2023 International Conference on Computing, Networking and Communications (ICNC) | 978-1-6654-5719-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICNC57223.2023.10074382

DIRS: Dynamic Initial Rate Setting in Congestion Control for Disaggregated Storage Systems

Xiaoqian Zhang†, Allen Yang†, Danlin Jia*, Li Wang*, Mahsa Bayati‡, Pradeep Subedi ‡, Xuebin Yao ‡, Bo Sheng† and Ningfang Mi*
*Department of Electrical and Computer Engineering, Northeastern University, Boston, USA
†Department of Computer Science, University of Massachusetts Boston, Boston, USA
‡Samsung Semiconductor Inc., San Jose, CA, USA

***Abstract*—This paper focuses on network congestion control in a disaggregated storage system. In such a system, the supporting network requires low latency and is extremely sensitive to network congestion. The existing congestion control algorithms for data center networks do not work well in our target system because of the unique characteristics of the network topology and the storage I/O workload. Motivated by the existing issues, we develop a new solution, DIRS, which dynamically sets the initial sending rate for each flow. Our scheme helps improve the effectiveness of the congestion control protocols, especially under heavy I/O traffic. It chooses an appropriate initial rate for a flow and mitigates the congestion from the beginning while not degrading the flow's network performance.**

I. Introduction

The disaggregated storage system is a promising direction for a storage framework in a large-scale data center. The computing devices and storage hosts are physically separated and connected through high-speed networks in such an architecture. It provides more flexible resource management, straightforward upgrade and maintenance, and other features that enterprise storage systems desire.

The connecting network becomes critical to the entire system's performance in a disaggregated storage system. There have been multiple options for the underlying network protocols in the prior work, such as Ethernet, Fibre Channel, RoCE [1], InfiniBand [2], or TCP/IP. In this paper, we consider RoCE (RDMA over Converged Ethernet) as the supporting network architecture which guarantees lossless data delivery. More details will be reviewed in Sec. II. Like traditional data center networks, disaggregated storage systems require extremely low latency to handle the storage I/O requests. Therefore, network congestion becomes a significant concern in such a system. Many prior works have aimed to solve the congestion issue in data center networks. However, they do not perform well in this disaggregated storage system because of the unique characteristics of the network topology and I/O workload. In this paper, we develop an enhanced scheme, DIRS, for the existing data center congestion control protocols. Our focus is the initial sending rate setting for a network traffic flow. Instead of using the network link capacity as the initial sending rate, our design captures the runtime congestion state and appropriately sets a lower sending rate for a new flow when the network is busy. Our solution explicitly addresses the existing issues, and the simulation-based evaluation shows significant performance improvement.

This work was partially supported by Samsung research grant and National Science Foundation Award CNS-2008072.

II. Background and Motivation

In this section, we first introduce the existing congestion control protocols for data center networks. Then, we present the unique network characteristics of the target disaggregated storage systems and examine the limitations of the existing solutions which motivate our design of DIRS.

A. Existing Congestion Control Schemes

We consider a disaggregated storage system built upon RDMA over Converged Ethernet (RoCE) network protocol. There are two major layers of congestion control:

1) ***Link Level Flow Control****:* In this layer, a congested switch will inform the upstream switch to take action to resolve congestion, e.g., in the Priority-based Flow Control (PFC) [3], a pause frame is sent to the upstream sender if the queue length of a switch exceeds a threshold. The upstream sender stops sending packets until receiving a resume frame from the switch. PFC ensures a lossless data center network by pausing upstream senders during network congestion.

2) ***End-to-end Congestion Control****:* In this traditional congestion control, a sender performs rate adjustments according to network congestion indicators, e.g., in TCP [4], packet loss is the indicator of network congestion. However, TCP congestion control does not perform well in low-latency data center networks. When packets are dropped, the congestion becomes severe. New congestion control schemes, such as DCQCN [5] and DCTCP [6], are proposed for data center networks. They deploy the Explicit Congestion Notification (ECN) [7] marking mechanism to resolve the issue of delayed congestion feedback for the end-to-end control schemes.

B. Characteristics of Disaggregated Storage Systems

This paper focuses on disaggregated storage systems where data storage (hosts) is formed by a pool of SSD devices and separated from computing devices (initiators). The I/O requests from users are received by the initiators and conducted on the data hosts. The associated data are transferred through internal high-speed RoCE networks.

