Early Marking for Controllable Maximum Queue Length in Data Center Networks

Wentao Wang*, Jiangping Han†‡, Rui Zhuang†, Kaiping Xue†‡, Qibin Sun†, Jun Lu*†
*Department of EEIS, University of Science and Technology of China, Hefei, Anhui 230027 China
†School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027 China
‡Corresponding author: J. Han (jphan@ustc.edu.cn), K. Xue (kpxue@ustc.edu.cn)

2023 32nd International Conference on Computer Communications and Networks (ICCCN) | 979-8-3503-3618-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCN58024.2023.10230181

***Abstract*—In data center networks (DCNs), numerous congestion control schemes utilize explicit congestion notification (ECN) to achieve low average queue delay. Such schemes generally mark packets based on the current queue length exceeding a marking threshold. However, due to the delay of ECN feedback, the queue length may further increase before the congestion notification is delivered to senders, which may lead to uncontrollable maximum queue length when bursts occur. In this paper, we propose an early ECN marking scheme based on prediction, E-ECN, to control the maximum queue length in DCNs. E-ECN uses predicted queue length rather than the current to indicate congestion with an advance time which offsets the hysteresis of ECN. We theoretically and experimentally demonstrate that early marking does not impact the throughput with appropriate selection of the advance time, and we provide guidelines for the selection in DCNs. Our simulation results show that E-ECN achieves shorter average queue delay and controllable maximum queue length in general with a bandwidth utilization guarantee. E-ECN greatly reduces queue overflow and improves the robustness of DCNs.**

***Index Terms*—Data Center Networks, Active Queue Management, Explicit Congestion Notification**

I. Introduction

Congestion detection is the key to end-to-end congestion control [1]–[4]. In the Internet, packet loss is usually used for congestion detection. A sender decreases its congestion window when packet loss is detected. However, due to the large traffic scale [5], [6] and the traffic demand for high bandwidth [7]–[11] and low latency [12]–[18] in data center networks (DCNs), the congestion detection based on packet loss cannot provide satisfactory quality of service [19], [20]. Using packet loss as a congestion signal can lead to high occupancy of switches, resulting in increased transmission delay. Frequent packet loss and retransmission can also reduce the effective throughput. To address these challenges and achieve low queues with a bandwidth utilization guarantee, an advanced congestion detection mechanism is needed for DCNs.

Explicit congestion notification (ECN) [21] is a congestion detection mechanism that is widely supported by commodity switches. By enabling ECN, switches can mark packets to indicate congestion rather than drop them. The switches can make congestion notification at the initial stage of queue buildup, and then senders can adjust their rate in time to avoid further congestion. Besides, ECN avoids the disadvantage of packet loss and can improve effective throughput and reduce the flow completion time (FCT). Because of the benefits of ECN, many congestion detection and control algorithms tend to utilize ECN in DCNs, such as DCTCP [22], DCQCN [23], and their enhanced schemes [24]–[26]. They generally utilize ECN to mark packets based on the queue length exceeding a marking threshold and can achieve less buffer occupancy and lower average queue delay.

However, despite the success of the ECN-based schemes in DCNs, they still have the defect of uncontrollable maximum queue length. Existing schemes react after the current (instantaneous or average) queue length reaches the threshold, thus suffering from the delay in the delivery of ECN [27]. During this delay, the queue length may further increase and reach its peak before the congestion notification is delivered to senders. Through analysis and experiment, we reveal that the peak of the queue is positively related to the input rate of the switch. When the rate increases, the peak exceeds the threshold more. Thus, while these schemes can effectively reduce the average queue length, they cannot provide adequate control over the peak of the queue. In addition, due to the limited switch buffer [28]–[31], these schemes may not avoid queue overflow when the input rate is high.

In this paper, we propose an early marking scheme based on prediction, called E-ECN, to achieve controllable maximum queue length in DCNs. Our innovative idea is that although the delay of ECN feedback cannot be eliminated, it can be neutralized by marking in advance. E-ECN predicts the queue length after an advance time and early marks packets based on the predicted value exceeding the threshold rather than the current queue length. When ECN notifies senders to reduce their rate and the queue falls, the actual queue length still does not exceed the threshold. Through this marking scheme, the delay of ECN feedback is offset by the advance time, and thus the peak of the queue can be controlled in time before reaching the threshold. We utilize a fluid model to analyze and demonstrate that, with appropriate selection of the advance time, early marking does not have a negative impact on the throughput of senders. Additionally, we provide guidelines for selecting appropriate parameter values in practical DCN scenarios based on our analysis.

We conduct multiple experiments to prove that E-ECN achieves shorter average queue delay and controllable maximum queue length in general with a bandwidth utilization guarantee. E-ECN limits 98.6% of the queue length below the threshold in steady state and achieves lower maximum

queue length distribution than other advanced active queue managements (AQMs) when bursts occur, thus enhancing robustness and burst tolerance in DCNs. Moreover, it helps to improve fairness and convergence speed at congestion points.

In summary, the contributions of this paper are as follows:

- We theoretically and experimentally reveal the uncontrollable maximum queue length problem of most schemes, which is caused by the hysteresis of ECN feedback and the marking decision based on the current queue state.
- We propose an early marking scheme named E-ECN, which marks packets based on the predicted queue state rather than the current to offset the delay of ECN feedback. We further prove that this early marking does not affect the throughput of senders.
- We use ns-3 [32] to evaluate E-ECN comprehensively. Our experimental results show that, compared to the existing advanced AQMs, E-ECN delivers a strong ability to suppress the peak of the queue. Additionally, it can further improve fairness and reduce FCT.

The rest of this paper is organized as follows. In Section II, we introduce the schemes based on ECN and their shortcomings and thus explain the motivation of our scheme. In Section III, we analyze and design the E-ECN scheme. Section IV presents comprehensive performance evaluations. At last, we conclude this paper in Section V.

II. Related Work and Motivation

This section discusses the related work on congestion detection in DCNs and illustrates the necessity of our scheme that early marks packets based on predicted queue length.

A. Schemes based on ECN

Congestion detection is the key to congestion control. In DCNs, congestion detection is often based on ECN [21] instead of packet loss to achieve more precise detection and fine-grained control. DCTCP [22] first uses ECN for congestion detection in DCNs, which marks packets based on the instantaneous queue length exceeding the threshold and adjusts the congestion window by the fraction of marked packets. Compared with TCP, DCTCP achieves lower buffer occupation and significantly reduces latency. DCQCN [23] is a congestion control scheme for converged Ethernet which marks packets based on double thresholds in switches and adjusts the rate for remote direct memory access (RDMA). When the queue length exceeds the high threshold, all incoming packets will be marked; when the queue length is lower than the low threshold, the packets are directly queued; when the queue length is between two thresholds, the packets are marked with a linear increasing probability. ECN* [33] is proposed to enhance the queuing performance of TCP, which marks packets by reference to the instantaneous queue length and utilizes dequeue marking to speed up the delivery of ECN.

