RLB: Reordering-Robust Load Balancing in Lossless Datacenter Networks

Jinbin Hu
Changsha University of Science and Technology
Changsha, China
jinbinhu@csust.edu.cn

Yi He
Changsha University of Science and Technology
Changsha, China
heyi@stu.csust.edu.cn

Jin Wang[*]
Changsha University of Science and Technology
Changsha, China
jinwang@csust.edu.cn

Wangqing Luo
Changsha University of Science and Technology
Changsha, China
luowangqing@stu.csust.edu.cn

Jiawei Huang
Central South University
Changsha, China
jiaweihuang@csu.edu.cn

ABSTRACT

Many existing load balancing mechanisms work effectively in lossy datacenter networks (DCNs), but they suffer from serious packet reordering in lossless Ethernet DCNs deployed with the hop-by-hop Priority-based Flow Control (PFC). The key reason is that the prior solutions are not able to correctly and timely perceive PFC triggering when making load balancing decisions. Once the forwarding path pauses transmission due to PFC triggering, the packets allocated on it are blocked, inevitably leading to out-of-order packets and retransmission. In this paper, we present a Reordering-robust Load Balancing (RLB) scheme with PFC prediction in lossless DCNs. At its heart, RLB leverages the derivative of ingress queue length to predict PFC triggering and proactively notifies the upstream switches to choose an appropriate rerouting path or perform packet recirculation to avoid reordering. As a building block for existing load balancing mechanisms, we have integrated RLB into Presto, LetFlow, Hermes and DRILL. The test results show that the RLB-enhanced solutions deliver significant performance by avoiding packet reordering. For example, it reduces the 99^{th} percentile flow completion time (FCT) by up to 58%, 67%, 72% and 54% over Presto, LetFlow, Hermes and DRILL, respectively.

CCS CONCEPTS

• **Networks** → **Network architectures**; **Data center networks**; **Routing protocols**.

KEYWORDS

Data Center, Lossless Networks, Load Balancing, Reordering

ACM Reference Format:
Jinbin Hu, Yi He, Jin Wang, Wangqing Luo, and Jiawei Huang. 2023. RLB: Reordering-Robust Load Balancing in Lossless Datacenter Networks. In *52nd International Conference on Parallel Processing (ICPP 2023), August 7–10, 2023, Holladay, UT, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3605573.3605617

1 INTRODUCTION

Modern datacenter applications such as Online Data Intensive (OLDI) services [1], Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) [2] and Non-Volatile Memory Express (NVMe) over Fabrics [3] require low-latency and lossless transmission to meet the increasing demands from customers. Even a single packet loss can greatly increase flow completion time (FCT) [3, 4]. To prevent buffer overflow, PFC mechanism is widely deployed in the converged enhanced Ethernet of datacenter networks [2–12]. PFC pauses the related upstream ports when the ingress queue length reaches a specified PFC threshold, and resumes transmission after the buffer occupancy decreases to the PFC threshold [9].

Datacenter networks enable rich parallel paths between host pairs and balance traffic among them to deliver high throughput. However, packets transmitted through multiple paths with different delay may arrive at the receiver out of order. Furthermore, limited by the small on-chip memory at network interface controller (NIC), lossless datacenter networks employ simple go-back-N retransmission scheme to deal with packet reordering. In the go-back-N scheme, the receiver's NIC discards the out-of-order packet and asks the sender to retransmit all packets that are sent after the last acknowledged packet, resulting in significant performance degradation.

In recent years, many load balancing schemes have been proposed to address the packet reordering problem [13–17]. Despite much progress in lossy datacenter networks, prior solutions cannot effectively avoid out-of-order packets in lossless datacenter networks. The key reason is that the existing load balancing schemes cannot correctly and timely perceive hop-by-hop PFC pausing when they make rerouting decisions in PFC-enabled datacenter networks. They have no consideration of the possibility that the selected forwarding path may be paused by PFC mechanism. Once the selected path is paused by PFC due to transient congestion, the later-sent

[*] Jin Wang is the corresponding author.

ICPP 2023, August 7–10, 2023, Holladay, UT, USA

ACM ISBN 979-8-4007-0843-5/23/08…$15.00
https://doi.org/10.1145/3605573.3605617

packets may arrive before the earlier-sent ones at the receiver, inevitably leading to packet reordering.

Given the above inefficiencies, we ask the following question: can we design a building block for the existing load balancing mechanisms to avoid packet reordering in lossless DCN? In this paper, we present RLB to answer this question affirmatively.

The key contribution of this work is to make the existing load balancing schemes still perform effective rerouting without out-of-order packets in lossless DCNs. We present RLB, a building block for existing load balancing schemes to eliminate packet reordering. RLB realizes its design goal by rerouting or recirculation[1] based on PFC prediction. Before making the final forwarding decision, RLB allows the existing load balancing mechanisms to deliberately consider whether the initial path selected by its own rerouting algorithm is potentially paused by PFC.

Specifically, RLB predicts PFC for each path by measuring the derivative of ingress queue length, independent of the egress queue. Once PFC is predicted to be triggered on a path, a congestion notification message (CNM) is generated as a warning message and directly sent to the upstream switch (§3.2.1). Then, the upstream switch considers how to choose appropriate forwarding paths for the arriving packets to avoid packet reordering (§3.2.2). If the difference in delays between the optimal and suboptimal paths selected by the load balancing mechanism is large, the upstream switch prefers to recirculate the packets from its egress port to its ingress port to let the packets stay on the switch for a little while to avoid reordering. Then the packets reconsider whether to choose the initial optimal path. On the contrary, if the difference in delays between the optimal and suboptimal paths is small, the arriving packets are rerouted to the suboptimal path with no PFC warning.

RLB is architecturally compatible with all existing load balancing schemes. We have integrated RLB into four existing load balancing schemes (i.e., Presto, LetFlow, Hermes and DRILL) with NS-3 simulator. The simulation results with realistic workloads indicate that RLB-enhanced solutions achieve significantly better performance than vanilla load balancing schemes. For example, under Web Search workload [19, 20], RLB+Presto, RLB+LetFlow, RLB+Hermes and RLB+DRILL reduce tail FCT by up to 58%, 67%, 72% and 54% compared to Presto, LetFlow, Hermes and DRILL, respectively.

The rest of this paper is organized as follows. Section 2 introduces background and motivation. Section 3 describes the design of our solution in detail. Section 4 evaluates the performance of RLB. Section 5 presents the related work, and Section 6 concludes this paper.

