JOURNAL OF
Optical Communications
and Networking

Fast and scalable all-optical network architecture for distributed deep learning

Wenzhe Li,[1] Guojun Yuan,[1,*] Zhan Wang,[1] Guangming Tan,[1] Peiheng Zhang,[1,2] and George N. Rouskas[3]

[1] *Institute of Computing Technology, CAS, No. 6 Kexueyuan South Road Zhongguancun, Haidian District, Beijing, China*
[2] *Institute of Intelligent Computing Technology, CAS, 88 Jinji Lake Avenue, Industrial Park, Suzhou, China*
[3] *Department of Computer Science, North Carolina State University, 890 Oval Drive, Raleigh, North Carolina 27695, USA*
**yuanguojun@ncic.ac.cn*

Received 10 November 2023; revised 16 January 2024; accepted 29 January 2024; published 22 February 2024

With the ever-increasing size of training models and datasets, network communication has emerged as a major bottleneck in distributed deep learning training. To address this challenge, we propose an optical distributed deep learning (ODDL) architecture. ODDL utilizes a fast yet scalable all-optical network architecture to accelerate distributed training. One of the key features of the architecture is its flow-based transmit scheduling with fast reconfiguration. This allows ODDL to allocate dedicated optical paths for each traffic stream dynamically, resulting in low network latency and high network utilization. Additionally, ODDL provides physically isolated and tailored network resources for training tasks by reconfiguring the optical switch using LCoS-WSS technology. The ODDL topology also uses tunable transceivers to adapt to time-varying traffic patterns. To achieve accurate and fine-grained scheduling of optical circuits, we propose an efficient distributed control scheme that incurs minimal delay overhead. Our evaluation on real-world traces showcases ODDL's remarkable performance. When implemented with 1024 nodes and 100 Gbps bandwidth, ODDL accelerates VGG19 training by 1.6× and 1.7× compared to conventional fat-tree electrical networks and photonic SiP-Ring architectures, respectively. We further build a four-node testbed, and our experiments show that ODDL can achieve comparable training time compared to that of an *ideal* electrical switching network.

https://doi.org/10.1364/JOCN.511696

1. INTRODUCTION

Deep learning has emerged as a primary tool for cognitive applications, such as image classification, object detection, and language translation [1]. The success of deep learning is driven by increasing training datasets and model size, which require a large amount of computation [2]. However, the training process may take tens of days with increased computation [3]. Therefore, distributed deep learning (DDL) training, which accelerates the speed of training with scalable parallel hardware, is becoming a popular solution. Among the parallelism mechanisms for DDL, data parallelism is a typical and widely used one [4,5]. A large number of parameter updates need to synchronize in data-parallel training, leading to significant communication overhead that may account for as high as 90% of the total time [3]. Thus, communication has become the major bottleneck of large-scale DDL frameworks.

One of the major collective communication operations in distributed training is Allreduce, which sums and averages the data among workers [6]. Lots of previous works [7–9] focus on the Allreduce algorithm optimization to achieve either bandwidth-optimal, such as ring Allreduce [10,11], or latency-optimal, such as recursive doubling (RD) or recursive halving and doubling (HD) Allreduce [12]. Moreover, it is also important to optimize the underlying network architecture for distributed training. High-bandwidth network solutions such as NVIDIA DGX [13] have been proposed to improve the performance of intra-node communication. However, inter-node architectures have much lower bandwidth and higher latency than intra-node, which has become the main bottleneck of the performance of DDL [14]. For inter-node network architecture, the conventional multi-layer electrical network suffers from high latency, which is caused by multi-hop transmission and limited switching bandwidth. Fortunately, with recent advancements in photonic technology, it is now possible to provide extra high bandwidth for intra-node communication [15,16]. In addition, a fast optical switch can adapt the topology dynamically to different communication algorithms. With the dynamic topology reconfiguration, network resources may be fully utilized, and latency overhead may be minimized. Therefore, all-optical switching architecture is a promising solution to optimize communication performance for generalized distributed training applications.

Traditional high-port optical circuit switches (OCSs) are relatively slow in that they take tens of milliseconds (relatively slow) to reconfigure; therefore, they are often used as supplementary elements in electrical networks [17–19]. These architectures can reduce the congestion in electrical networks, but their performance is limited by the bandwidth of the electrical network. All-optical networks based on slow OCSs usually reconfigure the network when the traffic pattern changes significantly in order to avoid frequent reconfiguration. In this case, the network needs an over-provision of circuits to guarantee all communication requirements during the period. As a consequence, the node must integrate more network interfaces, and the optical switch must provide more ports per application, resulting in low network utilization. An OCS network with fine-grain transmit scheduling, which reconfigures the network on a per-flow or per-destination basis, can increase network utilization, thereby improving the scalability of the network architecture. To achieve finer granularity of transmitting scheduling, system-level network reconfiguration time, which includes optical switch reconfiguration time, network control time, and link establishment time, should be in the order of microseconds even to nanoseconds. Fortunately, fast wavelength switching technology based on a tunable laser is a promising solution to implement nanosecond optical switching. Although the switching capacity of fast wavelength switching is limited by the number of available wavelengths, we can overcome this bottleneck by combining it with a scalable and wavelength-sensitive optical switch.

This paper proposes optical DDL (ODDL), a high-bandwidth and flat all-optical switching architecture that employs fine-grained topology reconfiguration to improve the performance of large-scale DDL. ODDL leverages fast optical switch technology and a highly efficient distributed control plane to achieve real-time network configuration to adapt the topology to the communication demand. Specifically, the distributed control plane includes an arbiter for initialization and control units implemented in each transceiver. The control units configure the optical circuit by tuning the wavelength of transmitters during the training and guarantee that the payload is transmitted in the correct timeslot. The transmission states are synchronized between control units of the source and the destination when a new circuit needs to be built to avoid disturbing ongoing payload transmission. Furthermore, the proposed distributed control plane does not need global synchronization, keeping the control overhead low and stable when the system scales up. Our work makes the following contributions:

- We introduce ODDL to facilitate distributed training applications via flow-based topology reconfiguration, which achieves single-hop communication for each flow and maximizes the link utilization with highly dynamic traffic pattern matching.
- We analyze the scalability of ODDL in terms of port and wavelength, which determine the scale of the system and the maximum parallelism of training supported by the architecture. The utilization of a multi-dimensional interconnection topology and the availability of abundant wavelength resources supported by optical devices allow ODDL to scale up to thousands of nodes and support high levels of parallelism.
- We design a fine-grained transmit scheduling implemented within the distributed control plane, including the optimized communication library and control unit embedded in the network interface of each node.
- We develop a detailed simulator with real training traces to evaluate the performance of ODDL as the system size increases, and simulation results show that ODDL speeds up the training time significantly compared with the electrical network and another optical solution.
- We build a four-node prototype to validate the feasibility of ODDL, and we show that it achieves an overall performance comparable to ideal one-tier electrical switching networks.

Overall, our solution takes advantage of a fast reconfigurable all-optical network to accelerate communication in distributed training and implements an efficient control plane for the network with flow-based transmit scheduling. The rest of the paper is organized as follows. In Section 2, we review earlier works on network architectures for distributed training. We describe the overall fast all-optical switching architecture for DDL in Section 3. In Section 4, we present the distributed control plane for the proposed architecture. We evaluate the performance of the proposed architecture with simulations and a four-node prototype in Section 5. We discuss potential applications and the future optimization direction of ODDL in Section 6, and we conclude the paper in Section 7.

2. RELATED WORK

The network architecture of DDL has been the subject of extensive research [20–25]. In [23], researchers study different electrical network topologies for DDL and compare the conventional fat-tree with BCube topology. Theoretical analysis and simulation results show that BCube can achieve higher performance than fat-tree topology in terms of synchronization time because servers in BCube offer more input/output ports and BCube achieves better load balance. This work proves that the performance of DDL can benefit from network topology optimization. However, the topology of the electrical network is fixed once the network devices are deployed, and the performance of DDL may be limited by electrical switching bandwidth and the fixed topology.

Other researchers explore the potential of the optical network to adapt the topology to traffic patterns in DDL. References [26–28] use OCSs to interconnect top-of-racks (ToRs) and dynamically configure the OCS to match the traffic pattern. In Ref. [29], SiP-based OCSs are inserted between ToRs and aggregation switches, and between servers and ToRs. This work utilizes the OCS for server regrouping and bandwidth steering, thus optimizing ring Allreduce and synchronized parameter server algorithms. The reconfigurable optical network can help alleviate network congestion or reduce transmission hops in the electrical network. As bandwidth requirements increase in distributed training, network performance will be limited by electrical switching capacity.

Google's Jupiter datacenter [30] enables topology reconfiguration with OCS components and topology engineering technology. Jupiter uses large-port micro-electro-mechanical systems (MEMSs) to interconnect aggregation blocks to form block-level direct-connect topology, which can support different speed links and heterogeneous blocks. Since it is difficult to make an accurate short-term prediction of datacenter traffic, reconfiguration usually takes place every few weeks. By implementing topology engineering with an OCS, the Jupiter datacenter reduces its minimum round-trip time (RTT) and flow completion time (FCT). Furthermore, Google TPU v4 utilizes an optical reconfigurable direct-connect network to improve the performance specifically for DDL tasks. This network is strategically employed to enhance availability and utilization for large DDL workloads. TPU v4 can adaptively reconfigure the static topology according to the specific parallelism and communication requirements before the training. However, TPU may suffer from long-distance transmission and link contention during the training.

SiP-ML [24] proposes two all-optical solutions for DDL. SiP-ML framework utilizes a commercial OCS and a high-speed SiP-Ring switch to fulfill the communication demand and implements job placement with consideration of the connectivity degree of the optical network. Since the commercial OCS has a high reconfiguration time, the reconfiguration granularity of the OCS is application-level. At the same time, SiP-Ring can reconfigure the network with the estimate of the global traffic matrix due to its sub-µs reconfiguration time. When the architecture is extended to a large network, the performance of the network may be constrained by wavelength contention in the optical ring. TopoOpt [31] employs a slow OCS to achieve static topology optimization for distributed training. TopoOpt jointly optimizes topology and parallelization strategy to improve network performance. However, as distributed training workloads can exhibit significant variations in traffic patterns over time, TopoOpt may face challenges in adapting to these time-varying traffic patterns in DDL.

