2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA) | 979-8-3315-0971-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISPA63168.2024.00028

DAR: Deadline-Aware Rerouting for Mix-flows in Datacenter Networks

Jinbin Hu[1], Rui Zhi[1], Shuying Rao[1], Ying Liu[1], Jin Wang[1, 2]
[1]Changsha University of Science and Technology, ChangSha, China
[2]Hunan University of Science and Technology, Xiangtan, China
jinbinhu@csust.edu.cn, jinwang@hnust.edu.cn
{ruizhi,shuyingrao,yingliu}@stu.csust.edu.cn

***Abstract*—In modern datacenter networks (DCNs), the booming online data-intensive applications generate mix-flows with or without deadlines. Balancing these heterogenous flows among parallel equal-cost paths to meet the tight deadlines is crucial. However, due to the unaware of deadlines, the existing load balancing mechanisms cannot choose suitable (re)routing path for mix-flows to meet their respective stringent requirements. In this paper, we propose a deadline-aware rerouting scheme called DAR, which applies different routing strategies for mix-flows. Specifically, DAR first perceives the deadline flows and then categorizes them based on the urgency of the deadline, and employs different (re)routing strategies to ensure that flows with more urgent deadlines are completed earlier. The NS-3 simulation results show that DAR effectively balances mix-flows. For example, compared to the state-of-the-art load balancing schemes, DAR reduces the deadline miss rate and the average flow completion time (AFCT) by up to 38% and 35.5%, respectively.**

***Index Terms*—Datacenter networks, deadline-aware, load balancing, routing**

I. Introduction

With advancements in cloud computing, big data, artificial intelligence, and other technologies, an efficient and stable DCN is a key facility for network services. Some online data-intensive applications (e.g., online conferencing, video calling, and online gaming) focus on end-to-end transmission latency [1]–[4]. These applications exhibit a low tolerance for latency, so they are referred to as deadline flows [5]–[8]. Missing the deadline leads to issues like frame drops or audio hitches in games, severely impacting real-time experience and possibly causing revenue loss. Additionally, some data center applications (e.g., data backup, VM migration, etc.) have flows without strict deadline requirements (non-deadline flows) [9], [10]. These flows are dedicated to optimizing both latency and throughput to ensure transmission success. Existing DCNs employ multipath topologies to satisfy the transmission requirements of different traffic. Therefore, it is critical to design a load balancing scheme for datacenter networks for mixed flows in multipaths.

However, existing load balancing schemes (e.g., ECMP [11], RPS [12], CONGA [13], LetFlow [14], etc.) do not consider the transmission of deadline flows. Both deadline flows and non-deadline flows exist in some applications in data center (e.g., real-time data analysis, online transaction processing, video streaming, etc.). Deadline flows need to complete transmission before deadline and non-deadline flows don't have strict transmission time requirements. When these flows are transmitted at the same time, prior non-deadline flows will occupy the link resources and block deadline flows. The AFCT of deadline flows is extended and the deadline miss rate increases significantly. For real-time applications, the data has already failed by the time it arrives. It severely impacts network performance and the user's real-time experience.

To solve the previous problem, we design a deadline-aware rerouting scheme DAR. It senses flows whose remaining flow completion time is close to the deadline. These flows will miss their deadline if not rerouted. DAR helps these flows switch to a low-latency path for fast completion. It also safeguards flows anticipated to meet their deadlines but risk missing them due to path congestion. Specifically, DAR recognizes whether flows have deadlines and calculates the urgency level of each deadline flow. Then it assigns fitting transmission paths for flows with different urgency levels. Once the switch recognizes a deadline flow with high urgency, DAR picks the path with the shortest transmission delay. For flows that are not deadline-sensitive, the path with the lowest relative transmission delay is selected for transmission while ensuring that flows with deadline requirements are prioritized for transmission.

