TCP Flow Classification and Bandwidth Aggregation in Optically Interconnected Data Center Networks

Houman Rastegarfar, Madeleine Glick, Nicolaas Viljoen, Mingwei Yang, John Wissinger, Lloyd LaComb, and Nasser Peyghambarian

***Abstract*—Optical functionality is being used to realize new data center architectures that minimize electronic switching overheads, pushing the processing to the edge of the network. A challenge in optically interconnected data center networks is to identify the large, bandwidth-hungry flows (i.e., elephants) and efficiently establish the optical circuits. Moreover, the amount of optical resources to be provisioned during the network planning phase is a critical design problem. Flow classification accuracy affects the efficiency of optical circuits. Optical channel bandwidth, on the other hand, directly relates to the additive-increase, multiplicative-decrease congestion control mechanism of the transmission control protocol and affects the effective bandwidth allocated to elephant flows. In this paper, we simultaneously investigate the impact of two important mechanisms on data center network performance: traffic flow classification accuracy and optical bandwidth aggregation (i.e., the consolidation of several low-capacity channels into a single high-capacity one by employing advanced modulation formats for short-reach communications). We develop a discrete-event simulator for a hybrid data center network, enabling the tuning of flow classification parameters. Our simulations indicate that data center performance is highly sensitive to the aggregation level. We could observe up to a 74.5% improvement in network throughput only due to consolidating the optical channel bandwidth. We further noticed that the role of flow classification becomes more pronounced with higher bandwidth per wavelength as well as with more hot-spot traffic. Compared to a random classification benchmark, adaptive flow classification could lead to throughput improvements as large as 54.7%.**

***Index Terms*—Bandwidth aggregation; Congestion control; Data center; Elephant flow; Flow classification; Machine learning; Mouse flow; TCP protocol.**

I. Introduction

The past few years have seen the emergence of novel data center proposals, taking advantage of both electrical and optical interconnects [1–11]. Optical interconnects, capable of ultra-high switching capacities, bit-rate transparency, and low power density, are promising candidates to meet the scale, footprint, and power density requirements of massive data centers. Purely optical interconnects, however, suffer from the lack of a viable, all-optical buffering technology and relatively low reconfiguration speeds based on commercially available optical micro-electro-mechanical (MEMS) switches. Thus, a dual- or multi-fabric data center design that combines the advantages of both electrical and optical switching is being investigated by the research community.

Data center traffic measurements point to a rich set of traffic patterns with differing characteristics and requirements [12–15]. Typically, there is a very large number of short-lived, delay-intolerant flows (mice) and a small number of long-lived, bandwidth-hungry flows (elephants). We define elephants as flows whose total size exceeds a threshold and mice as those whose size is below this threshold (in our analysis, this classification threshold is set to 100 MB). While the number of elephants is significantly smaller than the number of mice in a data center, the majority of bytes are carried in elephant flows. For the best network behavior, the flows should be directed and scheduled to satisfy the demands while optimizing the network performance. In an all-electrical data center network, proper flow placement involves attempting to distribute elephants uniformly across links, while in an optically interconnected data center (as depicted in Fig. 1), higher performance could be achieved when long-lived, high-bandwidth flows are assigned to optical links, delay intolerant or control flows are placed on electrical links, and proper resource sharing is put in place [16–18].

From another perspective, flow behavior in data centers is strongly affected by the dynamics of the network transport layer, with the transmission control protocol (TCP) being the dominant transport protocol in data centers. Figure 2 depicts the dynamics of TCP (Tahoe variant) that arise from the feedback congestion control mechanism envisioned in this protocol. In Fig. 2, a slow-start threshold (ssthresh) is used to regulate the flow transmission rate in two distinct regimes. Below this threshold, the flow congestion (send) window size (or equivalently its transmission rate) is doubled per round-trip time (RTT). This

Manuscript received March 4, 2016; revised June 16, 2016; accepted August 29, 2016; published September 26, 2016 (Doc. ID 260483).

H. Rastegarfar (e-mail: rastegarfar@gmail.com), M. Glick, M. Yang, J. Wissinger, L. LaComb, and N. Peyghambarian are with the College of Optical Sciences, University of Arizona, Tucson, Arizona 85721, USA.

H. Rastegarfar is currently with the Department of Signals and Systems, Chalmers University of Technology, 412 96 Gothenburg, Sweden.

N. Viljoen is with Netronome Systems, 2903 Bunker Lane, Santa Clara, California 95054, USA.

http://dx.doi.org/10.1364/JOCN.8.000777

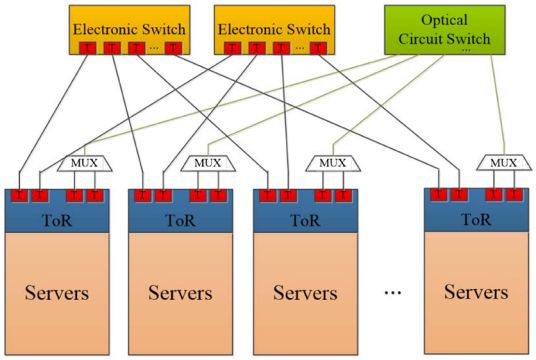

Fig. 1. Possible optically interconnected data center network architecture (T, transceiver; ToR, top-of-rack switch; MUX, wavelength multiplexer).

denotes an exponential increase of transmission rate and is called the slow-start phase. Once the send window size exceeds ssthresh, congestion avoidance starts and the flow rate is incremented linearly per RTT. When a flow's path is saturated, congestion occurs, and the flow congestion window will collapse to one segment worth of bytes and its status will be switched to slow start. However, ssthresh for the new iteration will be updated to one half of the maximum window size the flow could achieve at the congestion point. The additive-increase, multiplicative-decrease (AIMD) feature of the TCP congestion window size ensures fairness among flows and that links can operate at high utilization without being clogged due to sustained congestion [19,20]. In studying the performance of an optical data center with a combination of mice and elephants, the impact of TCP dynamics becomes a significant issue [17].