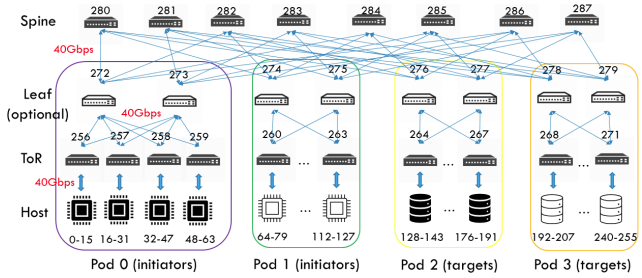


Fig. 1. Topology of Disaggregated System Simulation

In the rest of this subsection, we present the unique characteristics of the network and workload in the target system.

Long Hop Distance: First, in disaggregated storage systems, a traffic flow's source and destination are separated by a long hop distance. We assume the system adopts a Clos-like network topology which is popular in data center networks (shown in Fig. 1), with the initiators located on one side and the storage devices on the other side. Compared to regular data center traffic where the source and destination are randomly located, the I/O traffic in a disaggregated system traverses a longer hop distance yielding a larger RTT (Round Trip Time). Consequently, in congestion control schemes, when the destination sends a congestion notification to the source, it will take longer for the packet to arrive.

Short Flows in Workloads: In addition, we observe that there are usually a large amount of short I/O flows (read or write flows) in real-world workloads. For example, in the synthetic workloads we extracted from the Tencent trace [8], 51% of the flows are smaller than 124KB, and over 89% of flows are smaller than 457KB.

Multiple Parallel Sessions Finally, we find that a disaggregated storage system handles busy I/O requests, and multiple parallel network sessions exist between the same pair of sender and receiver. For example, in the Tencent I/O trace that we used in the evaluation, out of the 5057 write flows from ten initiators to one storage host, 4055(80.2%) of them have ongoing flows when the flows start.

C. Limitations of Existing Schemes

Considering the unique characteristics described above, we find two significant limitations while applying the existing congestion control schemes to the disaggregated storage systems. First, due to the long hop distance, when a sender receives congestion notifications from the receiver, transfers of short flows may have been completed. Then, these notifications are considered late notifications since senders do not get a chance to adjust sending rates. Any solution based on ECN marking becomes ineffective. We define a metric *ECN ineffective ratio* to represent the ratio of late congestion notifications and the total number of congestion notifications. For the Tencent traces, the ECN ineffective ratios of DCQCN are as high as 50.30%. The second limitation is that the existing congestion control schemes do not share the congestion information across the parallel network sessions between the same sender-receiver pair. As a result, new flows still start with the maximum sending rates, while the concurrent flows have received congestion notifications and reduced their sending rates.Both limitations of the existing solutions result in less effective control for network congestion.

III. Design of DIRS

In this section, we introduce the design of our congestion control scheme DIRS, a newly designed initial sending rate assignment technique.

Our solution consists of two technical components; both implemented on the sender's side. First, we create a *virtual flow* for each destination that captures the late congestion notifications for short flows and mimics the rate deduction in congestion control algorithms. Second, when starting a new flow, we apply a scheme that checks the lowest sending rate of the ongoing active flows with the same destination, including the virtual flow. And then, this lowest sending rate will be assigned to the new flow as its initial sending rate. The intuition is to enable information sharing among concurrent flows with the same destination. If the ongoing flows have already lowered their rates because of congestion, the new flow does not need to repeat the same rate adjustment process. Instead, the new flow can use the current lowest sending rate as its initial rate.

A. Capture Late Notifications with a Virtual Flow

In our design, we create a virtual flow (VF) structure for each possible destination on each participating node. The VF keeps a "sending rate" value as a regular flow. The purpose is to capture the late congestion notifications from the short flows and use them to adjust the VF's sending rate.

Let VF_i indicate the VF for destination node i. We use t'_i to record the timestamp of the last received congestion notification packet (CNP) from node i. We also keep two timers for each VF_i, a rate recovery timer, and a rate reset timer. The following Algorithm 1 shows the basic steps for processing a late CNP with a VF. We also use Fig. 2 to illustrate an example.