Overall, the paper makes the following contributions:

- We conduct an in-depth study to analyze the reasons why some existing load balancing schemes do not work well for rerouting deadline flows with high urgency. Then, we suggest the need to study load balancing schemes for sensing deadline flows.
- We propose a load balancing scheme that senses the urgency of deadline flows, called DAR. It senses deadline flows with different levels of urgency and selects different transmission paths for these flows so as to ensure that they are transmitted before the deadline.
- We have conducted NS-3 simulations in different scenarios. Compared to the traditional load balancing schemes, DAR reduces the deadline miss rate and the AFCT by up to 38% and 35.5%, respectively.

*Jin Wang is the corresponding author.

The rest of the paper is organized as follows. We describe the motivation in Section II. In Section III, we describe the design overview and details. In Section IV, we show the NS-3 simulation results. In Section V, we present the related works and then conclude the paper in Section VI.

II. Motivation

A. Blocked Deadline Flows

We performed simulations in a typical leaf-spine topology, as shown in Fig. 1. In it, a non-deadline data flow f_1 (in blue) is transmitted from the sender S_1 to the receiver R_1. Meanwhile, sender S_2 also sent a deadline flow f_2 (in yellow) to receiver R_2. The figure shows that non-deadline flow f_1 and deadline flow f_2 pass through node switch C_2 together. We used the four existing load balancing schemes to route flows separately. We found that deadline flow f_2 was blocked at the end of the queue by non-deadline flow f_1. The deadline flow f_2 had to wait to transmit until the non-deadline flow f_1 completed its transmission. When the f_2 reached the receiver R_2, the deadline had already been missed. We conclude that the existing load balancing scheme failed to route deadline flows in a timely manner, thereby missing their deadlines and consequently reducing network transmission performance.

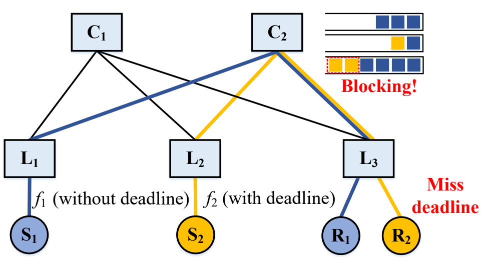
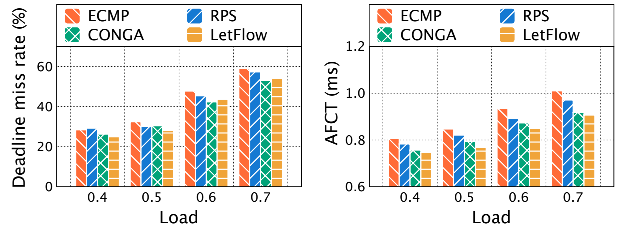


Fig. 1. Typical network topology. The earlier non-deadline flow f_1 (in blue) blocked the later deadline flow f_2 (in yellow)

B. Impact of Missing Deadline

We conduct tests on four load balancing schemes to assess their deadline miss rates and AFCT under different network loads. The outcomes of these experiments are illustrated in Fig. 2. Notably, the deadline miss rates and AFCT for all four load balancing schemes showed an increase as the network load grew. The growth rate accelerates significantly as the network transitions to heavy loads. On the one hand, when subjected to the same network load (for instance, a network load of 0.7), ECMP exhibited the highest deadline miss rate at 59%, closely trailed by RPS. The deadline miss rates of CONGA and LetFlow show at 53% and 54% respectively. We find that traditional load balancing schemes do not work well for deadline flows under heavy load networks. On the other hand, the AFCT of four load balancing schemes in descending order are ECMP, RPS, CONGA, and LetFlow in same load.

These findings highlight that conventional load balancing mechanisms do not differentiate between time-sensitive and regular traffic, thereby leading to missed deadlines for critical flows and diminishing overall network throughput. The poor performance of ECMP can be attributed to its equal-cost multi-path routing strategy, which does not account for the latency sensitivity of deadline flows. RPS, while attempting to distribute traffic evenly, lacks the sophistication to prioritize urgent traffic over less critical ones. CONGA's congestion-aware approach is more refined but may still struggle to meet tight deadlines due to its focus on avoiding congestion rather than explicitly prioritizing urgent flows. Similarly, LetFlow, despite its efforts to optimize for latency, might fall short in scenarios where the network is heavily loaded, as it does not provide explicit guarantees for deadline-sensitive traffic.



