Adaptive Queue Management Based On the Change Trend of Queue Size

Liangrui Tang and Yaomu Tan
State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources
(North China Electric Power University)
Beijing 102206, China
[e-mail: tanliangrui@163.com]
*Corresponding author: Yaomu Tan

Received July 14, 2018; revised September 6, 2018; accepted September 30, 2018; published March 31 2019

Abstract

Most active queue management algorithms manage network congestion based on the size of the queue but ignore the network environment which makes queue size change. It seriously affects the response speed of the algorithm. In this paper, a new AQM algorithm named CT-AQM (Change Trend-Adaptive Queue Management) is proposed. CT-AQM predicts the change trend of queue size in the soon future based on the change rate of queue size and the network environment, and optimizes its dropping function. Simulation results indicate that CT-AQM scheme has a significant improvement in loss-rate and throughput.

Keywords: Active queue management, dropping probability, network environment, response speed, loss-rate, throughput

1. Introduction

With the popularization of intelligent terminals, a large amount of real-time multimedia services which generate huge data traffic have emerged, such as VoIP, video conferencing, and web video browsing. Network with high traffic load leads to network congestion [1-4] and makes the buffer of forwarding router full. Queue size is an important factor affecting network quality of service [5-8]. When the queue is full, it will sharply increase the loss-rate, drastically reduce throughput, and greatly fluctuate transmission delay. It is essential to establish an effective mechanism of congestion control in network to reduce the frequency of congestion and improve network performance.

Traditionally, "drop tail" was used to control congestion in the Internet. It will drop packets at the end of the queue when buffer overflows. Therefore, "drop tail" is a passive queue management method. And it has the following drawbacks [9]:

- “Long time buffer overflow” phenomenon: The source receives notification of congestion only when buffer overflow occurs. The buffer will be near full for a long time which causes the long end-to-end delay.
- “Global synchronization” phenomenon: When network congestion occurs, TCP flows will notify their sources to reduce the size of transmission window at the same time, which results in link utilization slump. And when the link is detected to be idle, TCP flows will send a notification to their sources to increase the window size, which leads to congestion again.
- “Lock out” phenomenon: Because of the effect of network timing and “global synchronization”, one or several flows always occupies all the buffer space and no room for other flows.

For many shortcomings of "drop tail", scholars have proposed active queue management (AQM) mechanism which can effectively overcome the above problems. AQM is a technique which adopts a closed-loop congestion control mechanism at the nodes where congestion occurs based on router feedback. It can manage the congestion in the Internet through dropping or marking packets before buffer size becomes full. The event of packet dropping or marking will subsequently trigger a notification to the source to slow down its sending rate of packets. AQM, as an efficient means of network congestion technique, can significantly improve loss-rate, throughput, end-to-end delay and delay jitter.

RFC 2309 [9] recommends RED (Random Early Detection) algorithm as the recommended algorithm. RED optimizes its packet dropping function through the result of comparing the current average queue size with the maximum threshold (max_{th}) and the minimum threshold (min_{th}) [10-12]. **Fig. 1** shows how RED algorithm handles the incoming packets. When the average queue size is bigger than max_{th}, dropping all the incoming packets. While no packet dropped when the average queue size is lower than min_{th}. RED drops packets according to the current average queue size in the case of that the average queue size is between min_{th} and max_{th}.

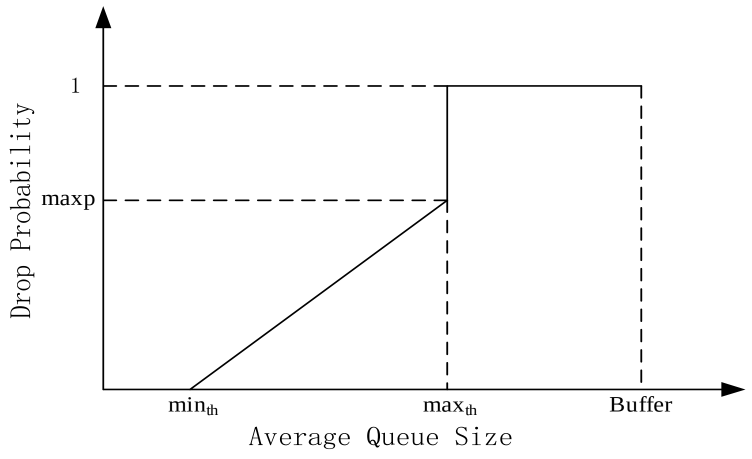

Fig. 1. RED Drop Curve

The main advantages of RED are as following:

- RED adopts a strategy of random early marking to handle incoming packets. The response of different sources to the congestion indication is more dispersed which can avoid the global synchronization phenomenon of the TCP flow.
- RED can better control the queue size than drop tail and greatly reduce the frequency of network congestion.
- RED has no bias against burst flows.
- RED can improve the utilization of network resources.

However, RED algorithm lacks self-adaptation mechanism and it is sensitive to parameters. Once the traffic load changes, the effect of congestion management will be seriously affected [10-12]. In order to improve the performance of the RED algorithm, many scholars proposed improved algorithms based on RED, such as ARED [13], TRED [14], MRED [15], URED [16], and Smart RED [17]. These RED-based algorithms overcome the shortcomings of RED to some extent. ARED [13] dynamically adjusts the parameter of maximum drop probability (*maxp*) to enhance the adaptability of algorithm. TRED [14], URED [16] and Smart RED [17] optimize the drop curve to achieve a better congestion control performance. For instance, URED uses a concave curve as its dropping curve when the average queue size is between min_{th} and $(min_{th}+ max_{th})/2$. TRED divided the average queue size into three sections, each section has its own dropping function. And Smart RED adjusts its dropping probability based on the traffic load instead of average queue size.