Algorithm 1: Receive a CNP from node i at time t

```
if t-t'_i >Δ then
    Decrease the sending rate of VF_i;
    Start/restart rate recovery timer for VF_i;
    Start/restart rate reset timer for VF_i;
    t'_i ← t;
else
    Discard this CNP
end
```

In Fig. 2, the destination is experiencing congestion and receives an ECN-marked packet of Flow1. But when the CNP (dashed line) is delivered to the sender, all the packets of Flow1 have been sent out. In the existing congestion control algorithms, there will be no sending rate deduction and this late CNP is ineffective. In our design, the sender, upon the

arrival of a late CNP, will decrease the sending rate of VF_i (line 2) following the same strategy in the congestion control algorithm. In Fig. 2, the VF's sending rate is initially the network link capacity (40Gbps). Assume the congestion control algorithm cuts the sending rate to half when the sender receives a CNP. We apply the same strategy to the VF's sending rate. Therefore, when the late CNP for Flow1 arrives, the VF's sending rate is reduced to 20Gbps.

When processing a late CNP, the sender also compares the arrival time to the timestamp of the last received late CNP (line 1). If the time difference is smaller than a threshold Δ, the CNP will be discarded. This mechanism prevents the VF's sending rate from over-deduction with consecutive late CNPs. In Fig. 2, Flow2 and Flow3 illustrate an example. Both are short flows and the starting times are close to each other. Each of them triggers a late CNP sent by the destination. The two consecutive late CNPs received by the sender represent the same congestion condition. With our design, the sender adjusts the VF's sending rate to 10Gbps for the late CNP from Flow2 but discards the other CNP from Flow3.

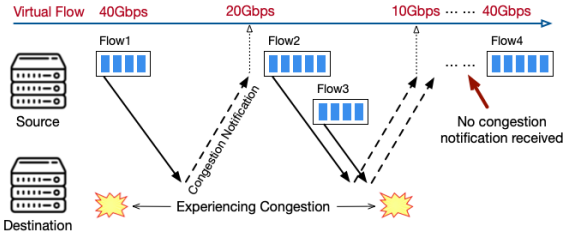


Fig. 2. Example of Using a VF to Capture Late Notifications

At the same time, the sender will start the two timers (lines 3–4). The rate recovery timer is part of the existing congestion control protocol. When it expires and the sender has not received any CNP, the sender will start a rate recovery process to increase the sending rate. We apply the same process to the VF_i's sending rate. Additionally, we include another rate reset timer in the design. The duration of this timer is longer than that of the rate recovery timer. Once the rate reset timer expires, the VF's sending rate will be reset to the network link capacity. The intuition of using this timer is that the VF does not carry out any traffic and is not guaranteed to receive feedback from the destination. The rate adjustment of VF based on the received late CNPs is only effective for a certain period. The reset of the VF's sending rate indicates the end of this effective period. In Fig. 2, assume after Flow2's late CNP, the rate reset timer expires, then the sending rate of the VF will become 40Gbps again.

B. Track Lowest Rate

The second component of our design is to let the sender track the lowest sending rate of all active flows with the same destination node, including the corresponding VF. When starting a new flow, the sender will assign the lowest sending rate as the new flow's initial rate rather than the network link capacity. For example, in Fig. 2, with the existing solution, all four flows will start with a 40Gbps initial sending rate. In our design, however, Flow2 and Flow3 will start with 20Gbps because of the existence of the VF.

Considering both concurrent flows with the same destination and the VF, our solution tracks two types of the lowest rate. We name them *instantaneous lowest rate* and *periodical lowest rate*. The instantaneous lowest rate is measured when assigning initial sending rates to new flows. Meanwhile, we schedule a time window to track the lowest sending rate of the active flows periodically, named the periodical lowest rate. In our design, the lower value of the instantaneous lowest rate and the periodical lowest rate will be assigned as the new flow's initial rate. Both rates are essential for setting an appropriate initial rate for the new flow. The following two cases explain our primary design intuitions.

