereby, is introduced to improve the performance of REM by 1) adaptively controlling the parameter to achieve fast response, and 2) evaluating marking/ dropping probability with each packet arrival to reduce the queue length jitter. AREM maintains queue length stability independently of traffic loads, round trip propagation delay, and bottleneck capacity.

AVQ [11] replaces the marking probability calculation with the computation of the capacity of a virtual queue. When a packet arrives in the real queue, the virtual queue is also updated to reflect a new arrival. The virtual capacity at each link is then modified such that total flows entering each link achieve the desired capability of the link. AVQ offers stability in the presence of feedback delays as well as the ability to maintain variation in queue length. However, AVQ cannot maintain the stable condition as link capacity scales upward. Scalable AVQ [12], therefore, has been proposed to obtain scalability by decoupling the control parameter from network condition parameters, such as link capacity, number of sources and propagation delay, and simply tuning the control parameters. SAVQ [13] has argued that there is a trade-off between high level of utilization and small queue length. It is difficult to achieve a fast system response and high link utilization simultaneously using a constant desired utilization value. An adaptive setting method for desired utilization value, hereafter, is proposed according to the instantaneous queue size and the given reference queue value. SAVQ stabilizes dynamics of queue while maintaining high link utilization. Virtual Queue and Rate-based (VQR) scheme [14] uses the virtual queue size and the aggregate flow rate to compute the marking probability. The VQR has proved the local stable condition for TCP based network with an arbitrary topology. However, this study has not presented any simulation to validate the system.

As mentioned above, an ideal AQM scheme must achieve the following objectives:

- Stability: The scheme must reduce queue oscillations and stabilize the queue length in terms of number of connections. The benefits of stabilized queues in a network are bounded delays and more certain buffer provisioning.
- Responsiveness: The scheme must respond quickly and accurately to changing network conditions.
- Efficiency: The scheme must maintain desired throughput (or desired utilization) and low loss rate over a wide range of load levels.

Most of the AQM schemes are designed to achieve one objective or at most two objectives at the same time. In this paper, a new hybrid algorithm, proportional rate-based control (PRC), is proposed to achieve all goals. We set up two desired utilization parameters, maximum and minimum threshold, to effectively control input rate under the desired level. We also incorporate the virtual queue into the proposed system to help regulate the queue size. The virtual queue aims to maintain high network utilization and does not further penalize source which are in the process of backing off in response to previous packet drops. The proposed scheme detects congestion based on both of the arrival rate of the packet and the virtual queue length. Two approaches of the algorithm are in mind: 1) drop the packet whose arrival rate above maximum desired rate; 2) use virtual queue to control the packet whose arrival rate exceeding minimum desired rate. Dropping alleviates the current or incoming network congestion promptly, but it might lead to large oscillatory behavior. On the other hand, using virtual queue might stabilize the queue behavior, but lead to more losses than desired. The PRC is shown to have smaller queue length and have lower loss rate in most cases when compared to other schemes. Adapting too fast may make the system respond quickly to changing network environment. However, it might result in large queue length oscillation. On the other hand, adapting too slowly might lead to more losses than desired, resulting in lower throughput. Motivated by the above analysis, the proposed scheme is designed to achieve fast adaptation while maintaining the stability of the system. To make readers more understand the queue control and the meaning of simulation parameters, we describe several well-known AQM equations of queue controller such as RED, DRED, SRED, AVQ and VRC. Table 1 compares them with various congestion indicators.

The rest of this paper is organized as follows. In Section 2, the rate-based control and its analysis are described. The simulation results of comparison between the existing well-known AVQ mechanisms and the PRC are shown in Section 3. Finally, Section 4 concludes this paper.

Table 1: Comparison of several well-known AQMs

congestion indicators	RED	SRED	DRED	VRC	AVQ	PRC
average queue size	∨					
instantaneous queue length		∨	∨		∨	
link utilization				∨	∨	∨
number of active flows	∨	∨				
input rate				∨		∨

II. PROPORTIONAL RATE-BASED CONTROL

Suppose r_{max} is the maximum desired arrival rate and r_{min} is the minimum arrival rate to trigger virtual queue. We define two desired utilization parameters, $\bullet_{max}$ and $\bullet_{min}$, where $0 < \bullet_{min} < \bullet_{max} < 1$. We have

$$r_{max} = C \times \bullet_{max} \tag{1}$$

$$r_{min} = C \times \bullet_{min} \tag{2}$$

where C is the link capacity.