Several control schemes have been proposed by researchers to optimize optical switching networks in data centers. CBOSS [32] uses a software-defined networking (SDN)-based approach as its control plane. In this scheme, the scheduling application computes the timeslots assigned to each source-to-destination flow. The upper-layer centralized controller within the software stack processes incoming requests, computes slots for these requests, and subsequently dispatches messages to the respective nodes. However, this multi-step process introduces additional control delays in the software stack. Another approach, presented in [33], leverages a field programmable gate array (FPGA)-based controller to mitigate control delays. This scheme monitors all states of the whole network, limiting the scalability of the control plane. The studies in [33,34] employ a sub-µs control scheme to achieve fast reconfigurable optical networks. Their approach of using the point-to-point configuration for all controllers and ring topology may suffer from low efficiency when scaled to large numbers of nodes.

The predictable nature of traffic in DDL enables the utilization of fine-grained reconfiguration. While this approach significantly improves bandwidth efficiency, it imposes a heavy burden on the control plane due to frequent configuration updates. In this study, we address this challenge by implementing configuration decisions at the hardware layer to minimize control delays. Additionally, recognizing the communication pattern inherent in the Allreduce algorithm, we propose a distributed implementation of the control process, avoiding global information collection and management. This design aims to further reduce control overhead and ensure optimal performance, especially when scaled to thousands of nodes (refer to Section 4).

To leverage the advantages of an all-optical network and exploit the predictability of traffic of distributed deep learning, this paper introduces a scalable and highly dynamic reconfigurable all-optical network architecture, where the circuit is scheduled on a per-flow basis to match the communicating demand of machine learning accurately. Moreover, to sustain an acceptable level of control overhead as the system size increases, this paper introduces a distributed control plane. This control plane strategically exploits the inherent traffic patterns, thereby reducing control overhead. It collaborates with the communication library, enhancing the precision of reconfiguration decisions to ensure optimal performance.

3. ODDL ARCHITECTURE

Some communication algorithms of Allreduce have been proposed to improve the performance of data parallelism. Among these, ring and HD Allreduce are the most widely used Allreduce algorithms [2,11,12]. As depicted in Fig. 1, the ring Allreduce only communicates with two adjacent nodes, achieving good throughput. However, it requires $2(N-1)$ steps, a quantity that scales linearly with the number of nodes N, resulting in low communication efficiency as the system expands. In contrast, as exemplified in Fig. 2, HD Allreduce requires $2 * \log_2 N$ steps. However, in conventional fat-tree networks, certain long-distance communication (e.g., all communications in step 1) may suffer from severe contention within multi-level electrical switches. We also observe that the traffic pattern characteristic of HD Allreduce exhibits periodicity and each node only communicates with one node per step. Consequently, we can utilize the dynamic reconfigurability inherent in optical switching networks to provide single-hop transmission for communications at each step. This method can improve bandwidth utilization and reduce network latency.

The overall architecture of ODDL is shown in Fig. 3. ODDL implements a highly dynamic all-optical switching network for distributed training. Specifically, we implement flow-based transmit scheduling with fast optical link reconfiguration and achieve scalable architecture with $N \times M$ liquid crystal on silicon (LCoS) wavelength selective switch (WSS) devices. The control plane is separated from the data plane, which allows for high utilization of optical links and efficient data transmission through control messages. In this work, the control plane, which consists of control units, the arbiter,

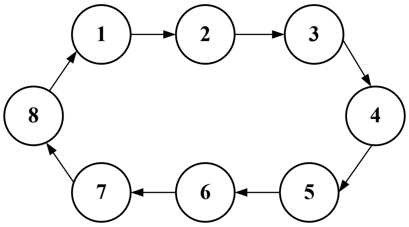


Fig. 1. Communication in ring Allreduce with eight nodes.

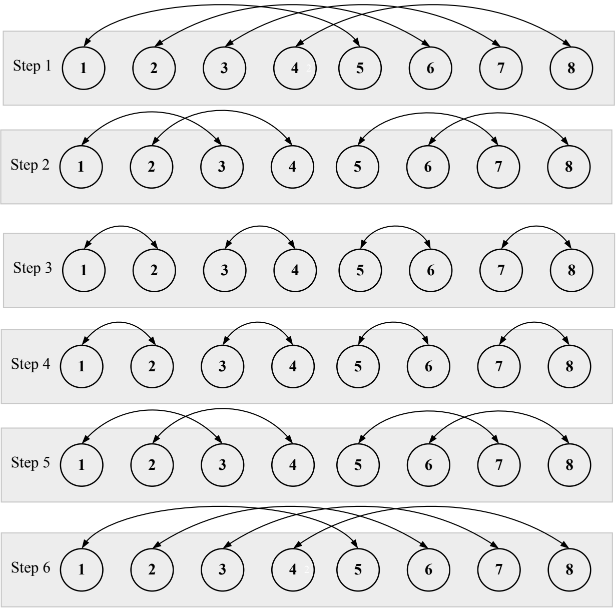


Fig. 2. HD algorithm with eight nodes.

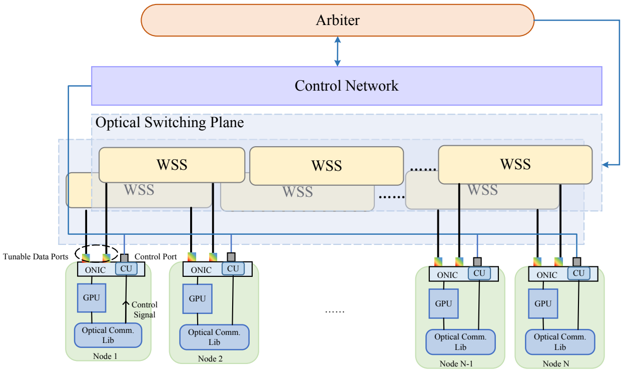


Fig. 3. Overall architecture of ODDL.

the control network, and the optical communication library, is responsible for network reconfiguration control and the transmission of small-size control messages. A scalable optical switch and high-speed tunable optical transceivers are used to implement the optical transmission plane, which provides a dedicated single-hop link for each traffic flow. At the end of the network, an optical network interface card (ONIC) is designed to ensure a reliable communication process during the reconfiguration of optical networks.

A. Reconfiguration Scheme

Current typical optical switch technology can be divided into two types according to switching speed: fast switching and slow switching. Fast switching technology, such as silicon photonics switches and tunable transceivers, can be used for fine-grained topology reconfiguration to match dynamically changing traffic. At the same time, current slow switching technology, such as a WSS, is commercially available and can offer enough ports to interconnect all needed resources for the application. Therefore, we utilize LCoS based WSS and tunable transceivers to implement topology reconfiguration in two dimensions. As shown in Fig. 4, the optical switch based on an LCoS-WSS is configured by the arbiter when a new training task arrives. Then, allocated nodes can be connected by different wavelengths (or wavelength groups), forming a subnetwork for the application. The optical switch based on an LCoS-WSS will not be reconfigured until training ends, to minimize reconfiguration overhead induced by configuring the optical switch.

Notice that the communication demand is mainly determined by the communication library in the software stack, and the predictability of traffic patterns simplifies the subnetwork setup process. Taking the eight-node HD algorithm as an example (shown in Fig. 2), Node 1 will communicate to Node 5, 3, and 2. Therefore, in the stage of optical switch reconfiguration, we only need to ensure the necessary communication defined in the HD algorithm, i.e., that Node 1 should be reachable to Node 2, 3, and 5 with specific wavelengths. During WSS reconfiguration, we simply ensure these connections are established with specific wavelengths (Table 1). Reconfiguring the WSS unlocks significant advantages for ODDL's scalability and flexibility. HD Allreduce can be implemented using only $\log_2 N$ wavelengths or wavelength groups, meaning tunable lasers need only to support this more compact set. This significantly enhances system scalability while maintaining efficient communication patterns. Furthermore, WSSs can be configured for multi-wavelength connections between ports to enable higher bandwidth. For instance, when an application requires 256 nodes, tunable lasers with 8 wavelengths can provide 1 wavelength for each destination for HD Allreduce. Conversely, if another application requests 16 nodes, the tunable laser can allocate 2 wavelengths per destination, doubling the bandwidth for those connections. This dynamic allocation contrasts with the operation of arrayed waveguide grating routers (AWGRs), which employ fixed wavelength routing. With AWGR, the number of wavelengths required in a tunable laser is determined by the size of the entire system, and the bandwidth cannot be adjusted due to the fixed routing rule.

During the training process, the optical circuit is reconfigured by tuning wavelength in transceivers according to communication demand. As an example shown in Fig. 4, Node 1 is connected to Node 5, Node 3, Node 2, Node 3, and Node 5 in turn by tuning the wavelength from λ_2 to λ_3, λ_1, λ_3, and λ_2, as shown in Table 1.