2 BACKGROUND AND MOTIVATION

2.1 Background

2.1.1 PFC Mechanism.
PFC is a hop-by-hop link-layer flow control mechanism, which is intended to eliminate packet loss due to congestion [2, 10, 21]. Fig. 1 shows the architecture of a typical PFC-enabled switch with shared memory. All packets are buffered in the shared memory pool. Each packet is counted in both ingress and egress queues. Once the ingress queue length exceeds the specified PFC threshold checked by memory management unit (MMU), the switch sends PFC PAUSE message to the upstream switch. Then the related upstream ports (or priorities) stop data transmission. When the given pause duration specified in the PAUSE message expires or a RESUME message is received once the queue length decreases to the PFC threshold, the upstream ports resume data transmission.

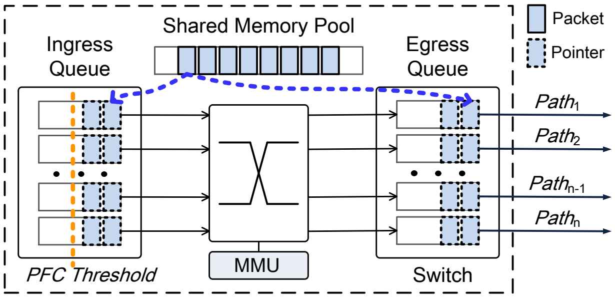


Figure 1: Architecture of shared memory switch.

However, PFC is a coarse-grained mechanism, which operates at port level and cannot distinguish between flows. This can potentially cause head-of-line (HoL) blocking and congestion spreading, etc., leading to serious performance damage for individual flows [2]. Specifically, suppose multiple flows from the same ingress at a switch, if one flow's egress port is paused by PFC from downstream switches, the other flows targeting different egress ports are also blocked, leading to HoL blocking. In addition, this back-pressure behavior potentially causes congestion spreading. Even if end-to-end congestion control protocols such as DCQCN [2], TIMELY [11], MP-RDMA [9], Swift [12] and PCN [3] are employed, PFC is inevitably triggered especially under the transient congestion due to uncontrollable bursty traffic.

2.1.2 Impact of Out-of-order Packets in Lossless DCN.
Although PFC mechanism can prevent packet loss due to buffer overflow, retransmission mechanism is still needed to recover the lost packets due to other reasons such as switch configuration errors, link failures or frame check sequence errors [22]. Since the memory of NIC is small, RoCEv2 protocol widely deployed in lossless DCN simply supports go-back-N algorithm. The retransmission starts from the first dropped packet, resulting in a waste of bandwidth. When an out-of-order packet arrives at the receiver, the receiver assumes that the next expected packet has been lost. Then, the receiver generates a negative acknowledgement (NAK) and sends it back to the sender to trigger retransmission. The work [2] indicates that the network throughput will decrease to nearly zero once the packet loss rate exceeds 1%, which seriously degrades the application performance.

2.1.3 Existing Load Balancing Schemes in Lossy DCN.
A rich body of work [13–17, 23, 24] has emerged on better load balancing for lossy DCN. However, path diversity due to congestion and asymmetry can easily cause packet reordering for finer switching granularity schemes. In recent years, many proposals strive to reduce out-of-order packets to improve load balancing.

[1] Packet recirculation is a mechanism that sends the deparsed packet at the end of egress pipeline to the ingress pipeline to repeat ingress processing without needing to replicate packet. The Portable Switch Architecture (PSA) implementation supports packet recirculation for further processing [18].

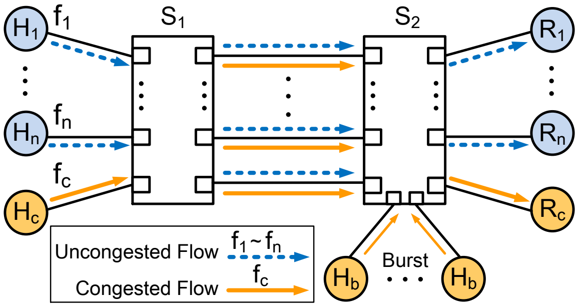


Figure 2: Typical network scenario in DCN. The uncongested flows $f_1 \sim f_n$ that are not responsible for congestion are sent from the source hosts $H_1 \sim H_n$ to the destination hosts $R_1 \sim R_n$ over multiple parallel paths, respectively. The congested flow f_c that is really responsible for congestion from H_c and the bursty flows from H_b are sent to the same receiver R_c.

Presto [14] selects paths in round-robin fashion for fixed-sized flowcells and re-assembles out-of-order flowcells back in order by utilizing reordering buffer. CONGA [13] and LetFlow [15] balance traffic at a flowlet granularity. If the inactive gap between flowlets is larger than the maximum difference of path delay, flowlets can be rerouted without packet reordering. Hermes [16] is resilient to network uncertainties. It makes deliberate rerouting decisions only if they bring performance gains. DRILL [17] performs micro load balancing at packet granularity in the divided symmetrical area to reduce disorder packets.

Although the above load balancing solutions work effectively in lossy DCN, they cannot achieve good performance in lossless DCN due to packet reordering. The reason is that they cannot correctly and timely perceive PFC pausing when choosing the optimal forwarding path. This motivates us to design a new building block to guarantee no out-of-order packets for the existing load balancing schemes.

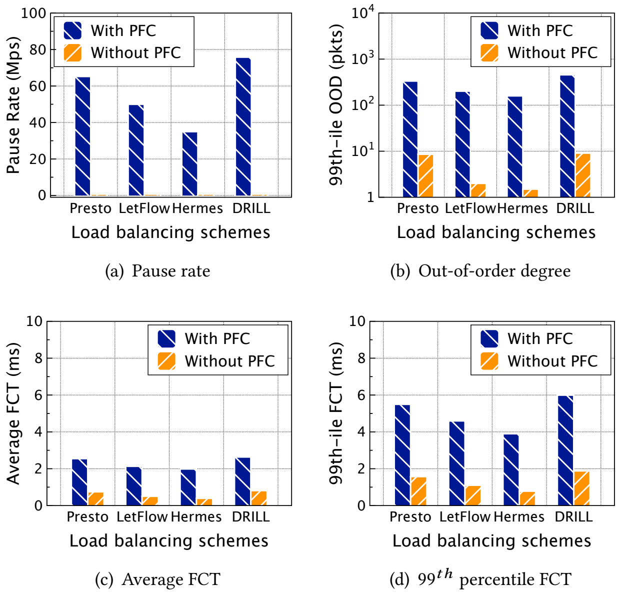
2.2 Motivation

Through an empirical analysis on the representative load balancing schemes with different switching granularities, we demonstrate how PFC mechanism affects the existing load balancing schemes.

