AgileDCN: An Agile Reconfigurable Optical Data Center Network Architecture

Dinh Danh Le, Liam P. Barry, Daniel C. Kilper, Philip Perry, Jingyan Wang and Conor McArdle

***Abstract*— This paper presents a detailed examination of a novel data center network (DCN) that can satisfy the high capacity and low latency requirements of modern cloud computing applications. This reconfigurable architecture called AgileDCN uses fast-switching optical components with a centralized control function and workload scheduler. By providing a highly flexible network fabric between racks that can be reconfigured to provide all-optical paths, network efficiency can be improved significantly. The simulation results show that the AgileDCN network can deliver TCP flow completion times significantly shorter than an equivalent electronic leaf-spine topology at high (70%) loads.**

***Index Terms*—Data center networks, Optical networking, Agile reconfigurability, Traffic scheduling.**

I. Introduction

A large-scale data centre network (DCN) must provide low latency and high capacity connectivity between thousands of servers and storage nodes. Currently, such a network uses optical links between network nodes that use electronic packet switching to create the required network paths. The lowest layer of this 3 tier network is composed of the Top-of-Rack (ToR) switches that connect the servers in each rack to the aggregation switches. The aggregation layer then connects to a network of powerful core switches. Each network layer therefore duplicates the switching hardware of the lower tier in order to preserve full bisection bandwidth across the network. This results in repeated conversion from the optical domain used for transmission to the electronic domain for switching within the node. The use of reconfigurable optical nodes to create all-optical paths to implement the routing purely in the optical domain offers the possibility of reducing latency and energy consumption while providing network capacity that can be managed more efficiently.

In this paper, we propose a reconfigurable DCN architecture based on two separate layers of high-radix space switches and multiple low-radix fast Arrayed Waveguide Gratings (AWGs) operating in conjunction with fast tunable lasers and optical receivers. The network-wide reconfigurability provided by the space switches makes the architecture highly adaptable to the typically diverse and dynamic traffic patterns observed in datacenters. The architecture proposes a novel use of coherent optical technologies (primarily used in optical backbone networks) to considerably reduce switching time.

Unlike electronic switches, optical switches do not have the ability to perform packet inspection or other intelligent processing functions, so optical switching in principle trades off ubiquitous processing for more efficient transport and switching. This is a trade off that is already being exploited in Software Defined Networking (SDN), by separating the data and control planes. We extend this principle to our proposed optically switched architecture by using a centralized control plane, which optimizes load-balancing of the incoming traffic, and creates the required all-optical paths through the network. This reconfiguration feature makes the architecture adaptable to widely varying traffic dynamics, which greatly reduces latency compared to a non-reconfigurable network.

To evaluate our architecture's network performance we compare to state-of-the-art electronic leaf-spine architectures, which aim for low latency and high intersection bandwidth by flattening the network topology to just two tiers of switching. A lower layer of leaf nodes interconnects the ToRs and leaf nodes are interconnected by a higher layer of high port-count spine switches. Detailed simulation studies show that the proposed architecture can achieve similar network bandwidths at significantly lower latencies (and lower TCP flow completion times) compared to electronic leaf-spine architectures.

II. Related Work

Different optical technologies have been explored for DCN scenarios. Micro Electro-Mechnaical Systems (MEMS) have been used to create all-optical networks (e.g., OSA [1], Mordia [2]) and also for hybrid electro-optical networks (e.g., Helios [3] and Lightness [4]). The current MEMS state-of-the-art components are, however, rather slow to reconfigure and have a low port-count which limits the scalability of such networks. The arrayed waveguide grating (AWG) however, offers greater scalability and can be used in conjunction with fast-switching tunable lasers to provide reconfiguration times in the order of tens of nanoseconds. Such AWG-based architectures have been used previously in DCNs [5]–[7]) and more recently, the NEPHELE project [8] has proposed fast AWG-based routers in an optical WDM ring configuration, coupled with a TDMA scheme, to provide an optical data center network capable of interconnecting a large number of ToRs. Free-Space Optics (FSO) technologies have also been proposed to enable thousands of direct optical connections between ToRs (FireFly

Dinh Danh Le, Liam P. Barry, Philip Perry and Conor McArdle {dinhdanh.le, liam.barry, philip.perry, conor.mcardle}@dcu.ie are with the School of Electronic Engineering, Faculty of Engineering and Computing, Dublin City University, Dublin 9, Ireland. Dinh Danh Le is also affiliated with Hong Duc University (HDU), 565 Quang Trung Street, Dong Ve Ward, Thanh Hoa (Vietnam).

Daniel C. Kilper dkilper@optics.arizona.edu is with the College of Optical Sciences, University of Arizona, 1630 E. University Blvd., Tucson, AZ 85721, U.S.A.

Jingyan Wang jingyan.wang@huawei.com is with Huawei Technologies Co Ltd, Huawei Industrial Base, Bantian, Longgang 518129. Shenzhen, Guangdong, China

[9], ProjecTor [10]), realizing a single central high-radix space switch as the basis of the optical DCN.

The present work most closely relates to [11] and [7]. In [11], a similar arrangement of AWG switches, as proposed here, is used as the basis for a large-scale DCN. The proposed architecture uses wideband non-coherent optical receivers which requires that inter-cluster traffic needs 2-hop routing. In contrast, our 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 hop, reducing switching complexity and latency. Additionally, [11] proposes a time-slotted system, where a single packet is transmitted in one time slot. In our architecture, larger transmission units, with a single common control packet, are used to gain a multiplexing efficiency. A similar transmission scheme for DCNs is proposed in [7], though the proposed switching architecture doesn't have the same degree of reconfigurability as the architecture proposed in this paper. Unlike previous studies of related architectures, which evaluate delay and throughput on a per packet basis, we evaluate TCP flow completion times (FCT) and compare to state-of-the-art electronic DCNs, which gives a more realistic representation of expected data center application performance.

A. Enabling Optical Technologies

This section briefly discusses the important optical technologies used to realize the proposed architecture, namely; wavelength tuneable lasers, optical space switches, arrayed waveguide gratings, and coherent optical receivers.

