AI and ML Applications Traffic: Designing Challenges for Performance Optimization of Optical Data Center Networks

Peristera Baziana[1*], Georgios Drainakis[2], David Georgantas[1], Adonis Bogris[3]

[1]Department of Informatics and Telecommunications, University of Thessaly, Greece
[2]School of Electrical and Computer Engineering, National Technical University of Athens, Greece
[3]Department of Informatics and Computer Engineering, University of West Attica, Greece
baziana@uth.gr*, gdrainakis@mail.ntua.gr, dageorgantas@uth.gr, abogris@uniwa.gr

***Abstract*— Artificial intelligence (AI) and machine learning (ML) are vastly becoming key parts of today's applications, supporting diverse human activities and decision making. New training models are used requiring an explosive traffic exchange among data centers (DCs) servers, putting tremendous pressure on DC network (DCN). Therefore, new DCN design challenges are required to efficiently handle the emerging AI/ML applications traffic. Contrary to existing works that do not account for these challenges, in this paper we introduce an optical intra-rack DCN architecture, as an optical local area network (LAN) paradigm, and an efficient wavelength division multiplexing access (WDMA) algorithm tailored to serve the AI/ML traffic. For the proposed WDMA scheme, we assume diverse AI/ML traffic classes according to the required service time and quality of service (QoS) delivery and we adopt appropriate access rules for each of them to achieve high DCN performance and access fairness. Simulation results show that our proposal reaches high bandwidth utilization (90% under nominal load 400 Gbps) and low end-to-end latency (< 300 ns across all loads for high/medium QoS traffic, and < 100 µs under nominal load for low QoS traffic), contributing to the agenda of new design challenges for optical DCNs suitable for AI/ML traffic.**

Keywords—AI/ML traffic, data center network, optical network, traffic burstiness, wdma protocol.

I. Introduction

The prevalence of ever emerging artificial intelligence (AI) and machine learning (ML) applications in all aspects of modern life has resulted to a tremendous increase of traffic in networks of all scales. Popular applications such as facial/voice recognition, Apple Siri, Amazon Alexa, Google Voice Search, ChatGPT etc. are typical paradigms in the AI/ML era. It is worth mentioning that within only the first three months of its release, ChatGPT had been used by more than 100 million users, growing faster than any other digital application [1]. For many AI/ML applications, datasets of diverse volume are acquired by multiple internet of things (IoT) sensing devices at the end-user side, overloading the access networks. These datasets are thereafter transferred to constantly growing data centers (DCs) at the edge/cloud for appropriate processing and storage, causing traffic increase in the intermediate networks in metro and wide area scale. The relevant AI/ML computation workload is mainly executed within DCs' domain, where large amounts of data are exchanged among servers for multiple deep neural networks (DNNs) training. Several training DNN parallelism techniques have been introduced: 1) the model parallelism according which models are partitioned (split) across many servers, 2) the data parallelism which distributes the datasets to multiple servers, and 3) the hybrid parallelism which combines the aforementioned techniques [2]. Moreover, many emerging neural network (NN) training models based on neuromorphic algorithms have been widely adopted for multiple AI/ML applications support, providing further traffic congestion conditions in DC networks (DCNs) [3-4].

In such a data-intensive framework due to the AI/ML dominance, new hardware accelerators such as graphics processing units (GPUs) and tensor processing units (TPUs) have been introduced to enhance the computational speed capabilities of DC servers. Nevertheless, AI/ML model training may require several days or weeks [5-7] to ensure high accuracy. Typical examples include large language models (LLMs) with trillions of parameters where training may last even for months like for the LLM of GPT4 [8-9]. Note that the number of required computations within DCs for AI/ML model training exponentially increases with 3.4 months doubling rate [10]. In this computation intensive content, Facebook and Google have announced that much of the required AI/ML computations within their DCs cannot be served by a single server as a result of the increasing size of datasets [11-12]. Therefore, computations are divided into several tasks and are distributed to multiple servers within a DC for concurrent processing, aiming for significant reduction of the response time. As an immediate result, the DCN traffic is further exploding, reaching Cisco's forecast about the DCs' Zettabyte era worldwide [13].