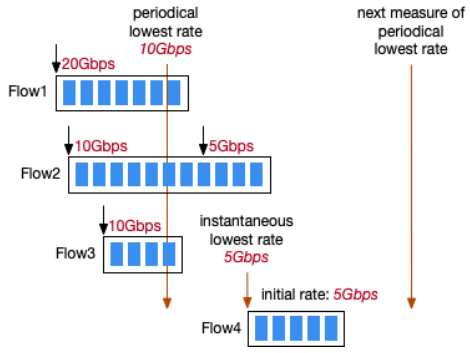


Fig. 3. An example of assigning initial sending rate (Case 1)

Case 1 - Assign instantaneous lowest rate to new flows: Refer to Fig. 3, Flow4 is a new flow that will be assigned with an initial rate. Before Flow4 is started, there are three active flows: Flow1 with the rate of 20Gbps, Flow2 with the rate of 20Gbps, and Flow3 with the rate of 10Gbps. The first periodical check records 10Gbps as the periodical lowest rate. But after that, Flow2 receives another CNP and further reduces its rate to 5Gbps. When starting Flow4, the instantaneous lowest rate is 5Gbps, by definition. Our solution will assign 5Gbps to Flow4 as its initial sending rate. In this case, the instantaneous lowest rate represents the up-to-date state.

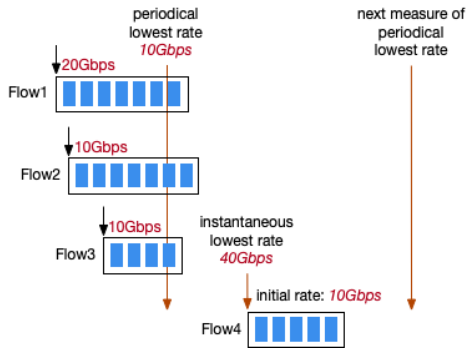


Fig. 4. An example of assigning initial sending rate (Case 2)

Case 2 - Assign periodical lowest rate to new flows: In Fig. 4, when the new flow (Flow4) starts without any active flows. All three flows have been completed when flow4 starts the transfer. The instantaneous lowest rate is 40Gbps at

this moment. The periodical lowest rate, in this case, is still 10Gbps. Flow4 will use 10Gbps as its initial sending rate. In this case, the periodical lowest rate more accurately reflects the congestion conditions, as it is measured shortly before Flow4 starts. The instantaneous lowest rate, on the other hand, misses the information when the other flows are just finished.

Overall, combining the instantaneous lowest rate and the periodical lowest rate, our solution assigns a low rate to the new flow as its initial rate that can better accommodate the network congestion conditions.

IV. Evaluation

In this section, we present our evaluation results. First, we introduce the RDMA-NS3 simulator and network configuration used to implement our scheme. Then we will describe the workloads utilized in the evaluation. Finally, we will present our scheme's performance metrics and evaluation results.

A. Simulation Environment

We implement our scheme on the RDMA-NS3 simulator from [9], which implements the RoCEv2 protocol and several other congestion control schemes, such as DCQCN, TIMELY, and DCTCP. We have mainly modified the simulator's implementation of network switches and configured a set of network parameters based on common settings and commercial network switch specifications, such as the network topology, link capacity, switch buffer size, and threshold parameters in ECN and PFC.

In particular, our simulation adopts a CLOS network topology as shown in Fig. 1. All link capacities are 40Gbps and the packet size is set to 1KB in our experiment. PFC and ECN mechanisms are enabled by default. All the other network configurations are based on the default DCQCN setting in [9].

B. Workload

The workload in our simulation consists of two groups of I/O traffic. The first group is heavy traffic that causes network congestion, and the second group is a set of light traffic that plays the role of *victim flows*. The purpose of examining the victim flows is to measure the impact of the congestion on regular traffic.

For the first group of traffic, we generate the I/O workload based on the characteristics extracted from real trace datasets and apply it to a setting of ten initiators with one data host. The following two workloads are considered in this paper:

- Tencent Workload: This workload is generated based on the Tencent trace from [8]. We generate a total amount of 164.78Gbps read traffic and 35.74Gbps of write traffic.
- Alibaba Workload: This workload is generated by the build-in flow pattern in the RDMA-NS3 simulator [9]. We generate three workloads with only write traffic and varying network loads of 32Gbps, 34Gbps, and 37Gbps.