1) Wavelength Tuneable Lasers:: Tuneable lasers (TL) have become a mainstream component in core and metropolitan optical networks for sparing and inventory concerns, and for the development of flexible wavelength routed networks. As WDM technology begins to be employed in optical access and data center networks these systems will also start to employ TLs. For initial systems that only use a limited number of wavelengths, thermally tuneable single mode lasers such as Distributed FeedBack (DFB) devices can be used, however the tuning time is limited to milliseconds [12]. As network capacity grows, lasers which can tune over much wider wavelength ranges and with faster tuning times will be required. In general, laser tunability can be achieved by either electronic tuning, thermal tuning, or mechanical tuning [13]. To date, mechanically tuned devices have not proved suitable for systems applications due to reliability, scalability, performance and speed issues, and thermally tuned devices also suffer from speed issues for certain applications, so electronically tuned devices are preferable for next generation networks. With electronically tuned lasers an electric field or current applied to the device modifies the refractive index and, in turn, changes the lasing frequency. There are many types of electronically tuneable devices which are all essentially variants of the Distributed Bragg Reflection (DBR) laser [14]. The main advantages of these devices for use in optical networks are the reasonably high output power (10 dBm), fast tuning time (<100ns), wide tuning range and low intensity and phase noise levels.

2) Optical Space Switches:: As detailed earlier, electronic switches are now experiencing scalability isses due to the demand for higher port-count switches. Since optical space switches create transparent optical paths, they do not need to regenerate the signals at the data rate and thus consume significantly lower power than their electrical counterparts [15]. MEMS-based optical space switches are now commercially viable and can offer up to 1000 ports with low loss [16]. Their reconfiguration time, however, is of the order of tens of microseconds which is not suitable for short lived connections that are common in data centres [17]. Silicon-based switches have been developed that are capable of reconfiguration times in the order of nanoseconds [18], but these currently do not scale to a sufficiently high port count with current technology.

3) Arrayed Waveguide Gratings:: An AWG is a passive device that uses the cyclic nature of optical interference to create a wavelength demultiplexer. That is, the spectrum in a fibre can be separated into different wavelength bands in a number of output fibres. They are typically made in a planar lightwave circuit and will demultiplex defined WDM channels in the input fibre into separate output fibres [19]. In the current context, then, an AWG can be used in conjunction with a tunable source to create reconfigurable paths between the source and any destination associated with one of the AWG's outputs.

4) Coherent Optical Receivers:: A coherent optical receiver consists of a tuneable laser, a 90 degree optical hybrid balanced photodetector and an electrical low pass filter. The tuneable laser is typically coupled with the WDM signal containing all wavelength channels, and tuned to the same wavelength as that which is to be received. These optical signals beat together on the photodiode to generate an electrical signal around baseband containing the electrical information (optical phase and intensity) on the required wavelength. The other wavelength signals may generate electrical signals but at frequencies outside the range of the electrical filter, so are eliminated. While coherent detection was experimentally demonstrated in 1979 [20], it was only with the development of digital coherent receivers in the early part of the 21st century [21] that commercial coherent systems were developed. The digital coherent receiver allows important functions such as phase and frequency tracking to be implemented in the electrical domain which greatly reduces complexity and aids practicality. The time it takes a coherent receiver to reconfigure is the same as the time it takes the tuneable laser to alter wavelength, and this will depend on the tuning mechanism employed for the tuneable laser (whether it is mechanical, thermal or electronic).

The rest of the paper is organized as follows. Section III describes the proposed DCN architecture, including the details of data plane and control plane. The simulation results and performance analysis are presented in Section IV. Section V concludes the paper.