Within AI/ML data-explosive context, DCNs are struggling to effectively handle the tremendously increasing traffic. The datasets large size as well as the traffic high variability and burstiness pose demanding requirements in order to deliver high DCN efficiency in terms of: 1) high bisection bandwidth for servers interconnection, 2) low end-to-end latency in µs scale, 3) fast clock data recovery (CDR), 4) fast switching time in ns

Authorized licensed use limited to: University of Thessaly. Downloaded on May 17,2026 at 13:35:40 UTC from IEEE Xplore. Restrictions apply.

scale, 5) efficient medium access control (MAC) for contention resolution, 6) low energy footprint, 7) cost efficiency, and 8) high reconfigurability. Both academy and industry put much of their efforts towards fulfilling these DCN performance requirements, while the optical networking and switching technology appears as the most promising for modern big-data DCNs as compared to the electrical one [14]. Indeed, many relative prototypes for optical DCNs have been proposed, presenting diverse servers' interconnection scenarios and switching technologies for either the intra-rack or inter-rack DCN domain [15-24]. Some of them have exploited ML and deep reinforcement learning (DRL) techniques as useful tools in order to improve DCN reconfigurability under variable traffic conditions, aiming to enhance DCN performance [22-24]. In these studies AI/ML is harnessed to optimize the functionality inside the DCN. To the best of our knowledge, few works have studied the DCN traffic that explicitly comes from AI/ML applications per se as well as its characteristics, despite its indisputable dominance in DCNs. In specific, in [25] the characteristics (packet size and interarrival time distributions) of traffic coming from a typical scientific computing application, like Hadoop, in a small-scale optical inter-rack DCN are experimentally measured, while results can be also extended for larger-scale DCNs. Moreover, in [26] a simulator model for the generation of DCN traffic originated explicitly by AI/ML applications is presented, based on traffic statistical modeling. Apart from the above-mentioned studies about optical DCNs traffic characteristics coming from big-data AI/ML applications, to the best of our knowledge in literature there is no specific proposal about the required DCN design directions in order to efficiently handle the traffic originated explicitly by AI/ML applications, in order to deliver high DCN performance and quality of service (QoS).

In this paper, we investigate the design challenges for optical DCNs in order to effectively serve the tremendously increasing traffic of AI/ML applications. Contrary to existing studies that - to the best of our knowledge- do not specifically study these challenges, we propose a novel optical intra-rack DCN as an optical local area network (LAN) model to interconnect servers that exchange traffic explicitly coming from AI/ML applications for computing and storage purposes. According to the suggested optical intra-rack DCN architecture, the rack servers are interconnected via an optical passive coupler using a multi-channel wavelength division multiplexing (WDM) system for control and data communication. In addition, we propose a WDM access (WDMA) protocol based on an efficient contention resolution mechanism effectively facing any possible collision over the WDM channels and at destination. In our DCN architecture and WDMA protocol, traffic variability and burstiness is taken under consideration based on the results of [25-26]. In specific, three traffic priority classes are assumed to adopt the AI/ML applications traffic demands diversity within a DCN in terms of response time and QoS delivery. Simulation results show that the proposed WDMA protocol provides: 1) high bandwidth exploitation (almost 100% for low loads and more than 90% under the nominal load of 400 Gbps), 2) low end-to-end latency across the intra-rack domain (lower than 300 ns across all loads for high/medium priority traffic, and lower than 100 μs under nominal load for low priority traffic), and 3) high access fairness among servers. Therefore, it is proved to be an efficient access proposal for optical intra-rack DCNs which mainly serve the ever emerging AI/ML applications. The remaining of the study is organized as follows: Section II presents the proposed intra-rack DCN architecture and WDMA protocol. The performance evaluation is given in Section III. Finally, Section IV outlines some conclusions.

II. AI/ML Traffic & Network Model

A. AI/ML Traffic Modelling

For our DCN traffic model, we utilize the Trafpy traffic generator [26], an open-source Python-based framework that realistically generates traffic patterns in existing DCNs. Trafpy assumes a flow-centric DCN traffic paradigm, where a single demand of AI/ML application's computation is considered as a flow. Each flow represents a specific task that demands data to be sent from a source to a destination server within the DCN. Each flow is described by the following characteristics: flow size, flow arrival (generation) time and source/destination server identifiers. Specifically, we utilize Trafpy's Tensorflow benchmark, as the most relevant for AI/ML-related traffic simulation. The generated traffic characteristics of such flows are depicted in Fig. 1a and 1b [26]. Flow sizes span from 15.88 to 91.84 MBytes with an average value of 49.99 Mbytes, as in Fig. 1a. The average and maximum inter-arrival time between flows is 0.842 and 10.8 msec, respectively, as in Fig. 1b.