(a) Deadline miss rate (b) AFCT

Fig. 2. Performance under different workloads

III. Design

A. Design Overview

DAR lies in categorizing deadline-constrained urgent flows into three distinct urgency levels and selecting appropriate paths for each level. Specifically, DAR transmits with the granularity of flow, determining the optimal routing for a flow based on its deadline and urgency classification to prevent packet disorder and missed deadlines. On the other hand, we calculate two urgency factors derived from the best-case and worst-case path conditions, and use these factors to classify flows into their respective urgency levels. This allows us to select the most suitable paths for flows of different levels, ensuring that urgent flows meet their deadlines and reducing the completion times of regular flows. The DAR mechanism deployed on the switch is composed of two main modules, each incorporating two sub-modules, as depicted in Fig. 3.

Deadline-aware Module: DAR categorizes the flows transmitted within the DCN into two categories deadline and non-deadline, within the deadline-aware sub-module. Specifically, the primary condition is to ensure timely completion of deadline flows, concurrently minimizing the transmission delay for all flows. Subsequently, DAR employs the urgency factor in the urgency rating sub-module to further categorize deadline flows into three levels: Level I, Level II, and Level III.

(Re)routing Module: DAR monitors the path state through the port detection sub-module, maintaining a path state table based on the output port queue lengths. It then selects the most suitable paths for both deadline and non-deadline flows, considering their urgency levels and the current path state, to prevent deadline flows from missing their deadlines due

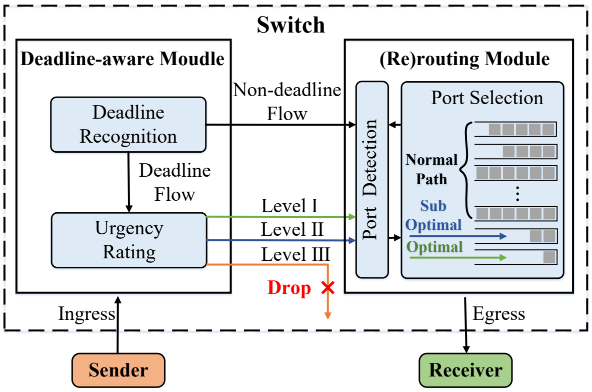


Fig. 3. DAR Overview

to contention with non-deadline flows. Additionally, DAR ensures low transmission delays for all flows.

B. Design Details

In this section, we present the design details of DAR, encompassing the methodology for classifying the urgency of flows with deadlines based on its urgency factor, and the strategy for selecting paths contingent upon the assessed urgency of the flows.

1) Deadline Recognition: In DAR, we adopt the method from D^{2}TCP [15] that emphasizes the necessity for applications to furnish deadline information for streams prior to initiating data requests. Specifically, when the source switch prepares to send a data request, it first determines whether the request is associated with a particular deadline. If so, the request is marked as a deadline flow. So, each hop in the network recognizes which packets carry deadline requirements and is thus able to prioritize these packets to ensure that they reach their destination within the specified time.