In ODDL architecture, commercial network interface cards (NICs) are no longer suitable for fast link reconfiguration, since they cannot ensure whether data flows are sent at the correct timeslot when the optical link is properly configured. Therefore, we design an FPGA-based ONIC to achieve fast and accurate reconfiguration. The block diagram of ONIC is shown in Fig. 5. Link reconfigurations based on wavelength switching and payload transmission are both managed by the control unit. Specifically, the datapath is allowed to transmit payload only when the optical link is correctly established. At the same time, the transceiver is allowed to switch wavelengths only when no payload transmission taking place. ONIC offers a reliable physical link to avoid payload retransmission caused

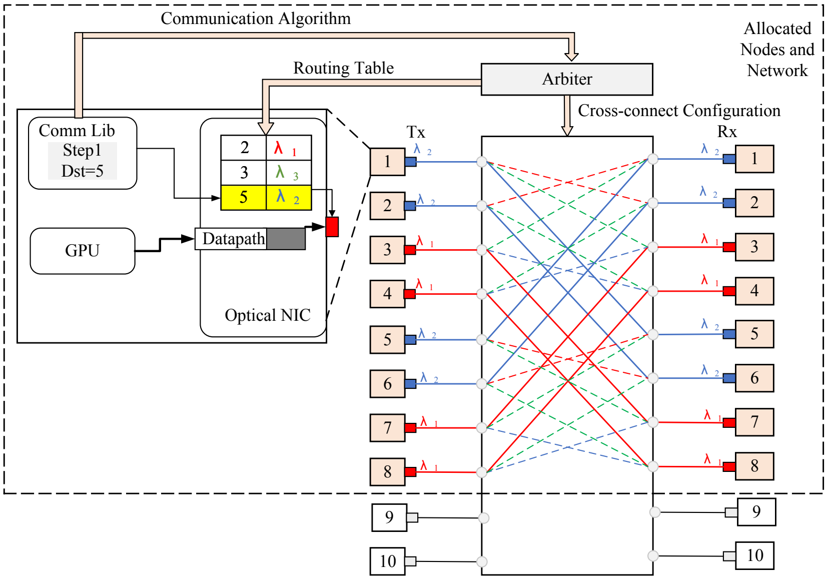


Fig. 4. Reconfiguration scheme example with eight nodes.

Table 1. Routing Table of a WSS for Eight Allocated Nodes

	Rx							
Tx	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1		λ_1	λ_3		λ_2			
2	λ_1			λ_3		λ_2		
3	λ_3			λ_2			λ_1	
4		λ_3	λ_2					λ_1
5	λ_2					λ_1	λ_3	
6		λ_2			λ_1			λ_3
7			λ_1		λ_3			λ_2
8				λ_1		λ_3	λ_2	

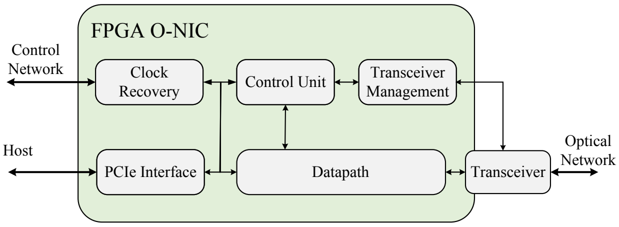


Fig. 5. Block diagram of the ONIC.

by packet loss. Furthermore, the clock recovery block is accomplished to handle the nanosecond precision clock and phase synchronization, thus supporting fast physical link establishment. The control unit synchronizes control messages with other nodes and the arbiter via the electrical control network, preferable for small-size control message transmission. The control plane will be further described in Section 4.

B. Hardware Implementation

There are two processes during network reconfiguration: optical link establishment and physical link establishment. Since optical link establishment mainly depends on the tuning speed of the laser, we apply a distributed feedback (DFB) tunable laser, which can switch wavelengths on the order of tens of nanoseconds. The tunable laser supports wavelength tuning in the range of the C band to be compatible with the LCoS-WSS device. Furthermore, the speed of physical link establishment mainly depends on the speed of clock and data recovery (CDR). The conventional CDR solution usually takes hundreds of milliseconds [35], making it impossible for fast link reconfiguration. However, with the recent clock phase caching technique, CDR locking time can be sub-nanosecond [36]. With fast tunable lasers and CDR techniques, the optical reconfiguration process can be implemented on the order of sub-microseconds, enabling shorter reconfiguration timeslots.

The growing maturity of LCoS technology has driven the commercialization of larger WSS configurations, such as 8 × 8 [37], 8 × 16 [38], and 8 × 24 devices [39]. Notably, recent developments have yielded contentionless 32 × 32 WSS products [40]. Furthermore, a larger-port $N \times N$ WSS can be constructed using a small-port WSS with the implementation of Clos topology [41], shown in Fig. 6. For example, a non-blocking 64-port WSS can be constructed with 8 8 × 16 ingress WSS modules, 16 8 × 8 middle WSS modules, and 8 16 × 8 egress WSS modules [41,42].

An optical switch based on an LCoS-WSS can route the signal by the port granularity and wavelength granularity to support two-dimensional reconfiguration. Additionally, an LCoS-WSS can be partly configured without disturbing current communication, so the application can be dynamically embedded. By reconfiguring the cross-connect of the optical switch, some ports are allocated to the arriving distributed deep learning application, and these ports are physically isolated from other ports; thus, the traffic of the application is isolated from other ongoing applications. As a result, communications and data can be more secure and stable. The configuration time of the optical switch is on the order of milliseconds, but it can be neglected in comparison with the overall training time of

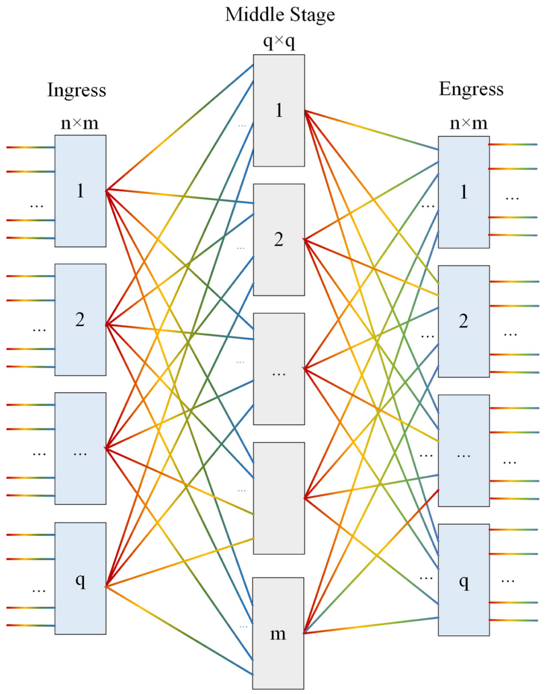


Fig. 6. $N \times N$ WSS optical switch based on the three-stage Clos topology.

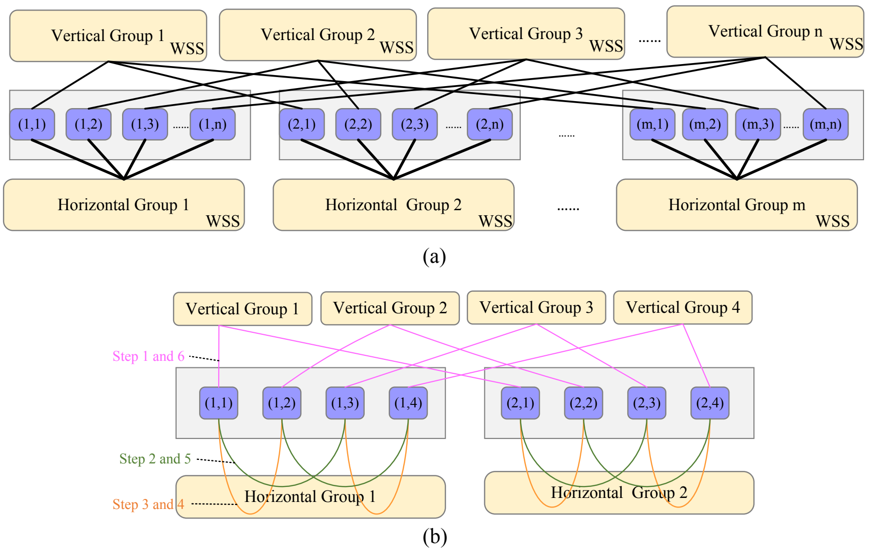
an application. Once the optical switch is configured, it can act as a wavelength routing switch. There is no extra switch configuration time when the traffic passes through the optical switch, which means the switching delay of traffic depends on wavelength switching speed.

C. Scalability Analysis

To connect all nodes in a large-scale system, ODDL can be extended in a multi-dimensional topology. Taking the two-dimensional case as an example, illustrated in Fig. 7(a), each node is individually connected to a horizontal switch and a vertical switch. Nodes within the same horizontal or vertical group are connected with a single optical switch. For instance, a 1024-node system requires 64 32-port WSSs. As the system size continues to increase, we can leverage a higher-dimensional topology. For instance, a 3-dimensional topology with 288 16-port WSSs can efficiently implement a 4096-node system. Additionally, driven by the continuous demand for flat topology and low latency in transport networks, larger-port WSSs with lower insertion loss are being developed [43]. This creates the potential for ODDL to utilize these WSSs to reduce topology dimensionality while maintaining system size in the future, resulting in a decreased number of ONICs needed.

To evaluate the impact of introducing a WSS on ODDL's performance, we perform a basic end-to-end link budget analysis. Considering a 1024-node system with an 8 × 8 × 8 topology, where the insertion loss of an 8-port WSS can be minimized to less than 6.8 dB [39], the insertion loss of single-mode fiber is around 0.5 dB per km. Given transmission distances of around 2 m (intra-rack) to 200 m (inter-rack) in modern HPC centers, the total insertion loss is estimated to be around 7 dB. At the same time, the tunable laser has a maximum output power of 4.5 dBm and receivers with a sensitivity of -10.5 dBm. In this case, the system can function without the need for an erbium-doped fiber amplifier (EDFA) for signal amplification. On the other hand, when the 1024-node system is constructed based on 32 × 32 topology, the insertion loss of a 32-port WSS ranges from 17 to 25 dB [40]. In this case, an EDFA is necessary to meet the requirements for receiving power. Similarly, if larger-port $N \times N$ WSS is applied, an EDFA is required for the system.

In a system with $n \times m \times k$ nodes, where $n = 2^i$, $m = 2^j$, and $k = 2^q$, communications in HD and RD Allreduce can be implemented with one-hop routing. As illustrated in Fig. 7(b), communications in steps 1 and 6 can be implemented with one hop using vertical interconnection, while communications in other steps can be implemented with one hop using horizontal interconnection. Similarly, when ODDL is extended into three-dimensional topology, all communication steps for



(a)

(b)

Fig. 7. 2D-based interconnection topology.

Allreduce with HD and RD algorithms can be implemented with one-hop using horizontal, vertical, or perpendicular interconnection. Moreover, collective operations, such as Allgather and ReduceScatter, can be implemented with the RD algorithm [44,45]. Since these operations exhibit the same communication steps when utilizing the same algorithm, they can also be implemented with one-hop routing in ODDL.