A traffic flow is essentially a sequence of traffic packets from a source to a destination server. Since Trafpy does not allow for such packet-level modelling, we partition each generated flow into packets, based on the packet traffic characteristics that are identified for Hadoop tasks [25], as the most relevant to AI/ML traffic. Specifically, almost 90% of the packets have a size of 1500 Bytes, while the remaining ones assume various sizes from 80 to 900 Bytes [25], as shown in Fig. 1c. Also, in 90% of the cases the packet inter-arrival time is less than 8 μs [25], as presented in Fig. 1d. Each packet is characterized by its source and destination server, QoS (priority) class, and packet size. For the QoS (priority) class, we assume the following classification rule: small-sized packets (0-200 Bytes) represent high priority packets that account for network control messages requiring immediate service. Medium (200-900 Bytes) and large (1500 Bytes) sized packets are considered as time-sensitive (e.g., AI/ML model configuration) and delay-tolerant (e.g., training data) AI/ML messages respectively that

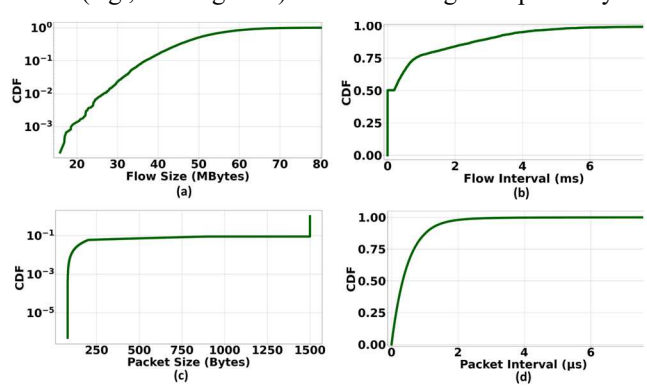

Fig. 1. AI/ML traffic characteristics within DCNs. Flow-level: (a) size and (b) arrival times. Packet-level: (c) size and (d) inter-arrival time.

Authorized licensed use limited to: University of Thessaly. Downloaded on May 17,2026 at 13:35:40 UTC from IEEE Xplore. Restrictions apply.

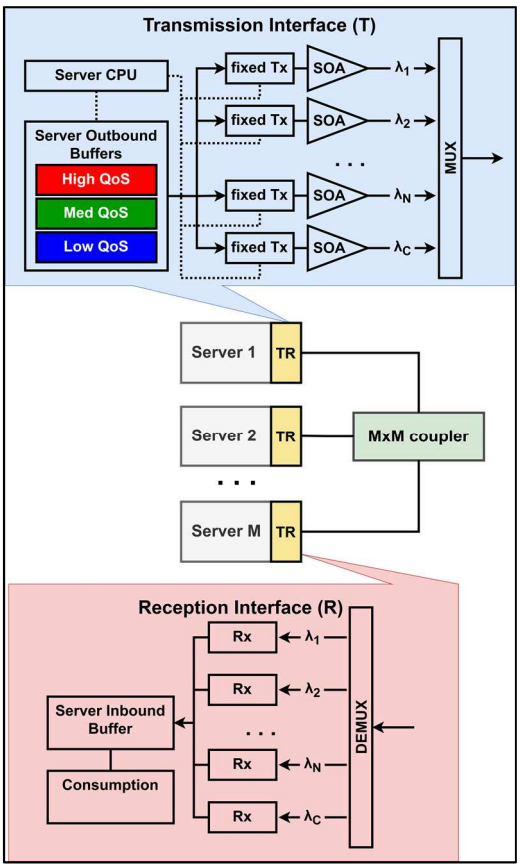

Fig. 2. Network and server architecture.

correspond to medium (med) and low QoS classes respectively, based on the assumptions of [16].

B. Network and Server Architecture

As presented in Fig. 2, our proposed solution assumes a typical intra-rack optical DCN architecture [16], where each rack consists of M servers and a total of $(N+1)$ wavelengths (channels). A total of N wavelengths ($\lambda_1, \lambda_2, \ldots \lambda_N$) are used for data packet exchange and one wavelength (λ_c) for the exchange of control information among servers. Each server is able to communicate with all the other servers within the rack via an $M\times M$ passive optical coupler (see Fig. 2). To communicate with the coupler, each server uses an optical fiber and a multiplexer-demultiplexer (MUX-DEMUX). To facilitate transmission (T) and reception (R) respectively, each server is equipped with $(N+1)$ burst mode fixed wavelength transmitters (Tx) and $(N+1)$ fixed wavelength receivers (Rx), each of which is tuned to a specific wavelength. After each laser a semiconductor optical amplifier (SOA) is used to ensure low switching on/off time T_S (almost 900 ps) with low power consumption [28]. For packet storage, each server accommodates a total of four electrical buffers (see Fig. 2). Specifically: a) three outbound buffers are used to store packets (corresponding to the three traffic priority classes high, med and low, as described in Sec. IIA) until their transmission, and b) one inbound buffer for packet reception and thereafter consumption. Finally, each server is equipped with a central processing unit (CPU), which in turn is responsible to run the suggested WDMA protocol which is given in Sec. IIC.

