Optical Switching and Networking 45 (2022) 100683

Contents lists available at ScienceDirect

Optical Switching and Networking

journal homepage: www.elsevier.com/locate/osn

DROAD: Demand-aware reconfigurable optically-switched agile data center network

Bartlomiej Siniarski [a,*], Dinh Danh Le [c], Conor McArdle [a], John Murphy [b], Liam Barry [a]

[a] School of Electronic Engineering, Dublin City University, Dublin, 9, Co. Dublin, Ireland
[b] School of Computer Science, University College Dublin, Belfield, 4, Co. Dublin, Ireland
[c] Ericsson, Athlone, Ireland

ARTICLE INFO

Keywords:
Data center networks
Arrayed waveguide grating
Optical interconnection

ABSTRACT

We present a Demand-aware Reconfigurable Data Center Network architecture design (DROAD) with integrated fast-switching optics and space switches that allows dynamic reconfiguration and separation of intra- and inter-cluster connections. The performance analysis results show a 64% improvement in average Flow Completion Time and a significant reduction in TCP session time, as well as a reduced number of sessions needed to be opened compared to traditional electrically-switched leaf-spine networks.

1. Introduction

The problem of scaling DCNs by increasing the number of switches to meet the traffic demand is expected to get worse due to emerging cloud workloads. Networks continue to experience high levels of congestion, even those that are built with high capacity in their switching fabrics [1]. This is due to inherent burstiness of flows that leads to inadmissible traffic in short time intervals, limited buffering capabilities, over-subscribing of the network and imperfect flow balancing. Therefore, networks need to match processing hardware speeds, providing sufficient bandwidth and extremely low latency. Historically, DCNs relied on packet-switched networks to connect their servers however, as scale and demand increased, the cost to build and manage these packet-switched networks is becoming too large. As a result of this change, new reconfigurable network topologies are gaining more attention from researchers and large cloud providers. Traditional ESNs continue to provide reasonable means to scale throughput with a large number of non-blocking switches that are directly connected to servers. However, the free scaling of electrical switches is expected to taper-off due to the slowdown of Moore's law [2,3]. Optically-switched architectures have in common that they reduce the static network provisioning requirements, thereby reducing the network's cost by presenting a means for bandwidth between hosts be updated periodically. These architectures reduce cost and complexity via scheduling methods, which can more dynamically manage bandwidth on optical paths in the data center. The integration of nanosecond optical switching devices in modern DCNs is therefore inevitable and we acknowledge a number of excellent network designs that were proven to work in production. **Helios** [4] was an early hybrid system using WDM for bursty low-latency traffic that delivered performance comparable to a non-blocking electrical switch with significantly less cost, energy, and complexity. Helios implements its traffic estimation and traffic demultiplexing features as part of the switch architecture. This approach makes traffic control transparent to end-hosts, but it requires modifications on all switches. **Solstice** [5] exploits sparse traffic patterns in DCNs to achieve fast scheduling of reconfigurable networks. Solstice takes advantage of sparsity and skewness observed in real datacenter traffic to provide x2.9 higher circuit utilization when compared to traditional schedulers in hybrid environments, while being within 14% of optimal, at scale. While, Solstice is shown to be effective among the preemptive algorithms, the number of reconfigurations required is still a major cause of inefficiency. **SIRIUS** [6] is an all-optical design that uses a single layer of optical gratings instead of multiple layers of electrical switches. The main motivation of SIRIUS design is to enable fast reconfiguration to support the bursty nature and high fan-out of emerging cloud workloads that require <10ns switching. Project Sirius is a demand-oblivious design that can perform end-to-end reconfigurations in less than 4 nano-seconds at 50 Gbps. **AgileDCN** [7] is another optical network design, which meets the requirements of high capacity and low latency networks. It is based on fast tunable lasers and AWGs for routing of intra-cluster traffic, with inter-cluster traffic being accommodated by optical space switches. In this paper we take the main design elements of AgileDCN and develop it further to evaluate the performance against other optically- and

* Corresponding author.
E-mail address: bartlomiej.siniarski@ucd.ie (B. Siniarski).

https://doi.org/10.1016/j.osn.2022.100683
Received 22 November 2021; Received in revised form 21 April 2022; Accepted 16 May 2022
Available online 26 May 2022

Acronyms

AWG	Arrayed Waveguide Grating
DCN	Data Center Networks
ESN	Electrically-switched network
FCT	Flow Completion Time
SDN	Software-Defined Network
ToR	Top of the Rack
VoQ	Virtual Queue
WDM	Wavelength-division multiplexing

electronically-switched DCNs. It is important to note that unlike electronic switches, optical switches do not have the ability to perform packet inspection or other intelligent processing functions, however this problem has been mitigated by the introduction of a SDN paradigm that works by separating data and control plane. **RODCA** [8] is an architecture that employs AWG routers and optical switches to deliver a flexible intra-cluster optical network, which is a Clos multi-stage network. The main goal of RODCA is to propose a network that adapts to traffic dynamics. The backbone of RODCA is a hierarchical optical DCN topology — several ToRs are interconnected through an AWGR to form a cluster, and several clusters can be interconnected through a higher-level AWGR. The design for the intra-cluster network includes a reconfigurable switching network that can be reconfigured at relatively coarse time scales, so that racks with mutually large traffic can be located within the same cluster, and gain bandwidth performance. Another architecture presented in Ref. [9] features fast optical switches in a single-hop topology with a centralized, software-defined optical control plane. The single-stage core topology is designed to be easily scaled up and scaled out without requiring major re-cabling and network reconfiguration. The use of OBS with two-way reservation allows the network to achieve the zero burst loss. The architecture has two layers namely the edge and core. The edge contains the electronic ToR, while the core comprises a group of fast optical switches.

To summarize, there is a wide range of data center specific technologies and scheduling ideas that enable efficient circuit switching in data center networks, with newer developments focusing on leveraging the benefits of faster optical circuit reconfiguration. In contrast, there has also been some recent work that discusses the idea of robust topology engineering e.g. reconfiguring circuits only every few minutes or even days. Notwithstanding, scaling current system designs can be problematic, in particular, due to the speed of the control plane and fan-out restrictions. Whereas one solution for the latter is free-space optics, those still face significant practical deployment issues in data center contexts. On the other hand, demand-oblivious system designs inherently overcome such control plane delays, but cannot adapt well to skewed demands. In their current form, they are not available as off-the-shelf hardware. Designing scalable, agile and demand-aware reconfigurable data centers is hence one of the main next challenges for future research in this area.