2.2.1 Why PFC Leads to Packet Reordering?

Since the existing rerouting algorithms cannot correctly and timely perceive PFC pausing, they pick paths in a PFC-oblivious manner for fine-grained granularities such as packets, flowcells or flowlets, resulting in poor performance in the lossless DCN. To concretely show the impact of PFC on load balancing performance, we investigate how the typical load balancing schemes (i.e., Presto [14], Letflow [15], Hermes [16] and DRILL [17]) work with and without PFC under a common scenario in lossless DCN.

Without loss of generality, as shown in Fig. 2, senders ($H_1 \sim H_n$, H_c) and receivers ($R_1 \sim R_n$, R_c) connect to the corresponding leaf switches (S_1, S_2), respectively. There are 40 equal-cost paths between the two leaf switches. In addition, multiple senders (in set H_b) connect to the leaf switch S_2. The capacity of each link is 40Gbps, and the link delay is 2μs. The switch buffer is set to 9MB. The PFC threshold at each ingress port is set to 256KB. The congestion control protocol DCQCN is enabled and the parameters are set to the default values recommended in [2]. We conduct NS-3 simulations to investigate the impact of PFC on packet reordering for four typical load balancing schemes with finer granularity.



(a) Pause rate

(b) Out-of-order degree

(c) Average FCT

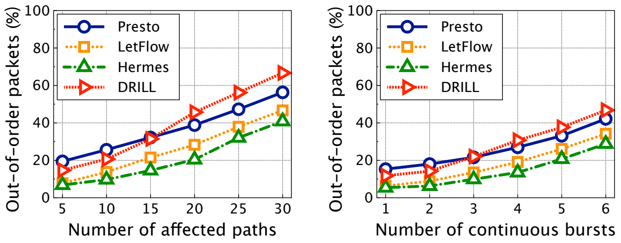
(d) 99^{th} percentile FCT

Figure 3: Performance of four load balancing schemes with and without PFC mechanism.

For this test, 100 servers $H_1 \sim H_{100}$ generate dynamic traffic $f_1 \sim f_{100}$ according to the realistic Web Search workload [25–28] with an average flow size of 1.6MB. Then each server in H_b generates 40 bursty flows with 64KB at line rate and sends them to the receiver R_c. Server H_c starts a long flow with 250MB as a congested flow f_c to R_c. By default, two continuous bursts are generated and f_c is transmitted over 5 parallel paths. In the first case, PFC mechanism is enabled, $f_1 \sim f_{100}$ can choose all of the parallel paths. Thus, they are likely to select the same paths as the congested flow f_c, and these paths have the risk of being paused by PFC due to bursty traffic. In the second case, PFC mechanism is not enabled, even though $f_1 \sim f_{100}$ are transmitted on the same paths as the congested flow f_c, $f_1 \sim f_{100}$ will not be affected by PFC pausing.

We measure the pause rate of PFC, 99^{th} percentile of out-of-order degree (OOD), average FCT and tail FCT. OOD is the difference between the sequence numbers of an out-of-order packet and the expected one. Fig. 3 compares these performances of four load balancing schemes with and without PFC mechanism under two scenarios.

Fig. 3 (a) shows that PFC is triggered under all load balancing schemes. Please note that the PFC pausing rate is zero when the PFC is not enabled. Fig. 3 (b) shows that the 99^{th} percentile of OODs under PFC pausing are larger than that of the second case without PFC pausing under four load balancing schemes. The reason is that the earlier-sent packets are blocked on the paths paused by PFC and then arrive later than the later-sent packets from a same



(a) Increasing affected paths

(b) Increasing bursts

Figure 4: Serious negative impact of PFC on load balancing with the increase number of affected paths and continuous bursts.

flow, leading to packet reordering and retransmission. Specifically, Presto [14] and LetFlow [15] inevitably choose the rerouting path with PFC PAUSE by randomly choosing rerouting path. The end-to-end Explicit Congestion Notification (ECN) and Round Trip Time (RTT) signals employed in Hermes [16] are difficult to feedback hop-by-hop PFC pausing in time. The local queue length used by DRILL [17] cannot timely sense the PFC pausing on the remote downstream switches.

Meanwhile, the out-of-order packets increase the average and tail latency as shown in Fig. 3 (c) and Fig. 3 (d). Compared with the case under PFC Pausing, the average FCT (AFCT) and 99^{th} percentile of FCT for Presto reduces by up to 68% and 72%, respectively. In addition, DRILL with the finest switching granularity suffers from the most serious reordering problem, because more rerouting increases the probability of selecting the paths that are potentially paused by PFC. Therefore, PFC PAUSE mechanism cripples the resilience of load balancing schemes against asymmetric networks.

2.2.2 *Why Packet Reordering Becomes More Serious?*

To show the serious negative impact of PFC on load balancing schemes, we further measure the ratio of out-of-order packets with the increased number of paused paths and continuous bursts. We control the number of affected paths that are paused by PFC through controlling the number of multiple paths that can be chosen by the congested flows. Worse still, on the one hand, as the number of parallel paths paused by PFC increases, the uncongested flows have a higher probability to choose the adversely affected paths. As a result, the existing load balancing rerouting algorithms are not able to guarantee no packet reordering. On the other hand, when the number of continuous bursts increases, the numbers of PFC triggering and paused paths increase correspondingly, resulting in more serious packet reordering. As shown in Fig. 4 (a) and Fig. 4 (b), the ratios of reordering packets dramatically increase with increasing affected paths and bursts, respectively. Therefore, four load balancing schemes perform inferior facing serious packet reordering. For DRILL, the ratio of out-of-order packets under the case of 30 affected paths is 76% higher than that of 5 affected paths.

3 THE RLB DESIGN

In this section, we first present the design rationale of RLB, which aims to avoid packet reordering for the existing load balancing schemes in lossless DCN. Then we describe the design of RLB in detail, including predicting module and rerouting module.

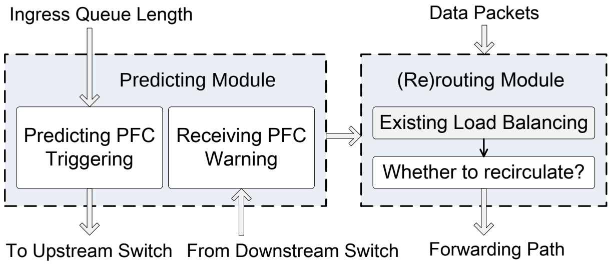


Figure 5: RLB overview.

3.1 Design Rationale

Basic idea: We first introduce the key idea of RLB. As discussed above, the existing load balancing schemes are not able to perceive PFC triggering in time. Their rerouting decisions cannot guarantee no out-of-order packets due to hop-by-hop PFC pausing in lossless DCN. Hence, before making accurate load balancing decisions, RLB predicts whether the best forwarding path initially chosen by the existing load balancing algorithm will be paused by PFC. Such port-based PFC PAUSE mechanism unavoidably leads to large queueing delay and severe packets reordering. To handle this problem, we utilize packet recirculation as a powerful technique to avoid out-of-order packets in the PFC-enabled networks.