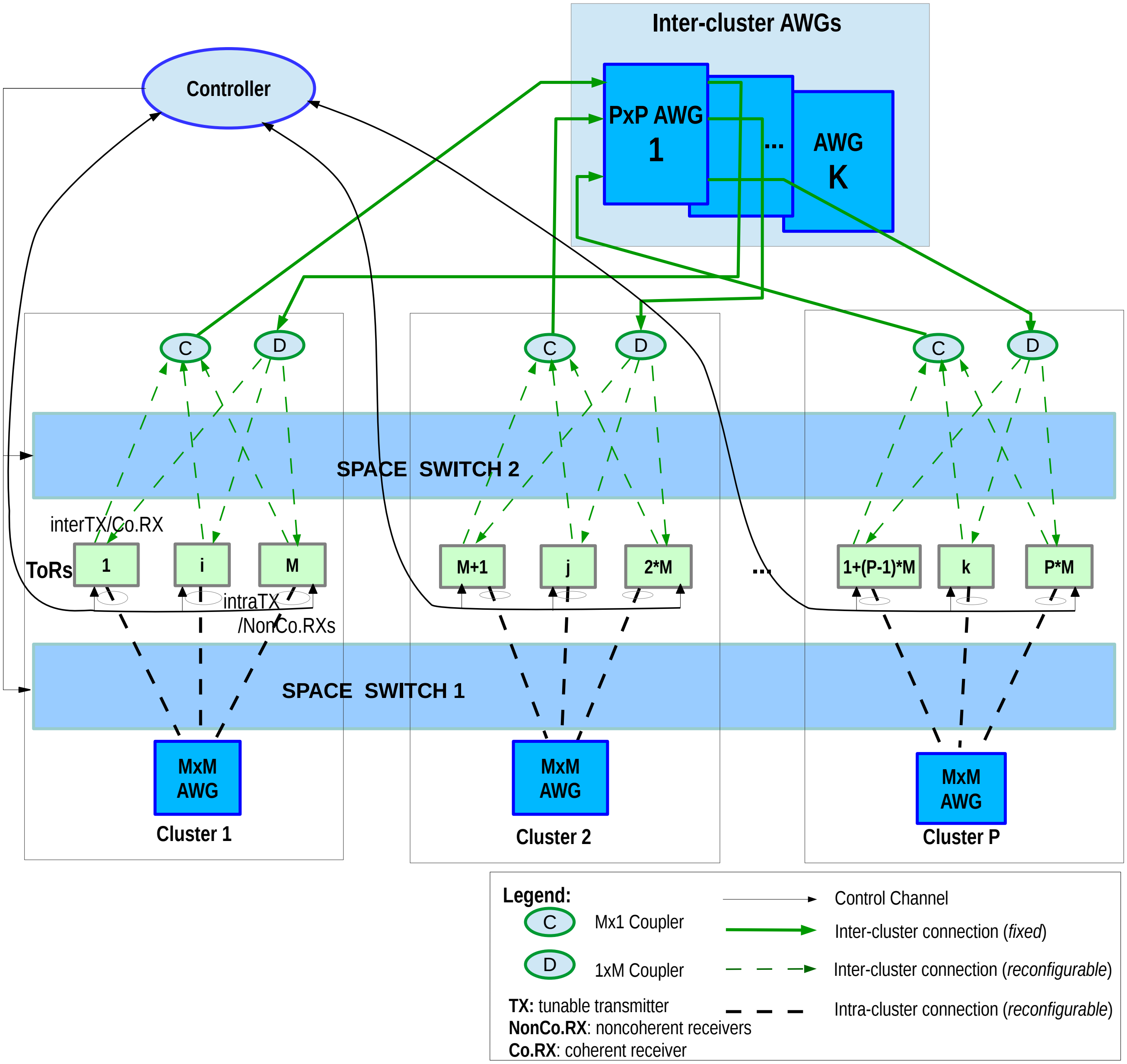


Fig. 1: AgileDCN: Agile Reconfigurable Optical Data Center Network Architecture

III. Agile Optical Data Center Network Architecture: AgileDCN

A. Overall Architecture

The proposed data center architecture, called AgileDCN, is shown in Fig. 1. The architecture provides two optically-switched data planes, which together provide a highly configurable large-scale DC network. The lower data plane carries intra-cluster traffic and consists of Space Switch 1 and a set of P AWG routers. Each AWG provides fast optically-switched connections between the M ToRs grouped into the same cluster. The upper data plane carries inter-cluster traffic and consists of Space Switch 2 and sets of optical couplers/decouplers and AWG routers, that together provide optically-switched connectivity between ToRs residing in different clusters. This topology, employing two independent space switches, means that all ToR-to-ToR connections (intra- and inter-cluster) are high-bandwidth, single-hop optical connections, facilitated by low-power AWG-based circuit routing. There is no electronic buffering in the inter-connection network, removing potential packet latency bottlenecks and further reducing network power consumption. Additionally, separation of intra- and inter-cluster connections enable a network reconfigurability that can optimally met very diverse datacenter traffic patterns. The details of the two connection types provided by these two data planes are as follows.

Intra-cluster Connections: Each ToR switch resides at the top of a server rack and provides direct electronically-switched connections between the servers residing in that rack (servers are not shown in the diagram). A set of M ToRs forms a cluster, where ToRs in a cluster are interconnected by an M M AWG router. Which

ToRs are grouped into which cluster is configurable. By changing the configuration of optical Space Switch 1, ToRs that communicate heavily can be connected into the same cluster.

Each ToR has $L < M$ tunable optical transmitters (TXs) for out-going intra-cluster connections (Fig. 2). These L optical channels are multiplexed (MUX) at the ToR onto a single fibre which connects, via Space Switch 1, to one AWG input port. Each TX channel at each ToR is tunable over the same wavelength range, covering L wavelength channels. All L TX wavelengths at a ToR must be distinct, at any point in time, to avoid wavelength contention in its MUX. Similarly, an output port of the AWG connects on a single fibre (via Space Switch 1) back to the ToR, where there is an $1 : L$ optical wavelength de-multiplexer (DEMUX) (Fig. 2). Each MUX output channel feeds an optical *non-coherent broadband receiver* (RX), each one receiving from a different ToR. Thus, a ToR can simultaneously transmit to L other ToRs in its cluster, using L different wavelengths. Which ToRs it transmits to is decided by the tuning of its optical transmitters. A ToR can also simultaneously receive from L other ToRs in its cluster. Which ToRs it receives from is determined by tuning at the transmitting ToRs.

Inter-cluster Connections: Connections are established using K inter-cluster links to K inter-cluster AWGs via Space Switch 2 (SS2). Specifically, each ToR has K inter-cluster optical transceivers. Unlike intra-cluster transceivers, each inter-cluster transceiver includes one tunable transmitter and one *coherent receiver*, the reason for which is explained shortly. The k^{th} inter-cluster tunable transmitter connects to $k^{th}1$ M Coupler (denoted as C) via SS2, the output port of the k^{th} Coupler connects to an input port of the k^{th} inter-cluster P P AWG; each output port of the k^{th} inter-cluster AWG connects to the $k^{th}M$ 1 Coupler (denoted as D), where the composed WDM signals are passively split and delivered at the ToRs via the associated coherent receiver. (Fig. 1 shows only the inter-connectivity using the first inter-cluster TX/RX to/from the first inter-cluster AWG, for the sake of simplification.)

AgileDCN separates the control plane and data plane. The control plane consists of a central controller which connects to each of the ToR switches by an out-of-band control channel[1]. The controller is responsible for managing (routing, wavelength assignment, traffic scheduling and switch configuration) for both intra- and inter-cluster traffic. The data plane solely performs data forwarding using pre-established connections configured by the controllers.

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 (from the WDM signal received at the AWG output) 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 AWG port can carry multiple wavelengths at a time, so long as they fit within the AWG channelization. We suppose W is the total number of wavelengths supported by the AWGs (both intra- and inter- AWGs). For a P P AWG ($P <= W$), we assume its Free Spectral Range (FSR) is equal to P times its channel spacing. As a result, at most $F = W{=}P$ wavelength channels can be simultaneously used for each port pair (input port, output port) of the AWG. This sets the routing and wavelength constraints for traffic routing and channel scheduling in our architecture, as discussed in Section III-C.

To support reconfigurable topology, the data plane leverages two large-scale space switches: Space Switch 1 (SS1) is placed between ToRs and intra-cluster AWGs, and Space Switch 2 (SS2) is placed between ToRs and the inter-cluster network. The purpose of the two space switches is to periodically re-group the ToRs into clusters, whenever the relative traffic volumes between the clusters exceed predefined thresholds. That scheme is detailed in the next sections.