2. Design

2.1. Overall architecture

The proposed architecture (as shown in Fig. 1), provides two optically-switched, configurable data planes. In each data plane, fast optical switching using AWG routers are combined with slower optical space switches that are reconfigured periodically to re-optimise the overall network topology in response to slow changes in bulk traffic flows. Per-packet optical switching is facilitated by the AWG routers and their associated optical multiplexers/demultiplexers. To support reconfigurable topology, the data plane leverages two large-scale space switches: Space Switch 1 (SS1) - placed between ToRs and intra-cluster AWGs, and Space Switch 2 (SS2) - placed between ToRs and the inter-cluster network. The purpose of the two space switches is to periodically re-group the ToRs into different clusters, whenever the relative traffic volumes between the clusters exceed predefined thresholds. The detailed discussion on cluster reconfiguration mechanism is included in Section 2.3.

The design of DROAD is related to some concepts presented in Refs. [7–9], in particular due to similar arrangements of AWG switches in the network. Nevertheless, we identify several significant differences between DROAD and previously proposed architectures:

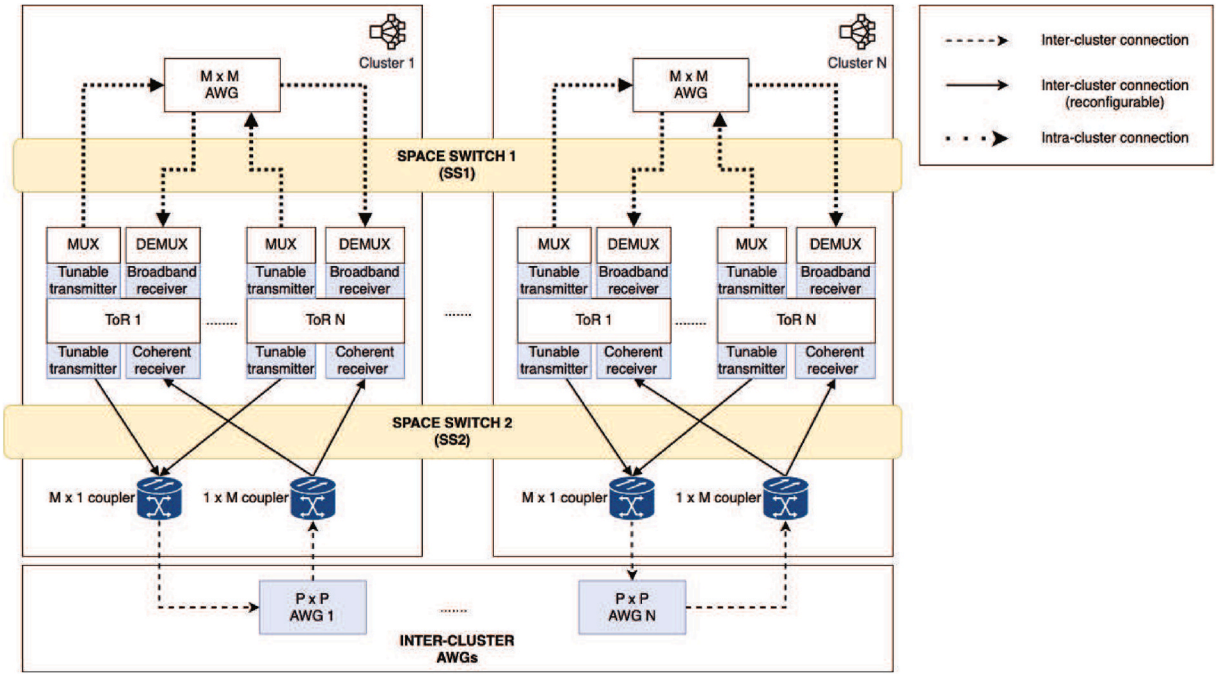

Fig. 1. DROAD network architecture.

1. The proposed architecture uses coherent receivers for inter-cluster channels at the ToRs, so that ToR to ToR traffic can be directly routed on a single optical hop, reducing switching complexity and latency. In contrast most architectures use non-coherent optical receivers, which requires that inter-cluster traffic needs two hop routing.
2. The proposed architecture uses larger transmission units, with a single common control packet to gain a multiplexing efficiency unlike that proposed in Ref. [8], where a time-slotted system is used with a single packet transmitted in one slot.
3. The proposed architecture is employing two independent space switches, which means that all connections between ToRs are high-bandwidth, optically transparent and facilitated by low-power AWG-based circuit routing. In contrast, the design of previously proposed architectures imposed electronic buffering in the interconnection network, which added potential packet latency bottlenecks and further degraded network power consumption.

The common element of DROAD and other architectures presented above is that the control plane consists of a central controller which connects to each of the ToR switches and it is responsible for managing (routing, wavelength assignment, traffic scheduling and switch configuration) of both intra- and inter-cluster traffic. The data plane performs data forwarding using pre-established connections configured by the controller. This is a well known design choice that is proven to work well in large-scale DCNs.

2.2. Intra- and inter-cluster connections

The envisaged architecture implements intra-cluster connections that provide direct electronically-switched connections between the servers in the rack and inter-cluster connections build using transceivers using one tunable transmitter and one coherent receiver. The reason for utilizing coherent receivers for inter-cluster communications is that the coherent receiver may filter out data transmitted to a particular ToR that is being sent to the specific ToR. The separation of intra- and inter-cluster connections allows optimal reconfiguration to meet diverse DCN traffic patterns. In particular we usually observe two main patterns in modern DCNs. In the first case the inter-cluster traffic dominates the whole DC and hence we can see that the communication degree of each ToR is bounded and hot ToRs exchange much of their data with only a few other ToRs. In the second case, the hot-spot traffic is the major traffic pattern, where hot ToRs communicate with most ToRs in the DC following a “fan-in/fanout” pattern while the “cold traffic” pattern is only popular among cold ToRs.