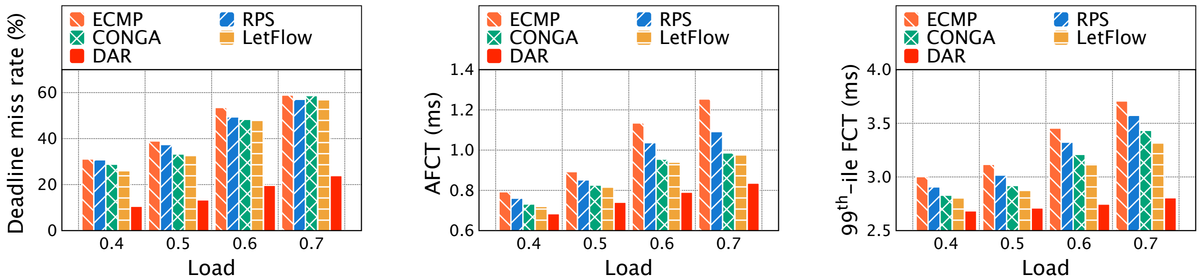
2) Urgency Rating: The DAR assumes that a flow should complete its transmission within its deadline. We assume that T_c is an upper bound on the time a flow can transmit to complete all data in the pessimistic scenario; D is the remaining time before the deadline. We compute the urgency factor d_1 as $d_1 = \frac{T_c}{D}$. If the time required for a flow to complete transmission of all data in the pessimistic case is exactly the same as the time remaining before the deadline (i.e., d_1 = 1), we consider this flow to be just about able to complete its transmission on time in the worst-case scenario of the link state. If d_1 > 1 (i.e., T_c > D), we consider this flow to be certain that it will not be able to complete its transmission before the deadline. If d_1 < 1 (i.e., T_c < D), then we are certain that the flow can fulfill the time constraints of the deadline. Although Tc can be calculated precisely, we believe that an approximation in D^{2}TCP can be used to avoid missing the flow's deadline. T_c can be approximated as $T_c = \frac{4B}{3W}$, where W denotes the current window size of the flow and it has B bytes to transfer.

On the other hand, DAR assumes T_o to be the flow completion time required for a flow to be transmitted in the optimal path, and we introduce a second urgency factor d_2 based on T_o as $d_2 = \frac{T_o}{D}$. If $d_2 \leq 1$ (i.e., $T_o \leq D$), we consider this flow to be strictly deadline-complete if it is transmitted in the path with the shortest transmission delay. If d_2 > 1 (i.e., T_o > D) represents that the time required for the flow to finish transmitting all the data in the optimal path is longer than the time remaining before the deadline, we consider that this flow will definitely miss the deadline. If the deadline will be missed even in the transmission of the optimal path, this flow will be canceled for transmission by DAR and will not participate in (re)routing afterwards. This helps to reduce bandwidth wastage caused by sending useless data, and can also reduce queuing delays for other flows. The calculation of To is similar to T_c, where we use the ratio of the remaining required transmission bytes to the current window size of the flow as the required transmission time in the optimal path as follows $T_o = \frac{B}{W}$, where B and W are consistent with what we mentioned earlier.

DAR considers the following cases for deadline flows. When d_1 > 1 and d_2 < 1, such deadline flows miss the deadline on the worst path, but routing it to the optimal path for transmission does not miss the deadline. Such flows need to choose the optimal path to reduce the transmission delay to meet the deadline, and DAR determines them as Level I flows. However, we also need to protect flows that are close to missing the deadline from failing to complete their transmission due to choosing a highly loaded path. Therefore, DAR adjudicates flows that meet d_1 < 1 and d_2 < 1 as Level II flows. When d_2 > 1, this indicates that even if the flow is routed to the optimal path, it will not be able to meet the flow's tight deadline requirement. For flows that cannot meet the requirement, we determine them as Level III flows.

3) Port Detection: One of the core functions in DAR is to detect the status of the output ports of the switches in the network, which is crucial for deciding which path a packet should follow. In order to effectively manage and monitor the status of output ports, DAR introduces a path status table. The purpose of this table is to keep a detailed record of the real-time status of each output port, which provides a basis for decision-making on path selection. The path status table contains two sets of information: the port number, which is used to uniquely identify each output port in the network; and the queue length of the port, which reflects the number of packets currently waiting to be processed on the port, indirectly reflecting the busyness and potential delay of the port. By continuously monitoring and updating this information, the DAR is able to quickly identify bottlenecks and congestion points in the network, allowing for more rational path planning.