B. Top-Of-Rack Switch Architecture

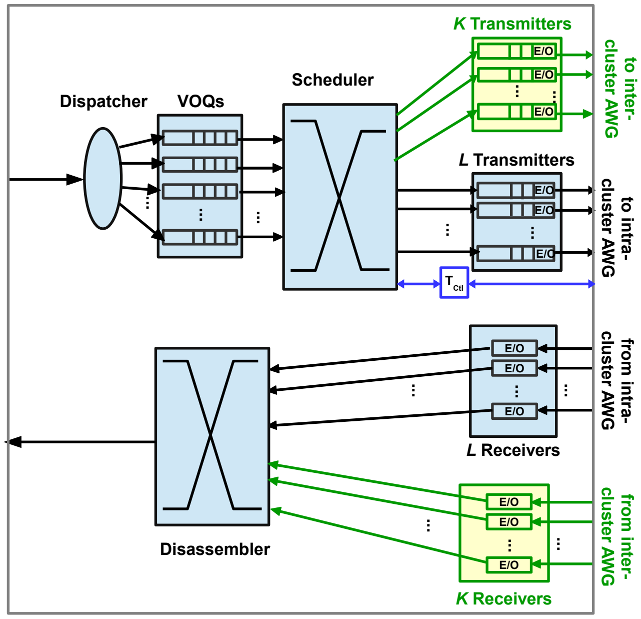


Fig. 2: Top-Of-Rack Switch Design

In this paper we are interested in evaluating DCN interconnects, so we model the DCN and its traffic from the ToR level but not the server level.

The ToR switch design is shown in Fig. 2. Each ToR has L optical tunable transmitters and L noncoherent broadband optical receivers for intra-cluster connections, connecting to its intra-AWG (in black), and K tunable transmitters and K coherent optical receivers for inter-cluster connections (in

[1] The control plane is modeled as a single centralized controller, but it can be implemented using multiple controllers to accelerate processing ability. The implementation detail is out of the paper's scope.

green) and one transceiver connecting to the controller (in blue).

A traffic scheduling scheme is applied for each unit of transmission (a bundle of packets, or burst) in an advanced channel reservation manner, so as to fully take advantage of the fast switching time of the optical components used. To this end, IP packets coming from servers are aggregated into bursts. We employ N 1 virtual output queues (VOQs), with N the number of ToRs of the DCN, so each ToR has a VOQ for every other ToR in the DCN. The Dispatcher module uses the IP address of the destination ToR to direct each packet to the correct VOQ. Burst assembly (burst aggregation) is timer-based, i.e., packets arriving during a fixed time period (beginning with the first packet arrival) are aggregated into the same burst. When the burst aggregation timer expires, a control packet (CP) is generated for the burst. The timer is restarted and the next burst begins its assembly when the next new packet arrives. The CPs are sent to the controller via a control channel and the controller then appends routing information and scheduling details and returns the updated CP back to the ToR switch. The Scheduler Module then uses the timeslot, 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. The present authors have applied a similar scheduling mechanism in [22].

C. Routing, Wavelength Assignment and Channel Scheduling

This section describes the main functionalities of the control plan, including Routing, Wavelength Assignment and Scheduling (or RWAS). 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, and receiver (RX) at the source ToR and destination ToR (this is the routing step). (Note that there is more than one TX/RX pair for each 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 (this is the wavelength assignment step). The final step is to assign a timeslot for the burst. This is the scheduling step.

The above tasks should satisfy the optical constraints incurred by the optical devices, as well as guarantee contention-free transmission. Firstly, only one data burst can be transmitted/received by a tuneable transmitter/receiver at any one instant. Secondly, at most $F = W = P$ bursts can be transmitted from an input port of the AWG to an output port of the AWG at the same time. Thirdly, tunable transmitters connecting to the same AWG port need to transmit on distinct wavelengths.

1) Control Packet Processing Framework: Fig. 3 shows the integrated control packet processing framework involving all three functionalities implemented in the controller.

Based on the relative position of the source and destination ToRs given in the control packet, the controller decides to allocate either *intra-cluster* or *inter-cluster* channels for the incoming burst. The following subsections describe the implementation.

The method uses the channel *horizon* to optimize the channel utilization. The term *horizon* of a channel is defined as the latest time at which the channel is free. Simply speaking, among multiple channels for use, the channel with the minimum horizon is favored, not only to maximize channel utilization but also to serve to split traffic load (load balance) among the available channels.

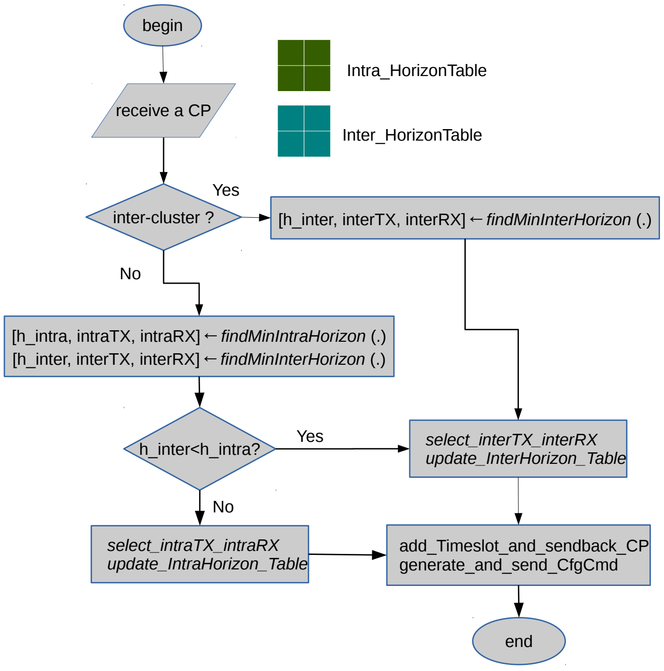

begin
receive a CP
🟩 Intra_HorizonTable
🟦 Inter_HorizonTable
inter-cluster ?
Yes
No
[h_inter, interTX, interRX] ← *findMinInterHorizon* (.)
[h_intra, intraTX, intraRX] ← *findMinIntraHorizon* (.)
[h_inter, interTX, interRX] ← *findMinInterHorizon* (.)
h_inter<h_intra?
Yes
No
select_interTX_interRX
update_InterHorizon_Table
select_intraTX_intraRX
update_IntraHorizon_Table
add_Timeslot_and_sendback_CP
generate_and_send_CfgCmd
end