C. WDMA Protocol

To ensure high performance in terms of high bandwidth utilization and fairness among servers transmissions, we exploit the rules first proposed in our previous MAC protocol for intra-rack optical DCNs [16] and we adapt them to allow for packets of variable size and QoS class that are relevant of AI/ML traffic (see Sec. IIA). The time axis is divided into equal cycles, synchronized among the $(N+1)$ wavelengths as shown in Fig. 3. The cycle duration T_c is equal to the transmission time of the largest possible packet (that is 1500 Bytes for AI/ML generated traffic), plus a guard band time of $(2\times T_S)$ to account for the required switching on/off times of burst mode lasers [28].

In each cycle over the control channel, M slots are further assumed, equal to the total number of servers, as Fig. 3 presents. Each server reserves its respective slot over the control channel per cycle to transmit its transmission requests. In other words, throughout system operation, a specific reserved slot per cycle is pre-assigned to each server. Let's consider cycle i as in Fig. 3, during which all servers announce their transmission requests for the next cycle $(i+1)$ by transmitting their control packets over their respective control slot, plus a few bits that denote the current traffic hierarchy matrix for fairness purposes [16] (denoted by their transmission time T_H in Fig. 3). In specific, the transmission request of each server consists of several control mini-packets, each corresponding to the respective data packet that is next-in-line for transmission at the server's high, med and low QoS buffers (see Fig. 2). Specifically, each mini-packet

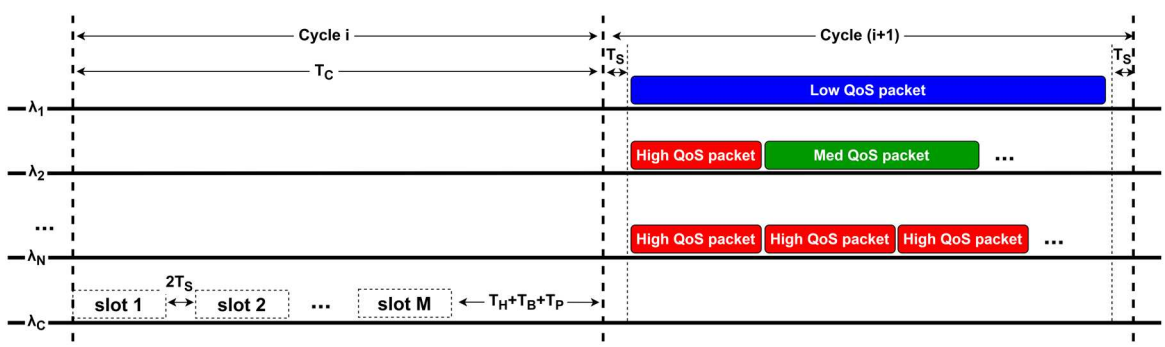

Fig. 3. Distributed WDMA protocol

Authorized licensed use limited to: University of Thessaly. Downloaded on May 17,2026 at 13:35:40 UTC from IEEE Xplore. Restrictions apply.

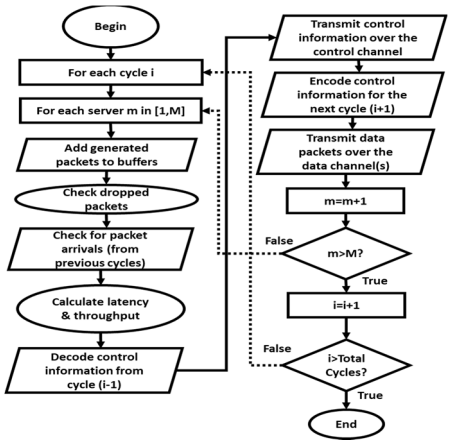

Fig. 4. WDMA protocol's block diagram.

includes information in terms of the respective data packet's size, QoS class, source and destination server. The traffic hierarchy matrix on the other hand is essentially a list of all servers' indexes that randomly changes in a per-cycle basis, ensuring fairness. The matrix is responsible to resolve potential transmission conflicts among servers. As such, in case of a transmission conflict (e.g., two servers request to send a large packet in the same channel), servers with higher matrix index will be favored over the ones with low index. Note that a guard band of ($2\times T_S$) is introduced between consecutive servers transmissions. Also, during each cycle additional time periods are reserved to allow for control mini-packets processing (T_B) and propagation delay (T_P) which are assumed 7 and 25 ns respectively [16].