The main contribution of this paper is to study the joint impact of TCP flow classification and bandwidth aggregation on the performance of a hybrid data center interconnect using both electrical and optical switches. Data center traffic is comprised of TCP mouse and elephant flows. With an efficient classification algorithm, it is feasible to allocate resources to flows according to their requirements and avoid over-provisioning (e.g., allocating an optical circuit to short mouse flows) and under-provisioning

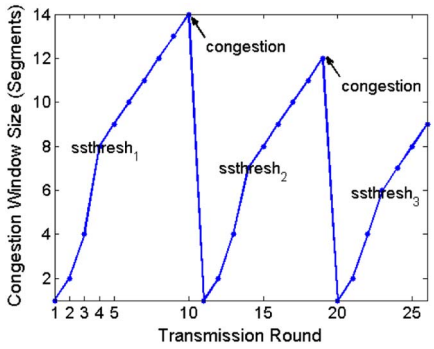

Fig. 2. Evolution of TCP congestion window (Tahoe variant). Each transmission round corresponds to one round-trip time.

(e.g., mapping a bulk data transfer to a resource-constrained electrical switch) problems. We build upon previous studies which suggest that optimized flow classification yields improvement in resource utilization [16,17,21]. We note that due to advances in processors, algorithms, and with the advent of big data, machine learning is being revived for applications in communication networks and has the potential to address the problem of rapid and accurate flow classification in data centers, in addition to adding the possibility of adaptability to traffic dynamics. We develop a Monte Carlo data center simulation framework, enabling the tuning of flow classification parameters, and model the performance of an adaptive flow classifier to examine its impact on network throughput. We consider a basic machine learning classifier as described in Section II.

The trade-offs of the various possible hardware configurations for aggregating bandwidth are an important question in current photonic implementations. The trade-offs for cost and energy are regularly discussed. However, there may also be additional trade-offs in network performance that need to be considered. Therefore, in addition, we investigate the performance gains of the TCP flow classification for a variety of scenarios with optical links of varied granularity. Our simulations indicate that data center throughput performance and the gains of flow classification are highly sensitive to the bandwidth aggregation level and the non-uniformity of network traffic.

The rest of this paper is organized as follows. In Section II, we provide an overview of machine learning-based traffic classification for accurate and rapid flow assignment in data center networks. To examine the impact of proper flow assignment in optical data centers, we introduce the data center control cycle in Section III. In Section IV, we detail our analysis framework and examine the impact of TCP flow classification on the hybrid data center performance. We especially look at the impact of aggregating the optical channel bandwidth from a TCP performance perspective. Finally, we summarize and conclude in Section V.

II. Adaptive Flow Classification in Data Center Networks

Due to the importance of elephant detection in data center networks, a variety of flow classification methods have been proposed [8,16,17,22]. These methods differ in terms of the location where classification takes place as well as the algorithm involved. From a location point of view, elephant flows can be detected either at the edge of the network or in its core. Helios [8] and Hedera [17] are examples of flow classification in the core of the network. In these proposals, all flows are monitored in network switches, and statistics are pulled from switches by the controller at regular intervals so that it can make decisions per an appropriate classification algorithm. This approach introduces significant signaling overhead and a substantial burden on the network controller, as the statistics per individual flow have to be delivered to the controller. Due to the limited bandwidth between the controller

and data center network switches as well as the limited resources of such switches (e.g., the limited number of flow table entries in OpenFlow switches), controller-based flow classification is not desirable for optically interconnected data centers. To cope with signaling overheads in controller-based classification schemes, sampling methods can be employed to decrease the load on the controller [22]. By sampling each flow and transferring a small portion of packets to the controller, elephants can be detected if the number of samples per flow exceeds a predefined threshold. In order to perform classification reliably, the sampling-based schemes need to accumulate enough samples, which may lead to unacceptable delays [16].

Unlike in-network traffic classification (i.e., classifying flows with the help of the network controller), classification at the edge of the data center network holds promise for low-overhead and speedy traffic classification. With in-network monitoring, flow behavior can be biased by network congestion, misleading the classifier, whereas flow classification at the edge can be more accurate due to decoupling the application behavior and network dynamics. Curtis *et al.* [16] propose end host-based elephant detection where classification is performed at the operating system (OS) level by monitoring end host socket buffers. When the socket buffer for a TCP flow exceeds a threshold, it is classified as an elephant. This requires a slight modification to the server OS for introducing buffer monitoring and elephant detection functions.

With the recent advances in fully programmable network interface cards (NICs), flow classification can now be performed on intelligent NICs that are equipped with hundreds of processing cores [23]. This would allow for more advanced edge-based classification algorithms without compromising the performance of the server operating system. Offloading classification procedures to intelligent NICs enables low overhead machine learning-based flow classification for improved speed and accuracy, in addition to adding adaptability to traffic dynamics. Machine learning is a form of computational intelligence that provides machines with the ability to learn and adapt without being explicitly programmed. Neural networks have existed as a form of machine learning since the late 1950s. However, they have only become useful for solving perceptual problems over the past decade. This has been due to algorithmic breakthroughs and the increase in applicable computational power from single instruction multiple thread (SIMT) graphics processing units (GPUs) and other processors. Programmable NICs have flexible multiple instruction, multiple data (MIMD) multi-core structures, and can host machine learning algorithms to solve a variety of networking problems [24–26]. In this section, we propose a basic machine learning-based traffic classifier that can be implemented on a programmable NIC for handling real data center traffic at very high speeds [27].

As shown in Fig. 3, the algorithm that we use for flow classification consists of three key parts: (1) the **Hash-Based Classifier**, which checks whether packets belong to a classified flow (this part runs at a high speed with low latency); (2) the **Feature Vector Storage**, which

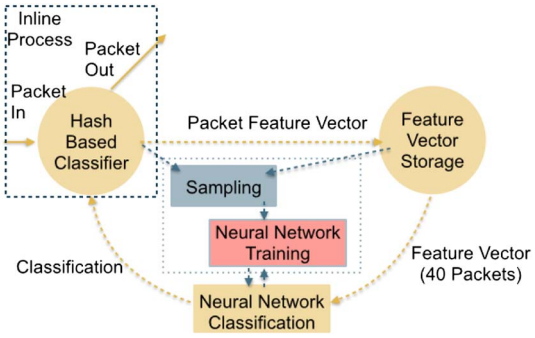

Fig. 3. High-level classification algorithm: The hash-based classifier is the only section that needs to operate at a line rate.

stores flow features using packets from unclassified flows (feature vectors are required to accumulate enough information to allow the machine learning algorithm to make a flow classification decision); and (3) the **Neural Network**, which classifies complete feature vectors. The feature vector is based upon previous work [28]. It includes the five-tuple (source IP address, destination IP address, source port, destination port, transport layer protocol), packet sizes, and a set of intra-flow timings within the first 40 packets of a flow (or roughly the first 30 TCP segments). This helps to improve the training speed and avoid the disappearing gradient problem when using gradient descent backpropagation [29–31].