Fig. 3: Control packet processing at the controller

To this end, the controller maintains *intra_horizon* and *inter_horizon* tables that keep track of the horizons for intra (respectively, inter) channels. This is possible as full information of all channel occupancies is available at the centralized controller. We divide the CP processing into *Intra-Cluster Scheduling* and *Inter-Cluster Scheduling*. If the data to be transmitted is between two ToRs residing at different clusters, then it is scheduled using *Inter-Cluster Scheduling*. In contrast, the intra-cluster data can be scheduled using either *Inter-Cluster Scheduling* or *Intra-Cluster Scheduling*, whichever is available sooner (minimum horizon). The following subsections detail the *Intra-Cluster Scheduling*, then the *Inter-Cluster Scheduling*.

2) Routing Using Intra-Cluster Connections: For intra-cluster bursts between ToR$_i$ and ToR$_j$, the routing task is to find a transmitter (TX) of ToR$_i$ and a receiver (RX) of ToR$_j$.

We use a technique called *mutual minimum horizon* to determine the TX/RX pair at both ends of the connection. Figure 4 illustrates an example where the horizons of all the TX channels at the source ToR$_i$ as well as all the RX channels at the destination ToR$_j$ are maintained at the controller. Upon reception of a CP requesting a time slot for its data burst, based on current horizons for all the channels between ToR$_i$ and ToR$_j$, the controller computes a TX/RX pair that has the mutual minimum horizons. In this example, TX number 2 (TX=2) and RX number 3 (RX=3) are selected, as they possess the channels with minimum horizons. The mutual horizon at both ends (h_{TRX}) will then be the greater among the two min-horizon values at both ends TX/RX (min_TX_h, min_RX_h, respectively) and the current time T_{cur}[2]

$$h_{TRX} = max(min\_TX\_h, min\_RX\_h, T_{cur}) \quad (1)$$

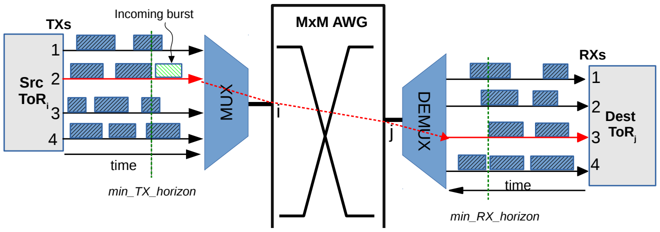


Fig. 4: Illustration of mutual minimum horizon scheduling using intra-cluster connection: the pair (TX=2, RX=3) is selected as it has the mutual minimum horizon at both ends of the connection (shown in red). Note that the time axis on RX's side of ToR$_j$ is in reverse direction to that on TX's side of ToR$_i$.

3) Wavelength Assignment: After having determined the input/output channels between a source ToR$_i$ and a destination ToR$_j$, the controller then finds a wavelength for the connection, and assigns a timeslot for the burst transmission. For wavelength assignment we take advantage of the cyclic routing property of AWG. Specifically, for the $P \times P$ AWG operating on W wavelengths, a pool of $F = W/P$ wavelengths can be shared for data transmission between an input/output port pair at the same time. These F wavelengths shared between the port pair (i, j) are given based on the following formula [11]:

$$\lambda_c = (i - j) \mod P + f \cdot P, 1 \leq i, j \leq P, \forall c \in [0, F - 1] \quad (2)$$

The wavelength for the light-path to be set up between the selected TX/RX pair is the one having the minimum horizon. To this end, we keep track of all these wavelengths' utilization using a *heap* (data structure) of their horizon values, in order to quickly pick up the wavelength λ with minimum horizon (h_λ) to assign for the lightpath. The final mutual horizon $T_{horizon}$ is determined as:

$$T_{horizon} = max(h_{TRX}, h_\lambda) \quad (3)$$

In this way we ensure no wavelength contention occurs on the entire lightpath between the selected TX/RX pair and at the same time we may make use of all available F wavelengths to set up concurrent lightpaths between the same ToR pair.

4) Timeslot Assignment: The final step is to assign a timeslot for the burst transmission. Fig. 5 illustrates an entire burst transmission cycle from its assembly at the source ToR until its transmission finishes. T_{start} and T_{end} are the start and end time of the timeslot assigned to the burst, respectively. T_{proc} is the processing time of a CP at the controller. T_{sw} (can be T_{sw_intra} or T_{sw_inter} depending if an intra or inter connection is to be established) represents the switching time of the connection via an AWG (which is actually the tuning time of the tunable laser and/or the detection time of the receiver involved in a burst transmission). T_{oh} is the aggregate time that a control packet takes to transmit from the ToRs to the controller, or from the controller to the ToRs; it is also the time a configuration command takes from its sending time at the controller until it arrives at the ToR switches. T_{trans} is the time needed to transmit the burst, which is calculated from the burst size and channel's data rate.

Finally, T_{syn} (synchronization time) is the time to guarantee system synchronization. It is the time needed for the incoming connection to wait until the ongoing transmissions on the same channel (if any) are completed. T_{syn} depends upon the mutual horizon $T_{horizon}$ computed by the Eq. 3, as well as T_{proc}, T_{oh} by the following formula:

$$T_{syn} = max(0, T_{horizon} - T_{cur} - T_{proc} - T_{oh}) \quad (4)$$

Accordingly, $T_{syn} = 0 \iff T_{horizon} \leq T_{cur} + T_{proc} + T_{oh}$, i.e., if the ongoing transmission will be completed before the CP arrives back to the ToR, then no waiting time is needed for synchronization at the ToR and the optical devices. Otherwise, an amount of time T_{syn} is required before setting up the new connection for the data burst.

Finally, the timeslot is allocated for the incoming burst using following formulas:

$$T_{start} = T_{proc} + T_{oh} + T_{syn} + T_{sw} \quad (5)$$

$$T_{end} = T_{start} + T_{trans} \quad (6)$$

As soon as the T_{start} and T_{end} have been determined, all the horizon tables are updated with relevant horizon being T_{end}.