Suppose the physical queue capacity is Q and the current queue size is q. The size of virtual queue (VQ) is less than the physical queue capacity. We denote k as the threshold of the virtual queue, where $0 < k < 1$. The capacity of the virtual queue can be described as:

$$VQ = Q \times k - q \tag{3}$$

At initialization, the current queue size q is set to be zero and the virtual queue is disabled. When a packet arrives in the real queue, the virtual queue is also updated to reflect the new arrival. Packets in the virtual queue are discarded when the virtual queue overflows. The proposed scheme allows for dropping packets before physical queue overflows. The setting of k is used to help detecting incipient congestion.

Fig. 1 depicts the component diagram. A rate estimator uses a packet list (Fig. 2) to calculate the arrival rate λ. Assume the maximum packets to be monitored in a packet list during one time unit is *MaxListQty*. The packet list consists of m recently seen packets, where $m \leq$ *MaxListQty*.

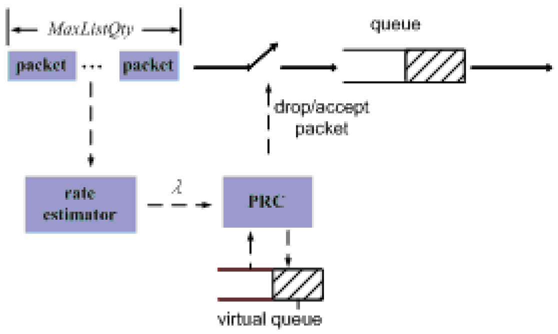


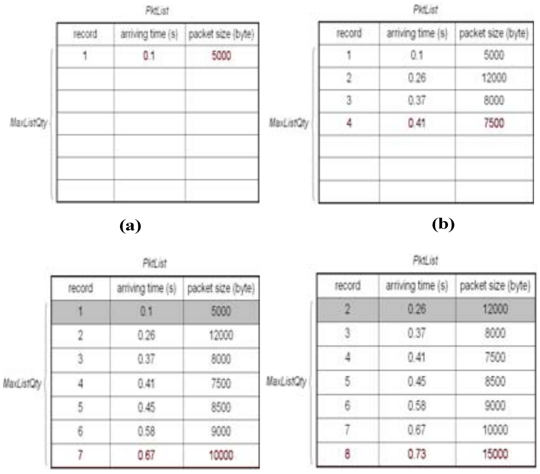
Figure 1. Component diagram

Each packet contains 2-tuple information (arrival time, packet size). Let $pkt(i).time$ and $pkt(i).size$ represent the arrival time and packet size of ith incoming packet, individually. The packet list starts out empty. As one packet arrives, as long as the list is not full ($m <$ *MaxListQty*), its timestamp is set to be the arrival time of the packet. Once the packet list is full, the first packet in the list must be removed to maintain $m <$ *MaxListQty*. The arrival time λ is given as follows.

$$\lambda = \frac{\sum_{i=1}^{m} pkt(i).size}{pkt(m).time - pkt(1).time} \tag{4}$$

Fig. 2 has the examples of arrival rate calculation. If λ is greater than r_{max}, the packet is discarded. If λ is between r_{max} and r_{min}, the virtual queue is enabled and the value of VQ is set to be $Q \times k - q$. Let b be the current packet size. If b is greater than VQ, the packet is dropped. Otherwise, the packet is admitted in physical queue and the value of VQ is updated with the current packet size b. The virtual queue is adapted to achieve the desired utilization and control queue oscillation within the desired range [11]. If λ is less than r_{min}, the virtual queue is disabled and the packet is allowed to the physical queue directly. The queue occupation is controlled under $Q \times k$. The PRC algorithm is described as follows:

```
At each packet arrival epoch do
IF $q > 0$
    {
        IF  $\lambda > r_{max}$
            {drop packet;}
        ELSE IF  $\lambda > r_{min}$
            {$VQ = (Q \times k) - q$;
            $VQ = VQ - b$;
            IF $VQ < 0$
                {drop packet;}
            ELSE
                {enqueue packet;}
            }
        ELSE  //input rate is less than minimum threshold
            {enqueue packet;}
    }
ELSE
    {enqueue packet;}
```



(a)

record	arriving time (s)	packet size (byte)
1	0.1	5000

(b)

record	arriving time (s)	packet size (byte)
1	0.1	5000
2	0.26	12000
3	0.37	8000
4	0.41	7500

(c)

record	arriving time (s)	packet size (byte)
1	0.1	5000
2	0.26	12000
3	0.37	8000
4	0.41	7500
5	0.45	8500
6	0.58	9000
7	0.67	10000

(d)

record	arriving time (s)	packet size (byte)
2	0.26	12000
3	0.37	8000
4	0.41	7500
5	0.45	8500
6	0.58	9000
7	0.67	10000
8	0.73	15000

Figure 2. Arrival rate calculation (a) $\lambda = 400$Kb/s (b) $\lambda = 839$Kb/s (c) $\lambda = 842$Kb/s (d) $\lambda = 1.12$Mb/s