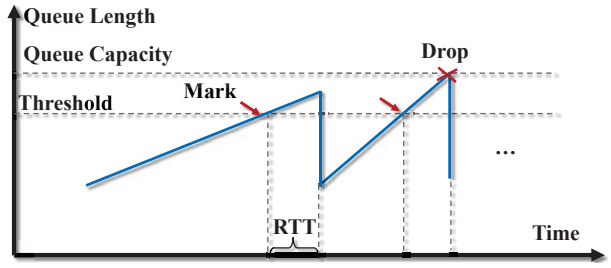
The ECN marking scheme in switches is the key to realizing high throughput and low latency. Excessive marking of packets impairs the throughput of senders, while a small number of markings causes queue buildup and increases queue delay. Therefore, numerous studies focus on more reasonable marking schemes to improve existing congestion control algorithms. HULL [24] is a scheme based on phantom queue (PQ) whose bandwidth is slightly lower than the real bandwidth. It uses the queue length of PQ to trigger marking and loses some bandwidth to achieve almost zero queue in the switch. ECN# [25] marks packets based on both instantaneous and persistent congestion states to handle the problem of round trip time (RTT) variations in DCNs. ACC [26] uses machine learning technology to automatically adjust the marking threshold. It uses the reinforcement learning algorithm, sets the queue length, switch output rate and bandwidth as input of the algorithm, and sets the thresholds and marking probability as output to optimize the ECN setting. These schemes are actually based on the current (instantaneous or average) queue length. However, these schemes do not take queue changes into account. Owing to the delay of ECN feedback, the queue status changes by the time congestion notification reaches the senders. Therefore, S-ECN [34] uses the slope of queue growth to mark packets with a linearly increasing probability, which is more sensitive to queue changes. However, its design does not consider the queue length and thus cannot cope with the situation of low queue growth rate but high queue length.

B. Motivation

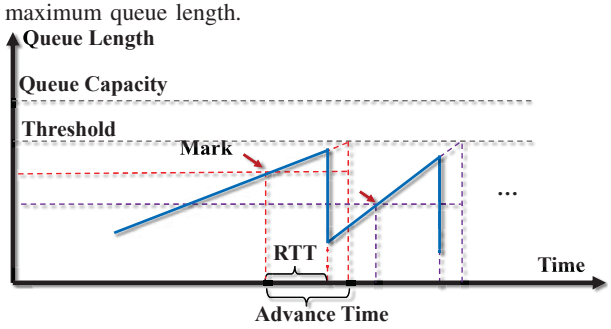
The motivation of our scheme comes from the uncontrollable maximum queue length caused by the hysteresis of ECN feedback and the marking decision based on the current queue length that most schemes use.

According to the ECN feedback process, when a switch marks the arriving packet, the marked packet is transmitted to the receiver, which then feeds back congestion notification to the sender by the acknowledgment (ACK) packet. After the sender receives the ACK packet, it reacts to congestion and cuts its congestion window to reduce send rate. When the low-rate data packets arrive at the switch, the input rate and queue length of the switch decrease accordingly. The entire process takes about one RTT, which is the hysteresis of ECN feedback. Such hysteresis of feedback is an inherent characteristic of ECN and thus is difficult to be eliminated.

However, most existing schemes are based on the current (instantaneous or average) queue length exceeding the threshold to mark packets. When the current queue length reaches the threshold, the ECN marking is triggered. Due to the hysteresis of ECN feedback, the queue length may further increase, exceed the threshold and reach its peak before the congestion notification is delivered to senders. The size of the peak depends on the input rate of the network. Assuming that the rate does not change during this period, the peak value is approximately equal to $threshold + rate \times RTT$. Fig. 1(a) shows how the queue length changes. The peak value of the queue length is larger than the threshold and is rate dependent. When the flow rate further increases due to a burst, the peak of the queue becomes far beyond the threshold, which may result in queue overflow and serious packet loss. Therefore, the schemes based on the current queue length have the defect



(a) Schemes based on current queue length fail to control the maximum queue length.



(b) Early marking based on predicted queue length limits the maximum queue length.

Fig. 1: The queue process of two ECN marking schemes.

of uncontrollable maximum queue length and lack adaptability to dynamic networks.

In order to further illustrate this problem, we conduct an experiment in which multiple senders simultaneously transmit data to a receiver through a bottleneck link. We choose DCTCP as the congestion detection and control algorithm. We change the number of flows and measure the average and maximum queue length of the bottleneck switch. Fig. 2 shows the result. Although the average queue length is relatively stable with the increase in the number of flows, the maximum queue length continuously increases linearly. This experimental result also confirms that such schemes based on the current queue length lack control over the maximum queue length.

Conclusion: The hysteresis of ECN is inherent and the delay of ECN feedback is difficult to be eliminated, while most existing schemes mark packets based on the current queue length. During the feedback period, the maximum queue length of the switch will exceed the threshold and is related to the flow rate. Therefore, the existing schemes cannot control the maximum queue length and is easy to cause queue overflow when a burst occurs.

III. The E-ECN Scheme

The design of E-ECN is motivated by the uncontrollable maximum queue length problem described above. The goal of E-ECN is to limit the peak of the queue and make it generally predictable.

The difficulty in solving this problem is that we cannot eliminate the hysteresis of feedback or speed up the transmission of ECN. Our innovative idea is that although the delay of ECN feedback cannot be eliminated, it can be neutralized

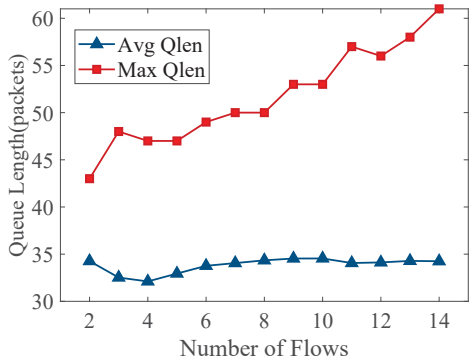


Fig. 2: The maximum and average queue length versus the number of flows.

by marking packets in advance. A switch can advance the marking time to offset the delay of ECN feedback. When senders are notified to reduce their rate and the queue length decreases, the peak of the queue is still within the predictable range. We denote the advance time as T_a. The peak value is approximately equal to $(threshold - rate \times T_a) + rate \times RTT$. If $T_a > RTT$, the peak value can generally be controlled below the threshold as shown in Fig. 1(b). We prove in the subsequent sections that by carefully selecting the advance time, this scheme does not impact the throughput of senders.