In the proposed architecture, the lower data plane carries intra-cluster traffic and consists of SS1 and a set of MxM AWG routers. Each AWG provides optically-switched connections between the ToRs grouped into the same cluster. The upper data plane carries inter-cluster traffic and consists of SS2 and sets of optical couplers/splitters and AWG routers, that together provide optically-switched connectivity between ToRs residing in different clusters. In case of the intra-cluster connections, each ToR switch is located at the top of the rack and provides direct electronically-switched connections between servers in that rack. Those ToRs form a cluster, where ToRs are interconnected by AWG router. In the envisaged architecture it is possible to configure the grouping of clusters by changing the configuration of SS1. For example, ToRs that communicate heavily with each other, can be connected into the same cluster and then separated in case of a minimal communication. To achieve the functionality of cluster formation, ToR switch design had to be redesigned. The modified ToR design is depicted in Fig. 2 and shows all input and output connections.

Each ToR has a number of tunable optical transmitters (TXs) for intra-cluster connections. The outputs of these TXs are combined into a single fiber that connects to one input of SS1, which has a dedicated path connected to a single input on one of the AWGs. Similarly, an output port

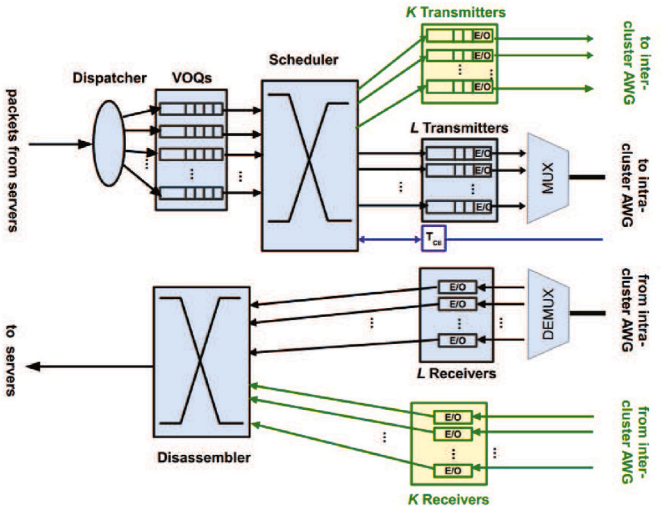

Fig. 2. ToR switch design.

of the AWG connects on a single fiber via SS1 back to the ToR. A ToR can simultaneously transmit to many ToRs in its cluster as each ToR is equped with an optical wavelength demultiplexer.

Inter-cluster connections are established using links to AWGs via SS2. Specifically, each ToR has a number of inter-cluster optical transceivers. Unlike intra-cluster transceivers, each inter-cluster transceiver includes one tunable transmitter and one coherent receiver. The reason for using coherent receivers for inter-cluster connections, instead of non-coherent receivers, is that the coherent receiver can filter out the data on the required wavelength that is being sent to a specific ToR; whereas that signal is simply discarded at the other ToRs that are not supposed to receive any data from that WDM signal. This feature makes it possible for the inter-cluster connections to benefit from one-hop connections via inter-cluster AWGs, just like intra-cluster connections. Each ToR has a pool of VoQs, one queue for each ToR in the DCN to which it will transmit data. The dispatcher module uses the IP address of the destination ToR to direct each packet to the correct VoQ. Then, the scheduler module uses the time-slot, size and output channel to control the transmitter appropriately. When bursts are received at the destination ToR, the packets are extracted by the disassembler module and routed to the correct server within the rack.

2.3. Cluster reconfiguration

Cluster reconfiguration is dependent on traffic variations being sampled. Periodically (every 10 ms in our experiments), every ToR sends its VoQ sizes to the controller to build a short-term traffic matrix. The controller decides to re-cluster the network when the ratio between the total bytes waiting for intra-cluster transmission (L_{inter}) to the total bytes waiting for intra-cluster transmission (L_{intra}) exceeds a threshold. The calculation method presented in Ref. [8] is used in our work i.e. $\beta \times L_{intra} \leqslant L_{inter}$, where a threshold parameter (β) is incorporated to make the reconfiguration decision. The value of β needs to be kept relatively low (0–10) as higher values of β will affect the latency in the network. When the condition for re-clustering is met, the groups of ToRs are collected (ToRs that have a larger mutual number of bytes waiting at the VoQs) and the new clusters are formed. As soon as reconfiguration is done, the controller re-initiates the network state and data transmission in the data plane.

2.4. Routing and scheduling

Given a data burst requesting a transmission between a source ToR and a destination ToR, the first step is to select a transmitter (TX) at the

source ToR and a receiver (RX) at the destination ToR. After having determined the TX/RX pair, the next step is to assign a suitable wavelength for the light-path between the selected TX/RX pair. The controller finds a wavelength for the connection, and assigns a time slot for the burst transmission. For wavelength assignment, we take advantage of the cyclic routing property of AWG. Specifically, for the MxM AWG operating on W wavelengths, a pool of F = W/M wavelengths can be shared for data transmission between an input/output port pair at the same time. The final step is to assign a time slot for the burst.

3. Simulation performance analysis

The performance of DROAD architecture was studied using OMNET++ [10] simulator. The full logic including ToR switches, transmitters, receivers, space switches, AWGs and the controller was implemented on the side of standard INET modules that provided the TCP/IP layers. To give the best indication of application-level performance and allow the comparison with other designs we used three different metrics:

1. **FCT** is used as the main performance metric since application performance is more sensitive to FCT rather than packet delay/loss statistics [11]. FCT is the time from when the first packet of a flow is sent (in TCP, this is the SYN packet) until the last packet is received. Intuition suggests that as network bandwidth increases flows should finish proportionally faster. For the current Internet, with TCP, this intuition is wrong. Latest research [11] shows that improvements in link bandwidth have not reduced FCT by much in the Internet over the past 25 years. With a 100-fold increase in bandwidth, FCT has reduced by only 50% for typical downloads. OMNET++ simulator is configured to measure the FCT for every flow in the network, which in practice consumes large amount of computing resources, however it allows the in-depth performance analysis.
2. **TCP Session Time** - there are at least two reasons why the TCP session time should be an important metric. Firstly, applications often leave a connection open long after data has been transmitted and the connection is no longer required. This typically happens when a connection is not deliberately closed as part of the transmission and is terminated later when the connection times out. This is an inefficient way to close a TCP connection. It is good practice to close a connection as soon as possible after data is transmitted, to prevent channels from being kept open needlessly. By closing connections promptly that do not need to be kept open for reuse, you can reduce energy consumption in your application. Despite that this is an application layer problem, we want to examine the impact of FCT times on TCP session time and consequently check if it is possible to close TCP sessions faster. Secondly, TCP session times can be affected by packet loss and subsequently packet re-transmissions, duplicated ACKs and eventually TCP timeouts, which are very harmful from the performance point of view. To achieve the meaningful results, we configure the simulator to close TCP sessions right after the last packet from each flow arrives at the destination (including configuration packets). We then measure the time for which the TCP session was opened for. Similarly to FCT measurements, we do this for each TCP session in the simulation.
3. **Number of active TCP sessions** - we assume that it is computationally costly for the network to support large numbers of simultaneous TCP sessions [12] and especially TCP sessions that last for long periods of time. Therefore, similarly to the above metric, we examine the amount of TCP sessions opened at any given time in both networks (DROAD and electrically-switched) for the entire duration of the simulation. Then we look at the total number of sessions that were opened in order to complete all flows during the simulation. In many applications such as web browsing, it is difficult to predict when exactly data transfers of a TCP connection will finish, since a client may initiate a new request at any time after receiving the previous response. Thus a common practice is to employ an application-layer timeout to close a TCP connection. For example, HTTP keep-alive timers, which are usually statically configured, are used by almost all of today's HTTP clients and servers. TCP connections are usually closed by exchanging FIN packets between two endpoints. The aforementioned timeout can cause FIN packets to be delayed by seconds to minutes after the transfer of the last user data packet. There are schemes proposed to overcome this issue such as Silent TCP connection Closure [13], however those are not implemented in today's networks as a modification to an endpoint operation systems (mainly user device) is required.

We generate 350,000 flows and use a hotspot model described in Ref. [14]. We assume that only 10% of ToRs (hottest) send 90% of bytes. These numbers comply with the DCN traffic characteristics reported in Refs. [15,16]. We consider 100% offered load to be the hottest ToRs sending at 100% of all its outgoing channels capacities. We vary the inter-arrival time to simulate different load levels in the range of 10–90%. In order to achieve meaningful and accurate results we test two networks initially: optically-switched (DROAD) and electrically-switched (traditional leaf-spine) using the same number of flows and flow sizes. Simulation parameters are presented in Table 1. Both networks were designed using identical number of clusters/spines (4), ToRs (8), data channels (8) and switch buffer sizes (16 MB). The data link rate is set to 10Gbps and number of wavelengths to 64 (only applicable to the optically-switched architecture). The intra- and inter-cluster switching time, as well as packet processing time is set to 0.1μs in both networks. The cluster reconfiguration time is set to 30μs, with the sampling interval of 10 ms. Equal-cost multi-path routing was used as the default load-balancing algorithm in leaf-spine network.

3.1. Flow Completion Time

3.1.1. Overall performance

First, we look at the comparison of FCTs for all flow sizes in the range of 300 bytes to 10 MB. Fig. 3 shows the overall performance comparison between the proposed optically-switched (DROAD) and traditional electronically-switched networks. This comparison provides us with an overall view of the network performance in both architectures. We can notice immediately that plotted FCTs for DROAD follow linearly increased pattern, where FCTs for ESN are heavily scattered for similar flow sizes. This is because in the DROAD architecture, packets are aggregated into larger bursts and then scheduled on what is essentially an end-to-end circuit with no packet loss, hence it can achieve lower FCTs. Packets can still be lost in the proposed scheme, mainly at the transmitting ToRs if the burst queue fills up, however it occurs infrequently. In addition to this, VoQs in the proposed architecture use shared memory for all destinations i.e. they don't have a fixed max size per destination, where ESNs would have a individual link transmission queue set per destination, which leads to a greater loss. The first subgraph in Fig. 3 shows an advantage of ESNs for up to 30% network load. For medium and high network loads, smaller flows start to suffer in ESN, while the larger flows are still showing acceptable FCTs. Due to the large number of simulations that were executed in this project, we handpicked the results for 30%,60% and 90% network loads, which constitute to typical network testing scenarios, hence Fig. 3 shows the comparison of FCTs for only three typical network loads, where the overall improvement was calculated by averaging FCT results for all network loads simulated (10%–90%). The full comparison of average FCTs for different network loads (10–90)% and flow sizes is also presented as a bar chart in Fig. 7.

In order to get a better understanding of the network performance and to achieve more detailed analysis we define three categories of flow sizes: mice (<10 KB), medium (10KB-1MB) and large (>1 MB). It is especially important to examine FCTs for different flow sizes as the overall average FCT results can only give us partial knowledge about the

Table 1
Simulation paramenters

Network architecture	Clusters/ spines	ToRs per cluster	Data channels	Switch buffer size	Wavelenghts	Data rate	Burst aggregation timeout	Packet processing time	Overhead	Reconfiguration time	Sampling interval
DROAD (proposed)	4	8	8	16	64	10	25μs	0.1μs	1us	30μs	10 ms
Traditional Leaf-spine	4	8	8	16	N/A	10	N/A	0.1μs	1us	N/A	N/A

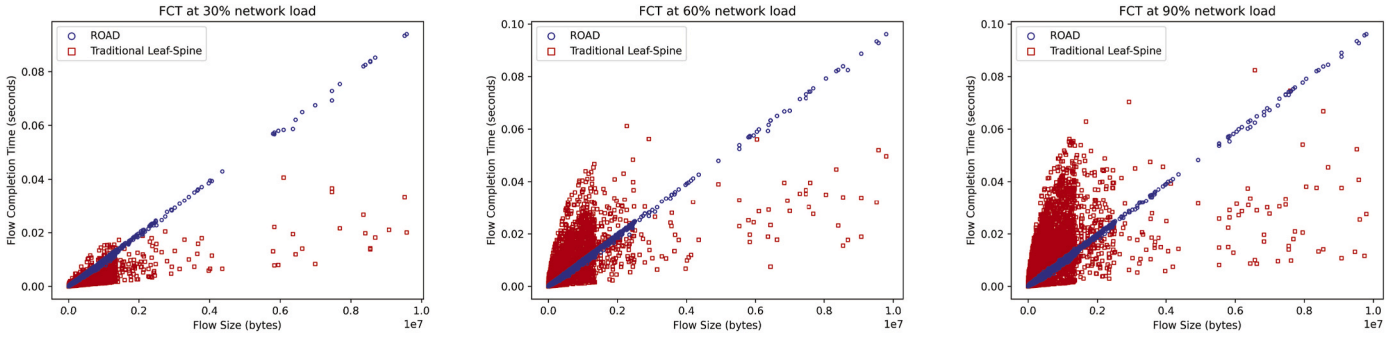

Fig. 3. FCTs for a full range of flow sizes (300 bytes–10 MB) and three typical network loads (30,60,90)%.

performance, where more fine grained analysis allows us to build more predictable networks. For example, some DCNs may be designed for specific traffic such as Internet of Things networks that mostly consist of very small, short-lasting flows. Results of those experiments are shown in Table 2.