To ensure orderly transmission, we first predict PFC triggering according to the derivative of buffer occupation, and then decide whether to recirculate or reroute packets according to the PFC warning messages and path delay. Specifically, on the one hand, if the initial optimal path chosen by the existing load balancing scheme will not be paused by PFC, RLB forwards packets to this path directly. On the other hand, the initial selected best path is likely to be paused by PFC. If the packets are forwarded on this initial path aggressively, they may be blocked and suffer from large queueing delay due to PFC PAUSE mechanism, resulting in packet reordering. On the contrary, if the packets give up this best path conservatively, they may waste an opportunity to transmit quickly from an uncongested path, because data transmission may be resumed soon by PFC RESUME message in case of transient congestion.

Therefore, once receiving a PFC warning message, RLB makes a tradeoff between the above two sides. Specifically, if the delay of initial best path is much smaller than the other parallel paths, RLB introduces a packet recirculation mechanism. After a packet goes through a recirculation, even though it spends a little more time on the switch, it obtains a new opportunity to decide whether the initial best path is still an appropriate one according to the PFC warning. RLB can recirculate the packet for multiple times before it finally chooses an appropriate forwarding path. Instead, if the delay of initial optimal path is close to the suboptimal parallel path, RLB decides to reroute the arriving packet to the suboptimal path directly. In this way, RLB is able to well preserve the original properties of existing load balancing schemes and avoid packet reordering simultaneously.

Design overview: Fig. 5 overviews RLB, which mainly contains two modules: predicting module and rerouting module.

- **Predicting Module (§3.2.1):** RLB predicts whether PFC will be triggered according to the increase rate of ingress queue length rather than the whole buffer occupation. Once there is a risk of triggering PFC, it directly sends a congestion notification message (CNM) to the upstream switch as PFC warning, which indicates that the corresponding path is likely to be paused by PFC. At the same time, the PFC warning message received from the downstream switch will be sent to the rerouting module for making load balancing decision.
- **Rerouting Module (§3.2.2):** The key for RLB to avoid packet reordering is to consider whether the initial best path selected by the existing load balancing schemes will be paused by PFC. If so, RLB decides whether to recirculate or reroute packets according to path delay. After recirculating, the packets have a new chance to decide whether the initial best path is still an appropriate forwarding path. For directly rerouting, the packets are protected from blocking. With such a scheme, RLB can effectively split traffic among multiple paths, without causing out-of-order packets. In addition, RLB, as a building block, is compatible with existing load balancing algorithms.

3.2 Design Details

3.2.1 *Predicting PFC Triggering.*

RLB first checks whether the ingress queue length exceeds a certain threshold at the current increasing rate and only performs prediction when there is congestion. Under congestion, RLB repeats the following two steps: (1) predicts whether PFC will be triggered, if it is true then (2) sends PFC triggering warning to upstream switches.

Predicting congestion: To predict PFC triggering, RLB first monitors the increasing speed of ingress queue length. This is done by calculating the derivative of the buffer occupation for each ingress queue within a certain time interval Δt (default link delay 2μs [10]). At this rate, we can calculate the time required for reaching the PFC threshold. If the remaining time is smaller than a given threshold, RLB sends PFC warning to upstream switches. Note that PFC warning is based on the prediction of future ingress queue length, rather than the current ingress queue size. As long as the packet passes through the potentially dangerous ingress port before PFC is actually triggered, or chooses other safe path after receiving PFC warning, it will not be blocked due to PFC PAUSE. Therefore, RLB allows packets to reconsider how to choose forwarding path before PFC is triggered. To eliminate out-of-order packets and avoid link utilization degradation, RLB needs to carefully calculate the dynamic threshold for sending PFC warning (§3.2.3).

Sending PFC warning: After RLB predicts PFC will be triggered in a near future with high probability, the next step is to send PFC triggering warning to upstream switches. To quickly react to the ephemeral congestion before PFC is actually triggered, switches send PFC warning message in advance through direct signal CNM of existing QCN mechanism, which is commonly available in commodity switches [29, 30]. However, QCN forwards packets based on link layer addresses and cannot directly send the CNM to upstream switches in IP-routed networks. To address this problem, RLB records the source MAC address of the incoming packets in the flow table and then propagates CNM to upstream switches hop-by-hop. Meanwhile, the identification number of ingress port that is predicted to trigger PFC is filled in the QCN field of CNM.

3.2.2 *Rerouting without Packet Reordering.*

On the one hand, if there is no predicted PFC warning, RLB forwards packets directly to the initial optimal path selected by the existing load balancing, ensuring in-order transmission. On the other hand, with the predicted PFC warning, RLB employs a deliberate rerouting based on existing load balancing to avoid packet reordering. Instead of making a hasty decision to choose the initial optimal path calculated by existing load balancing schemes, RLB makes a careful consideration based on whether PFC will be triggered.

Specifically, if a PFC warning message is received, RLB will not give up the initial optimal path immediately, but give packets more opportunities to choose an appropriate forwarding path without reordering through packet recirculation. RLB decides whether to recirculate or reroute packets to ensure the packets will not arrive at the receiver later than the subsequent packets in the same flow due to being blocked by PFC pausing. If the delay of initial optimal path is much smaller than other paths, packet recirculation has a new opportunity to decide whether the initial optimal path is still an appropriate one and can still make the packets reach the receiver faster than the subsequent packets in the same flow. Otherwise, the packets are rerouted to the other path without PFC warning and then they reach the receiver orderly.

Algorithm 1: Rerouting without Packet Reordering

```
Input:
    $h_{PFC}$: PFC warning;
    $t_{rc}$:  Measured delay of packet recirculation;
    $t_{RTT}$: Measured RTT of a path;
1  for every packet do
2      Select the initial optimal path $p$;
3      if receiving $p.h_{PFC}$ then
4          Select the suboptimal path $p_s$;
5          if $(p_s.t_{RTT}$-$p.t_{RTT}) > t_{rc}$; then
6              Recirculate packet; go to Line 3;
7          else
8              Replace $p$ with $p_s$; go to Line 3;
9      else
10         $p^* = p$;
11     return $p^*$ /* New routing path */
```

Algorithm 1 shows the rerouting logic of RLB. For each arriving packet, RLB chooses the new routing path based on the initial optimal path selected by existing load balancing schemes (line 2). If the initial selected path has PFC PAUSE warning, there are two cases. In the first one, the difference in delay between the initial optimal path and the suboptimal path is large enough to be worth recirculating packets to decide whether the best path still should be chosen (line 5-6). In the second one, the difference in delay between the initial optimal path and the suboptimal path is small, RLB takes the suboptimal path as the optimal one, and then repeats the above steps from line 3 to make decisions until choosing an appropriate path (line 8). RLB finally selects the path without PFC warning to forward packets (line 10).