The early marking uses the predicted queue length to trigger marking instead of the current queue length. We predict the queue length after the advance time, and incoming packets will be marked if the predicted value is greater than the threshold. According to our experience, the prediction of queue length does not need to be very accurate because from the goal of reducing queuing delay with a throughput guarantee, the accuracy of the prediction model is not the primary influencing factor. Using accurate but complex prediction models is not necessary and can result in higher computational expenses. Therefore, we use a low-complexity but effective linear prediction approach and can generally limit the peak of the queue to a controllable range.

In the rest of this section, we first introduce the specific design of E-ECN, then explain its benefits, and finally give the selection of parameters in the scheme and explain how this selection can achieve early marking without affecting the throughput.

A. Scheme Design

E-ECN adopts an early marking scheme based on predicted queue length instead of current queue length. A simple but effective way to predict is based on the queue growth rate. Therefore, estimation of the queue growth rate is the initial step. For easy implementation, we use the average rate in a time window as the estimate of the queue growth rate. The time window is denoted as T_i. We denote the queue length in the current and before the time window T_i as $Qlen$ and $Qlen\_last$, respectively. Then the queue growth rate is

estimated by:

$$rate = (Qlen - Qlen\_last)/T_i. \quad (1)$$

E-ECN marks packets based on the predicted queue length. We denote the predicted queue length after the advance time T_a as $Qlen\_pred$. The predicted value $Qlen\_pred$ is given by:

$$Qlen\_pred = Qlen + rate \times T_a. \quad (2)$$

When packets arrive, E-ECN uses the predicted queue length $Qlen\_pred$ instead of the current queue length for marking decisions. If $Qlen\_pred$ exceeds the threshold, the packet will be marked. The congestion control algorithm takes effect after about one RTT, and then the queue length decreases accordingly. The advance time T_a offsets the delay of ECN feedback so that the maximum queue length is controlled below the threshold generally. Algorithm 1 shows the implementation of E-ECN.

Algorithm 1: Package Processing before Enqueue

input: The packet P.

Get the current time T_now, the current queue length $Qlen$, the last update time T_last, and last update queue length $Qlen\_last$;
if *The Queue is full* **then**
  Drop the packet P;
end
if $T\_now - T\_last \geq T_i$ **then**
  $rate \leftarrow (Qlen - Qlen\_last)/(T\_now - T\_last)$;
  $T\_last \leftarrow T\_now$;
  $Qlen\_last \leftarrow Qlen$;
end
if $Qlen + rate \times T_a \geq threshold$ **then**
  Mark the packet P;
end
Put P into the queue

B. Benefits

E-ECN enjoys the following benefits from the early marking scheme based on predicted queue length.

Controllable maximum queue length: For the marking schemes based on the current queue length, when a congestion notification is made, due to the delay of ECN feedback, the queue length exceeds the threshold before senders reduce their rate. The maximum queue length is related to the rate and is not controllable for these schemes. E-ECN uses the predicted queue length to mark packets in advance. When it indicates congestion to the senders, the actual queue length is below the threshold, and the advance time neutralizes the delay of ECN feedback. When the senders reduce their rate, the queue length decreases, and the maximum queue length does not exceed the threshold. Therefore, E-ECN achieves controllable maximum queue length.

Adaptability: Queue length and queue growth rate are important parameters for marking decisions. The marking schemes based on the current queue length only use the queue length and do not consider the variation of flow rate, thus lacking adaptability to dynamic networks. S-ECN only uses the slope of the queue length and lacks judgment on the current congestion level, thus cannot cope with the situation of low queue growth rate but high queue length. The E-ECN scheme uses both the queue length and the slope of the queue length, which can accurately evaluate the level of congestion and the further evolution of congestion. Therefore, E-ECN achieves better adaptability in DCNs.

C. Guidelines for Choosing Parameters

The time window T_i and the advance time T_a are two important parameters for E-ECN. In this subsection, we show the guidelines for choosing them and explain that the throughput is not impaired by careful selection of T_a.

The time window T_i: T_i determines the accuracy of rate estimating and queue length predicting. On the one hand, a smaller T_i makes the average rate in the time window closer to the real queue growth rate, thereby increasing the sensitivity of the scheme to burst. On the other hand, if T_i is less than the transmission time of a packet, the queue cannot change within the time window, and the rate calculation will fail. Therefore, T_i should be larger than the transmission time of a package, i.e.,

$$T_i > \frac{MTU}{bandwidth}. \quad (3)$$

Otherwise, the predicted queue length is unreasonable, and false judgment of the burst is prone to be caused. According to our experiments, T_i is generally set to $1.1 \sim 1.5\frac{MTU}{bandwidth}$ to achieve satisfactory results.

The advance time T_a: T_a is the key to the effectiveness of the scheme. E-ECN aims at controllable maximum queue length with a bandwidth utilization guarantee. According to the analysis in this section, only when the advance time T_a is greater than RTT can the peak of the queue be theoretically limited below the threshold. However, excessive T_a leads to overcontrolling and reduces the throughput.

We use fluid model [22] to obtain a reasonable range of T_a and prove that early marking will not affect the throughput of senders. Here we use the DCTCP as the end host congestion

TABLE I: Variables in the fluid model.

Variables	Description
t	Time
N	Number of flows
Q	Queue length
Q_m	Queue length at the start of marking
W	Congestion window of a sender
W_m	Congestion window at the start of marking
W_{max}	Maximum congestion window
W_{min}	Minimum congestion window
RTT	Round trip time
C	Link bandwidth
K	Marking threshold
α	Fraction of marked packets
T_{amax}	Maximum T_a without affecting throughput

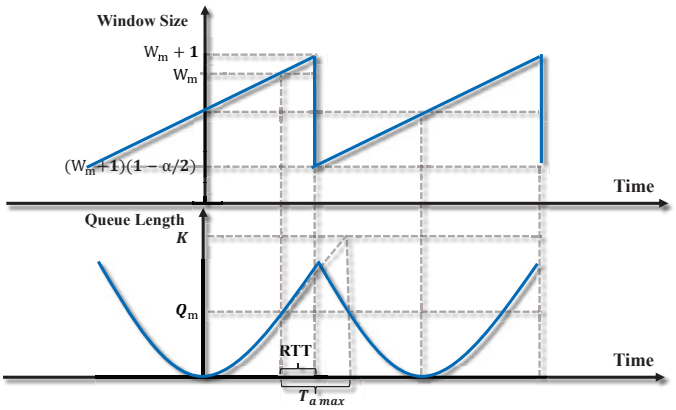


Fig. 3: Process of window size and queue length under the critical condition.