3.1.2. *Mice flows*

Fig. 4 shows the average FCT for mice flows for 30,60 and 90% network load. DROAD is able to reduce the FCT by 164 ms on average, which is very important in case of short-lived flows. Therefore, mice flows are handled very efficiently by DROAD, when compared to traditional leaf-spine network, especially at higher network loads. This is because mice flows are not affected by larger flows in the network as mice flows are assembled into a larger bursts and remain unlikely to get into a re-transmission state.

3.1.3. *Medium flows*

Fig. 5 shows the comparison of FCTs for medium size flows at 30,60 and 90% network load. As far as traditional leaf-spine network performs well at lower loads, network performance starts to degrade with 30% network load or higher. The average FCT improvement for medium sized flows is over 37% as presented in Table 2.

3.1.4. *Large flows*

Traditional leaf-spine network is managing larger flows more efficiently at network loads of up to 60% (Fig. 6) however, smaller flows are being affected by larger flows when traffic classification and flow priority functions are not implemented such as those in Ref. [17] or [18]. In addition to this, in traditional leaf-spine networks, packet transmission queues have busy periods, even at low network loads, but especially at higher loads, hence some flows will wait significantly longer to complete due to a small number of packets that have been delayed due to multiple re-transmissions. Fig. 6 shows that DROAD underperformed at 30%

Table 2
Average FCT (in seconds) results and calculated percentage difference for DROAD (proposed) and traditional leaf-spine architectures.

		Network load									
Flow size category		**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	
MICE	**DROAD (proposed)**	0.139	0.126	0.116	0.109	0.102	0.097	0.092	0.091	0.089	$\overline{FCT}$ **= 0.107**
	Traditional Leaf-Spine	0.019	0.060	0.117	0.236	0.299	0.359	0.394	0.449	0.511	$\overline{FCT}$ **= 0.271**
	Percentage difference	-86.465	-52.821	1.105	116.858	193.344	271.051	326.500	394.247	473.261	$\overline{\Delta}\%$ **= 181.898**
MEDIUM	**DROAD (proposed)**	3.779	3.611	3.518	3.464	3.460	3.432	3.456	3.506	3.537	$\overline{FCT}$ **= 3.529**
	Traditional Leaf-Spine	0.634	1.189	2.152	4.077	5.200	6.375	6.674	7.664	9.155	$\overline{FCT}$ **= 4.791**
	Percentage difference	-83.231	-67.076	-38.825	17.707	50.271	85.744	93.108	118.566	158.801	$\overline{\Delta}\%$ **= 37.229**
LARGE	**DROAD (proposed)**	17.154	16.586	18.119	17.581	17.542	17.300	17.209	17.369	17.082	$\overline{FCT}$ **= 17.327**
	Traditional Leaf-Spine	2.656	4.904	7.697	12.030	15.397	17.110	17.480	18.224	21.070	$\overline{FCT}$ **= 12.952**
	Percentage difference	-84.518	-70.434	-57.522	-31.573	-12.226	-1.094	1.574	4.922	23.348	$\overline{\Delta}\%$ **= -25.281**

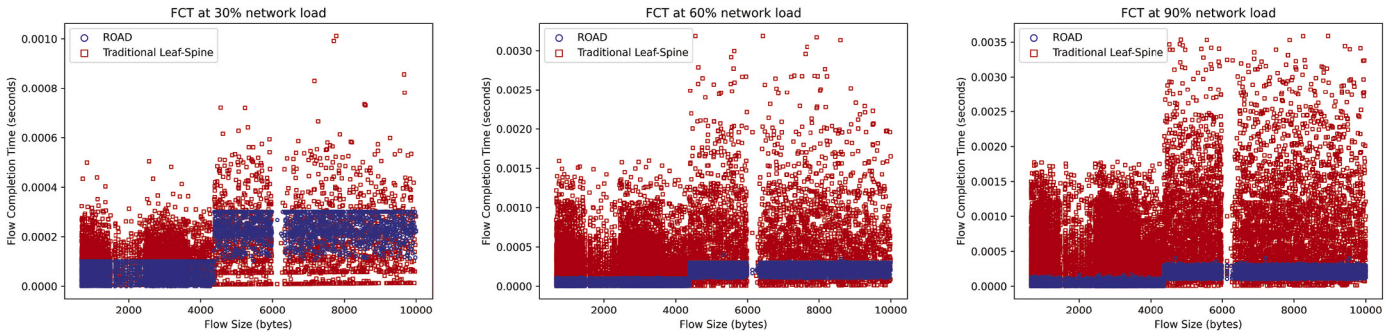

Fig. 4. FCTs for a mice flows (<10 KB) and three typical network loads (30,60,90)%.

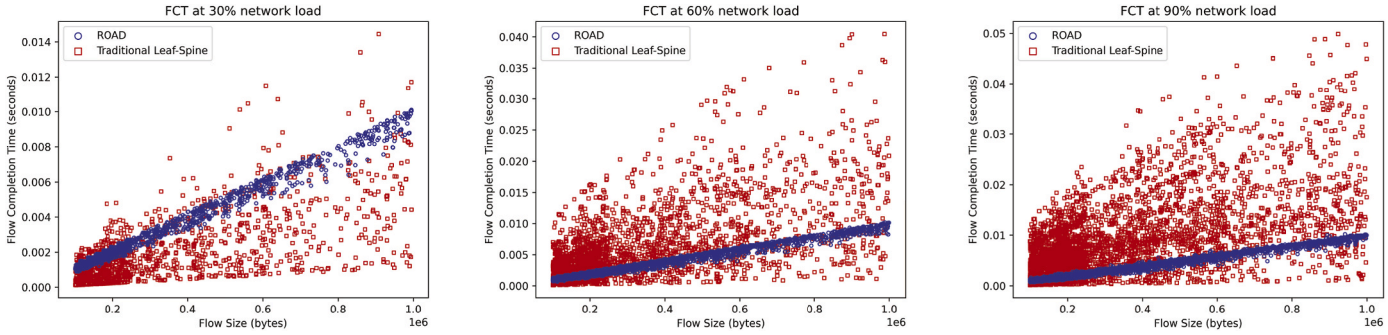

Fig. 5. FCTs for a medium flows (10KB-1MB) and three typical network loads (30,60,90)%.