4) Path Selection: In DAR, the switch dynamically assigns paths for non-deadline flows and deadline flows based on their urgency levels. Upon arrival of a flow at the output port, DAR initiates flow-level routing to prevent packet disorder. Specifically, upon receiving a packet from a Level I flow, DAR selects the optimal path from the path state table to



(a) Deadline miss rate (b) AFCT (c) 99^{th}-ile FCT

Fig. 4. Performance under Web Search

forward the packet, and this port remains dedicated to the flow until its completion. This prioritization ensures that the most urgent deadline flows are transmitted swiftly, minimizing the risk of deadline misses. For a packet from a Level II flow, DAR chooses the sub-optimal path for transmission. In contrast, upon encountering a Level III flow, DAR implements a strategy to preemptively drop packets, recognizing the inevitable deadline miss. When a non-deadline flow reaches the switch, DAR routes it through the optimal path in normal paths, catering to the low-latency demands of standard traffic.

IV. Evaluation

A. Experimental setup

Topology. We adopt a typical 3×3 leaf-spine network topology, with each leaf switch being connected to 32 terminal hosts. The network bandwidth of each link is 100 Gbps, transmission latency is 5ms, and the queue cache size of the switch is 256 packets.

Baselines. In our experimental evaluation, we compare the performance of DAR with the following four traditional load balancing schemes, ECMP, RPS, CONGA and LetFlow. All parameters of the algorithm are consistent with its paper.

Workloads. We design and simulate three representative real-world scenarios corresponding to Web Server, Web Search and Data Mining applications. The generation of data flows follows a Poisson distribution and the network workload range is set between 0.4 and 0.8. Table I details the distribution characteristics of flow sizes for the three realistic scenario-based workloads.

TABLE I: Flow size distribution of realistic workloads

	Web Search	Web Server	Data Mining
0-10KB	59%	63%	78%
10KB-100KB	3%	18%	5%
100KB-1MB	18%	19%	8%
> 1MB	20%	0	9%
Average flow size	1.6MB	64KB	7.41MB

B. Performance under Web Search Scenario

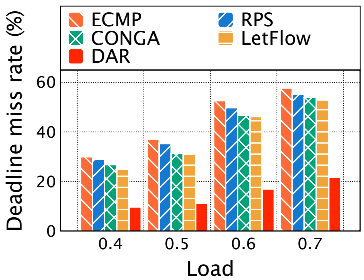
Applications in Web Search environments transmit about 62% of their data as short flows, including a large number of deadline flows. Fig. 4 (a) shows that DAR consistently exhibits the lowest deadline miss rate compared to the other four traditional load balancing schemes. At 70% network high workloads, DAR reduces its deadline miss rate by about 32.9%, 35%, 33.2% and 35% compared to LetFlow, CONGA, RPS, and ECMP, respectively. This is because DAR can prioritize low-delay queues for deadline flows and ensure that deadline flows with high urgency use the optimal path, effectively reducing the possibility of missing deadlines.

Similarly, compared to ECMP, RPS, CONGA and LetFlow, as shown in Fig. 4 (b), the AFCT increases as the network becomes more highly workloads, just at different growth rates. For high workload, the AFCT of DAR is the lowest, followed by LetFlow, CONGA and RPS. Under high workload (e.g., 0.7), DAR significantly reduces AFCT by about 33.3% compared to ECMP. This is because different types of flows in ECMP may be hashed to the same path, which increases the AFCT of the flows. However, DAR not only prioritizes the transmission of deadline flows but also ensures low-latency delivery for non-deadline flows, thereby achieving a notable decrease in the AFCT across all flows.

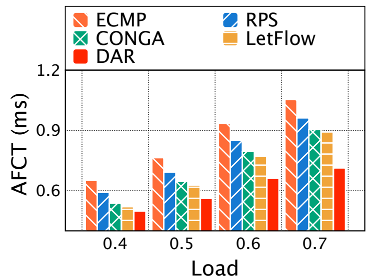
In addition, we also tested the 99^{th}-ile FCT at different loads. Fig. 4 (c) shows that the 99^{th}-ile FCT similarly increases with increasing load. As with AFCT, in descending order, ECMP, RPS, CONGA, LetFlow. DAR is the lowest compared to them under 0.7 workloads, decreasing by 24.3%, 21.5%, 18.3% and 15.4%, respectively. The results reveal that DALB significantly outperforms other load balancing mechanisms, particularly under high workloads, by substantially reducing AFCT. This is because DALB routes and forwards urgent deadline flows to faster paths, thereby reducing the queuing time for deadline flows and speeding up flow arrival and completion times.