There are some other AQM algorithms based on controller, such as AOPC [18], RQM [19], PI [20], REM [21], and adaptive REM [22]. RQM uses a queue controller to minimize the difference between the instantaneous queue size and target queue size. It can solve the problem of time-varying capacity in wireless networks. In [20-22], PI controller is used to overcome the steady-state error. These algorithms significantly improve the stability of queue size. The newly algorithms CoDeL and FQ-CoDeL proposed in [23-24] aim to reduce the delay. These schemes can achieve a higher bottleneck utilization while maintaining a low queuing delay over capacity-limited networks with large Round-Trip Time (RTT). In [25-27], algorithms are proposed to achieve the fairness among data flows. SRED proposed by T. J. Ott et al. [25] estimates the number of active flows and identifies flows that may be misbehaving. In [26], the authors proposed CHOKe which can protect TCP flows and suppress the number of UDP flows and allocate bandwidth reasonably for the two kinds of flows. In addition, many alternatives are proposed in the literature, such as BLUE [27], FRRED [28], and ACO [29]. Since different algorithm focuses differently, each has its own advantages and limitations.

Recently, Karmeshu et al. have proposed the AQMRD algorithm [30]. AQMRD introduces a new parameter, the change rate of average queue size, which reflects the current change direction of queue size. In AQMRD, not only the average queue size but also its change rate are taken into account to optimize the dropping function, which keeps queue size and transmission delay in a low value even in the network where traffic load changes frequently. However, parameters of AQMRD are not adaptable and the packet dropping strategy of AQMRD is so overaggressive that its loss-rate is too high, which seriously affect the efficiency of congestion management. In addition, AQMRD and many other AQM algorithms do not take into account the network environment which makes queue size change to manage congestion. It may greatly affect the response speed of the algorithm.

To improve the response speed of algorithm, we first discuss a mathematical model of network environment. And next establish a system model which can predict the change trend of queue size in the soon future based on the change rate of average queue size and the network traffic loads. Finally, a new algorithm based on the system model named CT-AQM (Change Trend-Adaptive Queue Management) was proposed. Simulation results indicate that CT-AQM can stabilize queue size within a reasonable range and has a significant improvement in packet loss-rate and throughput in different level of traffic load scenarios.

This paper is organized as follows: Section 2 discusses the system model for our proposed scheme while section 3 discusses the algorithm based on the system model. Simulations and analyses are presented in section 4. In the last section, we provide the concluding remarks.

2. System Model

2.1 Queue-State Model Analysis

Fig. 2 is a simplified topology diagram of data forwarding. Each node on the left represents a data flow. Node D on the right represents the destination of data flows. The link between the router and Node D is the output link and the value of its bandwidth is fixed. In the data forwarding process under AQM mechanism, all the data flows are aggregated to the forwarding router and the router discards some data packets to prevent network congestion before packets enter into the buffer. Packets entering the buffer will be sent to their destinations through the output link. So the relationship of size between the rates of packet arrive at the router and the bandwidth of the output link affects the frequency of network congestion.

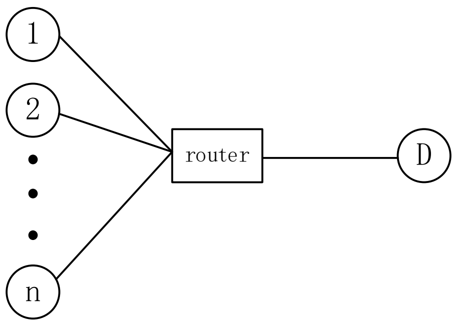

Fig. 2. Data Forwarding Topology

Assuming that the bandwidth of the output link is *C*, and the arrival rate of the *i*th data flow at time *t* is *ri(t)*. The total arrival rate of the data flow at time t is:

$$A(t) = \sum_{i=1}^{n} r_i(t) \tag{1}$$

If $A(t) < C$, the packet arrival rate is less than the output link bandwidth, packets will not accumulate in the buffer. If this state is maintained for a long time, the link utilization will be declined. For $A(t) > C$, the packet arrival rate is larger than the output link bandwidth, some packets will stay in the buffer and wait to be sent. It may lead to overflow event and even congestion in severe circumstance. The ideal situation is $A(t) \approx C$, the packet arrival rate is maintained near the output link bandwidth. In this case, the network performance is optimal.

Based on the above analysis, the relationship of size between the packet arrival rate *A(t)* and the output link bandwidth *C* causes the event of link idleness or congestion. Therefore, an effective queue management mechanism should adopt different packet discard strategies based on the relative sizes of *A(t)* and C. For $A(t) < C$, the packet dropping probability should be reduced to increase the link utilization. For $A(t) > C$, the packet dropping probability should be increased to ease or even avoid network congestion.

In order to achieve the above objectives, we establish the queue state model. First, the change rate of average queue size [30] is used to determine the current change direction of queue size. And next, the inflow/outflow ratio of the packet is calculated to measure network traffic load. Finally, judging queue status according to the above two factors. Queue status indicates the change trend of queue size in the soon future.

2.2 Change Rate Judgment

A new parameter *davg*, the change rate of average queue size, is proposed in [30]. It reflects the current change direction of queue size. Parameter *davg* is calculated as follows:

$$davg(n+1) = (1-\omega_q) \times davg(n) + \omega_q \times [q(n) - q(n-1)] \tag{2}$$

where *q(n)* and *davg(n)* are instantaneous queue size and the rate of change of the average queue size respectively at the time of the *n*th packet arrival. If $davg > 0$, the queue size is increasing. The larger the *davg*, the faster the queue size increases. Conversely, if $davg < 0$, the queue size is decreasing. The smaller the *davg*, the faster the queue size decays. For $davg \approx 0$, the queue size is stable. Then, its change direction is easily affected by the network environment.