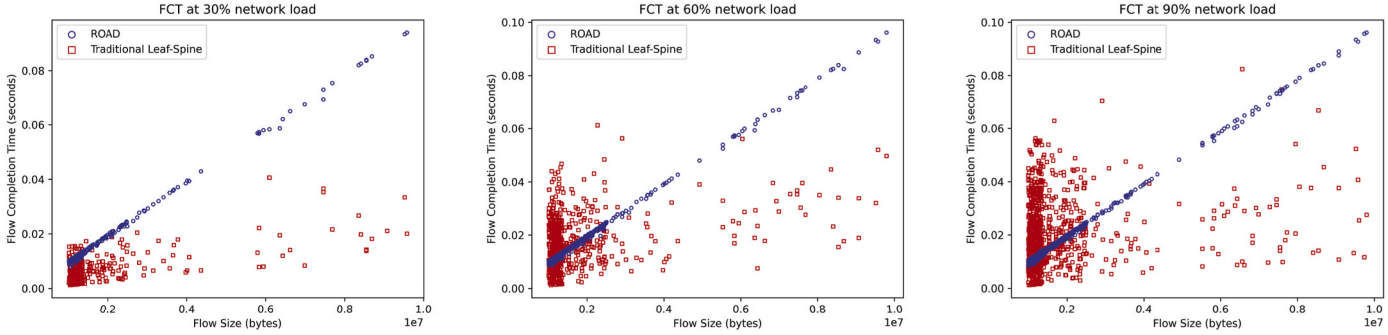

Fig. 6. FCTs for a large flows (>1 MB) and three typical network loads (30,60,90)%.

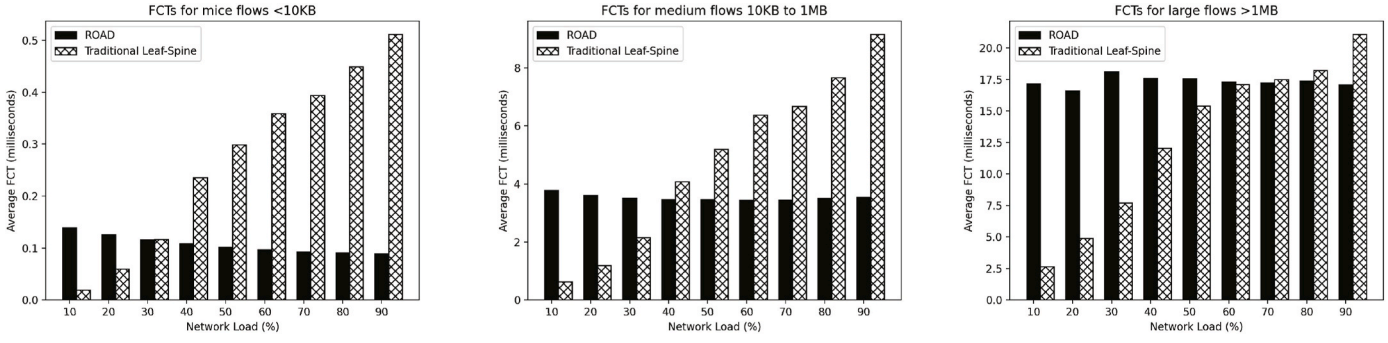

Fig. 7. Flow Completion Time for different range of flow sizes.

network load (57% slower), 60% network load (1% slower) and outperformed at 90% network load (23% faster). Table 2 shows more detailed results from which we can conclude that DROAD is slower in completing larger flows at lower network load and that it performs well at higher network loads, which is also the case for other flow sizes (mice and medium).

3.2. TCP session time

The analysis of average TCP session times is presented in Fig. 8. TCP

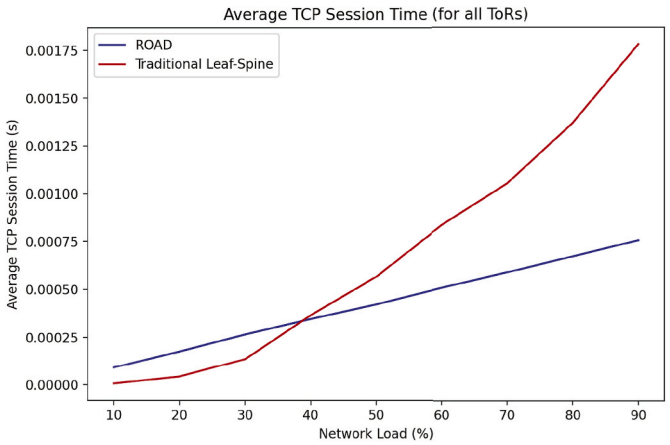

Fig. 8. Average TCP session time for all ToRs.

sessions stay open for similar amount of time for up to 40% network load (ESN outperforms DROAD by 0.14 ms on average) as the number of re-transmissions is negligible due to low packet loss in the network. DROAD shows significantly better performance at higher loads presenting predictable trend. For example at 90% network load, DROAD shows the overall difference of 1 ms (57% improvement) per TCP session in the network, meaning that the session are closed more efficiently.

By looking at Fig. 9, which shows the average TCP session time for each ToR in the network at 90% load, we can identify ToRs that are significantly more affected than others especially in ESN, however DROAD is showing more consistent performance especially in case of "hot" ToRs.

3.3. Number of active TCP sessions

The effects of network parameters on TCP behavior affects the energy consumption of the end-hosts and the DCN. More specifically, the presence of any bottlenecks in the network, causing loss of packets, produces, through the TCP congestion control, the re-transmission of lost segments, which causes the sender to wake-up an additional number of times, so wasting additional energy. Reducing the number of TCP sessions to minimum can save energy and allow other TCP sessions to be opened if needed.