Let $Q(t)$ be the queue size at time t. Suppose $A(t)$ is the data received by the router over $[0, t]$ and $F(t)$ is the cumulative data forwarded by the router over [0, *t*]. The difference $A(t)–F(t)$ is the queue size $Q(t)$ at time *t*. We have

$$Q(t)=\begin{cases}0, & \text{if } F(t)\geq A(t)\\ A(t)-F(t), & \text{if } F(t)<A(t)\end{cases} \tag{5}$$

$Q(t)$ changes depending on the difference between arrival rate and outgoing rate up to time *t*. Suppose $a(t)$ is the aggregate arrival rate and $f(t)$ is the aggregate departure rate at time *t*. The quantity, $a(t)–f(t)$, is the rate at which the queue grows. We can approximate $a(t)–f(t)$ to be the change in queue length $Q(t)$. Any instance of the input rate exceeding the output rate should increase the queue length which may lead to queue overflow. Since a considerable amount of time may elapse between the packet drop at the router and its detection at the source, the source continues to transmit at a rate that the router cannot support, thus resulting in a large amount of packet loss. The sources reduce their sending rate only after detecting the packet loss. The queue lengths grow and shrink rapidly due to the TCP congestion control algorithm. Using our scheme, a TCP flow can achieve stable queue oscillation with only a small amount of packet loss. Suppose $s(t)$ is the smoothed rate at time *t*. We define the smoothed rate to be the arrival rate regulated by the proposed PRC algorithm. To reduce the variability of queue size, we have

$$s(t)=\begin{cases}r_{\max}, & \text{if } a(t)\geq r_{\max}\\ a(t), & \text{if } a(t)<r_{\max}\end{cases} \tag{6}$$

If $a(t) \geq r_{max}$, the amount, $a(t)–r_{max}$, of the data at time *t* must be discarded and $s(t)$ is set to be r_{max}. Otherwise, $s(t)$ is set to be $a(t)$. Let $S(t)$ be the cumulative data regulated by the PRC up to time *t*. We have $Q(t) = S(t)–F(t)$. The queue length becomes empty when $S(t) \leq F(t)$.

Fig. 3 illustrates that the queue length is well-controlled by the PRC. Assume $S^*(t)$ is a straight line with slope r_{max} at a given point (t_0, d_0). We have

$$S^*(t)=r_{\max}(t-t_0)+d_0 \tag{7}$$

$$\hat{Q}(t)=S^*(t)-F(t)=r_{\max}(t-t_0)+d_0-F(t) \tag{8}$$

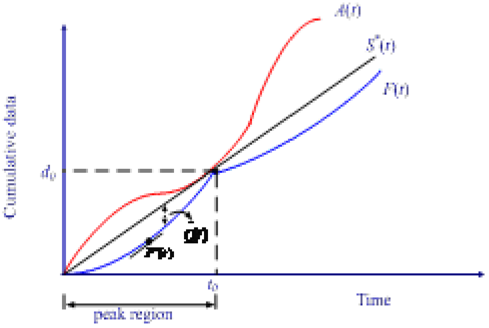


Figure 3. The proportional rate-based scheme

The instant velocity vector ***v*** of queue occupancy that has positions $x(t)$ at time *t* and $x(t+\Delta t)$ at time $t+\Delta t$, can be computed as the derivative of position:

$$v=\lim_{\Delta t=0}\frac{x(t+\Delta t)}{\Delta t}=\frac{\mathrm{d}x}{\mathrm{d}t} \tag{9}$$

The differential equation (8) governing queue length is given by:

$$\hat{Q}'(t)=r_{\max}-F'(t) \tag{10}$$

Suppose the region that requires $\hat{Q}(t)$ is the peak region. The slope of the tangent line is equal to the derivative of the function ($F'(t)$) at the marked point. We differentiate equation (10) to obtain:

$$\hat{Q}''(t)=-F''(t) \tag{11}$$

The first derivative of $F(t)$, $F'(t)$, can be represented as the transmission rate of a link at time *t*. Since the transmission rate of a link always remains constant, the second derivative, $F''(t)$, is therefore zero. Hence, the variance of $\hat{Q}(t)$ in a peak region is nearly linear. The current queue size can be adapted with the desired arrival rate. Using the PRC, the average queue lengths remain under desired level in any traffic condition.

III. SIMULATION RESULTS AND ANALYSIS

In this section we compare the performance of the proposed PRC with the other AQM schemes, such as RED, SRED, DRED, AVQ and VRC. We carried out simulation using the network simulator J-Sim [15]. Fig. 4 shows the simulation topology. In the first experiment, we study the convergence properties and queue sizes for the PRC alone. In the other experiments, we compare the PRC with other schemes. Table 1I lists all the simulation parameters along with their recommended values.