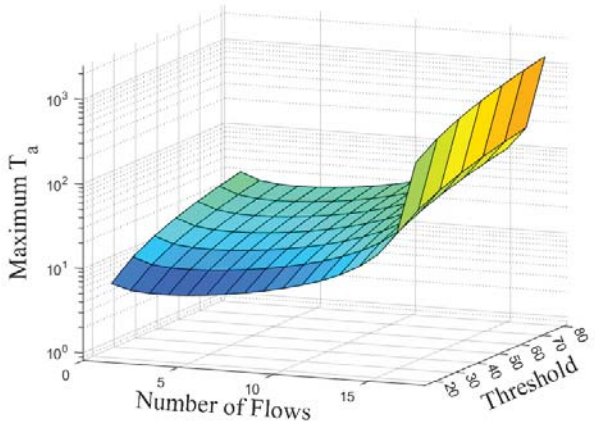


Fig. 4: The maximum value of T_a versus the number of flows with different thresholds.

control algorithm. Variables used in the fluid model are listed in Table I.

Fig. 3 shows the critical case where the lowest point of the queue is zero and T_a gets its maximum. If T_a is set to be greater than the maximum, the queue will be idle for a period, resulting in low link utilization and impaired throughput of senders.

The queue length versus time t can be calculated as:

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \frac{NW}{RTT} - C. \tag{4}$$

We set the queue to the lowest point when t equals 0, and the slope is 0 due to the continuity. The window W grows by 1 packet per RTT. Therefore, the window and queue length at time t are given:

$$W(t) = \frac{1}{RTT}t + \frac{C \times RTT}{N}, \tag{5}$$

$$Q(t) = \int_0^t \mathrm{d}Q = \frac{N}{2RTT^2}t^2. \tag{6}$$

According to the algorithm of DCTCP, the maximum value of window $W_{max} = W_m + 1$ and minimum values of window $W_{min} = (W_m + 1)(1 - \alpha/2)$. The average value of W makes the slope of Q zero. Hence,

$$\frac{W_{min} + W_{max}}{2} = \frac{C \times RTT}{N}. \tag{7}$$

The fraction of the marked packet is given by:

$$\alpha = \frac{W_m}{W_{min} + (W_{min} + 1) + \cdots + W_m}. \tag{8}$$

Here we only consider the case where the number of flows is not very large and assume that $\frac{C \times RTT}{N} >> 1$. Because when $N > C \times RTT$ even if the window of each flow is 1, the bandwidth can be fully utilized and there is no over control. Then we get $\alpha \approx \sqrt{\frac{2N}{C \times RTT}}$. We denote $\frac{C \times RTT}{N}$ as λ, then W_m and Q_m can be calculated:

$$W_m = \frac{4\lambda^2 - 4\lambda + \sqrt{2\lambda}}{4\lambda - \sqrt{2\lambda}}, \tag{9}$$

$$Q_m = \frac{N}{2} \times (\frac{\sqrt{2\lambda}\lambda - 4\lambda + \sqrt{2\lambda}}{4\lambda - \sqrt{2\lambda}})^2. \tag{10}$$

According to the E-ECN marking scheme:

$$Q_m + \left.\frac{\mathrm{d}Q}{\mathrm{d}t}\right|_{Q=Q_m} \times T_{amax} = K, \tag{11}$$

we can get:

$$T_{amax} = \frac{K}{N} \times A - \frac{1}{2A} \quad (in\ RTTs), \tag{12}$$

where:

$$A = \frac{2\sqrt{2\lambda} - 1}{\lambda + 1 - 2\sqrt{2\lambda}}. \tag{13}$$

Fig. 4 shows the maximum of T_a under the different thresholds. We can get that T_a has a wide range of values. In most cases, the maximum of T_a that does not have a negative impact on throughput is even greater than ten times the RTT, according to (12) and Fig. 4. In DCNs, RTT is not easy to measure on switches, so we generally set T_a to multiple base RTTs. According to Fig. 4, setting T_a to $1 \sim 5$ base RTTs can control the maximum queue length with a bandwidth utilization guarantee.

In Section IV-C, we conduct experiments to further discuss the effect of parameters on the scheme and confirm the above analysis and guidelines.

IV. Performance Evaluation

In this section, we evaluate the performance of E-ECN by ns-3 [32] simulation. We focus primarily on the following performance: the average and maximum queue length in steady state, fairness, convergence speed, the maximum queue length distribution when a burst occurs, throughput, and the flow completion time (FCT) in large-scale DCN. These are the requirements of traffic in DCNs.

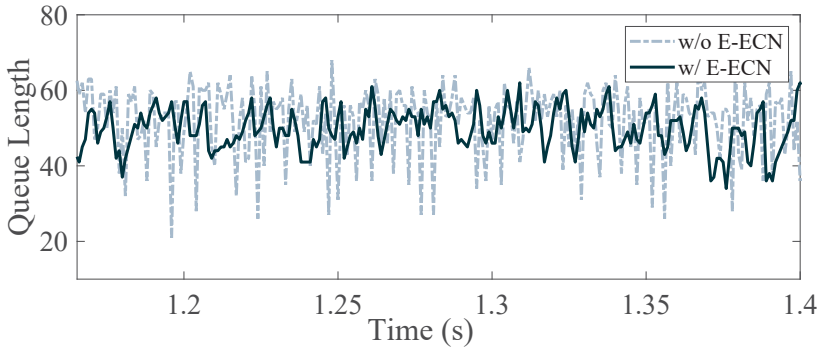


Fig. 5: The time series of queue length.

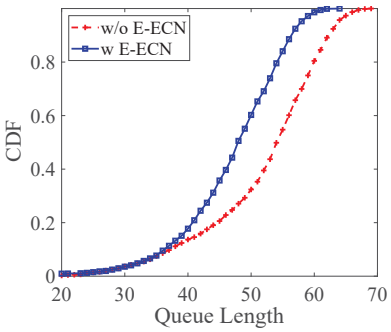


Fig. 6: The CDF of queue length.

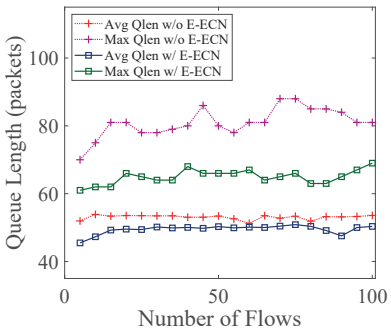


Fig. 7: The average and maximum queue Length.

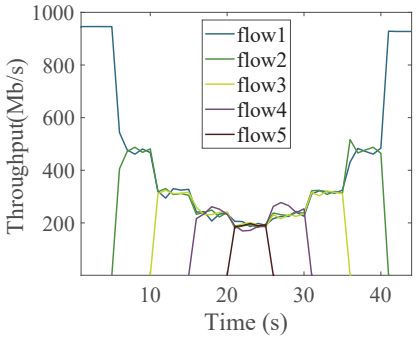


Fig. 8: Bandwidth sharing between flows.