We set the parameter $\lambda(\lambda > 0)$ which can be seen as a measuring index for *davg* to help us accurately measure the size of *davg*. And λ is fixed a priori to a small value. For $davg > \lambda$, the queue size is increasing rapidly. For $davg < -\lambda$, the queue size is shrinking rapidly. For $-\lambda < davg < \lambda$, the queue size is relatively stable and its change direction is not obvious. can be seen as measuring index for *davg*.

2.3 Network Environment Judgment

To judge the network environment, we introduce parameter γ represents the inflow/outflow ratio of packets within the previous period of ΔT . γ is calculated as follows:

$$\gamma = \frac{A}{C} \tag{3}$$

For $\gamma > 1$, the packet arrival rate is larger than the output link bandwidth. For $\gamma < 1$, the packet arrival rate is less than the output link bandwidth. For $\gamma \approx 1$, the packet arrival rate is maintained near the output link bandwidth. In order to accurately measure the size of γ , set:

$$\alpha = \frac{M \times N}{\Delta T \times C} \tag{4}$$

where *N* represents the size of one packet, *M* which is set in priori represents the number of packets to be sent within the period of *ΔT*, and α is a ratio of a sending rate to bottleneck link bandwidth. Then the following three conditions will occur:

$$\gamma > 1 + \alpha \tag{5}$$

$$\gamma < 1 - \alpha \tag{6}$$

$$1 - \alpha < \gamma < 1 + \alpha \tag{7}$$

If *equation (5)* happens, it indicates that the number of packets arriving at the router is much larger than the number of packets sent out within the period of *ΔT*. For bottleneck links, the network traffic load is too high at this time. If this state is maintained, it will lead to network congestion.

If *equation (6)* happens, it indicates that the number of packets arriving at the router is much smaller than the number of packets sent out within the period of *ΔT*. For the bottleneck link, the network traffic load is very low and the packets in the buffer will be sent out quickly. If this state is maintained, it will cause long-term link idle event and reduce the link utilization.

If *equation (7)* happens, it indicates that the number of packets arriving at the router is approximately the same as the number of packets sent out within the period of *ΔT*. For the bottleneck link, the network traffic load is moderate and the queue size will not change significantly.

2.4 Queue Status Judgment

The network environment will make queue size change, but its impact on the queue size has a certain lag. Queue status model use this characteristic to predict the change trend of queue size in the soon future.

When $davg < -\lambda$ while $\gamma < 1 - \alpha$, the queue size is rapidly attenuating and it will continuously show a rapid decay because of the low traffic load. We define the queue status as continuous attenuation at this time. In this queue status, the dropping probability should be reduced to avoid the link idle event.

When $davg < -\lambda$ while $\gamma \geq 1 - \alpha$, the queue size is rapidly attenuating but the network traffic load is relatively large. We define the queue status as attenuation at this time. In this queue status, on the whole, the queue size will decrease, but the decay rate will be gradually slow down. A conservative dropping strategy should be adopted to reduce unnecessary loss of packets.

When $-\lambda \leq davg \leq \lambda$, the queue size is relatively stable. We define the queue status as stable at this time. In this queue status, we can adopt a regular dropping strategy.

When $davg > \lambda$ while $\gamma \leq 1 - \alpha$, the queue size is increasing rapidly, however, the network traffic load is relatively low. Define the queue status as increase at this time. In this queue status, on the whole, the queue size will increase, but the increase rate will be gradually slow down. An active dropping strategy should be adopted to achieve the purpose of controlling the queue size.

When $davg > \lambda$ while $\gamma > 1 + \alpha$, the queue size is increasing rapidly, and will continuously show a rapid increase because of the large traffic load. We define the queue status as continuous increase at this time. In this queue status, the number of packets entering the queue should be strictly restricted so as to avoid large-scale packet loss event.

Based on the above analysis, there are 5 queue statuses as shown in **Table 1** In queue status model, the change rate of average queue size is like the direction of inertial moving of an object, and the inflow/outflow ratio of packets is like the direction of resultant external force received by the object. When the queue status is continuously attenuate or continuously increase, just as the direction of resultant external force coincides with that of inertial motion, the change rate of queue size will be continuous increased. When the queue status is stable, attenuate or increase, just as the direction of external force is inconsistent with that of inertial motion, the change rate will be slow down.

Table 1. Queue Status Classification

The Queue Status	$\gamma < 1 - \alpha$	$1 + \alpha \geq \gamma \geq 1 - \alpha$	$\gamma > 1 + \alpha$
$davg < -\lambda$	Continuous Attenuation	Attenuation	Attenuation
$-\lambda \leq davg \leq \lambda$	Stable	Stable	Stable
$davg > \lambda$	Increase	Increase	Continuous Increase

In summary, the queue status model first determines the change direction of queue size by parameter *davg*, measures the network traffic load through the inflow/outflow ratio of packet, and finally determines the queue status. The queue status reflects the change trend of queue size in the soon future.

3. Algorithm

When a packet arrives at the router, the average queue size is calculated. The average queue size is calculated as follows [11]:

$$avg(n+1) = (1-\omega_q)\times avg(n) + \omega_q \times q(n) \tag{8}$$

where *q(n)* and *avg(n)* are instantaneous queue size and the average queue size, respectively, at the time of the *n*th packet arrival. The dropping function is calculated as follows:

$$Pb = \begin{cases} 0 & avg < min_{th} \\ Pc & min_{th} \le avg \le max_{th} \\ 1 & avg > max_{th} \end{cases} \tag{9}$$

where min_{th} and max_{th} are the minimum threshold and the maximum threshold respectively, and $max_{th}= min_{th}/3$ [11]. For $avg > max_{th}$, there are too many packets waiting to be sent in the buffer, all the arriving packets will be discarded to relieve the congestion. For $avg < min_{th}$, the buffer is idle, all incoming packets are received to improve link utilization. For $max_{th} > avg > min_{th}$, the dropping probability is calculated according to the queue status.