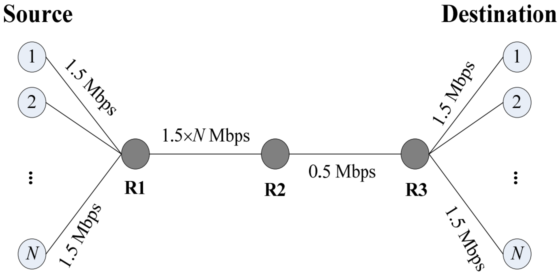


Figure 4. Simulation topology

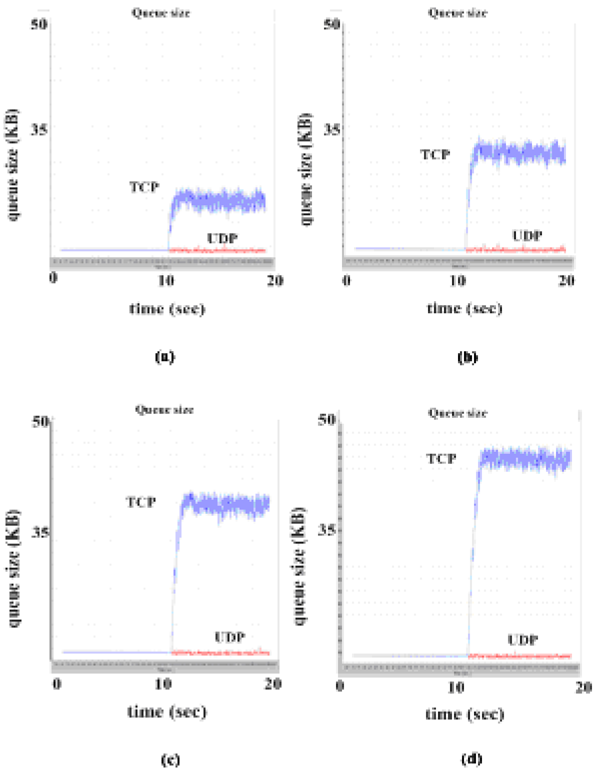
(a) *Experiment 1: Validity of parametric region for stability*

One TCP and one UDP traffic flow, which is generated from s1 and s2 individually, competes for a bottleneck link (R2-R3) with bandwidth of 0.5 Mbps. Initially at $t = 0$ sec, no traffic is generated. At $t = 10$ sec, two traffic flows are started from s1 and s2. Fig. 5 illustrates the PRC queue occupancy affected only by the variations of both□ ・ $_{\min}$ and ・ $_{\max}$, with no virtual queue involved. The difference between the value of ・ $_{\min}$ and ・ $_{\max}$ is fixed at 0.1. If both□ ・ $_{\min}$ and ・ $_{\max}$ are set high (Fig. 5(a) and (b)), the queue length oscillates severely due to excessive discard of arrival packets. If both ・$_{min}$ and ・$_{max}$ are set low (Fig. 5(c) and (d)), the queue can be stabilized around the small queue length, thus leading to underutilization. To combat large oscillatory behavior of the queue length, we incorporate the virtual queue into the proposed model. We study the sensitivity of queue occupancy to the virtual queue threshold parameter (k).

TABLE II.
SUMMARY OF SIMULATION PARAMETERS

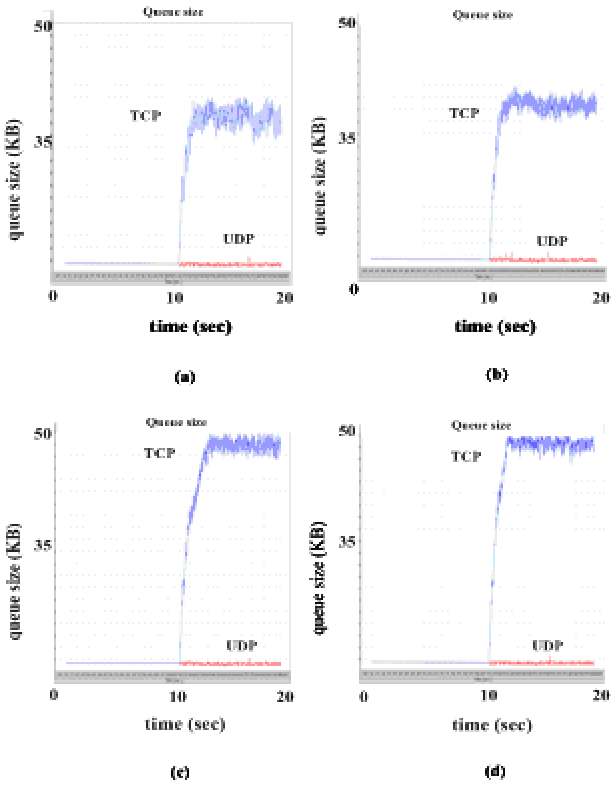
	Parameter	Value
RED	queue size Q	50,000 bytes
	maximum threshold max_{th}	30,000 bytes
	minimum threshold min_{th}	10,000 bytes
	maximum drop probability max_p	0.1
	queue weight w_q	0.002
SRED	buffer capacity B	50,000 bytes
	hit frequency $P(t)$	0.25
	maximum drop probability P_{max}	0.5
DRED	sampling interval Δt	0.05
	control gain α	0.05
	filter gain β	0.002
	control target T	0.5
	buffer size B	50,000 bytes
	no-drop threshold L	0.45
AVQ	damping factor ・ ・	0.15
	control factor ・	0.8
	・ desired utilization ・	1.0
VRC	K_D	0.0003
	K_P	0.0024
	K_I	0.0045