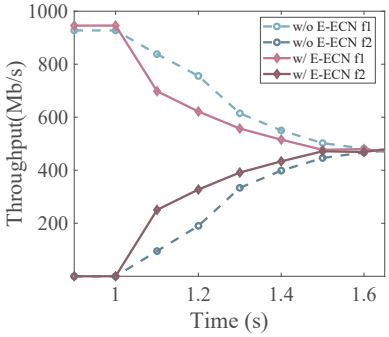


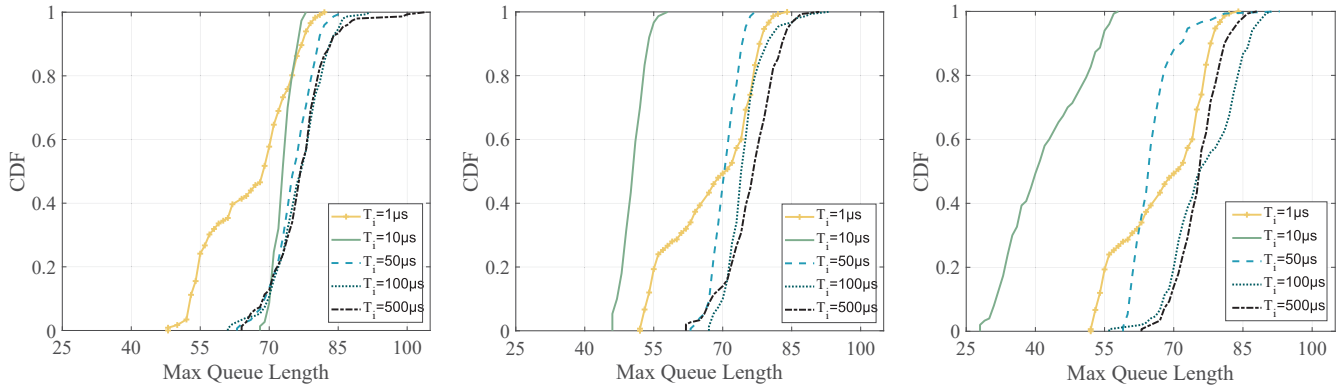
Fig. 9: Convergence speed of DCTCP with/without E-ECN.

A. *Performance Evaluation in Steady State*

We first evaluate the performance of E-ECN in the steady state, that is, the queue condition when multiple long-lived flows reach the same throughput. We use a many-to-one topology to evaluate the queuing performance in the steady state. The bandwidth of each link is 10Gbps, and the link delay is 1μs. In E-ECN, we set $T_i = 1\mu s$ and T_a to about 3 base RTTs. E-ECN can coordinate with other host congestion control algorithms. In this test, we use DCTCP as the end-to-end congestion control algorithm, which is more commonly used in DCNs. We set the marking threshold to 60 packets. We observe the average and maximum queue length and throughput of multiple long-lived flows when they reach the steady state.

Fig. 5 shows the time series of queue length of DCTCP with/without E-ECN and Fig. 6 shows the cumulative distribution function (CDF) of queue length. On the one hand, E-ECN reduces queue length and achieves lower queue delay than the marking scheme based on current queue length by early marking. With E-ECN, both the average and maximum queue lengths are lower than those without E-ECN. About 98.6% of the queue length is limited below the threshold by early marking packets in steady state, while about 20% of the queue length exceeds the threshold and is not limited without E-ECN. On the other hand, E-ECN achieves less queue jitter. With ECN, the queue length typically varies in the range of 40 to 60 packets, while the baseline is in a wide range. Thus E-ECN also helps to reduce delay jitter.

We next change the number of long-lived flows to further verify whether the maximum queue length is controllable when the input rate of the switch changes. We start with 5 flows and gradually increase to 100 flows in intervals of 5 flows. When they enter the steady state, we measure the average and maximum queue length. We still compare the performance difference of DCTCP with and without E-ECN. Fig. 7 shows the average and maximum queue length versus the number of flows. Without E-ECN, though the average queue length is below the threshold, the maximum queue length is uncontrollable and unpredictable. When the number of flows is less than 15, the maximum queue length increases approximately linearly with the number of flows. When there are a large number of flows, the maximum queue length is uncontrollable and unpredictable. E-ECN indicates congestion based on predicted queue length to limit the peak of the queue in time. Therefore, it can achieve more stable maximum queue length. As shown by Fig. 7, the maximum queue length of the E-ECN scheme is generally controlled around the threshold. Besides, E-ECN achieves lower average queue length in the steady state, which effectively increases the burst tolerance of the network. We also measure the link utilization with and without E-ECN, and the result is that they are both almost fully utilized. Therefore, it proves that E-ECN achieves the controllable maximum queue length with a bandwidth utilization guarantee.



(a) T_a=1RTT (b) T_a=5RTTs (c) T_a=10RTTs

Fig. 10: The CDF of maximum queue length under E-ECN with different parameters.

B. Fairness and Convergence Speed

Fairness is a traffic requirement that is sometimes as important as high bandwidth and low latency. Fair queueing needs to provide fair bandwidth allocation to network traffic by ensuring that each flow gets its fair share [35]. We use the convergence test to prove that E-ECN allows the flows to converge quickly to their fair share. We set five senders and a receiver connected by a switch and use DCTCP as the end-to-end congestion control algorithm. In the switch, we use the E-ECN marking scheme. We start each sender in turn at an interval of 5 seconds, and then stop them in reverse order. The bandwidth of each link is 1Gbps. Fig. 8 shows the bandwidth occupied by each flow in the switch. When multiple flows enter the switch, the bandwidth allocated to each flow is almost the same. When a new flow enters the switch, it can quickly get its fair share of bandwidth. After a flow stops, other flows can also use the remaining bandwidth quickly. E-ECN can achieve fair sharing of bandwidth in the switch.

To further evaluate the convergence speed of E-ECN, we then set up two flows to start one after the other. We still use DCTCP as the end-to-end congestion control algorithm and measure bandwidth convergence in a finer granularity with and without the E-ECN marking scheme. Fig. 9 shows the convergence speed of DCTCP with/without E-ECN. With E-ECN, the incoming flow can occupy bandwidth faster and then reach the fair share of bandwidth. While for DCTCP without E-ECN, it takes a longer time to converge to the fair share. This is because that E-ECN is more sensitive to changes in rate, and when the second flow is injected, it can make feedback earlier so that the window of the two flows can be adjusted to the fair share state as soon as possible. Therefore, E-ECN achieves more quick convergence to the fair share. As shown by Fig. 9, E-ECN reduces convergence time by 16%.