Additionally, as mentioned earlier, the transceivers reconfigure the topology by tuning different wavelengths (wavelength groups). Therefore, the number of available wavelengths can affect the scalability of ODDL. Since the nodes only need to communicate with a subset of other nodes during the data parallelism process, the wavelength requirement can be easily satisfied with the current LCoS-WSS and transceiver. Hence, for a system with $n \times m$ nodes, the number of wavelengths w should be no less than $\log_2(\max(n, m))$. For example, a system with 32×32 nodes using RD or HD algorithms requires at least 5 wavelengths. An LCoS-WSS device supporting 96 wavelengths in the C band has been mature and widely deployed for many years. Furthermore, LCoS-WSS will extend the wavelength range by covering the C+L band [46], thereby enabling each node pair to achieve higher bandwidth with a wider spectrum.

4. CONTROL PLANE

Since links are torn down and reestablished dynamically in ODDL, the control plane plays an important role in the quality of communication. Traditional control methods for all-optical networks usually make the reconfiguration decision according to the states of the global network, resulting in large control overhead and low network scalability. We propose a distributed control mechanism, which is implemented in the physical layer, to guarantee reliable communication. In addition, the communication library is optimized to support fast reconfiguration.

A. Optical Collective Communication Library

In the traditional communication library of distributed training, such as NCCL [47], the library needs to exchange essential channel information (i.e., Packet Sequence Number and Memory Key) with all other nodes for the following communication. It is inefficient and costly to transmit channel information through the optical network. Therefore, the optical collective communication library uses the electrical control network instead to exchange channel information. Then, the optical collective communication library uses the optical network for following payload transmission.

Due to the per-flow scheduling scheme, the completed signal of each flow will determine the validity of the reconfiguration. In DDL, the communication library manages the states of all communication. To get an accurate completed signal of one flow, the completed state is sent from the optical communication library to the control unit. Notice that there is a gap between two iterations due to a large amount of computation, the optical collective communication library informs the control unit to shut down the laser after each iteration, reducing power consumption. ODDL adds a new RDMA verb to support the delivery process of the completed states. The optical communication library guarantees the reconfiguration performance and the reliability of the underlying optical network. In addition, the optical communication library offers a standard interface for upper-layer distributed training frameworks to ensure compatibility with current training applications.

B. Distributed Control Mechanism

Due to the fact that ODDL reconfigures the topology on a per-flow basis, the control plane should be able to respond rapidly to the communication request. At the same time, the control overhead shouldn't grow linearly with system size. Therefore, in the proposed distributed control mechanism, each control unit manages its own transmission state and link state, and only exchanges control information with the destination nodes. With the distributed control mechanism, global information collection and network-level reconfiguration are avoided, thereby minimizing the control overhead and increasing the scalability. The optical switching network has a great advantage in handling predictable and bulk traffic. To guarantee the flexibility and effectiveness of the control plane, we integrate an electrical switching network for transmitting small-size control messages. Furthermore, the electrical network is also used to transmit other essential control messages, such as path-setup information during the initialization process in the communication library.

The central arbiter only handles the network reconfiguration before training, which is the first stage of reconfiguration. Moreover, the control mechanism of wavelength reconfiguration (the second stage) is implemented in the control unit in each ONIC.

In the first reconfiguration stage, shown in Fig. 3, the arbiter will send a wavelength-based routing table to the control units in the nodes that are allocated for the training task and initialize the state of the optical switch to interconnect all allocated nodes with a specific wavelength (wavelength group). Furthermore, a part of the ports of the optical switch can be reconfigured while other ports maintain cross-connection, allowing a new training task to be dynamically deployed without disturbing current tasks.

During the training process, the control units of nodes are in charge of fine-grained reconfiguration. After finishing the flow transmission, the physical link should be torn down, and a new link then be established for the next flow, which may cause packet drops during reconfiguration. However, in the physical layer, ODDL should offer a lossless fabric for the remote direct memory access (RDMA) protocol, which is essential for intranode communication in distributed training. Additionally, once a packet is dropped in ODDL, the network rebuilds the optical circuit for that packet and delays all following communication, resulting in higher retransmission overhead. In order to make a creditable reconfiguration decision, the control unit needs to check strictly two things: when all packets in one flow are received by the destination and when a new link is ready for the following communication. As shown in Fig. 8, due to the distributed control mechanism, the transmission state and link state only need to be synchronized between the source

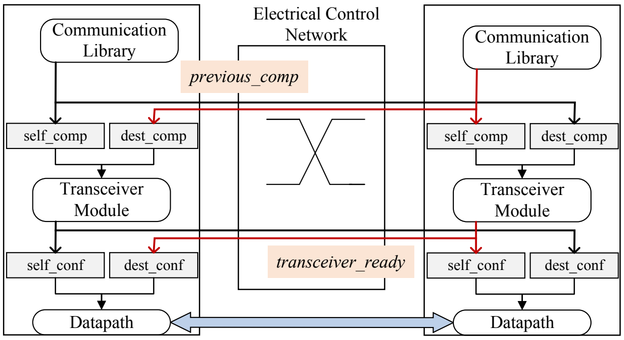


Fig. 8. Control message exchange between two nodes based on the distributed control mechanism.

and destination via a high-speed electrical control network. Only after the source and destination both complete the flow transmission in the previous step does the control unit prepare for reconfiguration. Similarly, the ONIC is allowed to send the payload only after receiving the signal, which means the physical link is ready. An overall process of training communication is described as follows.

1. The arbiter allocates the computing resource and corresponding network to the training task, and then configures the optical switch according to the allocation.
2. Allocated nodes exchange required channel information for all following RDMA transmissions through the control network.
3. The control unit configures the local transceiver for the flow based on the routing table.
4. When completing transceiver configuration, the control unit sets the flag *self_conf* to *true* and sends the *transceiver_ready* signal to the destination; the control unit receives the *transceiver_ready* signal and sets the flag *dest_conf* to *true*. If both flags are *true*, the payload begins to transmit.
5. When the payload transmission in this step finishes, the control unit sets the flag *self_completion* to true and sends the signal *previous_completion* to the destination of the following step. When the control unit receives the signal, it updates flag *dest_completion* to *true* and reconfigures the transceiver for the following step.
6. Repeat (4) and (5) until the training finishes.

As seen from the above reconfiguration process, after finishing communication in the last step, the inter-node network is reconfigured for data transmission in the next step. The time interval between adjacent transmissions is in the order of sub-milliseconds, consisting of a sub-microsecond duration for physical reconfiguration and tens of microseconds for control switching delay.

At the same time, the GPU processes the data (for example, Reduce operation), and then sends new data to the ONIC through the intra-node link. So the control delay and link reconfiguration delay can be partially overlapped with the time of data processing and transmission within the node, which can reduce the negative effect of the whole reconfiguration process. The control delay of one reconfiguration operation depends on the number of control messages and the transmission time of the electrical control network. In our control scheme, there are only two control message exchanges during the control process, shown in Fig. 8, so the control delay can be expressed as Eq. (1). T_{onehop} donates the average delay of one hop, and H_{avehops} donates the transmission hops of the control message. The average number of hops is related to the size of the electrical control network, which is based on the L3-fat-tree topology. Furthermore, the control message size is a mere 13 bits, composed of 12 bits for the address (supporting up to 4096 nodes) and 1 bit for the message type. This small control message effectively mitigates congestion in the control network. In addition, the number of control messages is minimized to further avoid contentions. Although T_{onehop} may increase with the expansion of the system size due to contention, the implementation of cut-through switching can further minimize switching delays within the control network. As the control packets are compact, they can be swiftly processed and forwarded without incurring significant buffering overhead. Consequently, control time remains relatively constant even as the system size scales, contributing to the robust scalability of the control plane. With several microseconds processing times in each electrical layer, the control delay of ODDL typically falls within the range of tens of microseconds:

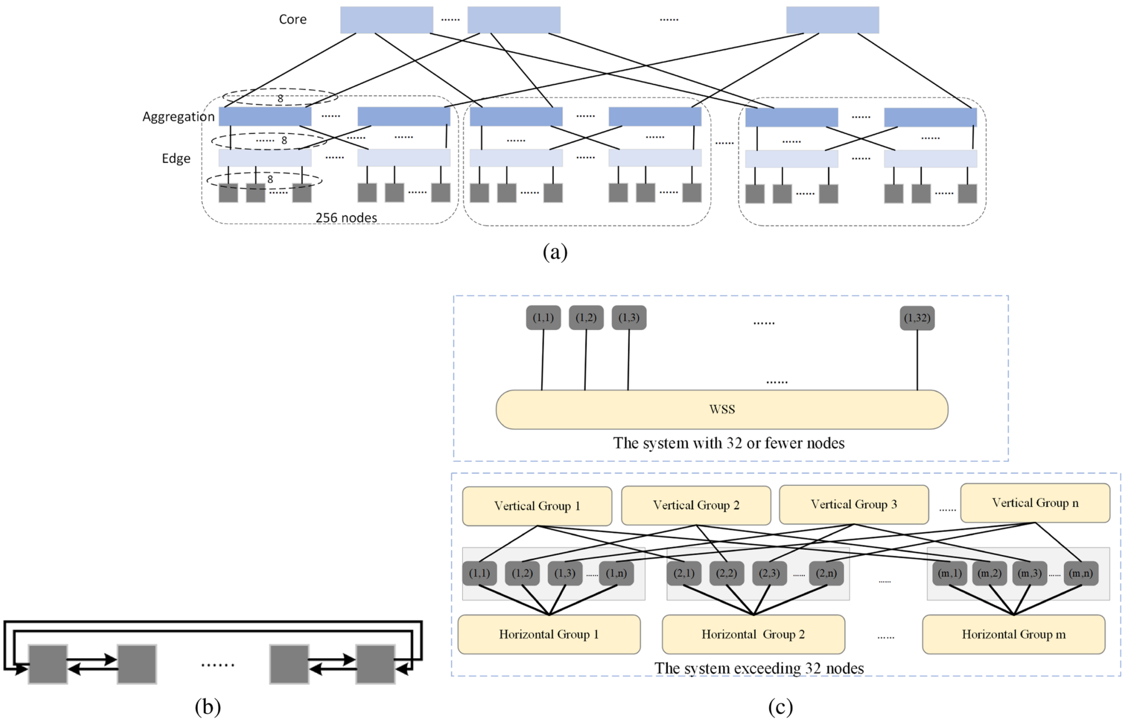
$$T_{\text{ctrl}} = 2 * T_{\text{onehop}} \times H_{\text{avehops}}. \tag{1}$$

5. EVALUATION

In this section, we evaluate the overall performance and the scalability of ODDL with software simulations built on real training traces. Also, we prototype the design on a four-node testbed to further verify the feasibility of ODDL.