Fig. 6 depicts PRC queue occupancy for different virtual queue threshold values (0.3, 0.5, 0.7, 0.9). Two desired utilization parameters, ・$_{max}$ and ・$_{min}$, are fixed at 0.7 and 0.4, respectively. We can clearly see that the PRC is effective at stabilizing and keeping the queue size around the control target ($Q\times k$). We also study the sensitivity of queue occupation to two desired utilization parameters, ・$_{max}$ and ・$_{min}$. We first vary the value of ・$_{min}$ (0.2, 0.4, 0.6, 0.8) while fixing the value of ・$_{max}$ to be 0.9 and the value of k to be 0.7. We can see from Fig. 7(a) and (b) that the PRC successfully controls queue length at small value of ・$_{min}$. A large ・$_{min}$ value (Fig. 7(c) and (d)) will slow down system reaction, thus causing oscillation of queue length around physical queue size. Alternatively, we fix the value of ・$_{min}$ to be 0.3 and the value of k to be 0.7. We use four different values of ・$_{max}$ (0.6, 0.7, 0.8 and 0.9) to study the sensitivity of queue occupation to ・$_{max}$. From Fig. 8, we found that the increasing ・$_{max}$ has minor effect of queue length oscillation. The packets whose arrival rate exceeding maximum desired utilization are discarded. The queue lengths remain small irrespective of the value of ・$_{max}$ that is used over 0.6. In both Fig. 7 and Fig. 8, we found that a lower bound 0.3 on the value of (・$_{max}$– ・$_{min}$) guarantees the stability of queue length.



(a) (b) (c) (d)

Figure 5. Queue occupation for PRC without virtual queue, experiment 1 (a) ・$_{min}$ = 0.8, ・$_{max}$ = 0.9 (b) ・$_{min}$ = 0.6, ・$_{max}$ = 0.7 (c) ・$_{min}$ = 0.3, ・$_{max}$ = 0.4 (d) ・$_{min}$ = 0.1, ・$_{max}$ = 0.2



(a) (b) (c) (d)

Figure 6. Queue occupation for PRC, experiment 1 (a) k = 0.3 (b) k = 0.5 (c) k = 0.7 (d) k = 0.9

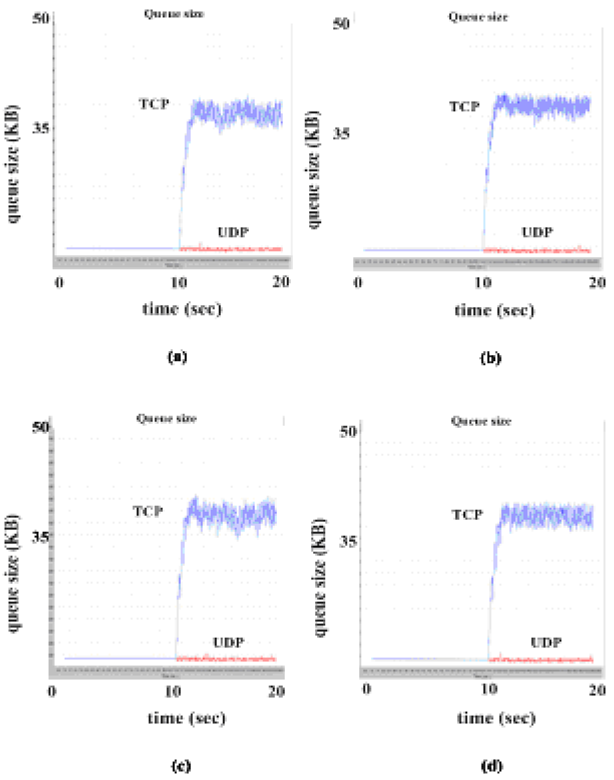


Figure 7. Queue occupation for PRC, experiment 1 (a) ・ $_{min}$ = 0.2 (b) ・ $_{min}$ = 0.4 (c) ・ $_{min}$ = 0.6 (d) ・ $_{min}$ = 0.8