The type of neural network used in our classifier is a fully connected multi-layer perceptron (MLP) with four hidden layers. MLPs are relatively simple to implement in high-dimensional situations without base knowledge of the intermediate features. MLPs have high levels of true negative classification, which is critical in order to ensure that mice do not flood the optical interconnect [32]. Due to the nature of mouse and elephant flow distribution in data centers (i.e., an overwhelming amount of mice compared to a much smaller number of elephants), there is a class imbalance problem. This can be overcome by training with a non-proportional amount of mouse and elephant flows. To ensure adaptability, we use an internal-self supervised teaching mechanism. We note that the accuracy, speed, and overheads of the neural network classifier would depend on the complexity of the machine learning algorithm, network flow processor hardware architecture, and implementation details. We have successfully assessed the performance of a baseline MLP-based traffic classification algorithm using real anonymized traffic from a university data center network. The results of this analysis are described in a separate publication [27]. We plan to perform more experiments for distinguishing trade-offs in classifier performance. In this paper, we model the performance of our classifier from the accuracy perspective and examine its impact on the efficiency of an optically interconnected data center.

From a network control perspective, we combine machine learning capabilities using edge intelligence with software-defined networking (SDN) to achieve network programmability and flexible resource allocation.

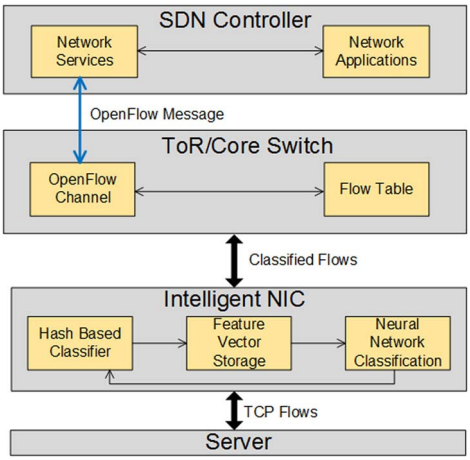

Fig. 4. Adaptive flow classification within SDN control framework.

Figure 4 provides a high-level diagram of network level interactions among various entities within the data center. Upon the start of a traffic flow in a server, data packets are inspected by the server's intelligent NIC, where the machine learning algorithm is implemented (including the hash-based classifier, feature vector storage, and neural network classification components). The intelligent NIC enables flow classification at the edge of the network. Once a flow is classified as an elephant, the SDN controller needs to trigger its elephant resource allocation routine for that flow at a proper scheduling instance. For this to become possible, the NIC manipulates the virtual LAN (VLAN) tag field of the newly detected elephant's packets (or any other unused field of the packet header that can be employed for OpenFlow matching) and sets it to a predefined value. Based on the VLAN tag content, the OpenFlow switches along the default path of the flow (top-of-rack (ToR)/core switch) perform matching in their flow tables and direct the first packet of the elephant flow they receive to the SDN controller. The controller identifies the new elephant flow using the VLAN tag information, performs resource allocation as appropriate, and reports its decision to the OpenFlow switches in terms of a set of routing rules to be installed as new flow entries in their flow tables. From this point on, the rest of the elephant flow's data packets are forwarded according to the updated flow tables.

III. Optical Data Center Architecture and Scheduling

Our goal is to study the impact of TCP flow classification on the performance of optical data center interconnects with varied bandwidth granularity. Optically assisted data centers are interesting in the sense that they enable network performance to be enhanced without resorting to expensive full-bisection bandwidth electrical interconnects. In an example optical data center network as depicted in Fig. 1, optical and electrical fabrics work synergistically to accommodate flows with different performance requirements. An under-provisioned electrical network provides all-to-all connectivity among computing nodes, enabling the transport of short-lived, delay-sensitive flows and control messages across the network. In addition, an optical circuit switching fabric is provisioned to enable point-to-point, high-bandwidth connectivity by accommodating long-lived bulk data transfers. Unlike the electronic switches that enable buffering and switching at nanosecond speed, high port count optical switches are usually based on MEMS technology and support reconfiguration speeds on the order of tens of milliseconds. Without loss of generality, in this paper, we assume that the electrical and optical networks are non-blocking and model each as a single switch. All server racks within the data center are connected to both electrical and optical networks with arbitrary numbers of electrical and optical links.

Scheduling traffic flows in a data center has a significant impact on the architecture performance. We consider the data center operation to be governed by control cycles [7–9]. Each control cycle involves (1) measuring current traffic demands, (2) estimating traffic for the newly started cycle, (3) calculating the optimal optical network topology, and (4) reconfiguring the network as required. Figure 5 depicts the scheduling tasks within a control cycle. The control cycle comprises a mandatory sequence of tasks and a secondary sequence should the optical circuit-switched network require reconfiguration. The control cycle should be long enough to compensate for scheduling and reconfiguration overheads.

A control cycle starts by measuring the number of elephant flows each rack has destined to other racks. Note that traffic classification becomes important here, as mistakenly perceived elephants can lead the scheduler to set up inefficient circuit paths, resulting in congestion in parts of the network and underutilization elsewhere. Once an elephant matrix has been constructed (with each entry denoting the number of elephants flowing between a pair of racks), the traffic estimation routine starts. A flow's current sending rate says little about its natural demand in an ideal non-blocking network [17]. Hence, the goal of traffic estimation is to make more intelligent flow assignment decisions by knowing the natural max–min fair bandwidth requirements of flows. TCP's AIMD dynamics try to achieve such a fairness. To calculate the share of bandwidth each rack can have to talk to other racks, the scheduler considers no host-imposed limitation (e.g., due to host

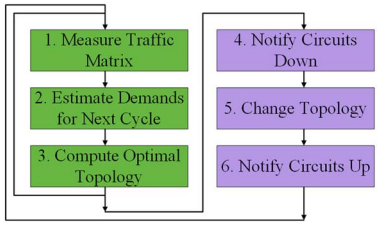

Fig. 5. Control cycle in an optically interconnected data center network.

disk access and processing) and only assumes flow rates get limited at the ToR level. The algorithm proposed in [17] can be used to estimate the traffic demands between rack pairs.