C. Performance under Web Server Scenario

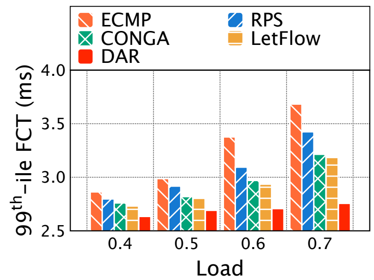
In Web Servers, each request and response cycle for user access is short. Previous work has summarized the workload distribution in Web Server, where 81% of flows are less than 100KB. Fig. 5 demonstrated the performance of DAR compared to existing load balancing schemes under Web Server scenario. The deadline miss rates of the other load balancing schemes still increase with network load in Fig. 5 (a), but DAR is always at its lowest level. Especially, under



(a) Deadline miss rate

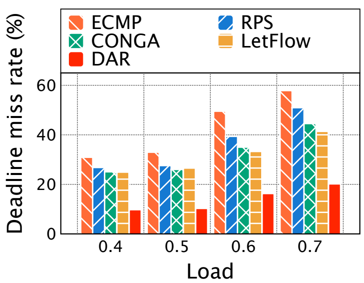


(b) AFCT

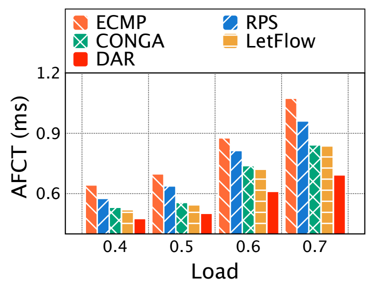


(c) 99^{th}-ile FCT

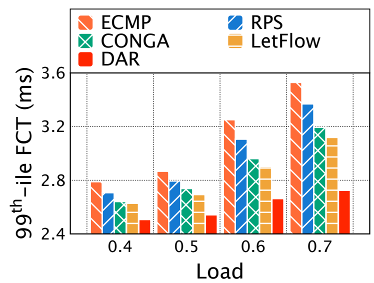
Fig. 5. Performance under Web Sever



(a) Deadline miss rate



(b) AFCT



(c) 99^{th}-ile FCT

Fig. 6. Performance under Data Mining

high load network (e.g., 0.7), the DAR is 36%, 33.6%, 32.3% and 31.2% lower than that of ECMP, RPS, CONGA and LetFlow, respectively. This indicates that DAR accelerates the transmission time of deadline flows, effectively reducing the overall deadline miss rate.

Fig. 5 (b) illustrates the AFCT under Web Server workloads, demonstrating that DAR significantly outperforms the other four load balancing schemes in different workload scenarios. Under workload of 0.7, the AFCT of DAR is reduced by up to approximately 32.4%. This performance is attributed to DAR's capability to aware and (re)routing deadline flows by preventing them from being blocked by non-deadline flows. By minimizing the queuing time for deadline flows, DAR accelerates the AFCT, showcasing its effectiveness in optimizing the overall flow processing speed.

Fig. 5 (c) demonstrates the 99^{th}-ile FCT under Web Server workloads. DAR achieves a significant reduction of roughly 25.1%, 19.5%, 14.3% and 13.4% in the 99^{th}-ile FCT relative to ECMP, RPS, CONGA and LetFlow at a workload intensity of 0.7. This is because DAR adeptly tackles the issue of excessive tail delays in deadline flows. By prioritizing these flows and minimizing their waiting times, DAR effectively alleviates the trailing delay problem, leading to a considerable decrease in the 99^{th}-ile FCT.