In brief, for the packet recirculation mechanism, only when the predicted PFC warning for the optimal path selected by the existing load balancing is received and the difference in delay between the optimal path and the suboptimal path is larger than the delay of packet recirculation, RLB will perform packet recirculation. After each recirculation, the packet will judge whether the above two conditions are met again. If yes, the packet continues to be recirculated, indicating that it will reach the receiver faster than rerouting. Otherwise, recirculation will stop to avoid the endless loop. Unlike prior load balancing routing mechanisms, which do not consider the negative impact of PFC on packet reordering in lossless DCN, RLB chooses routing paths after thoughtful consideration based on both PFC warning and path delay.

3.2.3 *Calculating PFC Warning Threshold.*

To predict PFC triggering, we theoretically analyze the value range of PFC warning threshold Q_{th}. Without loss of generality, we model the incast network scenario with the oversubscribe ratio of n:1, where n flows are simultaneously sent to one destination host through a single PFC-enabled switch. The link capacity is C, and the link delay between the source edge switch and the destination edge switch is d. The PFC threshold is Q_{PFC}, and the instantaneous queue length of the ingress port is $Q(t)$ at time t. The sending rate of each source host is $v_i(t)$ at time t, and the receiving rate of the destination host is $v_r(t)$. Thus, the difference between the sum of sending rate and the receiving rate can be defined as the queue length varying rate, which is calculated as $\sum_{i=1}^{n} v_i(t) - v_r(t)$. The change of queue length during the time interval t is calculated as $\sum_{i=1}^{n} \int_0^t v_i(t)dt - v_r(t)dt$.

We assume RLB triggers PFC prediction mechanism at time t_w when the ingress queue length increases to $Q(t_w)$, the PFC warning message is generated and sent to the upstream switch. During the PFC warning message transmission, the packets can still choose this related forwarding port, and the ingress queue length continues to increase for a short time d.

Thus, to ensure that PFC warning is sent before PFC triggering, the queue length $Q(t_w)$ should be satisfied the following condition

$$Q(t_w) < Q_{PFC} - \sum_{i=1}^{n} \int_{t_w}^{t_w+d} v_i(t)dt + d \times v_r(t). \tag{1}$$

The worst case is that all packets are rerouted to other paths once receiving PFC warning. To avoid the throughput loss due to queue emptying when the PFC warning is lifted at time t_r, the queue length $Q(t_r)$ also should be satisfied the following condition

$$Q(t_r) \geq d \times v_r(t) - \sum_{i=1}^{n} \int_{t_r}^{t_r+d} v_i(t)dt. \tag{2}$$

Thus, the PFC warning threshold Q_{th} is set in the range

$$\begin{aligned} Q_{th} \in [d \times v_r(t) - \sum_{i=1}^{n} \int_{t_w}^{t_w+d} v_i(t)dt, \\ Q_{PFC} - \sum_{i=1}^{n} \int_{t_w}^{t_w+d} v_i(t)dt + d \times v_r(t)). \end{aligned} \tag{3}$$

Since the future sending rate of other nodes is unknown, we set a conservative PFC warning threshold for RLB. In Equation 1 and Equation 2, the value of $v_i(t)$ is set to the maximum value of link capacity C. Thus, the conservative threshold Q_{th} is set in the range of $[\lfloor d \times C \rfloor, \lfloor Q_{PFC} - d \times C \times (n-1) \rfloor)$. The experimental results show that such PFC warning threshold is effective and robust to a wide range of traffic variations (§4).

4 EVALUATION AND ANALYSIS

We evaluate RLB in conjunction with four typical load balancing schemes (i.e., Presto [14], LetFlow [15], Hermes [16] and DRILL [17]) by conducting NS-3 simulations. Our evaluation seeks to answer the following questions:

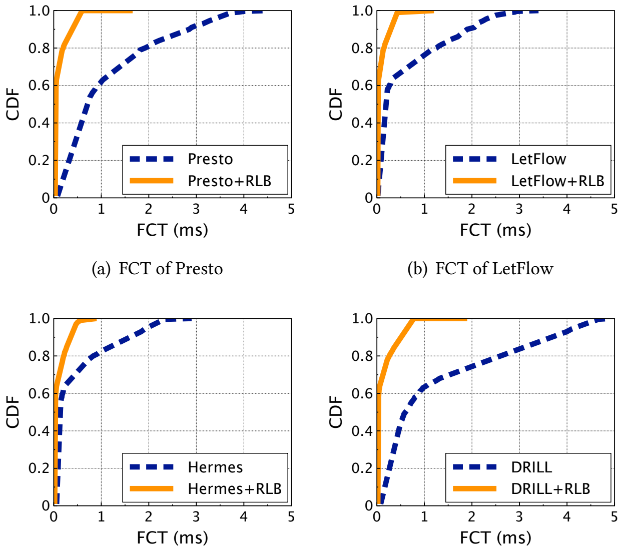
- **How does RLB perform under a symmetric and an asymmetric topology?** The experiments (§4.1 and §4.2) demonstrate the superior performance of RLB-enhanced solutions with realistic workloads. Specifically, RLB reduces the average flow completion times by 56%, 49%, 49% and 32% compared to Presto, LetFlow, Hermes and DRILL, respectively.
- **How sensitive is RLB to traffic patterns and conditions?** By varying the incast degree and response size in the bursty scenarios (§4.3), we show that RLB effectively predicts congestion, reduces out-of-order packets, and achieves persistent good performance under different traffic intensities.
- **How robust is RLB under different parameter settings?** Deep-dive experiments (§4.4) validate the effectiveness of RLB. The test results show that RLB's performance is stable under a variety of parameter settings, and RLB performs better than vanilla load balancing schemes.

Simulation settings: Unless otherwise specified, we conduct simulations on a symmetric leaf-spine topology with 12 leaf switches and 12 spine switches. There are 12 equal-cost parallel paths between any pair of leaf switches. Each leaf switch is connected to 24 hosts and 12 spine switches with 40Gbps links. Each link delay is set to 2μs [10]. Each switch enables PFC and the shared buffer size is 9MB. We use DCQCN [2] as the default transport protocol and set the related parameters as suggested in [2].