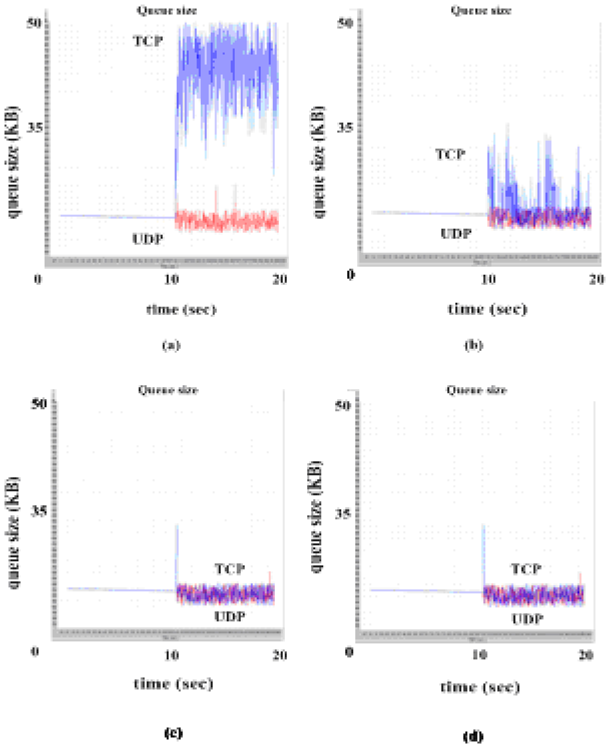


Figure 8. Queue occupation for PRC, experiment 1 (a) ・ $_{max}$ = 0.6 (b) ・ $_{max}$ = 0.7 (c) ・ $_{max}$ = 0.8 (d) ・ $_{max}$ = 0.9

(b) *Experiment 2: Validity and analysis of stability*

Initially at t = 0 sec, two TCP flows are started from s1 and s2, respectively. In this experiment, we compare the PRC with other well-known queue-based AQM schemes. Fig. 9 shows the queue occupation for each algorithm in the network. The initial queue size surges in all schemes due to initialization process of their algorithm. For SRED, its queue occupancy remains bounded away from empty, but its queue length shows periods of queue overflow. The reason is that the SRED discards packets with a load-dependent probability. The SRED ignores the consideration of queue length, thus causing the queue length close to the physical queue size. We found that the queue size is more variable for both RED and DRED algorithms. For RED, the queue length depends on the traffic load. In the status of light load, the queue length is small. The queue length increases as the load becomes heavy. Therefore, the queue length oscillates severely between the maximum and minimum thresholds. The DRED results in a tighter control of the queue size than the RED by using higher-level control mechanism to adjust the control targets dynamically. The queue length grows between the physical queue size and the no-drop threshold L. However, we found there are several traffic surges in the run. These surges possibly are at the cost of the complicated calculation of dropping probability and the insensitivity of the feedback control system. The PRC is designed with the objective of stabilizing the queue length around the target level, thereby minimizing occurrences of queue overflows and underflows. The PRC is effective at stabilizing the queue size from overflow and underflow. The queue occupation of PRC is stable around the target queue occupation (35,000 bytes).