In order to take a more proactive dropping strategy when the queue status is increase or continuous increase, mid_{th} is introduced [30], which is dynamically adjusted according to the change rate of the queue size. Here, mid_{th} is updated as follows:

$$mid_{th} = \begin{cases} mid_{th}+1 & davg < -\lambda \\ mid_{th} & -\lambda \le davg \le \lambda \\ mid_{th}-1 & davg > \lambda \end{cases} \tag{10}$$

When $davg < -\lambda$ and $\gamma < 1-\alpha$, the queue status is continuous attenuation, a conservative dropping strategy should be adopted to reduce unnecessary drops. Its *Pc* is calculated as shown in *equation (11-a)* where *1-davg* was used to be the index. The faster the queue size decays, the smaller the dropping probability will be.

$$Pc = \left(\frac{avg - min_{th}}{max_{th} - min_{th}} max_p\right)^{1-davg} \tag{11-a}$$

If $davg < -\lambda$ while $\gamma \ge 1-\alpha$, the queue status is attenuation, its *Pc* is computed as shown in *equation (11-b)*. Comparing with *(11-a)*, $1-(davg+\lambda)$ was used as the index to ensure that *(11-b)* is always greater than *(11-a)* when *avg* is the same.

$$Pc = \left(\frac{avg - min_{th}}{max_{th} - min_{th}} max_p\right)^{1-(davg+\lambda)} \tag{11-b}$$

When $-\lambda \le davg \le \lambda$, the queue status is stable, the queue size will not fluctuate greatly. A conventional dropping strategy can be used in this case. Its Pc is calculated as follows:

$$Pc = \frac{avg - min_{th}}{max_{th} - min_{th}} max_p \tag{11-c}$$

When $davg > \lambda$ while $\gamma \le 1-\alpha$, the queue status is increase, an proactive dropping strategy should be adopted to achieve the purpose of controlling the queue size. At this time, Pc is calculated as follows:

$$Pc = \begin{cases} \dfrac{avg - min_{th}}{mid_{th} - min_{th}} max_p & avg \le mid_{th} \\ 1 & avg > mid_{th} \end{cases} \tag{11-d}$$

When $davg > \lambda$ while $\gamma > 1+\alpha$, the queue status is continuous increase, the number of packets entering the queue should be strictly controlled to avoid overflow. Its Pc is computed as follows:

$$Pc = \begin{cases} \left(\dfrac{avg - min_{th}}{mid_{th} - min_{th}} max_p \right)^{\frac{1}{1+davg-\lambda}} & avg \le mid_{th} \\ 1 & avg > mid_{th} \end{cases} \tag{11-e}$$

$1/(1+davg-\lambda)$ was used as an index in *(11-e)*. The faster the queue size increases, the greater the dropping probability will be.

In summary, CT-AQM algorithm optimized the dropping function according to the average queue size and queue status. In addition, CT-AQM dynamically adjusts parameter max_p by ARED algorithm [13] to further control the average queue size within the target range between min_{th} and max_{th}.

4. Simulations and Results

4.1 Simulation Setup

We use *ns-2* simulator to evaluate the performance of our proposed scheme with existing schemes. The network topology in experiments with N FTP sources sending packets to N destinations over a network of two routers (R_1 and R_2) and a bottleneck link in between as shown in **Fig. 3**.

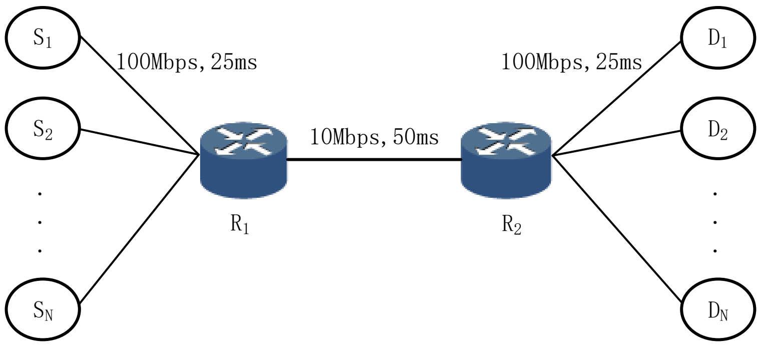

Fig. 3. Network Topology

Each link from source end to R_1 has capacity *100Mbps* and propagation delay of *25ms*. The bottleneck link has the capacity *10 Mbps* and propagation delay of *100ms*. Each FTP source sends a packet with a maximum packet size of *1,000 bytes* until the congestion control window allows sending of the packets. The FTP sources are set to deliver bulk data for a TCP object and the rate is not set manually in the simulation. Other simulation parameters are shown in **Table 2** [11]. And the initial settings of CT-AQM algorithm is the same as that shown in **Table 2**.

Table 2. Experiment Parameters

Parameter	**Value**
min_{th}	20 *packets*
max_{th}	60 *packets*
ω_q	0.002
max_p	0.1
Buffer size	80 *packets*
ΔT	0.5s
M in α	60 *packets*
λ	0.04

4.2 Performance Evaluation

In our experiments, we have simulated with 25, 35 and 50 FTP sources. These correspond to low, moderate, and high traffic load conditions. And the performance indexes such as queue size, loss-rate, throughput, delay and delay jitter of different algorithms in three traffic load conditions are compared to demonstrate the efficacy of the proposed algorithm.