C. Performance Evaluation in Bursts

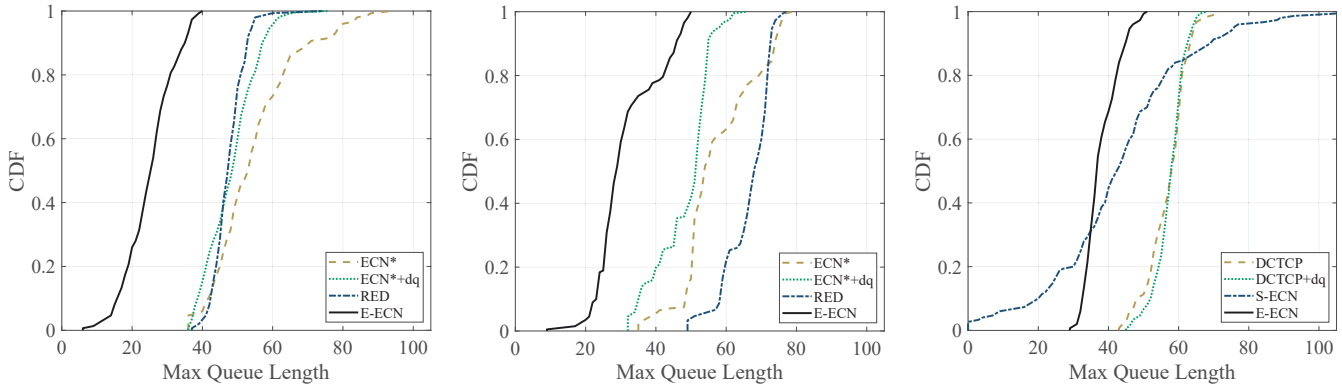
Our goal is not only to have a controllable maximum queue length in the steady state but also to provide adaptive control in the initial stages of bursts. Therefore we also have to evaluate the queuing performance of E-ECN under bursts. We make some long-lived flows enter the steady state, then inject multiple burst flows. The maximum queue length after the burst is related to the burst injection time. If bursts are injected while the queue is at peak, the maximum queue length increases significantly. Conversely, when bursts are injected at the bottom of the queue, the maximum queue length is better controlled. Therefore, we change the occurrence time of burst traffic and measure the maximum queue length distribution.

We first explore the effect of different the time window T_i and the advance time T_a on E-ECN. We set $threshold = 50$packets, $MTU = 1000$Bytes and $bandwidth = 1$Gbps. Fig. 10 shows the CDF of maximum queue length in E-ECN with different T_i and T_a.

On the one hand, we can confirm the impact of the time window T_i on E-ECN. It can be seen from the subplots of Fig. 10 that when $T_i = 1\mu$s, the CDF of the maximum queue length is different from the others, which is irregular and almost does not change with T_a. This is because $1\mu\text{s} < MTU/bandwidth$, according to the analysis in Section III-C, the time window T_i is less than the transmission time of a packet, and the queue cannot change within the time window, and the rate calculation fails which leads to incorrect prediction of queue length. When $T_i \geq 10\mu$s (larger than $MTU/bandwidth$), as shown by Fig. 10, the smaller time window T_i achieves lower maximum queue length distribution. When $T_i = 10\mu$s which is slightly larger than $MTU/bandwidth$, E-ECN obtains the best effect. The result confirms that the smaller T_i can achieve higher sensitivity to burst and better peak suppression capability, which conforms to the analysis of parameters in Section III-C.

On the other hand, we can also evaluate the role of advance time T_a in the scheme. As shown in Fig. 10, intuitively, a larger T_a causes earlier marking, and the queue will have a lower peak distribution. When T_a is about 5RTTs, E-ECN can control 45% of the maximum queue length below the threshold when a burst occurs. When T_a is about 10RTTs, 76% of the maximum queue length is controlled below the threshold.

We next evaluate E-ECN comparatively with other AQMs under different congestion controls. ECN* is used to enhance TCP. It is mentioned that dequeue (dq) marking can speed



(a) Reno (b) Cubic (c) DCTCP

Fig. 11: The CDF of maximum queue length under different AQMs and congestion control algorithms.

up the delivery of the ECN [33]. Therefore, we compare with ECN* and ECN*+dq under Reno and Cubic, besides, ECN-enabled RED [36]. S-ECN is a probabilistic marking method according to the slope of queue length growth. S-ECN mainly uses DCTCP as end host congestion control [34]. Therefore, we compare E-ECN with S-ECN, the marking scheme of DCTCP (hereinafter referred to as DCTCP) as well as DCTCP+dq under DCTCP congestion control. RED does not apply to DCTCP [22], so we do not evaluate RED under DCTCP congestion control. To be fair, all AQM thresholds are set to be 30 packets. In RED, the low threshold is half of the high threshold, and the high threshold is the same as the thresholds of other AQMs. In E-ECN, we set T_i = 10μs and T_a to be about 5RTTs.

Fig. 11 shows the evaluation results. When choosing Reno and Cubic for congestion control algorithms, ECN* and RED are schemes that mark packets based on the current (instantaneous and average) queue length and are therefore inadequate in limiting the maximum queue length. Although dq speeds up the delivery of ECN and improves ECN* by dequeue marking, most of the delay of ECN feedback still cannot be eliminated. Thus, ECN*+dq has limited performance in suppressing the peak of the queue. Compared with other schemes based on the current queue state, E-ECN achieves the lowest maximum queue length distribution and has better burst tolerance under Reno and Cubic. Under Reno with E-ECN, about 75% of the maximum queue length is below the threshold. Conversely, the maximum queue length in other AQMs exceeds the threshold.

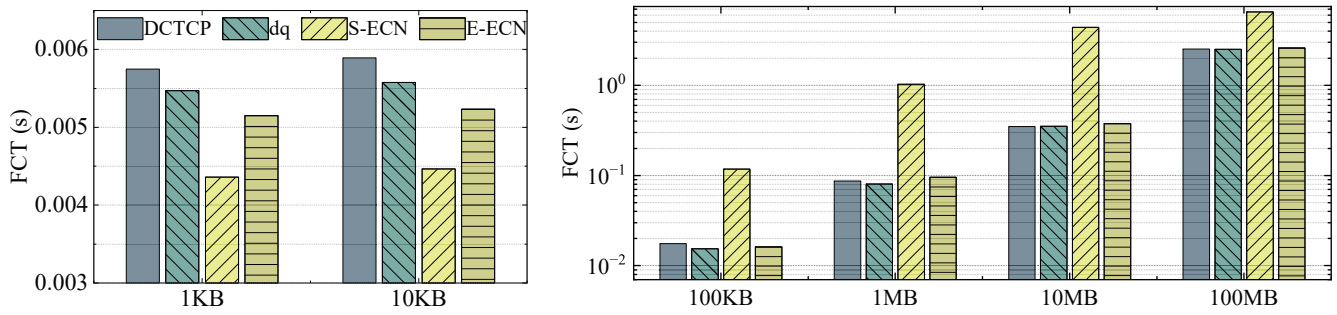
As shown by Fig. 11(c), when using DCTCP as the congestion detection and control algorithm, the maximum queue length has a high distribution, while its enhanced scheme DCTCP+dq slightly improves performance. As a pure rate-based scheme, S-ECN has a wider distribution of queue peaks, because it is not sensitive to the queue length. It may lead to over marking when the queue length is low but the rate is high, which reduces the link utilization, while may lack marking when the queue length is high but the rate is low, resulting in a large queue peak. Table II shows the average throughput of each scheme during the experiment. Except for S-ECN other schemes achieve almost full link utilization, while it affects the throughput of senders due to not using the queue length for marking decisions. E-ECN is a scheme that combines the queue length and the growth rate and uses them for congestion prediction. Therefore, E-ECN is able to sense the current congestion level well, and when a burst occurs, it can also predict the further change of congestion by queue growth rate and indicates the congestion in time. From Fig. 11 and Table II, it is concluded that E-ECN achieves controllable maximum queue length in general with a bandwidth utilization guarantee.