D. Performance under Data Mining Scenario

We analyze how the percentage of missed deadline flows, the AFCT of flows, and the 99^{th}-ile FCT vary under five different load balancing schemes in Data Mining scenario. In Fig. 6 (a), DAR significantly reduces the percentage of deadline flows missing deadlines among all flows, especially under high workloads. Specifically, in the data mining scenario under 0.7 workload, DAR reduces deadline miss rate by 38%, 30.8%, 24.4%, and 21.2% compared to ECMP, RPS, CONGA, and LetFlow, respectively. In addition, we find that the deadline miss rate in Data Mining scenarios is overall lower than Web Search for the same workload. This is because there are more short flows in the Data Mining scenario and the probability of missing a deadline is smaller.

Fig. 6 (b) shows that DAR is able to apply to data mining scenarios and maintains the lowest AFCT. The AFCT of each strategy tends to increase with workload. However, the increase in workload does not have a noticeable effect on the performance of DAR. Specifically, at high workloads (e.g., 0.7), DAR reduces the AFCT by 35.5%, 27.9%, 17.7% and 17.2%, respectively. This is because DAR prioritizes the deadline flows, which greatly reduces the AFCT of the deadline flows. But, it does not ignore the transmission needs of the non-deadline flows and selects the queue with the relatively lowest transmission latency for them. As a result, DAR enables an overall optimization of the average completion time of all flows in a data mining scenario.

We compare the performance optimization of DAR at the 99^{th}-ile FCT as shown in Fig. 6 (c). Compared with ECMP, RPS, CONGA and LetFlow, the 99^{th}-ile FCT of DAR is unaffected by workload, especially in highly loaded networks. Specifically, at high workloads (e.g., 0.7), DAR reduces 99^{th}-ile FCT by up to 22.8%. This is because ECMP's hashing mechanism causes short and long flows to be assigned to the same path. Later short flows are blocked by the long flows that came first, causing long-tailed delays. In contrast, DAR prioritizes the allocation of paths with low transmission delay for deadline flows so that they are not blocked by non-deadline flows. The relatively fastest paths are then selected for non-deadline flows to avoid long-tailed delays.

V. Related work

In recent years, many load balancing schemes for datacenter networks have been proposed to improve the transmission performance of datacenter networks [16]–[21].

Mechanisms based on centralized controllers. To minimize routing overhead and link congestion, SOFIA [16] determines the best threshold for differentiating between short and long flows and notifies all switches of the redirected flows. AuTO [17] uses machine learning to identify the priority of flows. It ensures the throughput of long flows while solving the problem of short flows being blocked.

Mechanisms based on host. Each flow is divided into several flowlets by Clove [18], which then uses ECN feedback to dynamically modify the path weights of the flowlets so that each flowlet chooses the path with the higher weight. ALB [19] forwards flowlets to paths with minimal congestion by utilizing delay-based congestion detection. It fixes the issue of mismatched source and destination host times and measures the RTT and unidirectional delay of each flow.

Mechanisms based on switch. By comparing the queue lengths of the two randomly chosen ports and the best port from the previous round, DRILL [20] choose the port with the least queue length as the forwarding port for the current packet. It attains microsecond load balancing and reacts swiftly to congestion. To guarantee that packets arrive in an ordered manner, ConWeave [21] performs RTT monitoring between the source and destination switches, enabling packet granularity rerouting.

VI. Conclusion

In this paper, we propose DAR, a deadline-aware load balance mechanism for mix-flows in datacenter networks. By categorizing flows into different urgency levels and employing an urgency factor to determine the optimal routing paths, DAR effectively prioritizes deadline flows while minimizing the impact on normal traffic. The NS-3 simulation findings show that DAR reduces the AFCT and the deadline miss rate by up to 35.5% and 38%, respectively, when compared to the traditional load balancing systems. We intend to continue refining the mechanism and testing DAR's performance in a range of real-world circumstances.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (62472050, 62473146, 62102046, 62072056), and by the Key Project of Natural Science Foundation of Hunan Province (2024JJ3017), and by the New Generation Information Technology Innovation Project 2023 (2023IT271), and by the Science and Technology Project of Hunan Provincial Department of Water Resources (XSKJ2024064-36).