A. Simulation

To evaluate the performance of ODDL and verify the proposed control plane, we simulated the ODDL and other electrical networks using the OMNeT++ framework [48]. We profile several DDL models on one NVIDIA GeForce 2080Ti GPU. Specifically, we collect the trace files of different training models using the function Timeline of Horovod [11]. The trace files collected include two types of information: Allreduce computation time before each Allreduce operation and communication volume without fusion. Then, we develop a simulator that implements gradient synchronization including the fusion process and Allreduce communication. ResNet50, VGG16, VGG19, and Lenet are used for training models. ImageNet1K is used as the training dataset. The batch size is set as 16. We measure ODDL in terms of the training time and bandwidth utilization and compare it with photonic architecture SiP-Ring and traditional fat-tree electrical networks. Figure 9 illustrates the topologies of these three networks. For the fat-tree network, we employ a switch radix of 16 and a 1:1 oversubscription ratio. When the number of nodes is 64 or smaller, the fat-tree network adopts a 2-layer topology. For systems exceeding 64 nodes, it employs a 3-layer fat-tree configuration. ODDL utilizes a single-dimensional topology for 32 nodes or fewer, switching to a 2-dimensional topology for larger systems.



(a)

(b)

(c)

Fig. 9. Topologies of the three networks used in the simulation: (a) fat-tree network with a 16-radix switch and 1:1 oversubscription, (b) SiP-Ring, (c) ODDL.

The fat-tree network employs ring and HD Allreduce algorithms, which are two typical Allreduce algorithms. SiP-Ring employs ring Allreduce [24] for data parallelism, and it can configure its topology into a ring to match the Allreduce algorithm. However, SiP-Ring encounters difficulties when employing the HD Allreduce algorithm due to bandwidth contention. For example, SiP-Ring with N nodes needs at least $N/2 - 1$ different wavelengths. ODDL only employs the HD algorithm in the simulation. To be noticed, ODDL can be reconfigured into ring topology to match ring Allreduce, which would perform similarly with SiP-Ring in this situation.

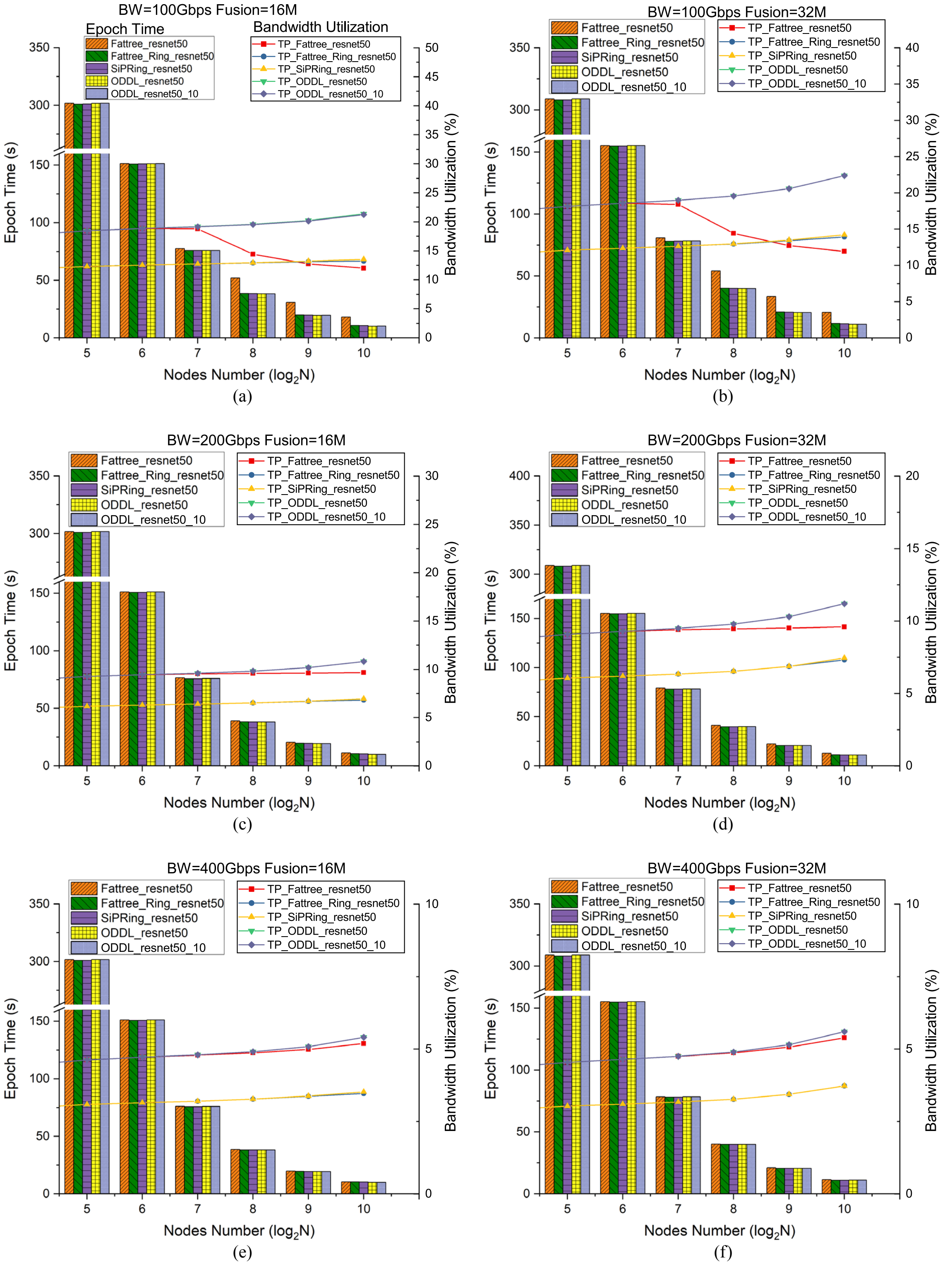
With sub-μs wavelength tuning and sub-μs CDR, the link reconfiguration time should be theoretically under 2 μs. The switching process time in the control network should be approximately 1 μs when using the Ethernet protocol. In our simulation, we have taken into account the significant impact of the control network on the overall performance of ODDL. Therefore, we set the link reconfiguration delay and switch process time of the control network at 5 μs (referred to as ODDL in the figures) and 10 μs (referred to as ODDL_10 in the figures), respectively, to evaluate the influence of the control plane. Additionally, we conducted simulations with different fusion values, which represent the data fusion threshold. A higher fusion value indicates a higher volume of communication in each Allreduce operation and a reduced number of Allreduce operations. The simulation parameters are summarized in Table 2.

Figures 10–13 present the results of epoch times, depicted as histograms, and bandwidth utilization, represented as curves, for 4 different models. These results are based on the utilization of fusion technologies with 16 MB and 32 MB, respectively. In the case of ResNet50, ODDL shows similar performance compared to SiP-Ring and fat-tree with ring Allreduce. However, fat-tree with HD Allreduce exhibits the worst performance when the network is scaled over 256 nodes among all candidate networks due to severe congestion. For the Lenet model, ODDL outperforms other networks when the fusion value is set as 16 MB. When the fusion value is increased to 32 MB, all networks perform similarly.

Table 2. Simulation Parameters

Parameters	Values
Fusion threshold	16 MB/32 MB
Fusion latency	1 μs
Batch size	16
Allreduce algorithm	HD
NIC bandwidth	100/200/400 Gbps
Link reconfiguration delay	5/10 μs
Switch process delay (fat-tree)	1 μs
Switch process delay (control network in ODDL)	5/10 μs

Utilizing the VGG16 model, which is characterized by a large number of parameters, ODDL with 1024 nodes demonstrates an acceleration in training speed ranging from 1.1× to 1.6× when compared to Sip-Ring, 1.1× to 1.55× when compared to fat-tree with ring Allreduce, and 2.4× to 7× when compared to fat-tree with HD Allreduce. The fat-tree with the HD algorithm demonstrates lower performance compared to the fat-tree with the Ring algorithm, attributed to increased long-distance traffic. At the same time, ODDL can reach approximately 85% bandwidth utilization under 100 Gbps bandwidth and maintains stability even with an



(a) (b) (c) (d) (e) (f)

Fig. 10. Performance of ODDL and two electrical networks with ResNet50.

increasing number of nodes. This advantage can be attributed to ODDL's ability to provide single-hop transmission for each flow, effectively avoiding congestion in large-scale networks. Similar results are observed with the VGG19 model, where ODDL with 1024 nodes accelerates the training speed by 1.1× to 1.7× when compared to the SiP-Ring, 1.1× to 1.6× when compared to fat-tree with ring Allreduce, and 2× to 7.1× when compared to fat-tree with HD Allreduce. ODDL presents a slightly higher performance advantage under VGG19 due to the larger parameter size of VGG19