Upon reception of the all control mini-packets during cycle *i*, each server runs the suggested distributed WDMA protocol to determine its transmission rights for the next cycle *(i+1)*. The algorithms that facilitate WDMA are presented in Fig. 4. In a cycle base, for each channel the WDMA protocol functions as follows: Firstly, it checks all servers (by priority i.e., by their matrix indexes) for potential transmission of high QoS packets. If such packets are in line for transmission they are immediately scheduled on the specific channel. If there are not enough available high QoS packets to fill the channel's bandwidth, the protocol checks for med QoS packet transmission from all servers that represent time-sensitive AI/ML-related messages. Finally, if no high nor med QoS packets are found, the channels are filled with low QoS packets that refer to AI/ML data and are therefore more delay-tolerant. Once the WDMA protocol has determined the transmission rights for each server during the cycle *(i+1)*, the data packets transmission is coordinated in a conflict-free and fair manner.

III. Performance Evaluation

To evaluate our proposed intra-rack optical DCN solution, we have performed a series of experiments using our developed Python-based network simulator. Throughout our experiments, we assume a traffic model that corresponds to AI/ML-related traffic [25-26], as described in Sec IIA.

A. Simulation Setup

In our simulation setup we assume an intra-rack optical DCN, with a total of M=16 servers and N=4 data channels, plus one control channel each @ 100 Gbps i.e. for a nominal bandwidth of 400 Gbps. T_S is set to 0.9 ns, for a total guard band of $2\times T_S$=1.8 ns [28]. The size of all buffers is set to 1 MByte [25]. The respective traffic flows are generated using the Trafpy software (Tensorflow benchmark for AI/ML traffic) [26]; flow partition to traffic packets is performed using the packet characteristics of Sec. IIA [25]. All experiments have a total duration of 1 s (simulation time). The following metrics are used to evaluate our solution's performance: 1) Throughput and dropping rate (bps) that denote the successfully transmitted and dropped packets (total bytes) respectively during a time period *t*, divided by the duration *t* for all servers, 2) Throughput per server (bps), which denotes the average throughput value for a single server, 3) Dropping probability (unitless) that is the ratio of the number of dropped packets to the total number of

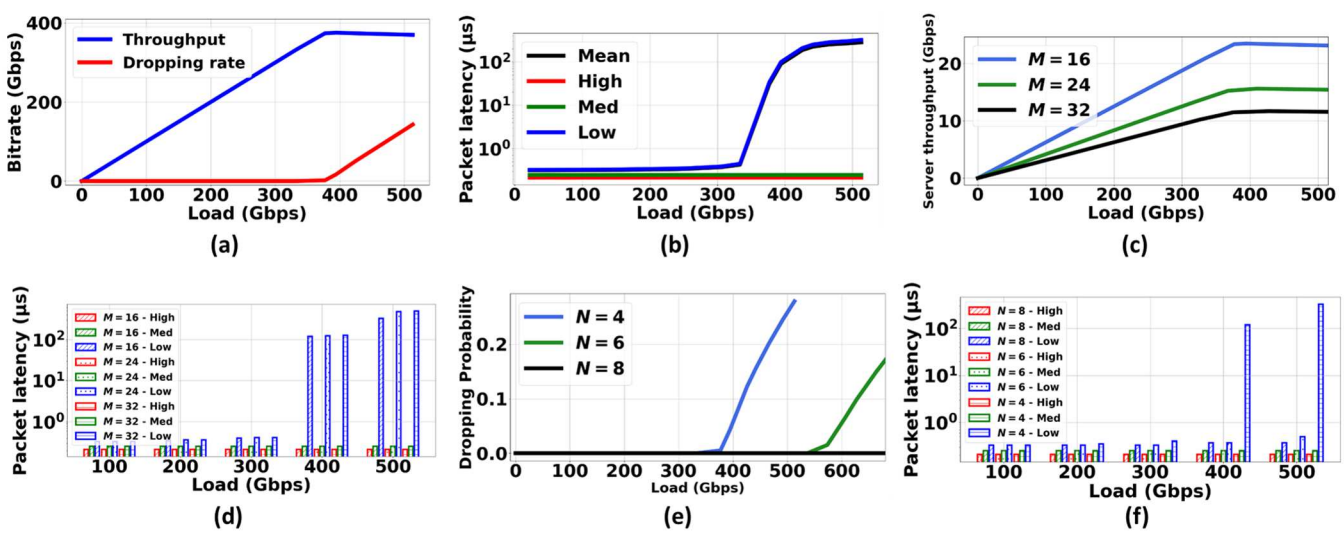

Fig. 5. For M=16 servers and N=4 data channels: (a) Total throughput, dropping rate, and (b) average and per QoS latency with respect to (w.r.t.) total offered load. For M=16, 24 ,32 servers and N=4 data channels: (c) Throughput per server, and (d) per QoS latency w.r.t. total offered load. For M=16 servers and N=4, 6, 8 data channels: (e) Dropping probability and (f) per QoS latency w.r.t. total offered load.