Based on the estimated traffic, the scheduler passes to new optical topology calculation. It greedily calculates a maximal matching between racks considering the traffic demands and the number of optical ports per rack. The greedy nature of the matching algorithm arises from the fact that in each iteration, circuit(s) will be set up between two nodes that have the highest estimated traffic demand. The output of the matching phase is a matrix that identifies the number of circuits (fibers) between each rack pair. Note that as data center traffic exhibits asymmetry, the circuit establishment is not symmetric. That is, if there are n circuits between rack i and rack j, it is not necessary to have the same number of circuits in the reverse direction. It is possible that some of the calculated circuits have already been established in a previous control cycle. There will not be a reconfiguration penalty for such circuits. While an optical circuit is being established, it will be inaccessible and cannot be used for data transfer. If a circuit has to be torn down, the flows using that circuit are migrated to the electrical portion of the network. When optical circuit establishment is complete, all flows that can use the circuit will try to exploit it for an enhanced share of bandwidth. In our study, we assume mice can share optical links with elephants due to potentially higher available bandwidths and thus faster completion times. During reconfiguration times, both mice and elephants can share the electrical network resources. Although the study of different resource-sharing policies in optically interconnected data centers is not within the scope of this paper, existing research suggests that sharing resources among different flow classes has the potential to improve the overall network performance [18].

Besides the abovementioned scheduling tasks that are carried out at the beginning of each control cycle, the scheduler performs some regular tasks during every time slot. These include handling new arrivals, classifying flows, and servicing all flows that exist within the data center. To handle arrivals, the scheduler considers the newly arrived flows during a time slot and assigns them to links. This entails finding the uplink (link from the source ToR to the core network) and the downlink (link from the core network to the destination ToR) with the maximum free capacity. Further, the scheduler considers all unclassified flows during any time slot. If an unclassified flow has sent segments beyond a predefined threshold (in our study, a flow can be classified if it has sent out at least 30 segments), the scheduler makes a decision on whether the flow should be treated as an elephant or not.

The service phase in each time slot involves detecting all links (electrical/optical) that reach the congestion point and performing the TCP Tahoe congestion control mechanism accordingly. When considering congestion for optical links, each wavelength is treated separately. A fiber link may still be far from its saturation point, but one or some wavelengths in such a fiber may have their capacity exhausted by TCP flows. After detecting congested links, the states of all affected flows will be updated, as explained in Section I. That is, they enter the slow-start phase with a congestion window size equal to the maximum segment size (MSS) and ssthresh equal to one half of congestion window size at the congestion point. Once congestions have been resolved, flows are shuffled (to implement fairness in service) and their desired number of segments (equal to the minimum of the number of segments in the send window and the number of outstanding segments of the flow) is transmitted across the network.

IV. Joint Impact of Flow Classification and Bandwidth Aggregation

In this section, we study the interplay of optical bandwidth granularity settings and TCP flow classification accuracy in an optically interconnected, hybrid data center network. We examine how the choice of optical channel bandwidth affects network throughput due to TCP dynamics. We also study the conditions necessary for high classification impact by tuning traffic characteristics and bandwidth granularity. We will see that depending on these two factors (i.e., traffic uniformity and optical channel bandwidth), flow classification can have a moderate to high impact on the overall data center network throughput.

For our study, we implemented a flow-level, discrete-event network simulator that can scale to thousands of servers. The network is modeled corresponding to Fig. 1, where all server racks connect to both electrical and optical switching fabrics. We assume an electrical network where hosts can perform non-blocking, all-to-all communications. The optical fabric in our simulator is modeled as a single MEMS switch, enabling fiber port to fiber port high-bandwidth connectivity.

We avoid performing packet-level simulations, as tracing network behavior at the packet level makes the analysis intractable for the several hundreds of thousands of flows that we consider. As a consequence of flow-level analysis, we cannot capture the impact of different packet sizes and buffer behaviors. Furthermore, we do not consider the bandwidth consumed by TCP ACKs, which is a small fraction of the data center bandwidth. Although these assumptions make our results relatively optimistic, we find such provisions necessary to study the flow classification impact at the scale of thousands of servers.

Our analysis models the behavior of a network with TCP traffic. The simulations proceed in a time-slotted fashion (discrete time ticks). Each time slot corresponds to a typical data center RTT of 100 μs [17]. During each time slot, the simulator performs several tasks, including accepting new flows into the system as well as updating flow rates based on their status and network congestion. We model the TCP AIMD principle and consider slow-start and congestion-avoidance phases when increasing flow rates. The slow-start threshold (ssthresh) is set to 64 kB. The TCP Tahoe congestion control mechanism is implemented in the "Service()" function denoted in Fig. 6. In this

```
1   for IT = 1 : ControlCycles
2     Calculate ElephantCount(N_r, N_r)
3     EstimatedTraffic ← Estimate(ElephantCount)
4     MatchingMatrix ← Match(EstimatedTraffic)
5     for T = 1 : MECT
6       time ← (IT − 1) × CC + T
7       Arrival(); Detect(); Service()
8     if MatchingMatrix implies changes
9       MigrateToElectrical(AffectedCircuits)
10      for T = MECT + 1 : MECT + (RT/RTT)
11        time ← (IT − 1) × CC + T
12        Arrival(); Detect(); Service()
13      FiberAssignment(NewCircuits)
14      MigrateToOptical()
15      for T = MECT + (RT/RTT) + 1 : CC
16        time ← (IT − 1) × CC + T
17        Arrival(); Detect(); Service()
18    else
19      for T = MECT + 1 : CC
20        time ← (IT − 1) × CC + T
21        Arrival(); Detect(); Service()
```

Fig. 6. High-level pseudocode of the data center simulator.

function, we make use of the mapping information that relates flows to links. In other words, for each individual link in the network (either electrical or optical wavelength link) during a new time slot (with duration equal to RTT), we project the required bandwidth based on the current window size/remaining segments of all flows that share it. If the aggregated demand across the link equals or exceeds its capacity, the link is treated as congested. Once all congested links have been detected, we update all flow rates per congested link and push them to the slow-start phase with a new ssthresh and window size equal to one segment. Finally, for each flow in the system, the desired number of segments is transmitted across the network. As congestions have been resolved before this step, no congestion will occur based on this bandwidth assignment.