Queue Size Analysis: Queue size comparison for each kind of traffic load for various algorithms can be seen from **Figs. 4(a)-(c)**. RED algorithm has the maximum fluctuation and the worst stability of queue size due to the lack of self-adaptive mechanism. ARED algorithm can stabilize the average queue size in low traffic load condition, but the stability of queue size deteriorates dramatically as the traffic load increases. It is because that the response speed of ARED algorithm is slow in the network with a lot of burst data flow. The average queue size of the AQMRD algorithm which is always below the minimum threshold is the most stable due to its aggressive dropping strategy. The average queue size is best stabilized into the target range between min_{th} and max_{th} when using CT-AQM scheme. Because CT-AQM algorithm can optimize its dropping function in advance by the impact of traffic load on queue size.

Loss-rate Analysis: A comparative study of loss-rates is shown in **Figs. 5(a)-(c)**. CT-AQM scheme is seen to give a lower loss-rate than existing AQM schemes for each traffic load. Because CT-AQM algorithm has a good adaptability to the burst flow and does not cause large-scale packet loss event. Since AQMRD algorithm obtains a lower queue size by significantly increasing its dropping probability, it has the highest loss-rate. The simulation results are consistent with the previous analysis.

Throughput Analysis: Throughput comparison for each scheme can be seen from **Fig. 6**. We can observe that the highest and most stable throughput, in different traffic load conditions, is obtained when using CT-AQM scheme. This result is achieved because the number of packets waiting in the queue can be kept within a reasonable range.

Delay and Delay Jitter Analysis: The mean and variance of end-to-end delay are shown in **Table 3**. μ and σ^2 represent the mean delay and delay jitter respectively. The mean of end-to-end delay is the largest using CT-AQM scheme, but the delay jitter of CT-AQM has been greatly improved compared to RED and ARED. It is because that the value of the delay is affected by the queue size. RED and ARED have a huge fluctuation in queue size fluctuates between 0 and buffer size. And the low value of queue size which results in low delay can pull down their average delay. However, the queue size of CT-AQM is focused on the range between min_{th} and max_{th} which leads to a higher mean of end-to-end delay and a lower delay jitter than RED and ARED. AQMRD reaches the lowest mean delay and delay jitter because aggressive dropping strategy which keeps the queue size in a low value. The data in **Table 3** is consistent with the previous simulation analysis.

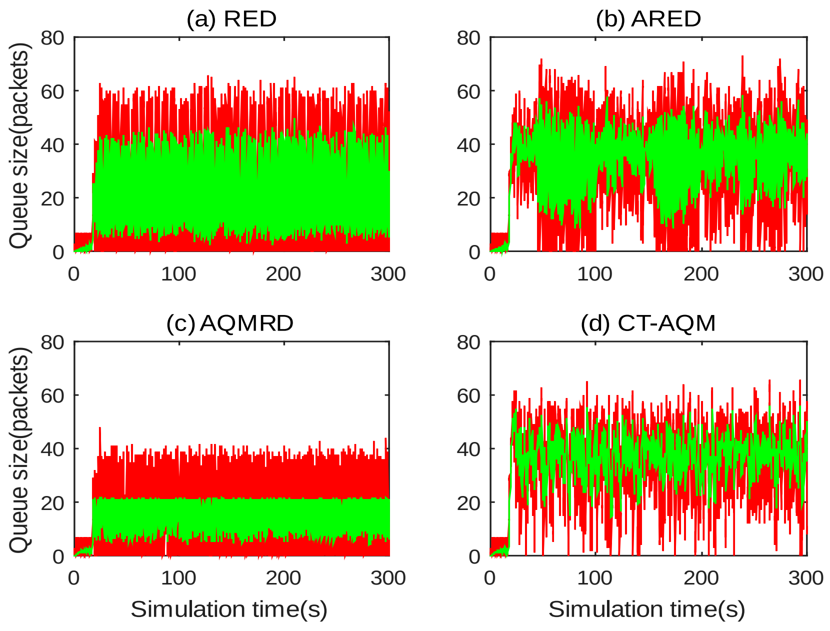

Fig. 4(a). Comparative changes in *q* (red color) and *avg* (green color) for N = 25

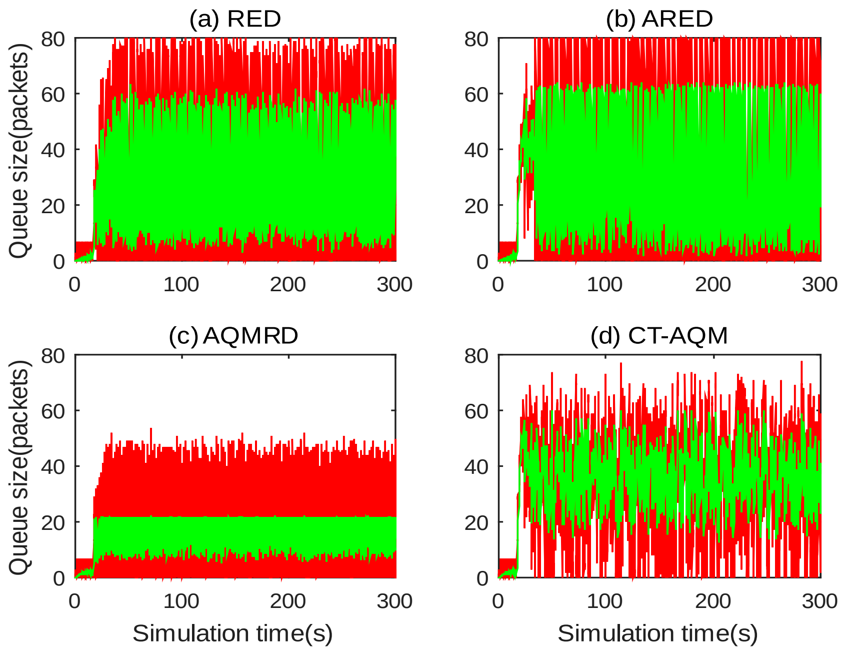

Fig. 4(b). Comparative changes in *q* (red color) and *avg* (green color) for N = 35