TABLE II: Average throughput of each scheme

	Throughput (Mbps)	Link Utilization (%)
DCTCP	971.017	94.825
ECN*	972.019	94.826
dq	971.017	94.825
RED	971.018	94.826
S-ECN	**835.054**	**81.548**
E-ECN	971.018	94.826

D. Large-scale Simulation

To simulate a real large-scale DCN, we simulate a 3-tier Fat-Tree topology including 128 nodes. Each core switch connects to the aggregation switches by the 100Gbps link. The bandwidth between aggregation switches and edge switches is 10Gbps and edge switches have 1Gbps links connecting to multiple servers. We choose DCTCP as the congestion control algorithm for end hosts. We compare E-ECN with DCTCP, DCTCP+dq, and S-ECN. All thresholds are set to be 50 packets. In E-ECN, we set T_i according to the bandwidth of the switch ports, and T_a to 0.3ms. To simulate complex traffic patterns in DCNs, we set the background traffic based on the measurements of [22]. A shorter flow completion time (FCT) is a requirement for all flows [37] and a comprehensive evaluation of the performance of the schemes. Therefore, we randomly inject flows of different sizes and measure their FCT with different schemes.



(a) Short flows

(b) Medium and long flows

Fig. 12: FCT comparison with different sizes of flows.

Fig. 12 shows the average FCT of each size flow. Fig. 12(a) shows the FCT of short flows which are usually completed in one RTT and are the main traffic in the DCNs. The main factor affecting the transmission delay of short flows is the queuing delay. By dequeue marking, dq can speed up the delivery of ECN and achieve a shorter average queue length. Thus, dq shortens the FCT compared to DCTCP. Compared to the schemes only based on queue length, the FCT of short flows under S-ECN and E-ECN are smaller because of shorter queue lengths. S-ECN has the advantage in reducing FCTs for short flows. In these schemes, S-ECN achieves the shortest FCT of short flows. However, S-ECN is actually an over-controlling scheme, which increases the FCT of long flows, as shown by Fig. 12(b). For long flows, throughput is the key to FCT. S-ECN overly indicates congestion, making the congestion window of the sender small. Although the queues of the switches on the path are at low occupancy, many bandwidth resources are not utilized. E-ECN marks packets based on the predicted queue length, so congestion can be indicated at the initial stage of queue establishment, achieving shorter queuing delay. In addition, by choosing the proper parameters, E-ECN does not over-control and impair the throughput. Therefore, E-ECN achieves a balance of throughput and latency.

TABLE III: Normalized packet loss rate of each scheme

	E-ECN	S-ECN	DCTCP	dq
Normalize PLR	1	0.8	1.76	1.41

To further evaluate the performance, we measure the packet loss rate (PLR) of the entire network in the test. Table III shows the normalized RLR. DCTCP lacks adaptability to dynamic networks and the queue may overflow when a burst occurs. The dq scheme speeds up the feedback of ECN, further reducing the PLR and enhancing robustness. E-ECN can detect the changes in the growth rate of the queue, thereby having a higher burst tolerance, and greatly reducing packet loss. Compared to DCTCP, E-ECN reduces packet loss by 43%. While S-ECN achieves less packet loss, as mentioned above this is achieved through over-control and therefore does not provide a better quality of service in DCNs.

V. Conclusion

In this paper, we theoretically and experimentally revealed the uncontrollable maximum queue length problem of most existing schemes, which is caused by the delay of ECN feedback and the marking decision based on the current queue state. This defect may lead to queue overflow due to the limited buffer of the switches in DCNs when burst traffic occurs. To solve this problem, we proposed an early marking scheme, named E-ECN. E-ECN marks packets based on the predicted queue state instead of the current queue length to indicate congestion in advance. The advance time offsets the delay of ECN feedback so that the maximum queue length can be controlled before it exceeds the threshold. We utilized a fluid model to demonstrate that, with appropriate selection of the advance time, E-ECN does not have a negative impact on throughput. Moreover, we provided guidelines for selecting appropriate parameter values. Our performance evaluation results show that E-ECN delivers a high tolerance to bursts. E-ECN achieves a lower maximum queue length distribution to handle bursts, and can effectively avoid packet loss caused by queue overflow.

Acknowledgment

The work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61972371, Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093, and the Fundamental Research Funds for the Central Universities.

References

[1] W. Cheng, K. Qian, W. Jiang, T. Zhang, and F. Ren, "Re-architecting congestion management in lossless ethernet," in *Proceedings of the 17th Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2020, pp. 19–36.

[2] S. Jha, A. Patke, J. Brandt, A. Gentile, B. Lim, M. Showerman, G. Bauer, L. Kaplan, Z. Kalbarczyk, W. Kramer, and R. Iyer, "Measuring congestion in high-performance datacenter interconnects," in *Proceedings of the 17th Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2020, pp. 37–58.

[3] Y. Zhang, Y. Liu, Q. Meng, and F. Ren, "Congestion detection in lossless networks," in *Proceedings of the 2021 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2021, pp. 370–383.

[4] X. Zhong, J. Zhang, Y. Zhang, Z. Guan, and Z. Wan, "PACC: Proactive and accurate congestion feedback for RDMA congestion control," in *Proceedings of the 2022 IEEE Conference on Computer Communications (INFOCOM)*, 2022, pp. 2228–2237.

[5] A. M. Abdelmoniem and B. Bensaou, "T-RACKs: A faster recovery mechanism for TCP in data center networks," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1074–1087, 2021.

[6] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, "HPCC: High precision congestion control," in *Proceedings of the 2019 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2019, pp. 44–58.

[7] J. Ros-Giralt, N. Amsel, S. Yellamraju, J. Ezick, R. Lethin, Y. Jiang, A. Feng, L. Tassiulas, Z. Wu, M. Y. Teh, and K. Bergman, "Designing data center networks using bottleneck structures," in *Proceedings of the 2021 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2021, pp. 319–348.