Table 3 shows the number of TCP sessions that were opened during our simulations for various network loads. In the DROAD architecture, flows are completed faster and re-transmissions/timeouts occur

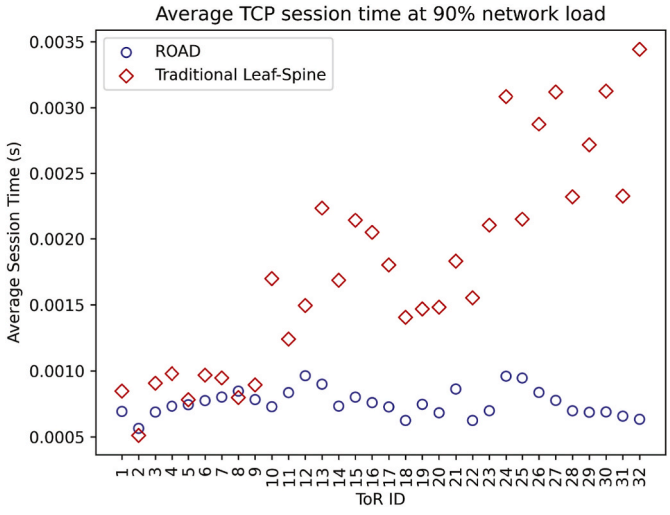

Fig. 9. Average TCP session time per ToR for 90% load.

Table 3
Active TCP sessions.

Network load	Total number of active TCP sessions		
	DROAD	Leaf-spine	Δ%
10%	6364	3400	−47%
20%	19617	9695	−51%
30%	40471	25606	−37%
40%	64336	65642	2%
50%	93855	115008	23%
60%	130899	185026	41%
70%	170537	260425	53%
80%	218106	357087	64%
90%	270588	476498	76%

infrequently therefore DROAD is able to reduce the number of active TCP sessions by 76% for higher network loads. The traditional ESN is still showing good performance for very low network loads (<30%). In order to better understand the simulation results we plot the number of active TCP sessions for the entire simulation time (1s) for 30,60,90% network loads in Fig. 10. It is interesting to observe that the number of TCP sessions opened is more consistent and stable in case of DROAD without large spikes as seen in case of traditional ESN. For example, at simulation time 0.3s–0.8s, there were over 40 simultaneous sessions opened at the time in case of traditional ESN, where only 10 remained opened for DROAD at 90% load.

4. Comparative study: DROAD vs SIRIUS

Simulation results are further compared with an optically-switched DCN architecture SIRIUS - a Microsoft Research project led by authors of [6]. In SIRIUS, each uplink port on a node is connected to a different grating and through it, can send traffic to a different set of destination nodes. This is in principle similar to DROAD topology, where each node can send traffic to a any other destination node. Gratings with 100 ports and lasers that can tune across 100 wavelengths are commercially available, so each node uplink can reach 100 other nodes through the corresponding grating. The SIRIUS topology is flat, so a simple direct routing had to be used to allow the communication between the nodes using only a fraction of a total uplink bandwidth. Sirius adopts a scheduler-less design, proposed by Chang et al. [19] as an extension of Valiant load balancing [20]. Traffic from a node, irrespective of its destination, is routed uniformly on a packet-by-packet basis across all other nodes, which then forward the traffic to its destination node. SIRIUS is depending on CMOS-based electrical switches for intra-rack connectivity in rack-based deployments and use all-optical links for inter-connections similarly to DROAD. Authors of SIRIUS simulate a rack-based deployment DCN with 128 racks, each comprising 24 servers and compare the performance of their network to the electrical network. Each ToR switch is equipped with 8 uplinks. Authors of SIRIUS evaluate a non-oversubscribed setup (ESN (ideal)) for electrically-switched network. It is important to note that SIRIUS solution is evaluated using a significantly larger network of ToR switches, hence we reconfigured our testing environment and ran 5 more simulations using 128 ToRs (instead of 32) and 50Gbps uplinks (instead of 10Gbps) for 10,25, 50,75 and 100% network loads accordingly.

Workload characteristics - in SIRIUS, a synthetic workload, modeled after published datacenter traces [[21,22]] is generated. Flow sizes are heavy tailed, drawn from a Pareto distribution with shape parameter 1.05 and mean 100 KB. This distribution creates a heavy-tailed workload where the majority of flows are small, but the majority of traffic is from large flows, as is commonly observed in production networks. Flows arrive according to a Poisson process with uniformly randomly chosen sources and destinations. Each simulation generates approximately 200,000 flows. In our simulation, we have matched the traffic characteristics to the above description. Results of our experiments are shown in Fig. 11. Both optical solutions outperform

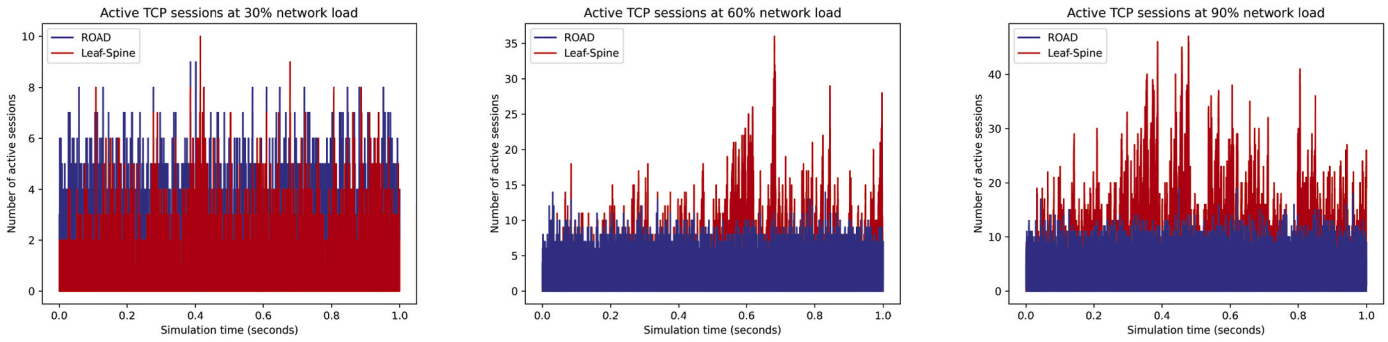

Fig. 10. Comparison of the number of active TCP sessions needed to complete all flows in the network.