Realistic workloads: We use four typical realistic workloads observed from production data centers, i.e., Web Server, Cache Follower, Web Search and Data Mining [26]. The average flow sizes range from 64KB to more than 7.41MB, and the distribution of flow sizes is scattered. Particularly, the Data Mining workload is more heavy-tailed with 83% of flows that are smaller than 100KB and 95% of all data bytes from around 3.6% of flows that are larger than 35MB [16]. All flows in Web Server workload are less than 1MB. The flows are generated between random pair of end-hosts according to Poisson processes. The traffic load is varying from 20% to 70% [16]. We implement RLB on top of Presto, LetFlow, Hermes and DRILL, and compare the performance of RLB-enhanced solutions with vanilla Presto, LetFlow, Hermes and DRILL, respectively.

4.1 Performance under Symmetric Topology

We first inspect the performance of RLB in conjunction with four load balancing schemes under symmetric topology. Fig. 6 shows



(a) FCT of Presto

(b) FCT of LetFlow

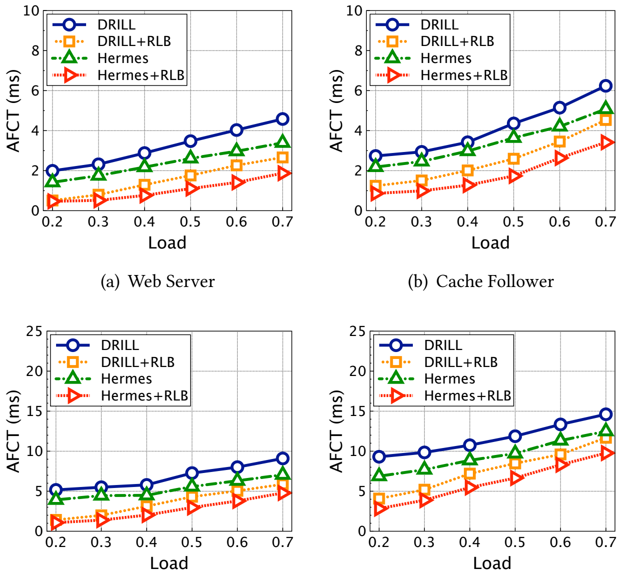
(c) FCT of Hermes

(d) FCT of DRILL

Figure 6: FCT of all flows in the symmetric topology under Web Search workload. The average load of the network core is 60%.

the cumulative distribution function (CDF) of FCT for the Web Search workload. The results indicate that RLB significantly reduces FCT and cuts the tail FCT compared with four load balancing schemes alone. For example, RLB-enhanced solutions cut the 99^{th} percentile FCT by up to 58%, 67%, 72%, and 54% over Presto, LetFlow, Hermes and DRILL, respectively. This is because the above load balancing schemes benefit from RLB to make thoughtful load balancing decisions according to the prediction of PFC triggering. The RLB-enhanced solutions effectively reduce the out-of-order packets through packet recirculation or timely rerouting.

Another observation is that DRILL without RLB suffers from longer tail delay than other load balancing schemes. This is because DRILL reroutes at finer-grained packet level, there are more paths potentially affected by PFC pausing when the packets of congested flows that really contributing to congestion are sprayed on more parallel paths. Hence, more packets are assigned on the paths that are likely to be paused for transmission. RLB solves this problem by predicting PFC triggering and making load balancing schemes with considering PFC pausing for packet rerouting or recirculation. After receiving a warning that PFC may be triggered on the current path, the arriving packets can avoid reaching the destination host later than the subsequent packets with larger sequence number in both cases. Specifically, in the first scenario, the current packet can be flexibility rerouted to the suboptimal equal-cost path with small difference in delay to avoid being paused by PFC. In the second scenario, the current packet can be recirculated instead of radically rerouting to the suboptimal path with large difference in delay. However, as the number of paths predicted to trigger PFC increases, the effective paths available to RLB decrease, resulting in limited the performance improvement. In brief, RLB successfully reduces the out-of-order packets.



(a) Web Server

(b) Cache Follower

(c) Web Search

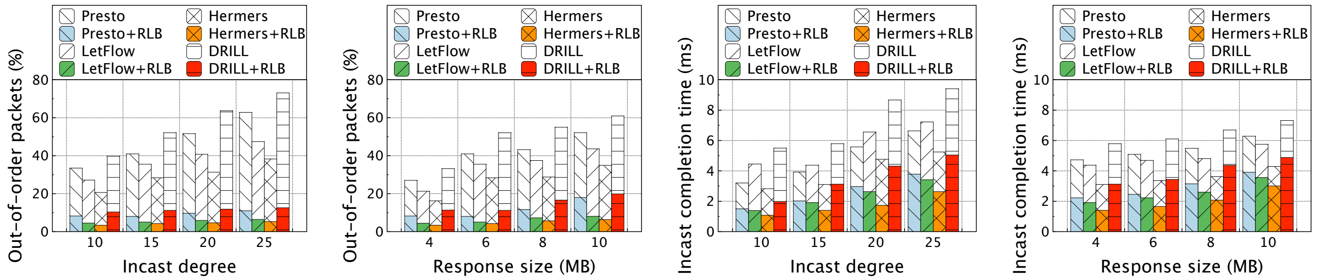
(d) Data Mining

Figure 7: AFCT of all flows for realistic workloads in the asymmetric topology with the varying load.

4.2 Performance under Asymmetric Topology

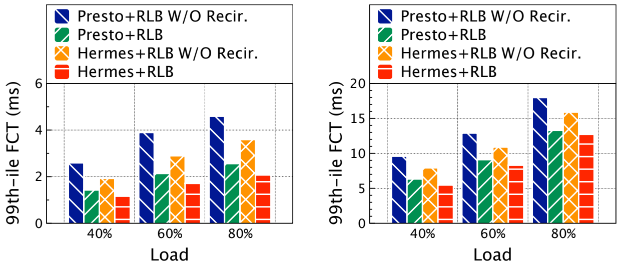
We further integrated RLB into DRILL and Hermes under four realistic workloads in an asymmetric topology with the varying load. We adopt the default symmetric topology and reduce the link capacity from 40Gbps to 10Gbps for 20% of randomly chosen leaf-to-spine links [16]. Fig. 7 shows the improvement of RLB in terms of average FCT as the load varies from 20% to 70% of the network capacity. We can see that DRILL and Hermes benefit from RLB across a wide range of loads. For example, DRILL+RLB outperforms DRILL by up to 42% and 28% at 0.6 load for Web Server and Data Mining workloads, respectively. For Cache Follower workload, Hermes+RLB is 58% and 34% better than Hermes at 0.2 and 0.6 load, respectively.