Figure 6 illustrates the high-level structure of our data center simulator (with N_r racks), where *ControlCycles* denotes the number of scheduling cycles that the network performance is simulated. *CC* is the number of time slots (i.e., RTTs) within a cycle. *MECT* is the number of time slots required for traffic measurement, estimation, and running a maximal matching algorithm. *RT* denotes the number of time slots for reconfiguring the optical switch hardware. The simulator accepts a traffic file as its input that includes the list of flows that arrive at the network during the course of the simulation. Each flow is characterized by source and destination racks, arrival time, and size. We are inspired by empirical studies on data center traffic patterns to populate the input traffic file [13]. We consider a Poisson flow arrival process. Each server in our simulations is assumed to generate, on average, 20 new flows per second (with 1 Gbps network interface capacity) [16]. Based on this, during each time slot, we examine each server to determine if a new flow has been generated. We assume 80% of flows remain within the rack where they are generated. Our traffic generation mechanism ensures that a flow's source is uniformly picked from the set of existing racks. As per destination, we model hot-spot communications in our data center. We assume that a flow is routed to a group of hot-spot racks (a small fraction of data center racks) with a probability of 0.9 and uniformly to any other rack with a probability of 0.1. If a flow is destined to the hot-spot group, its destination rack will be uniformly picked from the set of hot-spot racks. We denote the number of hot-spot racks as hot-spot size (HSS) and examine HSS = 4 and HSS = 8. For flow size distribution, we consider a rounded Pareto distribution. The flow size in bytes is calculated as

$$L = \left\lfloor \frac{x_m}{U^{1/\alpha}} \right\rfloor, \tag{1}$$

where U is a random variable uniformly distributed on (0,1), and $\lfloor\ \rfloor$ represents the floor function. x_m is the scale factor and denotes the minimum flow size, and α is the tail index. Based on [13], we consider $x_m = 100$ B and $\alpha = 1/3$, which leads to significant variability in flow size (infinite mean and variance). Once the flow size is determined, we divide L by the TCP maximum segment size (MSS = 1500 B) to determine the number of segments a flow contains. Our simulator considers MSS as the data unit.

We use two conditional probabilities to model the classification behavior. $P_{e|e}$ is the probability that a flow is classified as an elephant given that it is actually an elephant flow (i.e., true positive rate or elephant detection rate), and $P_{m|m}$ is the probability that a flow is classified as a mouse given that it is actually a mouse flow (i.e., true negative rate or mouse detection rate). When a flow is to be classified, we know its actual type due to its size in the input traffic file (again, we consider flows smaller than 100 MB as mice and as elephants otherwise). We generate a random number u uniformly over (0,1). If the flow is a mouse and $u \le P_{m|m}$, then it will be classified correctly as a mouse. If $u \ge P_{m|m}$, it will be misclassified as an elephant. If the flow is an elephant and $u \le P_{e|e}$, then it will be classified correctly as an elephant. If $u \ge P_{e|e}$, it will be misclassified as a mouse. Once the flow is classified (as either mouse or elephant), it will be migrated to the optical network should there be an optical circuit to its destination already in place.

In our simulations, we consider a data center with 32 racks of 48 servers (1536 hosts in total). We simulate the data center network for 60 control cycles. Each control cycle comprises 10,000 time slots. Ten percent of the control cycle duration is associated with circuit scheduling and reconfiguration overheads (100 ms). The reconfiguration overhead is an important parameter that affects the utilization of optical circuits. In general, various parameters affect the overhead duration, including hardware reconfiguration time, controller computational power, complexity of resource allocation algorithms, and the number of flows to be handled. Previous work points to scheduling intervals on the order of 100 ms or less for scheduling elephant flows [8,17]. Here we consider 100 ms as a typical overhead value, which comprises 75 ms computation and 25 ms hardware reconfiguration periods.

Each ToR switch in our analysis is equipped with 10 bidirectional electrical links and a bidirectional optical link with either 1 or 4 wavelengths. With a 10 Gbps network interface capacity per server, it is reasonable to consider

electrical link bandwidths of 10 Gbps and optical channel capacities of 25 Gbps or beyond. Simulating TCP performance in a data center network with such capacities would be cumbersome due to the excessively large number of traffic flows. A server of 1 Gbps capacity is reported to generate 20 new flows per second [16]. A server with ten times more capacity can generate a significantly larger number of flows. As in [16], we scale down link capacities and traffic demands simultaneously to make the simulations tractable. To scale down traffic, we have two options. One is to reduce the number of flows and the other is to scale down their size. Since we would like the ratio of traffic demand to network capacity to be preserved in our simulations, we only manipulate one dimension (i.e., the flow arrival rate). By preserving the size of the flows, we ensure that an individual flow's intrinsic demand is not affected in the scaling process.

We emphasize that TCP dynamics (as depicted in Fig. 2 for one flow) depend on several factors, including link bandwidth, number of flows sharing a link, their arrival times and sizes, and the routing mechanism across the network. By scaling down the network parameters, we may not achieve the exact performance (multiplied by a scaling factor), but our studies have shown that the trends we discuss in this paper are independent of network scaling. Hence, we consider electrical links with 1 Gbps capacity and optical links with 4 channels of 2.5 Gbps capacity (or a single channel of 10 Gbps capacity) to represent a scaled-down version of a real data center network (with 10 Gbps electrical links and 25 Gbps or possibly 100 Gbps optical wavelengths). The statistics reported in this paper are collected from the latter 40 control cycles of each simulation. Furthermore, each data point corresponds to the average of five simulation runs.