References

[1] J. Hu, H. Shen, X. Liu, J. Wang. RDMA Transports in Datacenter Networks: Survey. In Proc. IEEE Network, 2024, DOI: 10.1109/MNET.2024.3397781.

[2] P. Hamadanian, D. Gallatin, M. Alizadeh, K. Chintalapudi. Ekho: Synchronizing cloud gaming media across multiple endpoints. In Proc. ACM SIGCOMM, 2023, pp. 533-549.

[3] J. Hu, C. Zeng, Z. Wang, J. Zhang, K. Guo, H. Xu, J. Huang, K. Chen. Load Balancing with Multi-level Signals for Lossless Datacenter Networks. In Proc. IEEE/ACM Transactions on Networking, 2024, 32(3), 2736-2748.

[4] X. Wang, L. Ren, R. Yuan, L. T. Yang, and M. J. Deen. QTT-DLSTM: A Cloud-Edge-Aided Distributed LSTM for IIoT Big Data. In Proc. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(10), 7286-7298.

[5] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang and M. J. Deen. A Tensor-based Big Data-Driven Routing Recommendation Approach for Heterogeneous Networks. In Proc. IEEE Network, 2019, 33(1), 64-69.

[6] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Ssenguta, M. Sridharan. Data center tcp (dctcp). In Proc. ACM SIGCOMM, 2010, pp. 63-74.

[7] C. Wilson, H. Ballani, T. Karagiannis, A. Rowstron. Better Never Than Late: Meeting Deadlines in Datacenter Networks. In Proc. ACM SIGCOMM, 2011, pp. 50-61.

[8] J. Hu, J. Huang, J. Wang, et al. A Transmission Control Mechanism for Lossless Datacenter Network Based on Direct Congestion Notification. In Proc. Acta Electronica Sinica, 2023, 51(9), 2355-2365.

[9] 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. In Proc. IEEE/ACM Transactions on Networking, 2020, 29(2), 489-502.

[10] J. Hu, J. Huang, W. Lv, Y. Zhou, J. Wang and T. He. CAPS: Coding-based Adaptive Packet Spraying to Reduce Flow Completion Time in Data Center. In Proc. IEEE/ACM Transactions on Networking, 2019, 27(6), 2338-2353.

[11] Hopps, C.E. Analysis of an equal-cost multi-path algorithm, 2000.

[12] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella. On the Impact of Packet Spraying in Data Center Networks. In Proc. IEEE INFOCOM, 2013, pp. 2130-2138.

[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, pp. 503-514.

[14] 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, pp. 407-420.

[15] B. Vamanan, J. Hasan, and T. N. Vijaykumar. Deadline-aware datacenter TCP (D2TCP). In Proc. ACM SIGCOMM, 2012, pp. 115-126.

[16] F. De Pellegrini, L. Maggi, A. Massaro, D. Saucez, J. Leguay and E. Altman. Blind, adaptive and robust flow segmentation in datacenters. In Proc. IEEE INFOCOM, 2018, pp. 10-18.

[17] L. Chen, J. Lingys, K. Chen and F. Liu. Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. In Proc. ACM SIGCOMM, 2018, pp. 191-205.

[18] 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, pp. 323-335.

[19] Q. Shi, F. Wang, D. Feng and W. Xie. ALB: Adaptive load balancing based on accurate congestion feedback for asymmetric topologies. In Proc. Computer Networks, 2019, 157, 133-145.

[20] Ghorbani S, Yang ZB, Godfrey PB, Ganjali Y, Firoozshahian A. DRILL: Micro load balancing for low-latency data center networks. In Proc. ACM SIGCOMM, 2017, pp. 225-238.

[21] C. H. Song, X. Z. Khooi, R. Joshi, I. Choi, J. Li and M. C. Chan. Network Load Balancing with In-network Reordering Support for RDMA. In Proc. ACM SIGCOMM, 2023, pp. 816-831.