Authorized licensed use limited to: University of Thessaly. Downloaded on May 17,2026 at 13:35:40 UTC from IEEE Xplore. Restrictions apply.

generated packets, 4) Average packet latency (sec) and per QoS class packet latency (sec) that show the average and per priority class end-to-end latency respectively, for a successful packet transmission (from packet generation to packet reception).

B. Throughput and End-to-End Latency Evaluation

We firstly evaluate our suggested WDMA protocol using the default setup of M=16 servers and N=4 channels @ 100 Gbps. As depicted in Fig. 5a, our protocol achieves 99.99% bandwidth utilization up until a total load of 378 Gbps (that is 94.5% of the system's nominal capacity). In fact, for loads up to 378 Gbps there are no packet drops (the corresponding value of dropping rate is zero). This behaviour is achieved as a result of the strict transmission coordination schema introduced by our WDMA protocol that carefully schedules all transmissions per cycle, based on the control information received by each server during the previous cycle in regards to its AI/ML traffic transmission requests. The system reaches a saturation point at 378 Gbps; as the load increases, the total throughput remains fixed. This results in a linear increase on the dropping rate (see Fig. 5a). These findings are also reflected in the average and per QoS class end-to-end packet latency that is presented in Fig. 5b. For loads up to 332 Gbps, the average latency has a maximum value of 0.43 μsec. From that point onwards it increases exponentially (note the log scale in y-axis in Fig. 5b). Interestingly, the latency both for high and medium priority packets remains fixed at lower values regardless the load value at 0.21 and 0.25 μsec, respectively. This is due to our introduced effective transmission mechanism that prioritizes high and medium QoS packets over the low ones, in line with the low-latency requirements for AI/ML traffic for the respective classes (see Sec. IIC). As such, the low QoS packets' latency also shapes the average packet delay, as depicted in Fig. 5b.

C. Effect of Scaling the Number of Servers

We then study the effect of scaling the number M of servers per rack (M=16, 24, 32). Note that M corresponds to typical number of servers inside a rack [15]. The total number of data channels remains fixed (N=4). As presented in Fig. 5c, scaling results in a gradual decrease of the achieved throughput per server. For a nominal load of 400 Gbps the per-server throughput reaches a maximum value of 23.8, 15.7 and 11.8 Gbps for M=16, 24 and 32, respectively. Note though that the maximum total throughput i.e., the per-server throughput multiplied by the total number of servers remains fixed at 378 Gbps, in line with the findings of Sec. IIIB. Our results demonstrate that scaling the number of servers M has no effect on the total throughput. This behaviour suggests that our proposed protocol manages to effectively coordinate transmission in a conflict-free and fair manner, under various server population and traffic load densities. The effect of server scaling on the packet latency (analysed per priority class) is depicted in Fig. 5d. Scaling has no effect on high and medium QoS packets' latency, which remains fixed at 0.21 and 0.25 μs, respectively across all loads (as demonstrated in Sec. IIIB). This is particularly crucial as these packets correspond to control or AI/ML time-sensitive messages (see Sec. IIA) that are expected to be served with minimal latencies. On the other hand, low QoS packets are less prioritized since they refer to relatively delay-tolerant messages. Specifically, up until 300 Gbps they exhibit a maximum latency of 0.43 μsec. Then, scaling the number of servers from 16 to 24 and 32, will result in an increase of latency by 2.66%, 6.11% (@400 Gbps) and 46.69%, 52.67% (@500 Gbps), respectively. Therefore, WDMA effectively serves the diverse traffic QoS classes with fair manner providing significantly low end-to-end delay (lower of 300 ns) for high and medium priority class across all load values.