Based on our machine learning experiments [27], we set $P_{m|m}$ to 0.95 and vary $P_{e|e}$ (i.e., the elephant detection rate) between 0.5 and 0.95. Figures 7 and 8 depict the network throughput versus the elephant detection rate (EDR) for hot-spot sizes of 4 and 8 server racks, respectively. With the bandwidth settings in our examination and the mouse detection rate fixed at 0.95, varying the elephant flow classification accuracy does not exhibit remarkable

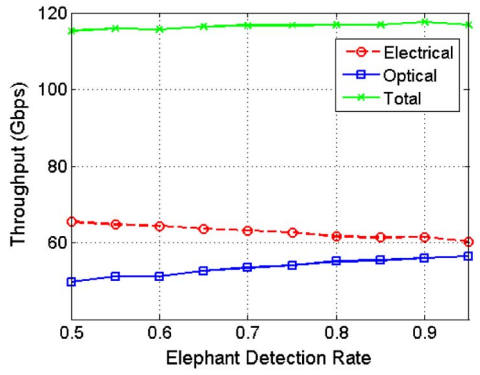

Fig. 7. Throughput versus elephant detection rate for HSS = 4 and four wavelengths per fiber (with the mouse detection rate fixed at 0.95).

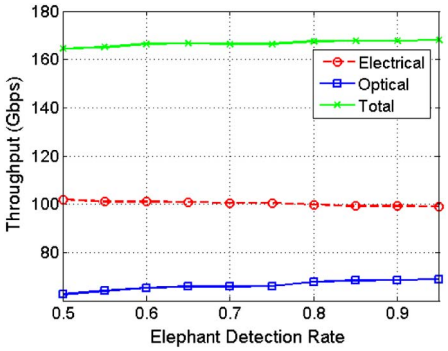

Fig. 8. Throughput versus elephant detection rate for HSS = 8 and four wavelengths per fiber (with mouse detection rate fixed at 0.95).

changes in throughput. *However, we will soon notice that performance gains could be significant compared to a random classification policy when the optical bandwidth becomes more abundant.* According to Figs. 7 and 8, the optical network throughput sees a 13.5% increase for HSS = 4 and 10.1% for HSS = 8 over the range of EDR values. This improvement can be attributed to a more efficient optical circuit setup due to better identifying long-lived flows. Furthermore, there exists a trade-off in the sense that an increase in the utilization of optical links results in a lower utilization of the electrical network.

Compared to HSS = 4, a more uniform traffic distribution with HSS = 8 allows for more connections to be set up simultaneously. The throughput increases significantly with doubling the number of hot-spot racks. The improvement is mostly observed in the electrical section of the network, which favors all-to-all connectivity. With an increase in HSS, the electrical throughput (averaged over elephant detection rates) is increased by 59.5%. The improvement in optical network throughput is 23.8%, and the overall data center network sees an improvement of 43.1%.

As we aim at studying traffic classification in conjunction with optical bandwidth settings in the network, we examine the impact of flow classification accuracy when enhancing the optical channel bandwidth in hybrid data centers. Fatter optical pipes (such as a consolidated 100 Gbps wavelength instead of four parallel 25 Gbps links) are more expensive, but it is interesting to note the performance with respect to TCP dynamics. Please note that our analysis assumes interconnectivity beyond the rack level and hence does not consider the inefficiencies associated with link aggregation protocols within the ToR switch. The downlinks of a ToR switch are assumed to operate at 10 Gbps (connected to servers with 10 Gbps interfaces), whereas the optical uplinks can operate at 25 Gbps (baseline case) or 100 Gbps (aggregated case). In either case, some electronic aggregation and framing should be performed. By optical aggregation, we do not consider an intermediate step going from 4×25 to 100 Gbps. Instead, we are interested in the possibility of employing advanced modulation formats in data centers to achieve higher capacity per channel. Current optical links in data centers carry binary modulated data. However, recent research

points to the feasibility of higher-order modulation formats suitable for short-reach applications [33,34]. With proper bandwidth settings, modulation schemes, processing, and drop in price points, it is feasible to achieve 100 Gbps capacity per channel in optically interconnected data centers.

With high-capacity wavelengths, it is feasible for flows to achieve higher transmission rates, as links will get congested less frequently. Figure 9 depicts the data center throughput for HSS = 4 (focusing on the optical and electrical sections of the network), comparing two scenarios. In one case, each fiber link carries four channels of 2.5 Gbps capacity (scaled-down version of 4 × 25 Gbps). In the other case, a fiber is assumed to include one wavelength channel only with a capacity of 10 Gbps (scaled down version of 1 × 100 Gbps). Our simulations point to the significant improvement of network performance with optical bandwidth consolidation. The majority of improvement in throughput comes from the optical section of the network that undergoes bandwidth aggregation. Optical bandwidth aggregation increases the average optical network throughput from 53.5 to 135.6 Gbps, corresponding to a 153.2% increase. The minimum and maximum improvements achieved in the overall network throughput due to optical channel bandwidth aggregation are equal to 57.9% and 74.5%, respectively.

Apart from performance gains, the impact of elephant detection accuracy on the optical network throughput is stronger for the case of aggregated optical bandwidth. With one wavelength of 10 Gbps per fiber, the throughput is equal to 121.7 Gbps for EDR = 0.5 and 148.9 Gbps for EDR = 0.95. This translates to a 22.4% improvement as compared to a 13.5% improvement using 4 × 2.5 Gbps links. According to Fig. 9, the optical fabric plays a significant role in the data center network with bandwidth aggregation. As a result, missing out on the low-congestion optical circuits due to classification inaccuracy and imperfect matching could incur a stronger penalty compared to the case where optical and electrical networks exhibit similar throughputs.

To quantify the combined advantages of adaptive flow classification (considering mouse detection rate in addition to elephant flow detection rate), we define the *throughput improvement factor* as the relative increase in the network throughput with adaptive, machine-learning-based flow classification (as described in Section II) compared to *random* flow classification. In adaptive flow classification, a flow can be classified using two conditional probabilities (0.95 mouse detection rate and a variable elephant detection rate) once a certain number of TCP segments (30 in our examination) have been examined by the network interface card. In random classification for flows that are longer than the 30 MSS threshold, a flow is classified with a probability corresponding to the proportion of mice in the data center traffic flow set. We consider this probability to be 0.9 (typical ratio of mouse flow count to total flow count based on data center measurements). If a flow has to be classified, it will be regarded as a mouse with a probability of 0.9.