Specifically, RLB-enhanced solutions always outperform the vanilla DRILL and Hermes as the load increases. Firstly, RLB performs better for the Web Server and Cache Follower workloads than Web Search and Data Mining. To explain this, note that the Web Server and Cache Follower workloads contain more small flows that cannot be controlled by the end-to-end transport protocol. Furthermore, the Data Mining workload has larger inter-flow arrival time and much more long flows that can be controlled by the transport protocol. Hence, PFC is more likely to be triggered in Web Server and Cache Follower workloads, and RLB has more chances to take effect. RLB is able to predict PFC triggering and make the rerouting or recirculation decisions cautiously based on the difference in path delay to avoid out-of-order packets. Secondly, we find that as the load increases, the room for improvement by RLB reduces slightly, which is a result of less available parallel paths for rerouting. Last but not least, the performance improvement of RLB on the existing load balancing mechanisms in asymmetric network is greater than that in symmetric network.



(a) Reordering with varying incast degree (b) Reordering with varying response size (c) FCT with varying incast degree (d) FCT with varying response size

Figure 8: Varying incast degrees and response sizes.



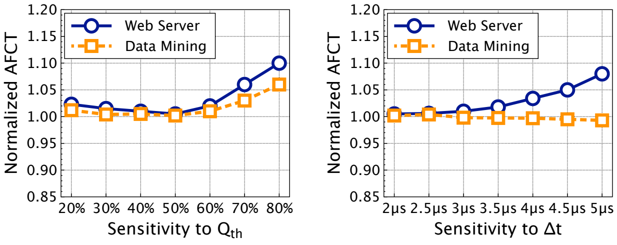
(a) Web Server (b) Data Mining

Figure 9: RLB deep dive for effectiveness of packet recirculation under realistic workloads. Here “Recir." refers to recirculation.

4.3 Performance under Incast Scenario

We next evaluate the effectiveness of RLB with different intensities of bursty traffic by varying incast degrees. In this test, a client makes simultaneous requests to fetch responses from multiple servers. By default, the number of involved responders is 15 and the total response traffic is 4MB in each incast initiation. We vary the incast degree from 10 to 25 and change the response size from 4MB to 10MB. We measure the ratio of out-of-order packets and the completion time of the last flow, as shown in Fig. 8.

The results indicate that RLB can help the load balancing schemes to significantly reduce the ratio of out-of-order packets and speed up flows even under stressed incast scenarios. For example, the ratio of out-of-order packets under the scenario where incast degree is 15 and the response size is 10MB is reduced by up to 51%, 66%, 75% and 47% over Presto, LetFlow, Hermes and DRILL, respectively. The incast completion time is improved by 22%~46% across different response sizes. The key reason is that RLB assists the load balancing schemes to avoid triggering spurious retransmissions due to disorder packets. It is also worth noting that, although RLB greatly alleviates the packet reordering problem, the tail FCT has not reduced in the same proportion due to packet recirculation delay. In addition, the end-to-end transport protocol is able to control the large flows to alleviate congestion. Therefore, the bursty traffic can be controlled by the transport protocol as the response size increases, the performance improved by RLB is smaller than that with the increase of incast degree.



(a) Sensitivity to Q_{th} (b) Sensitivity to Δt

Figure 10: Sensitivity to PFC predicting threshold Q_{th} and the calculating time interval Δt.

4.4 Performance Robustness

Effectiveness of recirculation: To avoid the side effects of PFC pausing, a simple method is to reroute packets directly when receiving the warning of PFC triggering. However, this is not always a good choice. Considering the different path congestion and PFC pausing duration, the benefit of packet recirculation on the original path is higher than that of aggressive rerouting. Now we investigate the benefits of packet recirculation using Web Server and Data Mining workloads. Fig. 9 (a) shows that packet recirculation bring about 37% and 28% improvement to the 99^{th} percentile of FCT under 80% load for Presto and Hermes, respectively. A similar trend is observed in Fig. 9 (b) as well.

Sensitivity of parameter settings: We further study how different parameter settings affect the performance of RLB. Fig. 10 shows the normalized AFCT for the optimal parameters under Web Server and Data Mining workloads with varying Q_{th} and Δt. First, we observe that the AFCT is relatively stable when theses two parameters are set close to the suggested values. Another observation is that the AFCT is increased under both workloads as the increase of Q_{th}, because PFC prediction is late and PFC may have been triggered when RLB takes effect. We also observe that the two workloads experience different trend as the calculating time interval increases. Since the Web Server workload is more bursty, PFC warning is generated more frequently, thus an aggressive parameter setting brings better performance due to rapid adaptation to network congestion. In contrast, the performance of Data Mining is less sensitive to Δt.

5 RELATED WORK

We classify previous work on balancing traffic in DCNs into two categories: load-balance routing schemes and multipath transport solutions.

Load balancing schemes: To make good use of multiple paths, a wide range of mechanisms perform rerouting at finer granularity. These include, but are not limited to: 1) using packet-level switching granularity to split traffic flexibly across parallel paths [16, 17, 31, 34], but potentially suffering from packet reordering problem; 2) using fixed flowcell-level switching granularity to reduce out-of-order packets [14]; 3) using flowlet-level switching granularity to dynamically adapt to network congestion [13, 15, 23, 24]; 4) using flow-level coarse granularity to avoid out-of-order delivery at the cost of low link utilization [32, 33]. However, these schemes are originally designed for lossy DCN and cannot work well in lossless DCN due to the adverse impact of PFC.

Multi-path transport solutions: Multi-path transmission schemes split flows into multiple subflows and assign them on the equal-cost paths to improve throughput. MP-RDMA [9, 35] distributes packets among parallel paths in a congestion-aware manner to achieve high throughput. IRN [36] explores the effective loss recovery schemes for lossy RDMA networks to abandon PFC mechanism. While these schemes effectively alleviate congestion, they still cannot completely avoid PFC triggering, so further efforts are needed to avoid reordering.

6 CONCLUSION

This paper presents RLB, that augments all existing load balancing algorithms to reduce packet reordering in lossless datacenter networks. By predicting PFC triggering, RLB decides whether to choose the initial best path selected by existing load balancing schemes, or reroute to other parallel paths, or recirculate packets to obtain more opportunities to choose appropriate forwarding paths without packet reordering. The evaluation results indicate that, RLB can effectively reduce out-of-order packets and significantly reduce the tail flow completion time by up to 72% compared with the state-of-the-art load balancing schemes.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (62102046, 62132022, 62072056), the Natural Science Foundation of Hunan Province (2020JJ2029, 2022JJ30618, 2021JJ30867), and the Scientific Research Fund of Hunan Provincial Education Department (22B0300).

REFERENCES

[1] M. Alizadeh, A. Greenberg, D. A. Maltz, et al. Data Center TCP (DCTCP). In Proc. ACM SIGCOMM, 2010.

[2] 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 Proc. ACM SIGCOMM, 2015.