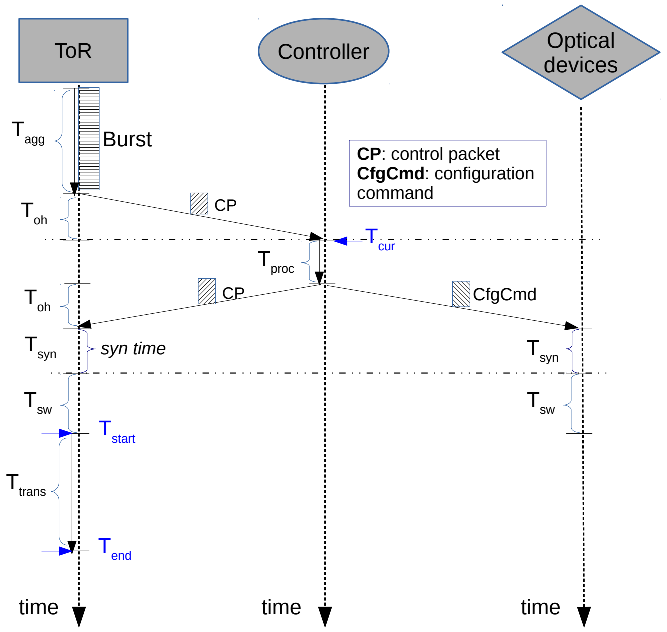


Fig. 5: A burst transmission cycle

[2] We take the current time into account when computing the mutual horizon because the case $h_{TRX} < T_{cur}$ means that the channels are now *idle*, and ready to use anytime from T_{cur}.

5) Inter-Cluster Scheduling: Unlike intra-cluster connectivity, with each ToR having multiple TXs/RXs to connect to its intra-cluster AWGs, each ToR uses a single inter-cluster TX/RX to connect to one of the K inter-cluster AWGs. As a result, scheduling over inter-cluster connectivity is slightly different from intra-cluster scheduling. The main difference is the routing step. Specifically, we find the mutual horizon for each inter-cluster TX/RX channel and the corresponding wavelength to be used. The final horizon and selected wavelength will be chosen as the one having the minimum mutual horizon. The other steps are the same as presented for intra-cluster scheduling.

Once the controller has processed the CP and made the appropriate reconfiguration decisions, the CP is returned to the ToR and a number of reconfiguration commands (CfgCmds) are sent to the optical nodes (e.g. space switches). These control messages are of the order of 32 bytes long and can therefore be communicated to the switching elements in the order of nanoseconds.

D. Cluster Reconfiguration

Cluster reconfiguration is based on sampling dynamic traffic fluctuation. Periodically (e.g., every 100ms) 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 of inter-cluster to the total bytes of intra-cluster exceed a threshold. In the simulation, we set this threshold to the number of clusters, i.e, P.

Clustering Heuristics: When the above-mentioned condition for re-clustering is met, we apply a simple greedy heuristic that was proposed in [11]: collect the groups of M ToRs which have larger mutual number of bytes waiting for transmission between them (at the VoQs) and repeat the procedure until all P new clusters are formed.

When the network is in reconfiguration, all ongoing data transmission in the data plane are paused until the reconfiguration is done. The controller postpones processing control packets, all CPs arriving at the controller are queued until reconfiguration is done. As soon as reconfiguration is done, the controller re-initiates the network state, data transmission in the data plane is resumed, and the waiting CPs at the controller are popped to be processed, and network resumes normal operation. Pausing all transmission is used here for simplicity. More advanced reconfiguration algorithms should be possible, which only pause those connections impacted by the reconfiguration.

IV. Performance Analysis

A packet-level simulation model of the AgileDCN architecture (including ToR switches, transmitters, receivers, space switches, AWGs and the controller node) was built in OMNeT++ [23]. The full logic of packet forwarding, burst generation and CP generation was implemented in the ToR model. The logic associated with routing, scheduling and optical node configuration was modelled in the controller. The TCP/IP layers were included by use of the standard OMNeT++INET library, which enabled flow completion times (as opposed to just IP packet latency) to be simulated. This gives the best overall indication of application-level performance and allows a comparison with state-of-the-art leaf-spine networks.

A. Simulation Model

We scale proposed AgileDCN architecture to $P = 8$ clusters, each consisting of $M = 16$ ToR switches for a total of 128 ToRs in the network. Each ToR has a total of 8 10Gbps optical channels, in which we use the same number of intra-cluster channels and inter-cluster channels, e.g., $K = L = 4$ and a 10Gbps control channel.