For the second group of traffic, we inject different amounts of light flows with fixed flow sizes as the victim flows. We measure their network performance to evaluate the impact of network congestion.

Combing these two groups of traffic, we use the following sets of workloads in our simulations. For the Tencent workload, we generate 5 traces by injecting 5000 victim flows with different flow sizes (6KB, 12KB, 18KB, 24KB, and 30KB). For the Alibaba workload, we generated 3 traces by injecting 5000 victim flows with a fixed victim flow size of 12KB.

C. Evaluation Metrics

We evaluate the effectiveness of DIRS against DCQCN through the number of received PAUSE frames from all switches and the network performance through the victim flows' throughput. The total number of PFC pause frames sent out in the network indicates the overall network congestion level. Therefore, comparisons of the total numbers of pause frames will show the effectiveness of DIRS. In addition, we measure and compare the network throughput of the victim flows to show the direct network performance improvement. [1]

D. Evaluation Results

Both Fig. 5-(a) and (b) show DIRS can better mitigate network congestion with a significantly reduced number of pause frames. For the Tencent and Alibaba traces, compared to DCQCN, DIRS decreases the pause number by 37.1% on average and 89.88% respectively.

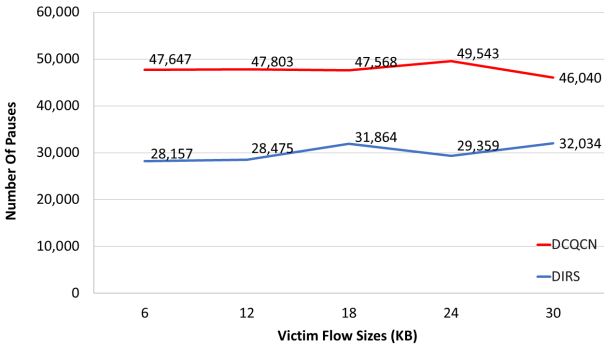
Fig. 6 illustrates the victim flows' throughput of DIRS and DCQCN. DIRS is overall superior to DCQCN in the tested cases. For the Tencent traces, DIRS improves the victim flows' throughput by 54.34% on average. Victim flows' throughput is more beneficial from DIRS with the increasing sizes of the injected victim flows. When the victim flow size is 30KB, the throughput of DIRS surpasses DCQCN by 2.35 times. For the Alibaba traces, victim flows' throughput is enhanced through DIRS by 18.89% on average. Above all, our simulation results show that DIRS is an effective solution that reduces the pause frames and confines the negative impact of the congestion.

V. Related Work

The recent solutions to data center congestion control can generally be divided into two categories, Rate-Control and Packet-Scheduling. In a Rate-Control solution, a congested switch marks the outgoing packets, the receiver sends a congestion notification to the sender once receiving marked packets, and the sender adjusts its rate accordingly. Some well-known marking techniques include the ECN [7], TCP [4], and QCN [10]. A few rate adjustment solutions have been proposed based on ECN marking, e.g., DCTCP [6], PCN [11]. and DCQCN [5]. Another work TIMELY [12] uses RTT as the indicator for rate adjustment.

Packet-Scheduling-based schemes schedule data transmissions by assigning different levels of priority to packets to resolve network congestion. These schemes aim to maximize

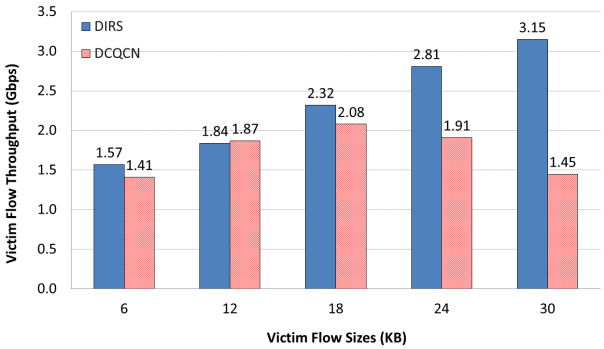
[1]Note that in an RDMA lossless network, congestion control does not help improve the performance of the flows that cause the congestion. The benefit is for the entire network. Thus we use victim flows' throughput to represent the network performance metric.