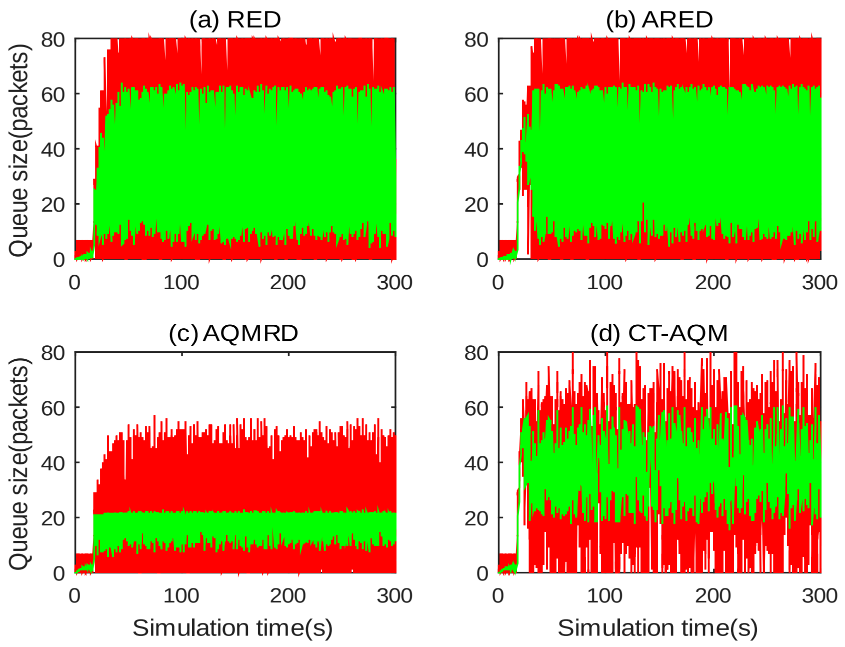

Fig. 4(c). Comparative changes in *q* (red color) and *avg* (green color) for N = 50