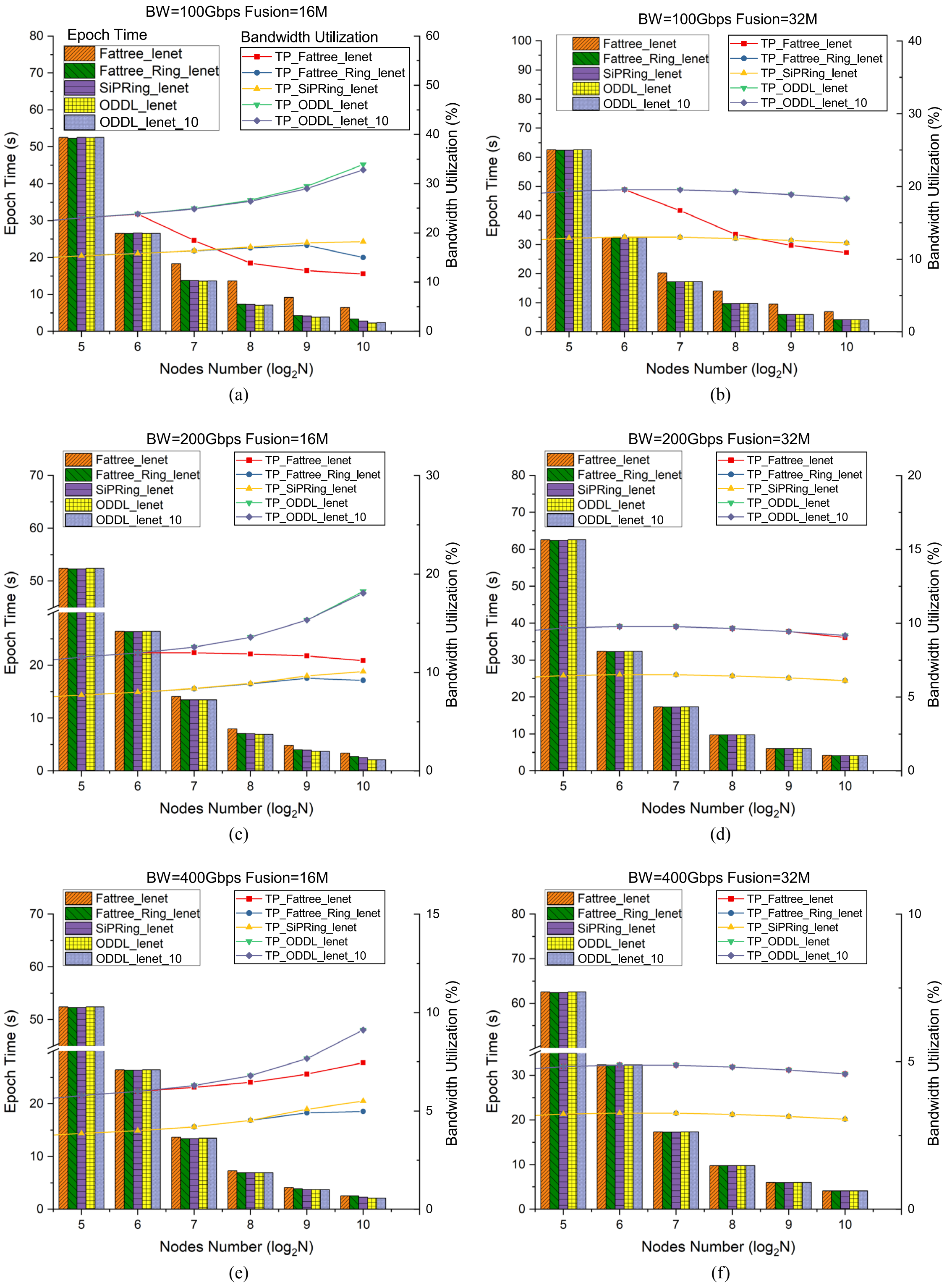
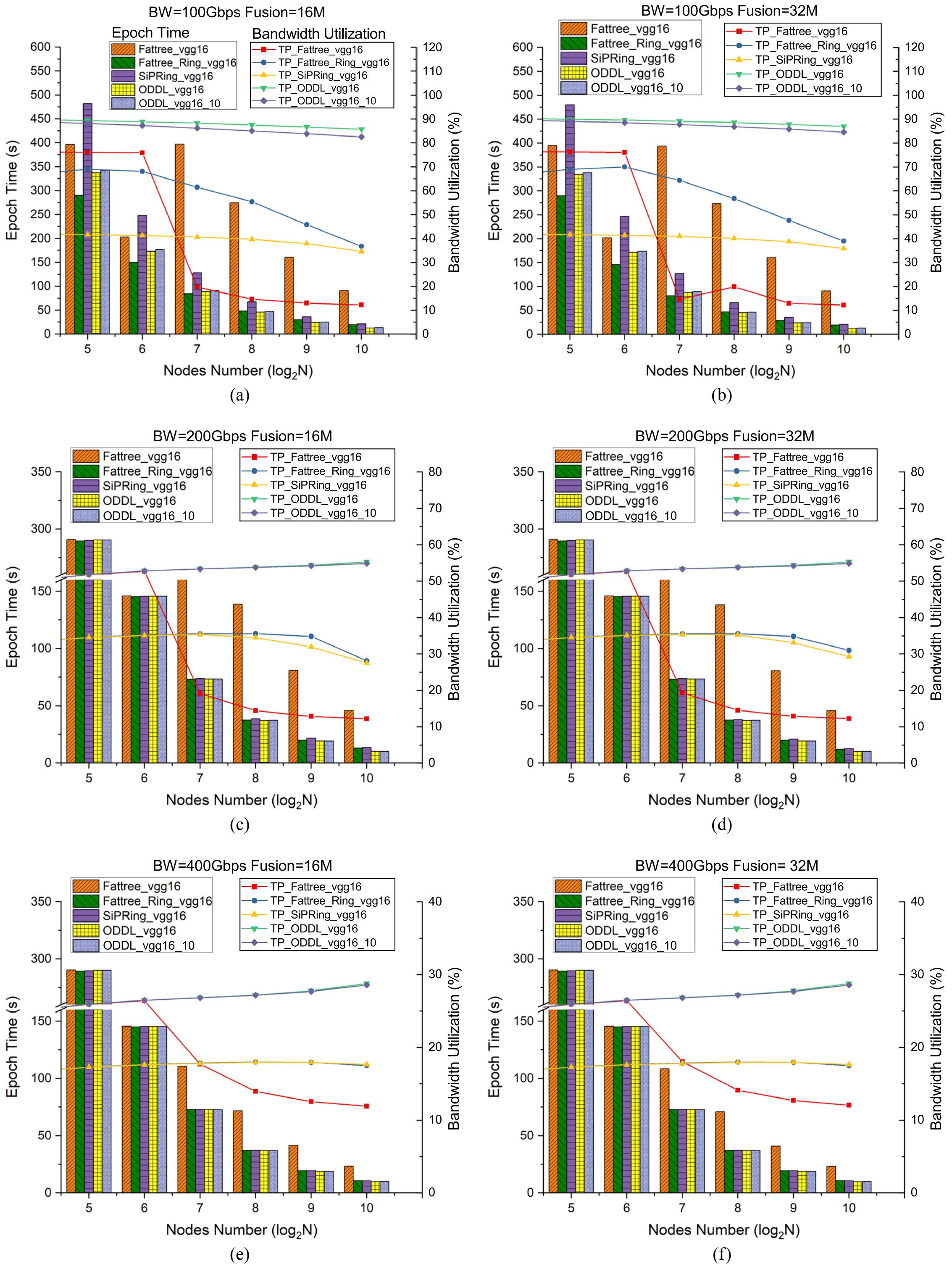


Fig. 11. Performance of ODDL and two electrical networks with Lenet.

compared to VGG16. In addition, when the reconfiguration time and control switch time increase to 10 μs, the difference in one epoch time of ODDL is less than 4%, and ODDL_10 still performs better than other networks under VGG16 and VGG19. This is because control overhead is partially overlapped with the time of data processing and transmission within the node. Therefore, we can conclude that reconfiguration overhead of a few microseconds is acceptable for the

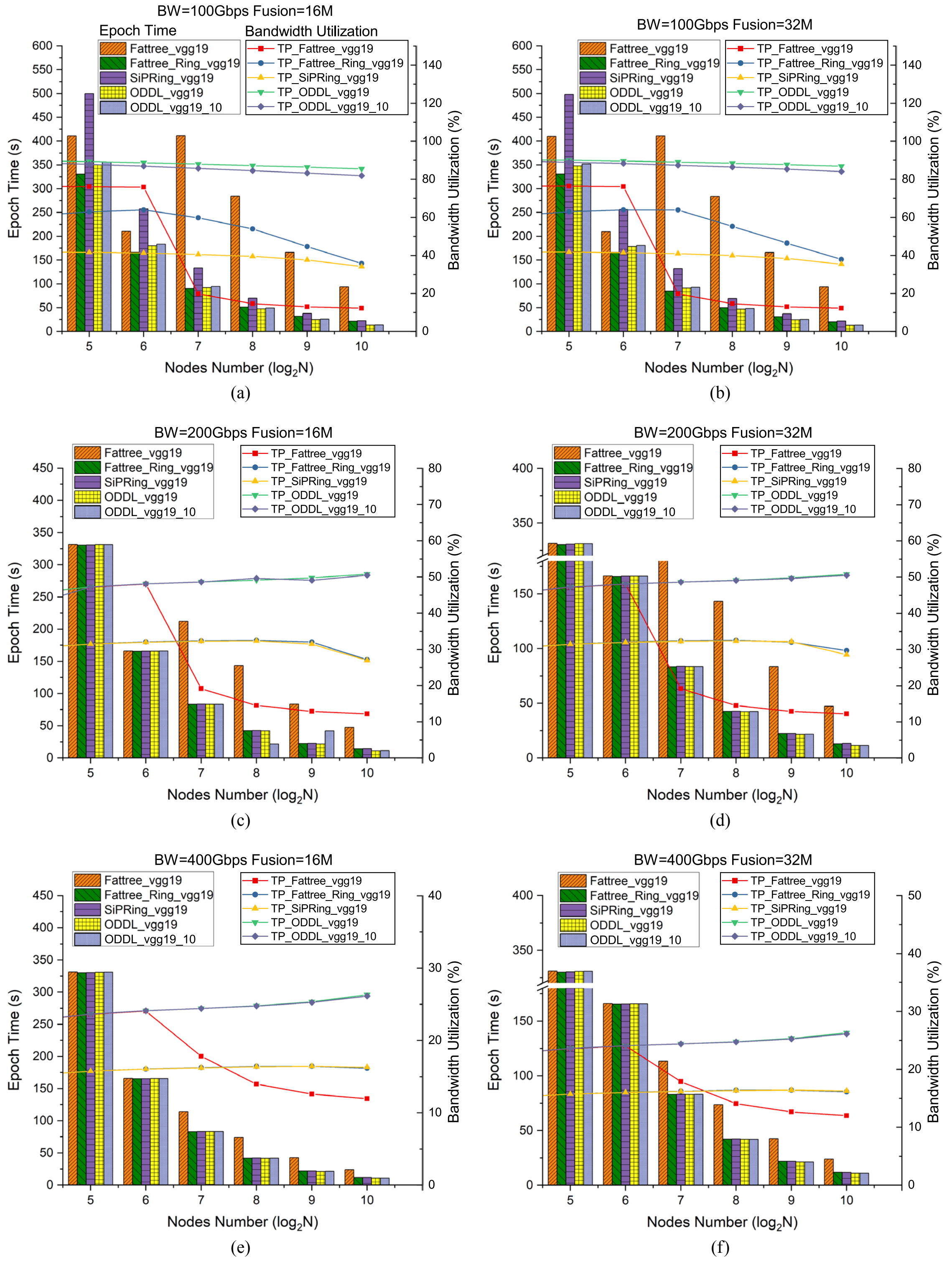


(a) (b) (c) (d) (e) (f)