[3] W. Cheng, K. Qian, W. Jiang, T. Zhang, and F. Ren. Re-architecting Congestion Management in Lossless Ethernet. In Proc. USENIX NSDI, 2020.

[4] K. Qian, W. Cheng, T. Zhang, and F. Ren. Gentle Flow Control: Avoiding Deadlock in Lossless Networks. In Proc. ACM SIGCOMM, 2019.

[5] Y. Zhu, M. Ghobadi, V. Misra, and J. Padhye. ECN or Delay: Lessons Learnt from Analysis of DCQCN and TIMELY. In Proc. ACM CoNEXT, 2016.

[6] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye, and M. Lipshteyn. RDMA over Commodity Ethernet at Scale. In Proc. ACM SIGCOMM, 2016.

[7] W. Bai, A. Agrawal, A. Bhagat, M. Elhaddad, N. John, J. Padhye, et al. Empowering Azure Storage with 100×100 RDMA. In Proc. USENIX NSDI, 2023.

[8] J. Xue, M. U. Chaudhry, B. Vamanan, T. N. Vijaykumar, and M. Thottethodi. Dart: Divide and Specialize for Fast Response to Congestion in RDMA-based Datacenter Networks. IEEE/ACM Transactions on Networking, 28(1):322-335, 2020.

[9] Y. Lu, G. Chen, B. Li, K. Tan, Y. Xiong, P. Cheng, J. Zhang, E. Chen, and Thomas Moscibroda. multipath Transport for RDMA in Datacenters. In Proc. USENIX NSDI, 2018.

[10] C. Tian, B. Li, L. Qin, J. Zheng, J. Yang, W. Wang, G. Chen, and W. Dou. P-PFC: Reducing Tail Latency with Predictive PFC in Lossless Data Center Networks. IEEE Transactions on Parallel and Distributed Systems, 31(6):1447-1459, 2020.

[11] 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 Proc. ACM SIGCOMM, 2015.

[12] G. Kumar, N. Dukkipati, K. Jang, et al. Swift: Delay is Simple and Effective for Congestion Control in the Datacenter. In Proc. ACM SIGCOMM, 2020.

[13] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, G. Varghese. CONGA: Distributed Congestion-Aware Load Balancing for Datacenters. In Proc. ACM SIGCOMM, 2014.

[14] K. He, E. Rozner, K. Agarwal, W. Felter, J. Carter and A. Akellay. Presto: Edge-based Load Balancing for Fast Datacenter networks. In Proc. ACM SIGCOMM, 2015.

[15] E. Vanini, R. Pan, M. Alizadeh, P. Taheri and T. Edsall. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. In Proc. USENIX NSDI, 2017.

[16] H. Zhang, J. Zhang, W. Bai, K. Chen, and M. Chowdhury. Resilient Datacenter Load Balancing in the Wild. In Proc. ACM SIGCOMM, 2017.

[17] S. Ghorbani, Z. Yang, P. Godfrey, Y. Ganjali, and A. Firoozshahian. DRILL: Micro Load Balancing for Low-Latency Data Center Networks. In Proc. ACM SIGCOMM, 2017.

[18] The P4.org Architecture Working Group. P4$_{16}$ Portable Switch Architecture (PSA). https://p4.org/p4-spec/docs/PSA-v0.9.0-draft.html#sec-recirculate.

[19] Z. Liu, K. Chen, H. Wu, S. Hu, Y. Hu, Y. Wang, G. Zhang. Enabling work-conserving bandwidth guarantees for multi-tenant datacenters via dynamic tenant-queue binding. In Proc. IEEE INFOCOM 2018.

[20] W. Bai, S. Hu, K. Chen, K. Tan, Y. Xiong. One more config is enough: Saving (DC) TCP for high-speed extremely shallow-buffered datacenters. IEEE/ACM Transactions on Networking, 2020, 29(2), 489-502.

[21] IEEE 802.1 Qbb - Priority-based Flow Control. https://1.ieee802.org/dcb/802-1qbb/.

[22] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, Z. Lin, V. Kurien. Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis. In Proc. ACM SIGCOMM, 2015.

[23] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford. Hula: Scalable Load Balancing Using Programmable Data Planes. In Proc. ACM Symposium on SDN Research, 2016.

[24] N. Katta, A. Ghag, M. Hira, I. Keslassy, A. Bergman, C. Kim, and J. Rexford. Clove: Congestion-Aware Load Balancing at the Virtual Edges. In Proc. ACM CoNEXT, 2017.

[25] W.Bai, S. Hu, K. Chen, K. Tan, Y. Xiong. One More Config is Enough: Saving (DC)TCP for High-speed Extremely Shallow-buffered Datacenters. In Proc. IEEE INFOCOM 2020.

[26] J. Hu, J. Huang, Z. Li, Y. Li, W. Jiang, K. Chen, J. Wang and T. He. RPO: Receiver-driven Transport Protocol Using Opportunistic Transmission in Data Center. In Proc. IEEE ICNP, 2021.

[27] J. Hu, J. Huang, Z. Li, J. Wang and T. He. A Receiver-Driven Transport Protocol with High Link Utilization Using Anti-ECN Marking in Data Center Networks. IEEE Transactions on Network and Service Management, DOI:10.1109/TNSM.2022.3218343, 2022.

[28] J. Zhang, W. Bai, K. Chen. Enabling ECN for datacenter networks with RTT variations. In Proc. ACM CoNEXT, 2019.

[29] IEEE. 802.1Qau – Congestion Notification. http://www.ieee802.org/1/pages/802.1au.html.

[30] A. Saeed, V. Gupta, P. Goyal, M. Sharif, R. Pan, M. Ammar, E. Zegura, K. Jang, M. Alizadeh, A. Kabbani, and A. Vahdat. Annulus: A Dual Congestion Control Loop for Datacenter and WAN Traffic Aggregates. In Proc. ACM SIGCOMM, 2020.

[31] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella. On the Impact of Packet Spraying in Data Center Networks. In Proc. of IEEE INFOCOM, 2013.

[32] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: Dynamic Flow Scheduling for Data Center Networks. In Proc. USENIX NSDI, 2010.

[33] T. Benson, A. Anand, A. Akella, and M. Zhang. MicroTE: Fine Grained Traffic Engineering for Data Centers. In Proc. ACM CoNEXT, 2011.

[34] J. Hu, J. Huang, W. Lv, W. Li, J. Wang and T. He. TLB: Traffic-aware Load Balancing with Adaptive Granularity in Data Center Networks. In Proc. ACM ICPP, 2019.

[35] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving Datacenter Performance and Robustness with Multipath TCP. In Proc. ACM SIGCOMM, 2011.

[36] R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy, and S. Shenker. Revisiting Network Support for RDMA. In Proc. ACM SIGCOMM, 2018.