D. Effect of Scaling the Number of Channels

In this section , we conduct a performance evaluation of our proposed WDMA protocol's behavior, when the number of data channels N is increased (N=4, 6, 8). The total number of servers remains fixed (M=16) and so does the bitrate of each channel (@ 100 Gbps). The effect of data channel scaling on the packet dropping probability is presented in Fig. 5e. For loads up to 378 Gbps, the dropping probability has a maximum value of 0.6% for N=4, in line with the results presented in Sec. IIIB. From that point onwards, the protocol reaches its saturation point (for a load equal to 94.5% of the nominal capacity) as previously discussed for Fig. 5a, resulting in the linear increase of dropping probability. For loads up to 400 Gbps (which is the nominal capacity for N=4), the dropping probability does not exceed 5%. Our protocol exhibits similar performance when the number of channels are increased to N=6 (nominal capacity of 600 Gbps). The saturation point reaches a load of 565 Gbps, which corresponds to 94% of the nominal capacity. For N=8, dropping probability is zero, since the system does not reach saturation for loads up to 680 Gbps. These findings showcase our WDMA protocol's ability to effectively exploit the available channel bandwidth, when the number of channels is increased, grace to its (maximum) throughput utilization logic that is described in Sec. IIC. The effect of data channel scaling on the packet latency is depicted in Fig. 5f. Similarly to what was demonstrated in the previous sections, scaling has no effect on the high and medium QoS packet latency, across all loads. It however benefits low QoS packets, which in turn experience a decrease of latency by 5.71%, 17.5%, 99.69% and 99.84% at load values 200, 300, 400 and 500 Gbps, respectively, when the channels are increased from 4 to 6. A further increase to N=8 however, has only a minor additional effect on latency decrease (99.89% at 500 Gbps).

E. Comparison with State of the Art (SotA)

In this section we provide a comparative analysis between our protocol (WDMA) and other proposed solutions (SD-POIRN [19] and POTORI [30]) suitable for intra-DCNs.

TABLE I. PROTOCOL COMPARISON

Protocol	***Norm. Load = 0.5***		***Norm. Load = 1***	
	Throughput (Gbps)	Latency (ms)	Throughput (Gbps)	Latency (ms)
WDMA (proposed)	**32**	**<0.01**	**64**	**<0.1**
SD-POIRN	32	0.1	60	<1
POTORI	30	1	55	<10

Authorized licensed use limited to: University of Thessaly. Downloaded on May 17,2026 at 13:35:40 UTC from IEEE Xplore. Restrictions apply.

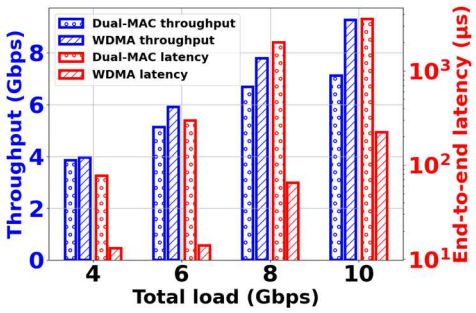

Fig. 6. WDMA vs. Dual-MAC comparison

Initially we present a numerical comparison of achieved throughput and end-to-end packet latency of all protocols, based on [16]. The results are shown in Table I. Our protocol achieves up to 16% better bandwidth efficiency (in terms of throughput to offered load ratio) and at least 10 times latency reduction, as a result of its coordination mechanism that offers collision-free transmission. To validate the numerical results, we performed a simulation-based comparative study between WDMA and Dual-MAC [15] which is also suitable for intra-DCNs (with the setup discussed in Sec. IIIA). For fairness, we have downgraded our initial solution (in terms of bitrate and number of channels) to match the specifications of the other solutions. As shown in Fig. 6, WDMA achieves an up to 35% better bandwidth efficiency (particularly in high offered loads). In parallel, the latency is reduced by a factor of 10 for high loads; for lower loads the reduction can reach up to 100.

IV. Conclusion

We have presented an intra-rack optical DCN architecture based on the LAN paradigm, along with our efficient WDMA protocol that is meticulously designed to serve AI/ML traffic. Contrary to SotA, we have evaluated our solution using realistic AI/ML traffic datasets from Trafpy, an established DCN traffic generator, with various packet QoS profiles. Our results suggest that WDMA achieves an almost loss-less transmission for loads up to 94.5% of the system's nominal capacity, across a wide range of configurations (number of servers and data channels per rack). Our protocol ensures that the end-to-end packet latency for high and medium priority AI/ML packets is kept at sub-μsec level regardless the traffic offered load. As a future work, our planned activities include: a) augmenting our datasets with traffic patterns from social networks, academic etc., b) further developing our DCN architecture to account for both intra- and inter-rack DCN traffic, and finally, c) conducting a performance evaluation of our protocol in an actual testbed that includes multiple racks, each equipped with up to 8 servers, and d) exploring our protocol tolerance when intra-DCN scales up.

References

[1] Ubs: Chatgpt may be the fastest growing app of all time, 2023. Online: https://aibusiness.com/nlp/ubs-chatgpt-is-the-fastest-growing-app-of-all-time#close-modal.

[2] M. Khani et al., "SiP-ML: high-bandwidth optical network interconnects for machine learning training," in Proc. SIGCOMM 2021, pp. 657–675.

[3] C.D. Schuman, "Opportunities for neuromorphic computing algorithms and applications," Nature Computational Science, vol. 2, pp. 10–19, 2022.