Fig. 12. Performance of ODDL and two electrical networks with VGG16.

system, further validating the effectiveness of the proposed control scheme.

We can conclude from these results that for the model with small-size parameters like Lenet and Resnet, the ODDL exhibits slightly higher or similar performance than other candidate networks. When utilizing the larger model like VGG16 and VGG19, ODDL exhibits a more obvious advantage over other networks when system size increases. This is mainly because fat-tree will suffer from contention with a larger payload and large-size system. While ODDL can avoid the contention by achieving one-hop routing. At the same time, ODDL achieves better bandwidth utilization than other



(a) (b) (c) (d) (e) (f)

Fig. 13. Performance of ODDL and two electrical networks with VGG19.

networks in all cases, which can alleviate the limitation of bandwidth. With the trend of increasing the size of the training model, ODDL has the potential to achieve good network performance.

In summary, ODDL demonstrates slightly higher or comparable performance to other networks for models with small-size parameters, such as Lenet and ResNet. However, its advantages become more prominent with larger models like VGG16 and VGG19, especially as the system size increases. ODDL's ability to provide one-hop routing effectively addresses contention issues, ensuring robust scalability in contrast to fat-tree. Additionally, ODDL consistently achieves superior bandwidth

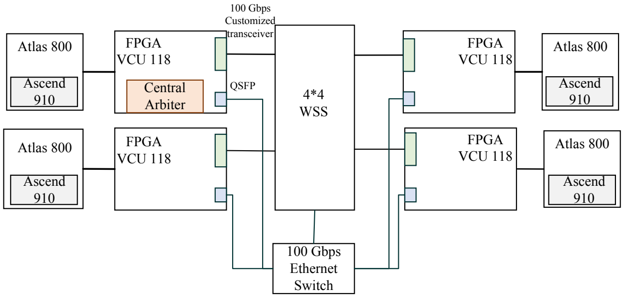


Fig. 14. Diagram of the four-node ODDL prototype.

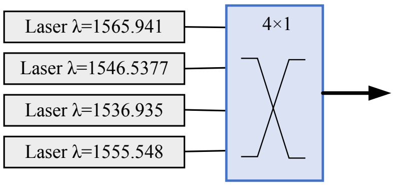


Fig. 15. Schematic diagram of the tunable laser prototype. The working wavelengths are switched by a fast 4 × 1 silicon photonic switch.

utilization in all scenarios, mitigating bandwidth limitations. Given the prevailing trend of escalating model sizes, ODDL shows promising potential for achieving excellent network performance.

B. Testbed

We build a four-node prototype to validate the feasibility of ODDL and compare its performance with an *ideal* electrical switching network. We use ImageNet as the dataset and HD as the communication algorithm. As shown in Fig. 14, we train model Resnet50 on 4 servers equipped with Ascend 910 GPU, and each server is connected to one FPGA board, which serves as an optical network interface. One of the ports on the FPGA board is connected to a WSS with a customized 100 Gbps tunable transceiver, which is based on a laser array and fast optical switches to support fast wavelength switching. The WSS is configured before training and remains unchanged during the training process. The diagram of the tunable laser in the prototype is depicted in Fig. 15. The delay characteristics of optical devices in the prototype are shown in Table 3. Since the WSS is only reconfigured once per training task (which typically lasts tens of minutes to hours), its tens of milliseconds reconfiguration time has little impact on overall training time. Throughout the training process, the WSS primarily introduces transmission time, which is deemed negligible. The photonic switches employed within the tunable laser demonstrate a switching speed of approximately 10.5 ns, as illustrated in Fig. 16. This tens of nanoseconds switching time is fast enough to satisfy the flow-based reconfiguration requirement, further proving ODDL's feasibility.

Another port on the FPGA board is connected to an Ethernet switch with a commodity 100 Gbps transceiver for the transmission of the control message. The function of the central arbiter is implemented using one of the FPGA boards. Two working wavelengths are $\lambda_1 = 1565.941$ nm and $\lambda_2 = 1546.5377$ nm. Two electrical networks use the RoCE protocol with 100 Gbps bandwidth to support RDMA. The one-tier electrical network connects all nodes with one 100 GE switch, representing an *ideal* network condition. As part of our ongoing research, we are developing a larger-scale prototype further to investigate the advantages of ODDL in practical scenarios.

Table 3. Time Cost of the Devices in the Prototype

Device	Time
Wavelength switching of the tunable laser	10.5 ns
Transmission of the WSS	Negligible
Reconfiguration of the WSS	56 ms

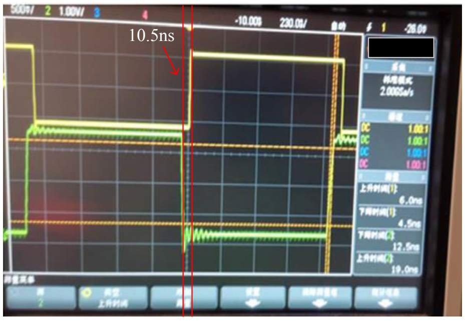


Fig. 16. Switching speed of the 4 × 1 silicon photonic switch used for the tunable laser prototype.

Table 4. Comparison of the Four-Node Prototype

Network	Training Time (min)	Accuracy (%)
ODDL	97.8	80.2
One-tier electrical network	97.3	80.2

Table 4 shows the training time of 20 epochs of Resnet50 with different networks. ODDL achieves the equivalent overall performance as the *ideal* electrical switching network, demonstrating that ODDL is efficient and feasible.

6. DISCUSSION

This paper explores the optimization of data parallelism with the flow-based scheduling all-optical network. The parameter synchronization in data parallelism is implemented by Allreduce collective communication, and Allreduce with one specific communication algorithm, such as RD or HD, presents a predictable traffic pattern and large communication volumes. ODDL can utilize the predictability of the traffic to reconfigure the topology, which can reduce the network latency and minimize the control overhead. In today's large-scale distributed training, model parallelism is also employed in addition to data parallelism. For model parallelism, the communication process mainly includes Allreduce and Allgather collective communication [49]. Allgather with RD algorithm (common algorithm for Allgather) doubles data volume after each step, resulting in a high requirement on network bandwidth. ODDL can alleviate the bandwidth limitation by providing high-bandwidth and single-hop transmission.

Therefore, the performance of Allgather could be optimized via ODDL.

Furthermore, the pre-trained model tends to adopt mixture-of-experts (MoE) to scale the model capacity. Each MoE layer requires two All-to-All collective communications. In the All-to-All collective communication, one node sends different chunks of the data to different destinations, and all nodes will receive different data chunks from other nodes. To reduce network contention caused by many-to-one communication, all-to-all is typically implemented with multiple point-to-point communications [50]. For the implementation of the all-to-all operation in ODDL, a hierarchical all-to-all algorithm [51] can be employed. This algorithm aggregates data chunks in the same dimension, mitigating communication across groups. Subsequently, the all-to-all algorithm is decomposed into all-to-all operations within each dimensional group. Similar to the reconfiguration scheme for HD and RD algorithms, ODDL dynamically reconfigures tunable lasers to sequentially communicate with all other nodes within each group. With this method, all communication steps can be implemented with one-hop routing as well. The optimization of all-to-all in ODDL will be explored in future research.

7. CONCLUSION

In this paper, we present a scalable and fast all-optical network for distributed training, along with a distributed control plane that enables fine-grained scheduling with minimal control overhead. With 1024 nodes, 100 Gbps bandwidth, and the VGG19 benchmark, ODDL significantly outperforms: it accelerates training by 1.6× compared to the traditional fat-tree network and 1.7× compared to the optical solution. In addition, the four-node ODDL prototype achieves equivalent overall performance to that of an *ideal* electrical switching network.

Funding. National Key Research and Development Program of China (2021YFB0300700); National Natural Science Foundation of China (61972380); Jiangsu Science and Technology Project (BE2022051-2); National Science Foundation (CNS-1907142).