Figure 10 depicts the improvement that can be achieved due to adaptive classification for HSS = 4. With 4 × 2.5 Gbps wavelengths per fiber, an average improvement equal to 8.6% is achieved. However, the impact of classification is much more significant with bandwidth aggregation. With 1 × 10 Gbps per fiber, the improvement is 40.3% for EDR = 0.5 and 54.7% for EDR = 0.95 (average improvement: 47.7%). This example shows that flow classification is highly sensitive to the bandwidth settings within the data center.

We also study the impact of optical bandwidth aggregation for HSS = 8. Figure 11 depicts the optical and electrical network throughput versus EDR for 4 × 2.5 and 1 × 10 Gbps optical link configurations. As for the case of HSS = 4, the majority of improvement in throughput comes from the optical network. With bandwidth aggregation, a maximum overall network throughput of 245.2 Gbps could be achieved. Comparing the cases of one and four wavelengths per fiber, the minimum and maximum network throughput improvements are equal to 37.8% and 46%, respectively.

Finally, Fig. 12 depicts the improvement that can be achieved due to adaptive classification as compared to random classification for HSS = 8. With 4 × 2.5 Gbps links, an average improvement equal to 7.8% is achieved. With bandwidth aggregation, the improvement is much higher and an average improvement of 35.3% could be achieved. Comparing hot-spot sizes of 4 and 8, we observe that improvements due to bandwidth aggregation are less significant for a larger HSS value. In other words, more uniformly

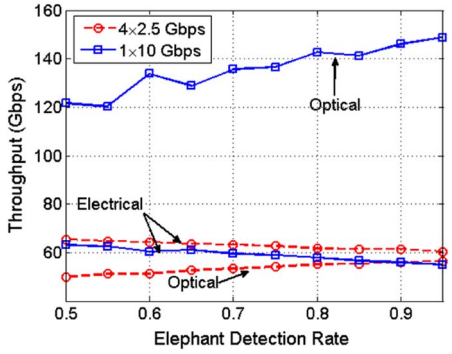

Fig. 9. Impact of bandwidth aggregation on electrical and optical network throughput for HSS = 4.

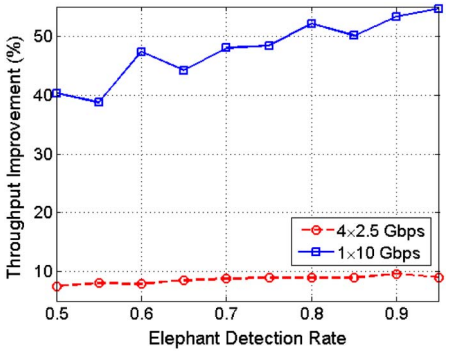

Fig. 10. Throughput improvement percentage due to classification accuracy for HSS = 4.

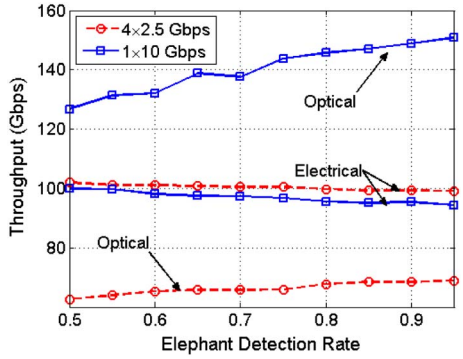

Fig. 11. Impact of bandwidth aggregation on electrical and optical network throughput for HSS = 8.

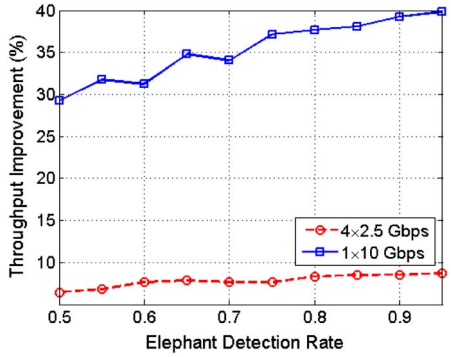

Fig. 12. Throughput improvement percentage due to classification accuracy for HSS = 8.

distributed traffic makes the performance less sensitive to the faulty detection of elephants and optical circuits become less crucial. In the latter case, more optical circuits are established, although with a lower utilization, and the traffic has more opportunities to be delivered. Hence, missing out on one specific optical circuit will have a less significant impact compared to missing out on one optical circuit with HSS = 4. It would be interesting to quantify this trend by varying HSS over a range of feasible values.

V. Conclusion

Optical switching is attractive for the emerging massive-scale data centers because of its bandwidth, power, and footprint advantages over electronics. A challenge is how to integrate such a technology into traditional, electronically switched architectures and optimize the overall network efficiency and performance. The optical fabric is best suited for stable and long-lived elephant flows. The accuracy of the correct detection of such flows and the amount of bandwidth allocated to them upon detection are two critical questions that we addressed in this paper.

Our simulations, modeling the TCP dynamics arising from the AIMD mechanism for congestion control, pointed to the crucial impact of optical bandwidth aggregation (i.e., the consolidation of several low-capacity channels into a single high-capacity one) on performance. The network throughput could be increased by as large as 74.5% when four wavelength channels within a fiber were aggregated into a single channel of four times more capacity. We also noticed that flow classification is more significant in certain scenarios. The role of flow classification accuracy becomes significant with higher bandwidth aggregation due to the greater penalty of missing out on the valuable high-capacity, low-congestion resources. Furthermore, traffic patterns exhibiting higher non-uniformity benefit more from accurate classification, as in such scenarios, scarce circuits play a critical role in data transfers. Compared to a random classification benchmark, adaptive flow classification could lead to throughput improvements as large as 54.7%.

While our analysis focused on a specific optical data center architecture and schedule, our methodology can be applied to many different scenarios, and we plan to extend our analysis to other optically interconnected data center designs. From a modeling point of view, it will be helpful to envision mechanisms for accommodating buffer behavior and actual link bandwidths without sacrificing scalability. Future work should also investigate the trade-offs between latency, power consumption, and accuracy of the machine learning-based flow classification. Furthormore, an interesting extension of this work could be the study of more advanced predictive analytics, where instead of binary flow classification, a richer set of information, including estimated flow sizes and the dependencies of flows within applications, is passed on to the data center network scheduler.