[8] M. Kheirkhah and M. Lee, "AMP: An adaptive multipath TCP for data center networks," in *Proceedings of the 2019 IFIP Networking Conference (IFIP Networking)*, 2019, pp. 1–9.

[9] Y. Gao, Q. Li, L. Tang, Y. Xi, P. Zhang, W. Peng, B. Li, Y. Wu, S. Liu, L. Yan, F. Feng, Y. Zhuang, F. Liu, P. Liu, X. Liu, Z. Wu, J. Wu, Z. Cao, C. Tian, J. Wu, J. Zhu, H. Wang, D. Cai, and J. Wu, "When cloud storage meets RDMA," in *Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2021, pp. 519–533.

[10] S. Agarwal, R. Agarwal, B. Montazeri, M. Moshref, K. Elmeleegy, L. Rizzo, M. A. de Kruijf, G. Kumar, S. Ratnasamy, D. Culler, and A. Vahdat, "Understanding host interconnect congestion," in *Proceedings of the 21st ACM Workshop on Hot Topics in Networks (HotNets)*, 2022, pp. 198–204.

[11] Y. Zhang, X. Nie, J. Jiang, W. Wang, K. Xu, Y. Zhao, M. J. Reed, K. Chen, H. Wang, and G. Yao, "BDS+: An inter-datacenter data replication system with dynamic bandwidth separation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 918–934, 2021.

[12] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "TIMELY: RTT-based congestion control for the datacenter," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM)*, 2015, pp. 537–550.

[13] G. Kumar, N. Dukkipati, K. Jang, H. M. G. Wassel, X. Wu, B. Montazeri, Y. Wang, K. Springborn, C. Alfeld, M. Ryan, D. Wetherall, and A. Vahdat, "Swift: Delay is simple and effective for congestion control in the datacenter," in *Proceedings of the 2020 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2020, pp. 514–528.

[14] B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout, "Homa: A receiver-driven low-latency transport protocol using network priorities," in *Proceedings of the 2018 ACM Conference on Special Interest Group on Data Communication*, 2018, pp. 221–235.

[15] D. Gibson, H. Hariharan, E. Lance, M. McLaren, B. Montazeri, A. Singh, S. Wang, H. M. G. Wassel, Z. Wu, S. Yoo, R. Balasubramanian, P. Chandra, M. Cutforth, P. Cuy, D. Decotigny, R. Gautam, A. Iriza, M. M. K. Martin, R. Roy, Z. Shen, M. Tan, Y. Tang, M. Wong-Chan, J. Zbiciak, and A. Vahdat, "Aquila: A unified, low-latency fabric for datacenter networks," in *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2022, pp. 1249–1266.

[16] Y. Zhang, G. Kumar, N. Dukkipati, X. Wu, P. Jha, M. Chowdhury, and A. Vahdat, "Aequitas: Admission control for performance-critical rpcs in datacenters," in *Proceedings of the 2021 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2022, pp. 1–18.

[17] S. McClure, A. Ousterhout, S. Shenker, and S. Ratnasamy, "Efficient scheduling policies for Microsecond-Scale tasks," in *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2022, pp. 1–18.

[18] M. A. Qureshi, Y. Cheng, Q. Yin, Q. Fu, G. Kumar, M. Moshref, J. Yan, V. Jacobson, D. Wetherall, and A. Kabbani, "PLB: Congestion signals are simple and effective for network load balancing," in *Proceedings of the 2022 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2022, pp. 207–218.

[19] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 640–656, 2017.

[20] A. M. Abdelmoniem and B. Bensaou, "Hysteresis-based active queue management for tcp traffic in data centers," in *Proceedings of the 2019 IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 1621–1629.

[21] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "The addition of explicit congestion notification (ECN) to IP," 2001, RFC 3168, Accessed on Feb., 2023. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8311

[22] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in *Proceedings of the 2010 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2010, pp. 63–74.

[23] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, "Congestion control for large-scale RDMA deployments," in *Proceedings of the 2015 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2015, pp. 523–536.

[24] M. Alizadeh, A. Kabbani, T. Edsall, B. Prabhakar, A. Vahdat, and M. Yasuda, "Less is more: Trading a little bandwidth for Ultra-Low latency in the data center," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, 2012, pp. 253–266.

[25] J. Zhang, W. Bai, and K. Chen, "Enabling ECN for datacenter networks with RTT variations," in *Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2019, pp. 233–245.

[26] S. Yan, X. Wang, X. Zheng, Y. Xia, D. Liu, and W. Deng, "ACC: Automatic ECN tuning for high-speed datacenter networks," in *Proceedings of the 2021 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2021, pp. 384–397.

[27] V. Addanki, O. Michel, and S. Schmid, "PowerTCP: Pushing the performance limits of datacenter networks," in *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2022, pp. 51–70.

[28] W. Bai, S. Hu, K. Chen, K. Tan, and Y. Xiong, "One more config is enough: Saving (DC)TCP for high-speed extremely shallow-buffered datacenters," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 489–502, 2021.

[29] V. Addanki, M. Apostolaki, M. Ghobadi, S. Schmid, and L. Vanbever, "ABM: Active buffer management in datacenters," in *Proceedings of the 2022 ACM Special Interest Group on Data Communication (SIGCOMM)*, 2022, pp. 36–52.

[30] P. Goyal, P. Shah, N. K. Sharma, M. Alizadeh, and T. E. Anderson, "Backpressure flow control," in *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2022, pp. 779–805.

[31] A. Sanaee, F. Shahinfar, G. Antichi, and B. E. Stephens, "Backdraft: a lossless virtual switch that prevents the slow receiver problem," in *Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2022, pp. 1375–1392.

[32] "Network Simulator 3," https://www.nsnam.org/about/, accessed on Feb., 2023.

[33] H. Wu, J. Ju, G. Lu, C. Guo, Y. Xiong, and Y. Zhang, "Tuning ECN for data center networks," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2012, pp. 25–36.

[34] D. Shan, F. Ren, P. Cheng, R. Shu, and C. Guo, "Micro-burst in data centers: Observations, analysis, and mitigations," in *Proceedings of the 2018 IEEE International Conference on Network Protocols (ICNP)*, 2018, pp. 88–98.

[35] Z. Yu, J. Wu, V. Braverman, I. Stoica, and X. Jin, "Twenty years after: Hierarchical Core-Stateless fair queueing," in *Proceedings of the 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2021, pp. 29–45.

[36] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[37] P. Taheri, D. Menikkumbura, E. Vanini, S. Fahmy, P. Eugster, and T. Edsall, "RoCC: Robust congestion control for RDMA," in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, 2020, pp. 17–30.