REFERENCES

1. M. Cho, U. Finkler, M. Serrano, *et al.*, “BlueConnect: decomposing all-reduce for deep learning on heterogeneous network hierarchy,” IBM J. Res. Dev. **63**, 1:1–1:11 (2019).
2. R. Mayer and H.-A. Jacobsen, “Scalable deep learning on distributed infrastructures: challenges, techniques, and tools,” ACM Comput. Surv. **53**, 3 (2020).
3. D. Narayanan, A. Harlap, A. Phanishayee, *et al.*, “PipeDream: generalized pipeline parallelism for DNN training,” in *Proceedings of the 27th ACM Symposium on Operating Systems Principles* (2019), pp. 1–15.
4. G. Wang, S. Venkataraman, A. Phanishayee, *et al.*, “Blink: fast and generic collectives for distributed ML,” arXiv, arXiv:1910.04940 (2019).
5. N. Dryden, N. Maruyama, T. Moon, *et al.*, “Aluminum: an asynchronous, GPU-aware communication library optimized for large-scale training of deep neural networks on HPC systems,” in *IEEE/ACM Machine Learning in HPC Environments (MLHPC)* (2018), pp. 1–13.
6. Z. Tang, S. Shi, X. Chu, *et al.*, “Communication-efficient distributed deep learning: a comprehensive survey,” arXiv, arXiv:2003.06307 (2020).
7. H. Zhao and J. Canny, “Butterfly mixing: accelerating incremental-update algorithms on clusters,” in *Proceedings of the 2013 SIAM International Conference on Data Mining* (SIAM, 2013), pp. 785–793.
8. A. Agarwal, O. Chapelle, M. Dudík, *et al.*, “A reliable effective terascale linear learning system,” J. Mach. Learn. Res **15**, 1111–1133 (2014).
9. H. Li, A. Kadav, E. Kruus, *et al.*, “MALT: distributed data-parallelism for existing ML applications,” in *Proceedings of the 10th European Conference on Computer Systems* (2015), paper 3.
10. A. Gibiansky, “Bringing HPC techniques to deep learning,” Tech. Rep. (Baidu Research, 2017).
11. A. Sergeev and M. Del Balso, “Horovod: fast and easy distributed deep learning in TensorFlow,” arXiv, arXiv:1802.05799 (2018).
12. R. Thakur, R. Rabenseifner, and W. Gropp, “Optimization of collective communication operations in MPICH,” Int. J. High Perform. Comput. Appl. **19**, 49–66 (2005).
13. “NVIDIA DGX,” https://www.nvidia.com/en-us/data-center/dgx-systems/.
14. B. Klenk and L. Dennison, “Why data science and machine learning need silicon photonics,” in *Optical Fiber Communication Conference (OFC)* (2020), pape M4F.6.
15. M. Wade, M. Davenport, M. De Cea Falco, *et al.*, “A bandwidth-dense, low power electronic-photonic platform and architecture for multi-Tbps optical I/O,” in *European Conference on Optical Communication (ECOC)* (2018).
16. R. Meade, S. Ardalan, M. Davenport, *et al.*, “TeraPHY: a high-density electronic-photonic chiplet for optical I/O from a multi-chip module,” in *Optical Fiber Communication Conference (OFC)* (2019), paper M4D.7.
17. N. Farrington, G. Porter, S. Radhakrishnan, *et al.*, “Helios: a hybrid electrical/optical switch architecture for modular data centers,” in *Proceedings of the ACM SIGCOMM 2010 Conference* (2010), pp. 339–350.
18. G. Michelogiannakis, Y. Shen, M. Y. Teh, *et al.*, “Bandwidth steering in HPC using silicon nanophotonics,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)* (Association for Computing Machinery, 2019).
19. Q. Cheng, M. Glick, and K. Bergman, “Chapter 18 - Optical interconnection networks for high-performance systems,” in *Optical Fiber Telecommunications VII*, A. E. Willner, ed. (Academic, 2020), pp. 785–825.
20. L. Luo, P. West, J. Nelson, *et al.*, “PLink: efficient cloud-based training with topology-aware dynamic hierarchical aggregation,” in *Proceedings of the 3rd MLSys Conference* (2020).
21. Z. Zhu, M. Y. Teh, Z. Wu, *et al.*, “Distributed deep learning training using silicon photonic switched architectures,” APL Photonics **7**, 030901 (2022).
22. A. Sapio, M. Canini, C.-Y. Ho, *et al.*, “Scaling distributed machine learning with in-network aggregation,” in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2021), pp. 785–808.
23. M. Wang, Y. Cui, S. Xiao, *et al.*, “Neural network meets DCN: traffic-driven topology adaptation with deep learning,” Proc. ACM Meas. Anal. Comput. Syst. **2**, 26 (2018).
24. M. Khani, M. Ghobadi, M. Alizadeh, *et al.*, “SiP-ML: high-bandwidth optical network interconnects for machine learning training,” in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (2021), pp. 657–675.
25. M. Glick, Z. Wu, S. Yan, *et al.*, “Flexible optical interconnects for efficient resource utilization and distributed machine learning training in disaggregated architectures,” Proc. SPIE **12027**, 1202703 (2022).
26. C. Wang, N. Yoshikane, F. Balasis, *et al.*, “Acceleration and efficiency warranty for distributed machine learning jobs over data center network with optical circuit switching,” in *Optical Fiber Communication Conference (OFC)* (2021), paper W1E.3.
27. L. Liu, H. Yu, G. Sun, *et al.*, “Online job scheduling for distributed machine learning in optical circuit switch networks,” Knowl.-Based Syst. **201–202**, 106002 (2020).

28. T.-N. Truong and R. Takano, “Hybrid electrical/optical switch architectures for training distributed deep learning in large-scale,” IEICE Trans. Inf. Syst. **E104.D**, 1332–1339 (2021).
29. Z. Zhu, S. Yan, M. S. Glick, *et al.*, “Silicon photonic switch-enabled server regrouping using bandwidth steering for distributed deep learning training,” in *Optical Fiber Communication Conference (OFC)* (2021), paper Th5H.3.
30. L. Poutievski, O. Mashayekhi, J. Ong, *et al.*, “Jupiter evolving: transforming Google’s datacenter network via optical circuit switches and software-defined networking,” in *Proceedings of the ACM SIGCOMM 2022 Conference* (Association for Computing Machinery, 2022), pp. 66–85.
31. W. Wang, M. Khazraee, Z. Zhong, *et al.*, “TopoOpt: co-optimizing network topology and parallelization strategy for distributed training jobs,” in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2023), pp. 739–767.
32. N. Benzaoui, J. M. Estarán, E. Dutisseuil, *et al.*, “CBOSS: bringing traffic engineering inside data center networks,” J. Opt. Commun. Netw. **10**, B117–B125 (2018).
33. M. Szczerban, N. Benzaoui, J. Estaran, *et al.*, “Real-time control and management plane for edge-cloud deterministic and dynamic networks,” J. Opt. Commun. Netw. **12**, 312–323 (2020).
34. H. Santana, A. Mefleh, and N. Calabretta, “Fast-controlled and time-slotted photonically interconnected edge computing and time-sensitive networks,” in *Conference on Lasers and Electro-Optics* (Optica Publishing Group, 2023), paper SF2M.6.
35. K. Clark, H. Ballani, P. Bayvel, *et al.*, “Sub-nanosecond clock and data recovery in an optically-switched data centre network,” in *European Conference on Optical Communication (ECOC)* (IEEE, 2018).
36. K. A. Clark, D. Cletheroe, T. Gerard, *et al.*, “Synchronous sub-nanosecond clock and data recovery for optically switched data centres using clock phase caching,” Nat. Electron. **3**, 426–433 (2020).
37. L. Zong, H. Zhao, Z. Feng, *et al.*, 8 × 8 flexible wavelength cross-connect for CDC ROADM application,” IEEE Photonics Technol. Lett. **27**, 2603–2606 (2015).
38. Z. Yuan, W. Li, R. Yang, *et al.*, “8 × 16 wavelength selective switch with full contentionless switching,” IEEE Photonics Technol. Lett. **31**, 557–560 (2019).
39. P. D. Colbourne, S. McLaughlin, C. Murley, *et al.*, “Contentionless twin 8 × 24 WSS with low insertion loss,” in *Optical Fiber Communication Conference* (Optica Publishing Group, 2018), paper Th4A.1.
40. Huawei, “Huawei OSN 9800 P32 brochure,” https://carrier.huawei.com/~/media/cnbgv2/download/products/networks/wdm-otn/osn-9800-p32-en.pdf.
41. J. Lin, T. Chang, Z. Zhai, *et al.*, “Wavelength selective switch-based Clos network: blocking theory and performance analyses,” J. Lightwave. Technol. **40**, 5842–5853 (2022).
42. C. Clos, “A study of non-blocking switching networks,” Bell Syst. Tech. J. **32**, 406–424 (1953).
43. Y. Ma, L. Stewart, J. Armstrong, *et al.*, “Recent progress of wavelength selective switch,” J. Lightwave Technol. **39**, 896–903 (2021).
44. R. Thakur and W. D. Gropp, “Improving the performance of collective operations in MPICH,” in *European Parallel Virtual Machine/Message Passing Interface Users’ Group Meeting* (Springer, 2003), pp. 257–267.
45. K. S. Khorassani, C.-H. Chu, Q. G. Anthony, *et al.*, “Adaptive and hierarchical large message all-to-all communication algorithms for large-scale dense GPU systems,” in *IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* (IEEE, 2021), pp. 113–122.
46. S. Yamamoto, H. Taniguchi, Y. Kisaka, *et al.*, “First demonstration of a C + L band CDC-ROADM with a simple node configuration using multiband switching devices,” Opt. Express **29**, 36353–36365 (2021).
47. A. A. Awan, K. Hamidouche, A. Venkatesh, *et al.*, “Efficient large message broadcast using NCCL and CUDA-aware MPI for deep learning,” in *Proceedings of the 23rd European MPI Users’ Group Meeting (EuroMPI)* (Association for Computing Machinery, 2016), pp. 15–22.
48. A. Varga, “Discrete event simulation system,” in *Proceedings of the European Simulation Multiconference (ESM)* (2001).
49. A. Castelló, M. F. Dolz, E. S. Quintana-Ortí, *et al.*, “Analysis of model parallelism for distributed neural networks,” in *Proceedings of the 26th European MPI Users’ Group Meeting* (2019).
50. C. Hwang, W. Cui, Y. Xiong, *et al.*, “Tutel: adaptive mixture-of-experts at scale,” arXiv, arXiv:2206.03382 (2022).
51. S. Rajbhandari, C. Li, Z. Yao, *et al.*, “DeepSpeed-MoE: advancing mixture-of-experts inference and training to power next-generation AI scale,” in *International Conference on Machine Learning* (PMLR, 2022), pp. 18332–18346.