B. Traffic Generation

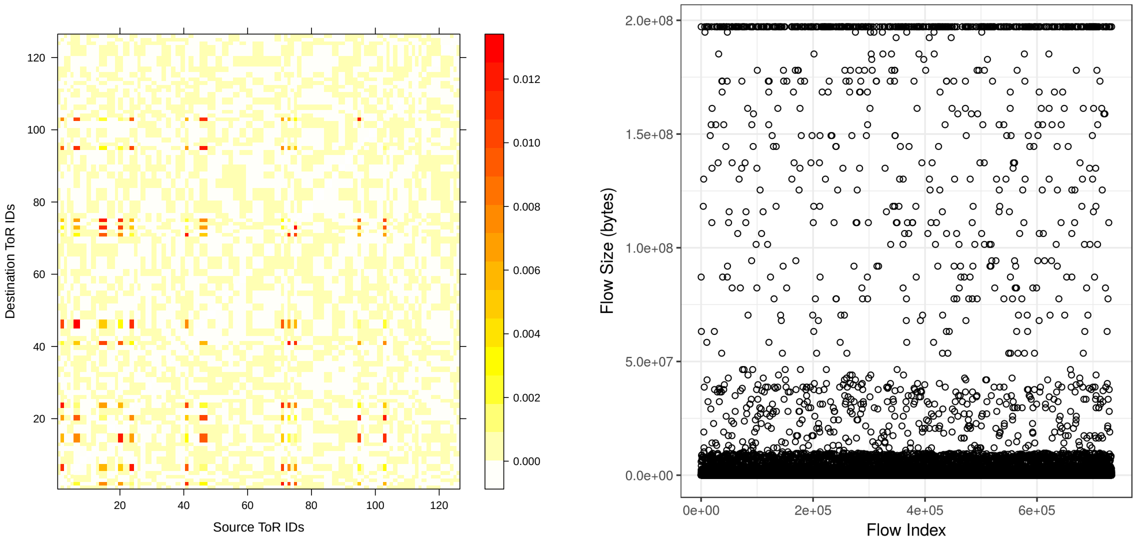
We rely on previous work on DC traffic characteristics and make use of state-of-the-art methods [6], [24] to generate traffic models for use in our simulations. Accordingly, we assume that the input traffic matrix is very sparse and skewed wherein only a small proportion of nodes are responsible for sending/receiving a substantial amount of traffic. Specifically, we assume only 10% of the ToRs (the *hottest* ToRs) send 90% of bytes and 60% of the ToRs (the *active* ToRs) send/receive traffic. These numbers comply with the DC traffic characteristics reported in [10], [25], [26]. The distribution of traffic sent/received by the active ToRs is modeled by a *Hotspot* model described in [6], in which traffic is dominated by *hotspots* by diagonal blocks of the traffic matrix. This Hotspot model means that each ToR has the same aggregate traffic volume, but that the volumes are not balanced across the cohort of destination ToRs for a particular source ToR. Figure 6a shows the heatmap of a $128 \quad 128$ traffic matrix used for simulations.

Furthermore, as applications running in datacenters mainly use TCP, we simulate input traffic as TCP flows. The traffic matrix solely, however, contains the portion of traffic sent/received betweeen the ToR's pairs, but lacks information about flow arrival time and size distributions. The different load levels required for a particular traffic matrix are created by varying the arrival rate used in the Poisson process that generates the TCP flows. The source and destination ToR pairs are selected by using the pairing probabilities in the traffic matrix and the flow sizes are taken from the literature [24].

We consider 100% offered load to be the hottest ToRs sending at 100% of all its (inter- and intra-) outgoing channels capacities. We vary the inter-arrival time to simulate different load levels in the range of f10%, 20%, .., 80%g. We generate TCP flows and map them to ToR pairs using the Deficit Round Robin strategy, as used in [24].

C. Simulation Parameters

All the key parameters are shown in Table I. The control packet processing time (T_{proc}) is the time spent by the controller to process a control packet (CP), which we assume to be 1 s (worst case), as modern hardware based controllers, through parallelism, can process up to 20 million flows per second [26]. The switching time of intra-cluster transmission (T_{intra}) includes the tuning time of the associated tunable



(a) Heatmap of a 128 × 128 traffic matrix

(b) Flow size distribution

Fig. 6: (a) Heatmap of a 128 × 128 traffic matrix and (b) flow size distribution [24] used in the simulation

transmitter at the source ToR; meanwhile the switching time of inter-cluster transmission (T_{inter}) depends on the reconfiguration time of the coherent receivers used at the ToRs. To evaluate the performance of the proposed DCN architecture we vary the above switching times over the range 0.1 s, 1 s and 10 s. The intension is to investigate the proposed architecture's performance over a representative range of fast tunable transmitter technologies (class 1: 10< s).

The overhead time (T_{oh}) accounts for propagation delay, NIC delay, O/E/O conversion delay and processing delay for a control packet and/or configuration command at the ToR switches. All these delays are in the few nanoseconds range so the aggregate delay of 1 s is compatible with these delays. We assume a data rate of 10Gbps for all the optical channels as well as the control channel. For burst generation, we investigated AgileDCN with different burst assembly timeouts (f25, 50, 75, 100, 150g s. The reconfiguration time (i.e., the switching time of the space switches) is set at 30 s.

For TCP/IP parameters, we use TCP Reno variant with maximum segment size (MSS) of 1460 bytes, which results in the typical maximum transmission unit (MTU) of IP packets of 1500 bytes. We set the buffer size for each ToR as 16MB which is in the region of what is found in current commodity data center switches. Simulation time is set to ensure settling and that depends on load, i.e., 10 seconds for loads 50%, and up to 20 seconds for loads higher than 50% (up to 80%).

D. Simulation Parametric Study

In this section we study the effect of *burst aggregation time* and different optical *component switching time* on the performance of AgileDCN. To evaluate the performance from the perspective of the application running on the servers, we use the flow completion time (FCT). This is defined as the time elapsed between the transmission of the first bit of a particular flow from the source ToR and the reception of the last bit at the destination ToR. For each value of offered load, there will be a large number of flows of varying durations, so the FCT is averaged across all the flows for each load setting. In the following, all figures showing results for average FCTs have a 95% simulation confidence interval.

TABLE I: Simulation Parameters

Parameter	Symbol	Value
Number of clusters	P	8
ToR switches per cluster	M	16
Total data channels	X	8
Intra-cluster channels	L	4
Inter-cluster channels	K	4
Number of wavelengths	W	64
Data rate		10Gbps
Burst aggregation timeout	T_{agg}	{25, 50, 75, 100, 150} s
Control packet processing time	T_{proc}	1 s
Intra-cluster switching time	T_{sw_intra}	{0.1,1,10} s
Inter-cluster switching time	T_{sw_inter}	{0.1,1,10} s
Overhead	T_{oh}	1 s
Sampling interval		100ms
Reconfiguration time		30 s
Buffer Size per ToR		16MB

1) Effect of Burst Aggregation Timeout: Figure 7 shows the simulation results in terms of average FCTs for three typical loads (30%, 50%, 70%) with different burst aggregation timeouts $T_{agg} = f25; 50; 75; 100; 150g$ s. In this experiment, we use $T_{sw\_intra} = T_{sw\_inter} = 0{:}1$ s. The other parameters are kept the same as Table I. The results show that for low loads (e.g., 30%, 50%), the performance is better with a smaller timeout. In contrast, for high load (70%) the best performance is with timeout = 100 s and worse for the other values.