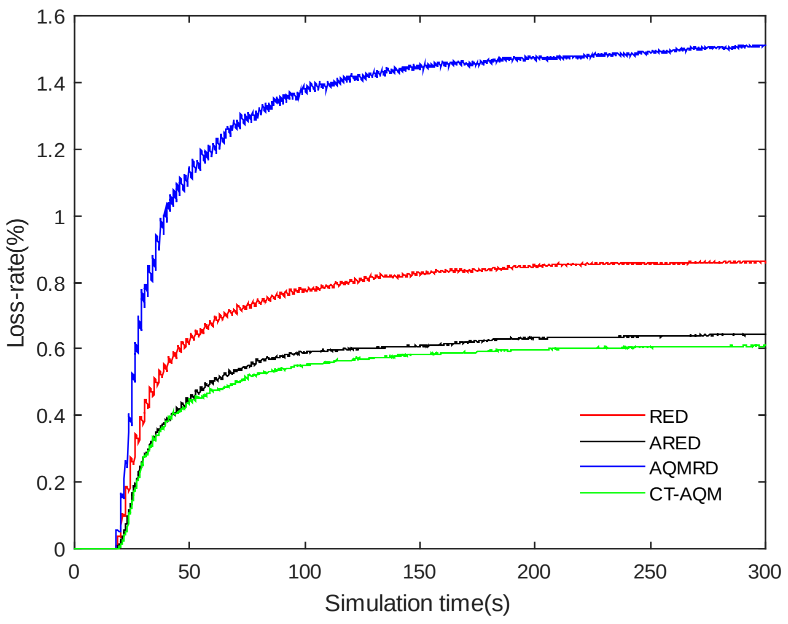

Fig. 5(a). Comparison of loss-rate for N = 25

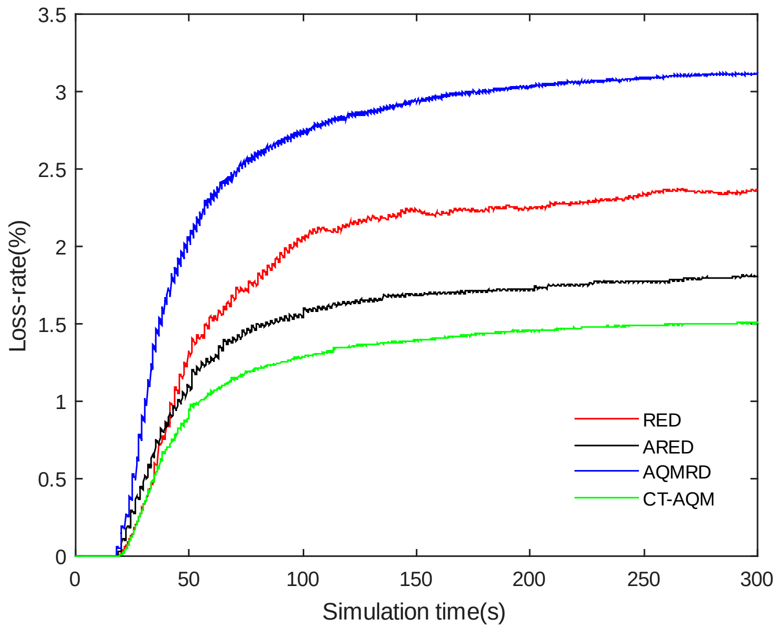

Fig. 5(b). Comparison of loss-rate for N = 35

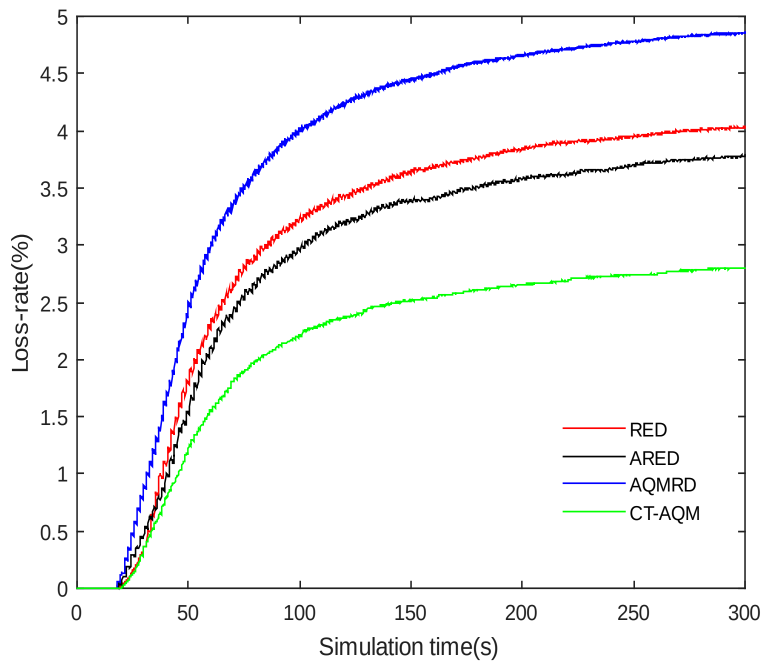

Fig. 5(c). Comparison of loss-rate for N = 50

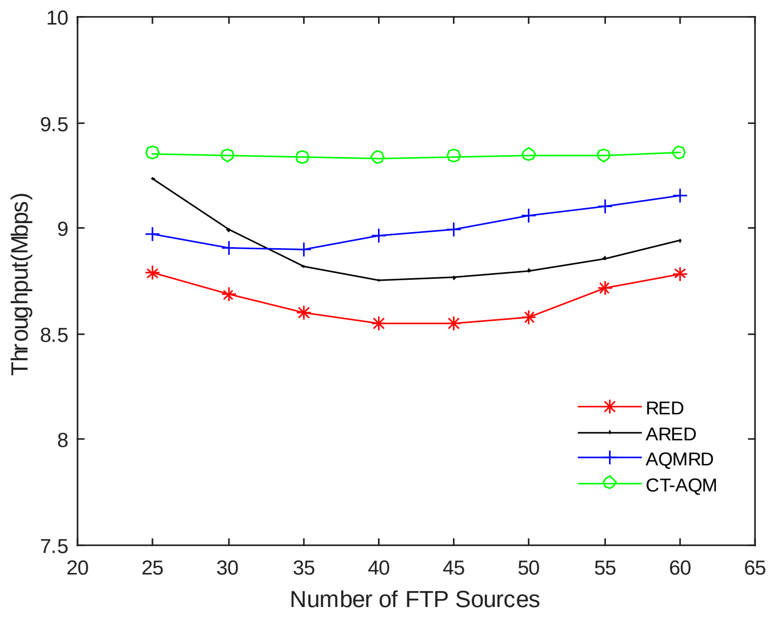

Fig. 6. Comparison for throughput vs N

Table 3. Mean and Variance of End-to-end Delay

AQM Algorithms	Mean and Variance of End-to-end delay (ms)					
	i=25		i=35		i=50	
	μ	σ^2	μ	σ^2	μ	σ^2
RED	119.4	287.4	125.7	534.0	128.7	640.1
ARED	129.4	192.7	127.8	590.8	130.2	631.2
AQMRD	112.2	85.1	113.1	103.3	114.6	107.3
TAQM	131.3	117.2	129.8	231.7	131.2	294.6

In summary, compared with existing algorithm, CT-AQM stabilize the average queue size into the target range between min_{th} and max_{th} and greatly improves loss-rate and throughput. However, CT-AQM allows more packets enter into the queue which results in the higher delay than other algorithms.

5. Conclusions

In this paper, we presented a novel algorithm, CT-AQM, that optimizes its dropping function depend on the queue status. Our scheme can significantly improve response speed because CT-AQM optimizes its dropping function according to its queue status that is determined not only depend on the change rate of average queue size but also depend on the network traffic load. Simulation results show that our scheme achieves the lower loss-rate as well as the higher throughput for each load level. However, CT-AQM does not perform as well as the other schemes in the end-to-end delay metric. It is because that CT-AQM allows more packets enter into the queue which results in the higher delay than other algorithms. The AQMRD performs well for end-to-end delay but worst in loss-rate due to its aggressive dropping strategy. These results conform to our previous analysis. The further work of our scheme is to find a more reasonable dropping function based on queue status model to reduce the delay and delay jitter.

References

[1] V. Jacobson, "Congestion avoidance and control," *Acm Sigcomm Computer Communication Review*, vol. 25, no. 1, pp. 157-187, Jan. 1995. Article (CrossRef Link).

[2] N. F. Huang, G. Y. Jai, H. C. Chao, Y. J. Tzang, and H. Y. Chang, "Application traffic classification at the early stage by characterizing application rounds," *Information Sciences*, vol. 232, pp. 130–142, May 2013. Article (CrossRef Link).

[3] Chaudhary, Pooja, and S. Kumar. "A review of comparative analysis of TCP variants for congestion control in network," *International Journal of Computer Applications*, vol. 160, no. 8, pp. 28–34, February 2017. Article (CrossRef Link).

[4] S. Duan, V. Shah-Mansouri, Z. Wang and V. W. S. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9847-9861, December 2016. Article (CrossRef Link).

[5] C. Y. Chen, T. Y. Wu, W. T. Lee, H. C. Chao, and J. C. Chiang, "QoS based active dropping mechanism for NGN video streaming optimization," *Knowledge Engineering Review*, vol. 29, no. 4, pp. 484–495, September 2014. Article (CrossRef Link).

[6] Xiao, Kefan, S. Mao, and J. K. Tugnait. "MAQ: A multiple model predictive congestion control scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2614-2626, April 2017. Article (CrossRef Link).