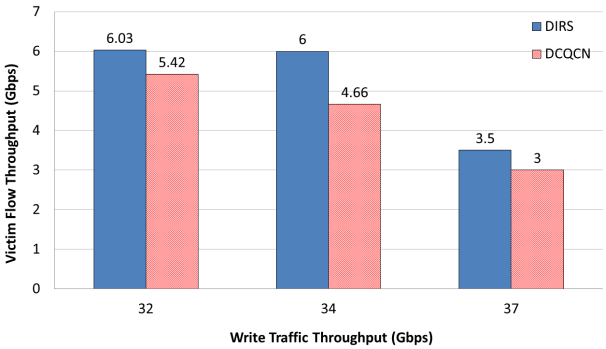
(a) Tencent Workload with Injection of 5000 Victim Flows

(b) Alibaba Workload with Injection of 5000 Victim Flows

Fig. 5. DIRS Total Number of Pauses Received by Switches



(a) Tencent Workload with Injection of 5000 Victim Flows



(b) Alibaba Workload with Injection of 5000 Victim Flows

Fig. 6. DIRS Average Network Throughput of Victim Flows

deadline meet rate for deadline flows and minimize average flow completion time for non-deadline flows [13]–[17].

In this paper, we propose a Rate-Control based scheme, DIRS, that overcomes the limitations of the current congestion control schemes to better resolve network congestion with the newly designed initial sending rate assignment technique.

VI. Conclusion

This paper presents a congestion control scheme that improves the existing schemes based on the characteristics of disaggregated storage systems. DIRS introduces an effective initial sending rate assignment technique that significantly improves performance compared to the existing scheme.

References

[1] IBTA, "RDMA over Converged Ethernet (RoCE)," https://cw.infinibandta.org/document/dl/7781, 2014.
[2] IBTA, "InfiniBand," https://www.mellanox.com/pdf/whitepapers/IB_Intro_WP_190.pdf.
[3] "Priority-based Flow Control," https://1.ieee802.org/dcb/802-1qbb/.
[4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *SIGCOMM '98*, 1998.
[5] 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 *ACM Conference on Special Interest Group on Data Communication*, 2015.
[6] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center TCP (DCTCP)," in *SIGCOMM '10*, 2010.
[7] S. Floyd, "TCP and explicit congestion notification," *Comput. Commun. Rev.*, vol. 24, pp. 8–23, 1994.
[8] Y. Zhang, P. Huang, K. Zhou, H. Wang, J. Hu, Y. Ji, and B. Cheng, "Tencent block storage traces (SNIA IOTTA trace set 27917)," in *SNIA IOTTA Trace Repository*, G. Kuenning, Ed. Storage Networking Industry Association, Oct. 2018.
[9] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, K. Frank, M. Alizadeh, and M. Yu, "HPCC: High precision congestion control," in *In Proceedings of the ACM Special Interest Group on Data Communication*, 2015.
[10] IEEE 802.1Qau, "Congestion Notification," https://1.ieee802.org/dcb/802-1qau/.
[11] W. Cheng, K. Qian, W. Jiang, T. Zhang, and F. Ren, "Re-architecting congestion management in lossless ethernet," in *NSDI '20*, 2020.
[12] R. Mittal, V. Lam, N. Dukkipati, E. Blem, H. M. G. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, "TIMELY: RTT-based congestion control for the datacenter," *Comput. Commun. Rev.*, vol. 45, pp. 537–550, 2015.
[13] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker, "pFabric: minimal near-optimal datacenter transport," in *SIGCOMM 2013*.
[14] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal, "Fastpass: a centralized "zero-queue" datacenter network," in *SIGCOMM 2014*.
[15] P. Gao, A. Narayan, G. Kumar, R. Agarwal, S. Ratnasamy, and S. Shenker, "phost: distributed near-optimal datacenter transport over commodity network fabric," in *The 11th ACM Conference on Emerging Networking Experiments and Technologies*, 2015.
[16] M. Handley, C. Raiciu, A. Agache, A. Voinescu, A. Moore, G. Antichi, and M. Wójcik, "Re-architecting datacenter networks and stacks for low latency and high performance," 2017.
[17] B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout, "Homa: a receiver-driven low-latency transport protocol using network priorities," in *ACM Special Interest Group on Data Communication*, 2018.