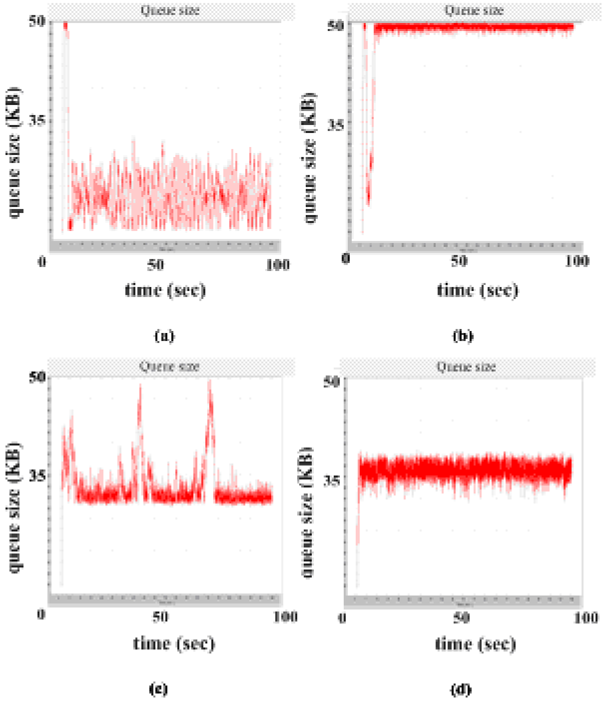


Figure 9. Queue occupation, experiment 2 (a) RED (b) SRED (c) DRED (d) PRC

(c) *Experiment 3: Validity and analysis of responsiveness*

Initially at t = 0 sec, the TCP traffic is generated from s1 to s4. To simulate a sudden change in a network condition, the flows from s5 to s7 are introduced in the system at t = 100 sec and the flows from s8 to s10 are started at t = 150 sec. The throughput for various queue-based AQM schemes is shown in Fig.10. The throughput is defined as the number of packet sizes (bytes) per second processed by R1-R2 link. Alternatively, the real throughput is defined to be the rate at which packets are forwarded through the congested link (R2-R3). The approximation of throughput to real throughput can be considered as an indicator of adaption. We found that the PRC maintains relatively high levels of adaption even at high load. Since the PRC filters the incoming traffic with the rate-based control, its throughput remains adaptive to changes in real throughput. For the other three schemes, the system takes some time to respond to departures and new arrival, thus making the throughput sluggish to changes in real throughput.

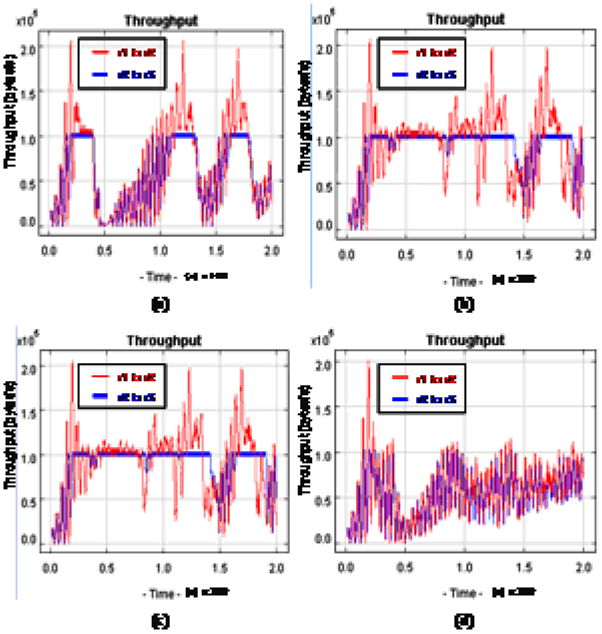


Figure 10. Throughput, experiment 3 (a) RED (b) SRED (c) DRED (d) PRC

(d) *Experiment 4: Validity of analysis of efficiency*

In this experiment, we examine the performance of four rate-based AQM schemes for different numbers of TCP connections, N, ranging from 20 to 200. The performance is evaluated in terms of average queue length and packet loss rate. In Fig. 11 (a), we found that the average queue length of AVQ is very sensitive to the value of N. The reason is that the AVQ does not control the queue length directly. For the other three schemes, the average queue length has not varied much from the target length regardless of the value of N. All the three schemes have the capability of regulating the queue length to the target level. In Fig. 11 (b), the loss rates of four rate-based AQM schemes grow smoothly as the value of N increases. The loss rate of VRC becomes slightly higher than the other schemes. The effect of increasing the value of N is similar to that produced by decreasing the capacity of bottleneck. The VRC tends to discard more packets as the capacity of bottleneck decreases. The PRC outperforms the other three schemes as the value of N increases.

IV. CONCLUSIONS

In this paper, a simple and robust AQM scheme, called proportional rate-based control (PRC) algorithm, is presented. As opposed to queue length or average length based AQM, the proposed mechanism stabilizes the oscillation of queue size by regulating the input rate and the queue size, respectively, with dropping and using virtual queue. The implementation complexity of the PRC algorithm is comparable to other well-known AQM schemes. To adequately show the reaction ability of PRC, we experiment the convergence of queue length to the desired utilization parameters (• $_{min}$ and • $_{max}$) and the virtual queue threshold (k). From simulation results, we found that the proposed scheme is able to respond in a timely manner to fluctuations in traffic characteristics. The queue size oscillation can be stabilized with moderate losses by keeping the values of two parameters, • $_{min}$ and • $_{max}$, proportionally. The simulation results validate the effectiveness of the early control in the PRC scheme, which arrives at stable queue occupation around the target value and a small loss rate in most of the cases.