[7] C. F. Lai, H. G. Wang, H. C. Chao, and G. F. Nan, "A network and device aware QoS approach for cloud-based mobile streaming," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 747–757, June 2013. Article (CrossRef Link).

[8] Xu, Changqiao, J. Zhao, and G. M. Muntean. "Congestion control design for multipath transport protocols: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2948-2969, Fourthquarter 2016. Article (CrossRef Link).

[9] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the internet," United States, April 1998. Article (CrossRef Link).

[10] Patel, Sanjeev. "Performance analysis and modeling of congestion control algorithms based on active queue management," in *Proc. of 2013 International Conf. on Signal Processing and Communication (ICSC)*, pp. 449-454, Dec. 12-14, 2013. Article (CrossRef Link).

[11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, August 1993. Article (CrossRef Link).

[12] T. Bonald, M. May, and J. C. Bolot. "Analytic evaluation of RED performance," in *Proc. of IEEE INFOCOM 2000. Conf. on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, pp. 1415-1424, March 26-30, 2000. Article (CrossRef Link).

[13] S. Floyd, R. Gummadi and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's active queue management," *Tech. Report*, UC, Berkeley, CA, August, 2001. Article (CrossRef Link).

[14] C. W. Feng, L. F. Huang, C. Xu and Y. C. Chang, "Congestion Control Scheme Performance Analysis Based on Nonlinear RED," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2247-2254, December 2017. Article (CrossRef Link).

[15] G. Feng, A. K. Agarwal, A. Jayaraman and K. S. Chee, "Modified RED gateways under bursty traffic," *IEEE Communications Letters*, vol. 8, no. 5, pp. 323-325, May 2004. Article (CrossRef Link).

[16] C. M. Patel, "URED: Upper threshold RED an efficient congestion control algorithm," *in Proc. Of 2013 Fourth International Conf. on Computing, Communication and Networking Technologies (ICCCNT)*, pp.1-5, July 4-6, 2013. Article (CrossRef Link).

[17] A. K. Paul, H. Kawakami, A. Tachibana and T. Hasegawa, "An AQM based congestion control for eNB RLC in 4G/LTE network," in *Proc. of IEEE Conf. on Electrical and Computer Engineering (CCECE)*, pp. 1-5, May 15-18, 2016. Article (CrossRef Link).

[18] J. Wang, L. Rong, and Y. Liu, "A robust proportional controller for AQM based on optimized second-order system model," *Computer Communications*, vol. 31, no. 10, pp. 2468-2477, June 2008. Article (CrossRef Link).

[19] K. Chavan, R. G. Kumar, M. N. Belur, and A. Karandikar, "Robust active queue management for wireless networks," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1630-1638, November 2011. Article (CrossRef Link).

[20] S. Athuraliya, V. H. Li, S. H. Low and Qinghe Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48-53, May 2001. Article (CrossRef Link).

[21] C. V. Hollot, V. Misra, D. Towsley and W. B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proc. of IEEE Conf. on Computer and Communications Societies*, pp. 945-959, April 22-26, 2001. Article (CrossRef Link).

[22] J. Sun, M. Zukermant, and M. Palaniswamit, "An adaptive REM for improving AQM performance," in *Proc. of IEEE International Conf. on Communications*, pp. 75-79, May 19-23, 2008. Article (CrossRef Link).

[23] K. Chamil, N. Kuhn, F. Gorry, and R. David, "Tackling Bufferbloat in capacity-limited networks," in *Proc. of 2015 European Conf. on Networks and Communications*, pp. 381-385, June 29-July 2, 2015. Article (CrossRef Link).

[24] I. Jarvinen and M. Kojo, "Evaluating CoDel, PIE, and HRED AQM techniques with load transients," in *Proc. of IEEE Conf. on Local Computer Networks*, pp. 159-167, Sept. 8-11, 2014. Article (CrossRef Link).

[25] T. J. Ott, T. V. Lakshman, and L. H. Wong, "SRED: stabilized RED," in *Proc. of IEEE INFOCOM '99. Conf. on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, pp. 1346-1355, March 21-25, 1999. Article (CrossRef Link).

[26] A. Tang, J. Wang, and S. Low, "Understanding choke: throughput and spatial characteristics," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 694-707, August 2004. Article (CrossRef Link).

[27] W. C. Feng, D. Sahaz, D. Kandlurz and K. G. Shin, "BLUE: A new class of active queue management algorithms," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 298-312, September 2000.

[28] Z. M. Chen, T. N. Diep Pham, C. K. Yeo and B. S. Lee, "FRRED: Fourier robust RED algorithm to detect and mitigate LDoS attacks," in *Proc. of Zooming Innovation in Consumer Electronics International Conference (ZINC)*, pp. 13-17, May 31-June 1, 2017. Article (CrossRef Link).

[29] S. Chebli, A. Elakkary, and N. Sefiani, "PID controller tuning using multi-objective ant colony optimization applied to TCP/AQM networks," in *Proc. of 2017 23rd International Conf. on Automation and Computing (ICAC)*, pp. 1-6, September 7-8, 2017. Article (CrossRef Link).

[30] Karmeshu, S. Patel, and S. Bhatnagar, "Adaptive mean queue size and its rate of change: Queue management with random dropping," *Telecommunication Systems*, vol. 65, no. 2, pp. 287-295, June 2017. Article (CrossRef Link).

Liangrui Tang received the Ph.D. degree in Communication and Information System from Beijing University of Posts and Telecommunications. Now, he is a Professor in State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources of North China Electric Power University, focusing on the research of communication in power system, wireless communications and optical network communication.

Yaomu Tan received B.Sc. in School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, China, in 2016. He is currently a master in electronic and communication engineering in North China Electric Power University. His research interest is congestion control of computer network.