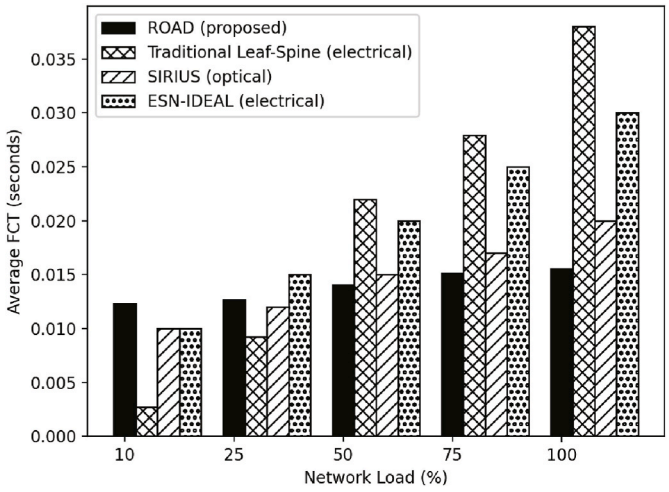

Fig. 11. Average FCTs for DROAD network design (optical and electrical) vs SIRIUS (optical and electrical).

electrically-switched networks at higher network loads (>50%). The results show an advantage of our proposed solution compared to SIRIUS at higher network loads and a slight disadvantage of our solution compared to SIRIUS at lower network loads. FCTs are comparable between our proposed solution and SIRIUS if we consider the average achievable FCT for different loads.

5. Conclusions

Proposed architecture (DROAD) was evaluated in terms of the average FCT achieved, the average session time and the number of active TCP sessions needed to be opened to complete flows. The architecture is based on fast tunable lasers and optical space switches that allow fast routing of inter-cluster traffic. A proposed scheduler allows us to make routing decisions, configure the network and schedule traffic efficiently. The simulation results show that DROAD can provide 64% FCT improvement overall, 13.7% decrease in the average total number of active TCP sessions opened at the time and the reduction of TCP session times by even 57% for high network loads compared with a leaf-spine topology. In this work we show that DROAD performs exceptionally well at higher network loads, especially with small and medium-size flows.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has emanated from research supported in part by a research grants 18/EPSRC/3591 and 13/RC/2077, from Science Foundation Ireland (SFI); co-funded under the European Regional Development Fund.

References

[1] A. Singh, J. Ong, A. Agarwal, Jupiter rising: a decade of Clos topologies and centralized control in google's datacenter network, Comput. Commun. Rev. 45 (4) (2015) 183–197, https://doi.org/10.1145/2785956.2787508.
[2] H. Ballani, P. Costa, I. Haller, K. Jozwik, K. Shi, B. Thomsen, H. Williams, Bridging the Last Mile for Optical Switching in Data Centers, 2018. W1C–3.
[3] S.K. Moore, Another step toward the end of moore's law: samsung and tsmc move to 5-nanometer manufacturing-[news], IEEE Spectrum 56 (6) (2019) 9–10.
[4] N. Farrington, G. Porter, S. Radhakrishnan, H.H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, A. Vahdat, Helios: a hybrid electrical/optical switch architecture for modular data centers, in: Proceedings of the ACM SIGCOMM 2010 Conference, 2010, pp. 339–350.
[5] H. Liu, M.K. Mukerjee, C. Li, et al., Scheduling techniques for hybrid circuit/packet networks, in: Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, 2015, pp. 1–13.
[6] H. Ballani, P. Costa, R. Behrendt, D. Cletheroe, I. Haller, K. Jozwik, F. Karinou, S. Lange, K. Shi, B. Thomsen, et al., Sirius: a flat datacenter network with nanosecond optical switching, in: Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, 2020, pp. 782–797.
[7] D.D. Le, L.P. Barry, D.C. Kilper, P. Perry, J. Wang, C. McArdle, Agiledcn: an agile reconfigurable optical data center network architecture, J. Lightwave Technol. 38 (18) (2020) 4922–4934.
[8] C. Liu, M. Xu, S. Subramaniam, A reconfigurable high-performance optical data center architecture, in: 2016 IEEE Global Communications Conference (GLOBECOM), IEEE, 2016, pp. 1–6.
[9] M. Imran, M. Collier, P. Landais, K. Katrinis, Software-defined optical burst switching for hpc and cloud computing data centers, J. Opt. Commun. Netw. 8 (8) (2016) 610–620.
[10] A. Varga, Discrete event simulation system, in: Proc. Of the European Simulation Multiconference (ESM'2001), 2001, pp. 1–7.
[11] N. Dukkipati, N. McKeown, Why flow-completion time is the right metric for congestion control, Comput. Commun. Rev. 36 (1) (2006) 59–62.
[12] B. Tuffin, P. Maillé, How many parallel tcp sessions to open: a pricing perspective, in: International Workshop on Internet Charging and QoS Technologies, Springer, 2006, pp. 2–12.
[13] F. Qian, S. Sen, O. Spatscheck, Silent tcp connection closure for cellular networks, in: Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, 2013, pp. 211–216.
[14] J. Wang, C. McArdle, L.P. Barry, Energy-efficient optical hpc and datacenter networks using optimized wavelength channel allocation, in: 2015 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), IEEE, 2015, pp. 1–8.
[15] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, R. Chaiken, The nature of data center traffic: measurements & analysis, in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, 2009, pp. 202–208.
[16] T. Benson, A. Akella, D.A. Maltz, Network traffic characteristics of data centers in the wild, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, 2010, pp. 267–280.
[17] M. Hamdan, B. Mohammed, U. Humayun, A. Abdelaziz, S. Khan, M.A. Ali, M. Imran, M.N. Marsono, Flow-aware elephant flow detection for software-defined networks, IEEE Access 8 (2020) 72585–72597.
[18] R.B. Basat, G. Einziger, R. Friedman, Y. Kassner, Optimal elephant flow detection, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.

[19] C.-S. Chang, D.-S. Lee, Y.-S. Jou, Load balanced birkhoff–von neumann switches, part i: one-stage buffering, Comput. Commun. 25 (6) (2002) 611–622.
[20] L.G. Valiant, G.J. Brebner, Universal schemes for parallel communication, in: Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing, 1981, pp. 263–277.
[21] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, M. Sridharan, Data center tcp (dctcp), in: Proceedings of the ACM SIGCOMM 2010 Conference, 2010, pp. 63–74.
[22] A. Greenberg, J.R. Hamilton, Vl2: a scalable and flexible data center network, in: Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, 2009, pp. 51–62.