[4] M. Moralis-Pegios et al., "Photonic Neuromorphic Computing: Architectures, Technologies, and Training Models," in Proc. OFC, 2022.

[5] B. Shah, H. Bhavsar, "Time Complexity in Deep Learning Models," Procedia Computer Science, vol. 215, pp. 202-210, 2022.

[6] M. Cho et al., "Blueconnect: Decomposing all-reduce for deep learning on heterogeneous network hierarchy," IBM J. Research and Development, vol. 63, no. 6, pp. 1:1 - 1:11, Nov.-Dec. 2019.

[7] D. Narayanan et al., "PipeDream: generalized pipeline parallelism for DNN training," in Proc. SOSP '19, Canada, 2019, pp. 1–15.

[8] OpenAI. Gpt-4 technical report, 2023.

[9] Gpt-4 has a trillion parameters-report, 2023. Online: https://the-decoder.com/gpt-4-has-a-trillion-parameters/

[10] AI and compute. Online: https://openai.com/research/ai-and-compute

[11] N. Farrington, A. Andreyev, "Facebook's data center network architecture," in Proc. IEEE Optical Interconnects Conference, 2013.

[12] S. Kanev et al., "Profiling a warehouse-scale computer," in Proc. ACM/IEEE 42nd Annual ISCA, 2015.

[13] Cisco Visual Networking, "The Zettabyte Era–Trends and Analysis," Cisco white paper, 2017

[14] Y. Zhao et al., "Optical Switching Data Center Networks: Understanding Techniques and Challenges," Computer Networks and Communications, vol. 1, no. 2, pp. 272-291, 2023.

[15] Y. Zheng, X. Sun, "Dual MAC based hierarchical optical access network for hyperscale data centers," J. Lightwave Technology, vol. 38, pp. 1608–1617, 2020.

[16] P. Baziana, G. Drainakis, "Collision-free distributed MAC protocol for passive optical intra-rack data center networks," J. Optical Communications and Networking, vol. 14, pp. A1-A23, 2022.

[17] X. Xue et al., "Automatically reconfigurable optical data center network with dynamic bandwidth allocation," J. of Optics, vol. 23, 114003, 2021.

[18] J. L. Benjamin, G. Zervas, "Scaling PULSE data center network architecture and scheduling optical circuits in sub-microseconds," in Proc. OFC2020, USA, 2020.

[19] Y. Cai et al., "SD-MAC: Design and evaluation of a software-defined passive optical intrarack network in data centers," Trans. Emerging Telecommunications Technologies, vol. 33, no. 8, e3764, 2019.

[20] P. Baziana et al., "Software-defined optical intra-data center network and access control Strategy," Optical Switching and Networking, vol. 45, 100679, 2022.

[21] Z. Cao et al., "Hi-LION: Hierarchical large-scale interconnection optical network with AWGRs," J. Optical Communications and Networking, vol. 7, A97–A105, 2015.

[22] R. Proietti et al., "Machine-Learning-Aided Bandwidth and Topology Reconfiguration for Optical Data Center Networks," in Proc. OFC2021, USA, 2021.

[23] H. Guo et al., "Machine Learning Assisted Optical Network Resource Scheduling in Data Center Networks," in Proc. ONDM 2019, 2019.

[24] C. Zhi et al., "The flexible resource management in optical data center networks based on machine learning and SDON," J. Optical Switching and Networking, Vol. 39, 100594, Nov. 2020.

[25] X. Guo et al., "QoS-aware data center network reconfiguration method based on deep reinforcement learning," J. of Optical Communications and Networking, vol. 13, no. 5, 2021.

[26] C.W.F. Parsonson, et al., "Traffic generation for benchmarking data centre networks," Optical Switching and Networking, vol. 46, 100695, 2022.

[27] X. Xue and N. Calabretta, "Synergistic switch control enabled optical data center networks," IEEE Commun. Mag. 60(3), 62–67 (2022).

[28] J. Benjamin, et al., "PULSE: optical circuit switched data center architecture operating at nanosecond timescales," J. Lightwave Technol. 38, 4906–4921 (2020)

[29] M. Ghobadi, Optimizing AI Systems with Optical Technologies. Online: https://people.csail.mit.edu/ghobadi/slides/Scaling%20AI%20Systems%20with%20Optical%20I_O_Manya_Ghobadi.pdf

[30] Y. Cheng et al., "POTORI: a passive optical top-of-rack interconnect architecture for data centers," J. Opt. Commun. Netw. 9, 401–411 (2017).

Authorized licensed use limited to: University of Thessaly. Downloaded on May 17,2026 at 13:35:40 UTC from IEEE Xplore. Restrictions apply.