Acknowledgment

This work was supported by the National Science Foundation (NSF) Center for Integrated Access Networks (CIAN) under grant no. EEC-0812072.

References

[1] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 1021–1036, 2012.

[2] C. Kachris, K. Kanonakis, and I. Tomkos, "Optical interconnection networks in data centers: Recent trends and future challenges," *IEEE Commun. Mag.*, vol. 51, no. 9, pp. 39–45, 2013.

[3] B. R. Rahimzadeh, G. Zervas, Y. Yan, and D. Simeonidou, "Griffin: Programmable optical datacenter with SDN enabled function planning and virtualisation," *J. Lightwave Technol.*, vol. 33, no. 24, pp. 5164–5177, 2015.

[4] H. Liu, M. K. Mukerjee, C. Li, N. Feltman, G. Papen, S. Savage, S. Seshan, G. M. Voelker, D. G. Andersen, M. Kaminsky, G. Porter, and A. C. Snoeren, "Scheduling techniques for hybrid circuit/packet networks," in *11th Int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT)*, Dec. 2015, pp. 339–350.

[5] P. Samadi, V. Gupta, J. Xu, H. Wang, G. Zussman, and K. Bergman, "Optical multicast system for data center networks," *Opt. Express*, vol. 23, no. 17, pp. 22162–22180, 2015.

[6] Z. Cao, R. Proietti, M. Clements, and S. J. B. Yoo, "Experimental demonstration of dynamic flexible bandwidth optical data center network with all-to-all interconnectivity," in *European Conf. on Optical Communication (ECOC)*, Sept. 2014, pp. 1–3.

[7] G. Wang, D. G. Andersen, M. Kaminsky, M. Kozuch, T. S. E. Ng, K. Papagiannaki, M. Glick, and L. Mummert, "Your data center is a router: The case for reconfigurable optical circuit switched paths," in *Proc. ACM HotNets-VIII*, Oct. 2009.

[8] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. ACM SIGCOMM*, Aug. 2010, pp. 339–350.

[9] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. S. E. Ng, M. Kozuch, and M. Ryan, "c-Through: Part-time optics in data centers," in *Proc. ACM SIGCOMM*, Aug. 2010, pp. 327–338.

[10] G. Porter, R. Strong, N. Farrington, A. Forencich, C.-S. Pang, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating microsecond circuit switching into the data center," in *Proc. ACM SIGCOMM*, Aug. 2013, pp. 447–458.

[11] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G. M. Voelker, G. Papen, A. C. Snoeren, and G. Porter, "Circuit switching under the radar with REACTOR," in *Proc. ACM/USENIX NSDI*, Apr. 2014, pp. 1–15.

[12] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *10th ACM SIGCOMM Conf. on Internet Measurement*, Nov. 2010, pp. 267–280.

[13] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM*, Aug. 2009, vol. 39, pp. 51–62.

[14] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of datacenter traffic: Measurements & analysis," in *9th ACM SIGCOMM Internet Measurement Conf. (IMC)*, Nov. 2009, pp. 202–208.

[15] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *Proc. ACM SIGCOMM*, Aug. 2015, pp. 123–137.

[16] A. R. Curtis, W. Kim, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *IEEE INFOCOM*, Apr. 2011, pp. 1629–1637.

[17] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. ACM/USENIX NSDI*, Apr. 2010, paper 19.

[18] K. Kanonakis, Y. Yin, P. Ji, and T. Wang, "SDN-controlled routing of elephants and mice over a hybrid optical/electrical DCN testbed," in *Optical Fiber Communication Conf. (OFC)*, Mar. 2015, paper Th4G.7.

[19] J. Gao and N. S. V. Rao, "TCP AIMD dynamics over Internet connections," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 4–6, 2005.

[20] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," IETF RFC 5681, Sept. 2009 [Online]. Available: https://tools.ietf.org/html/rfc5681.

[21] H. H. Bazzaz, M. Tewari, G. Wang, G. Porter, T. S. E. Ng, D. G. Andersen, M. Kaminsky, M. A. Kozuch, and A. Vahdat, "Switching the optical divide: Fundamental challenges for hybrid electrical/optical datacenter networks," in *2nd ACM Symp. on Cloud Computing (SOCC)*, Oct. 2011, paper 30.

[22] S. Shirali-Shahreza and Y. Ganjali, "Empowering software defined network controller with packet-level information," in *IEEE Int. Conf. on Communications Workshops (ICC)*, June 2013, pp. 1335–1339.

[23] Netronome, "Silicon solutions" [Online]. Available: https://netronome.com/products/silicon-solutions/overview/.

[24] T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surv. Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[25] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *IEEE Conf. on Local Computer Networks 30th Anniversary (LCN)*, Nov. 2005, pp. 250–257.

[26] A. Sivaraman, K. Winstein, P. Thaker, and H. Balakrishnan, "An experimental study of the learnability of congestion control," *Comput. Commun. Rev.*, vol. 44, no. 4, pp. 479–490, 2014.

[27] N. Viljoen, H. Rastegarfar, M. Yang, J. Wissinger, and M. Glick, "Machine learning based adaptive flow classification for optically interconnected data centers," in *18th Int. Conf. on Transparent Optical Networks*, July 2016, paper Mo.C3.4.

[28] G. J. Stark, N. J. Viljoen, and N. Viljoen, "Inter-packet interval prediction learning algorithm," U.S. patent US9042252 B2 (May 26, 2015).

[29] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Int. Conf. on Artificial Intelligence and Statistics*, May 2010, pp. 249–256.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167, 2015.

[31] L. V. der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[32] J. Wang, J. You, Q. Li, and Y. Xu, "Extract minimum positive and maximum negative features for imbalanced binary classification," *Pattern Recogn.*, vol. 45, no. 3, pp. 1136–1145, 2012.

[33] K. Szczerba, P. Westbergh, M. Karlsson, P. A. Andrekson, and A. Larsson, "70 Gbps 4-PAM and 56 Gbps 8-PAM using an 850 nm VCSEL," *J. Lightwave Technol.*, vol. 33, no. 7, pp. 1395–1401, 2015.

[34] P. Ji, D. Qian, K. Kanonakis, C. Kachris, and I. Tomkos, "Design and evaluation of a flexible-bandwidth OFDM-based intra-data center interconnect," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, 3700310, 2013.