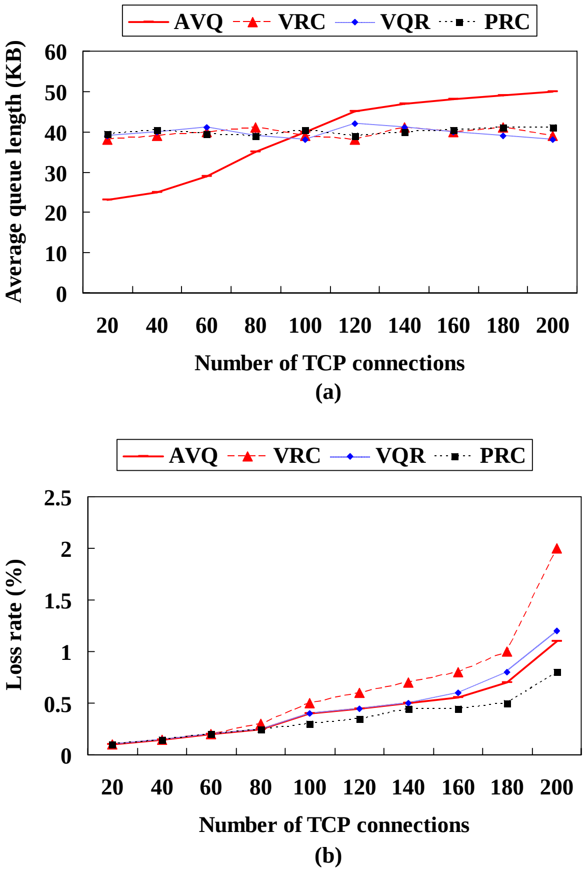


Figure 11. Performance comparison for various numbers of TCP connections, experiment 4 (a) average queue length (b) loss rate

REFERENCES

[1] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. Networking*, vol.1, no.4, August 1993, pp. 397–413.

[2] T. J. Ott, T. V. Lakshman and L. H. Wong, “SRED: Stabilized RED,” *IEEE INFOCOM 1999–The Conference on Computer Communications, n*o.1, March 1999, pp.1346–1355.

[3] J. Aweya, M. Ouellette, and D. Y. Montuno, “A control theoretic approach to active queue management,” *Computer Networks*, vol.36, July, 2001, pp. 203–235.

[4] L. Tan, Y. Yang, W. Zhang and M. Zukerman, “On control gain selection in dynamic–RED,” *IEEE Communications Letters*, vol. 9, no. 1, Jan 2005, pp. 81–83.

[5] E. Lochin and B. Talavera, “Managing network congestion with a Kohonen-based RED queue,” *ICC 2008–IEEE International Conference on Communications*, vol. 31, no. 1, May 2008, pp. 5586–5590.

[6] C. Joo, S. Bahk and S. S. Lumetta, “A hybrid active queue management for stability and fast adaptation,” *Journal of Communications and Networks*, vol. 8, no. 1, March 2006, pp. 93–105.

[7] S. Chen, Z. Zhou and B. Bensaou, “Stochastic RED and its applications,” *ICC 2007–IEEE International Conference on Communications*, no. 1, June 2007, pp. 6361 – 6366.

[8] E. C. Park, H. Lim, K. J. Park, and C. H. Choi, “Analysis and design of the virtual rate control algorithm for stabilizing queue in TCP networks,” *Computer Networks*, vol. 44, 2004, pp. 17–41.

[9] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, “REM: Active queue management,” *IEEE Network*, no.3, May/June 2001, pp.48–53.

[10] J. Sun, M. Zukerman and M. Palaniswami, “An adaptive REM for improving AQM performance,” *ICC 2008–IEEE International Conference on Communications*, vol. 31, no. 1, May 2008, pp. 75 – 79.

[11] S. S. Kunniyur and R. Srikant, “An adaptive virtual queue (AVQ) algorithm for active queue management,” *IEEE/ACM Transactions on Networking*, no.2, March 2004.

[12] L. Tan, Y. Yang, C. Lin, N. Xiong and M. Zukerman, “Scalable parameter tuning for AVQ,” *IEEE Communications Letters*, vol. 9, no. 1, Jan 2005, pp. 90 – 92.

[13] C. N. Long, B. Zhao and X. P. Guan, “SAVQ: Stabilized adaptive virtual queue management algorithm,” *IEEE Communications Letters*, vol. 9, no. 1, Jan 2005, pp. 78 – 80

[14] L. Zhu and N. Ansari, “Local Stability of a New Adaptive Queue Management (AQM) Scheme,” *IEEE Communications Letters*, vol. 8, no. 6, June 2004, pp.406-408.

[15] J-Sim, see http://www.j-sim.org/

Chin-Ling Chen received his BS degree from National Taiwan University in 1988, the Master degree in Management Information System from University of Wisconsin, Milwaukee, in 1992 and the Ph.D degree in Information Management from National Taiwan University of Science and Technology, 1999. Since 1999 spring, he joined the faculty of Department of Information Management at National Pingtung Institute of Commerce, Taiwan. He is currently an Associate Professor at that affiliation. His research interests include Internet QoS, network technology and network management. He is a member of IEEE and IEICE.