The results can be explained as follows. For high loads (e.g., 70%) T_{agg} is low and bursts and CPs are generated more frequently, which contributes to higher overall processing time and configuration time, meaning that the bursts need to more frequently wait for the optical device configuration to complete, hence higher FCT. In contrast, if T_{agg} is set to be too high, less CPs are generated to be processed, but higher queuing time is incurred by individual packets when being collected into a burst, which leads to high completion time as a whole. So a moderate timeout gives a better balance resulting in lower FCTs (100 s is shown to be the best option for this case).

For low or medium loads (50%), the same situation occurs, however, by having the same processing speed of the controller and switching time of the optical devices, the queuing time at the burst aggregation stage dominates the time contribution to FCT. As a result, lower T_{agg} also lowers FCT and vice versus.

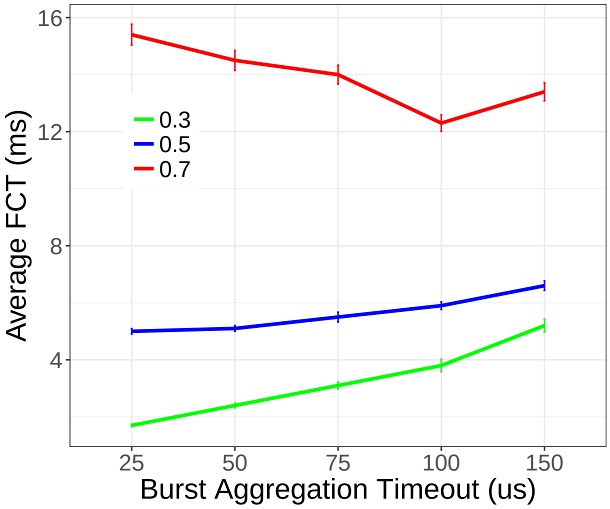


Fig. 7: Effect of burst aggregation timeout. The three lines show FCTs versus burst aggregation timeouts for three typical loads (30%, 50%, 70%).

We also conducted simulations using T_{sw_intra} = T_{sw_inter} = 1 s, the same tendency was obtained. The results suggest that an adaptive setting of T_{agg} for various loads can lead to better overall performance.

2) Effect of Switching Time: This subsection evaluates the performance of AgileDC with different switching times, in order to see how it adapts to different optical technologies. To this end, we vary the switching time T_{sw_intra} = T_{sw_inter} = $f0{:}1; 1; 10g$ s, we fix load = 50% and T_{agg} = 50 s and keep the same values for the other parameters, as per Table I.

The results are shown in Table II.

TABLE II: Effect of Switching Time

T_sw(s)	FCT(ms)	Gap(%)
0.1	**5** $\pm$ 0:1	*0*
1	**5.1** $\pm$ 0:1	*2*
10	**6.3** $\pm$ 0:13	*26*

The second column of the table shows FCT with a 95% confidence interval, while the third column shows the performance gaps (%) in term of FCT for the case of T_{sw}=0.1 s. Accordingly, there is a slightly higher FCT (2%) when using T_{sw}=1 s; while we sacrifice about 26% performance when using T_{sw}=10 s. The results indicate that class-2 tunable transmitters and (coherent) broadband receivers can provide reasonable FCTs, when compared to higher-speed components (class 1). For this reason, in the following simulation set we use moderate-speed tunable TXs/RXs operating at 1 s switching time.

E. AgileDCN versus Leaf-Spine

In this subsection, we compare AgileDCN with the state-of-the-art two-tiered electronic DCN architecture Leaf-Spine where every leaf switch is connected to each of the spine switches in a full-mesh topology as shown in Fig.8. In simulations, AgileDCN uses the parameters shown in Table I with Time T_{agg} = 100 s and T_{sw_intra} = T_{sw_inter} = 1 s. To have an equivalent setting for both architectures, Leaf-Spine uses the parameters with values shown in Table III.

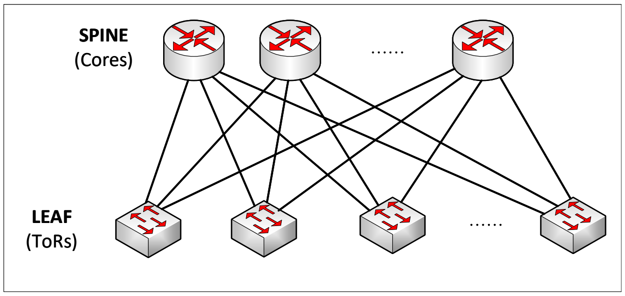


Fig. 8: Leaf-Spine Topology

TABLE III: Simulation Parameters for Leaf-Spine

Parameter	Value
Number of leafs (ToRs)	128
Number of spines	8
Number of data channels per leaf	8
Number of data channels per spine	128
Data link rate	10Gbps
Packet processing time (T_{proc})	0.1 s
Buffer Size per switch	32MB
Load-balancing technique	RPS [27]

The other inputs (e.g., TCP/IP, traffic load) are the same as in AgileDCN and both architectures use FIFO queue management.

Note that for Leaf-Spine we use Random Packet Spraying (RPS) [27] for load balancing although the equal-cost multi-path (ECMP) has been used as the *de-facto* routing algorithm in these data centers, especially in Clos-based architectures like Fat-Tree. RPS uses a Round Robin approach to uniformly distribute packets of a flow across all possible equal-cost paths, while ECMP enforces packets of a flow to stick with one of those paths. In fact we implemented both techniques and since simulations showed that ECMP is by far inferior to RPS, we only show the performance of Leaf-Spine architecture using the superior RPS.

The results are shown in Figs. 9-10. In the these figures, we consider 100% offered load network load to occur when the busiest ToRs are sending at their full (inter- and intra-) outgoing channels capacities. Fig. 9 reveals that leaf-spine can achieve lower FCTs than AgileDCN's at low loads, however it is by far inferior to AgileDCN for high loads. Specifically, for loads from 40% to 70%, Leaf-Spine suffers increasingly high FCTs, as opposed to AgileDCN, and the difference is greatest at high load (70% average load). On average, AgileDCN can reduce FCT by up to 90% compared to Leaf-Spine, at high loads. Besides, the graph on the right shows the same goodputs (or application throughputs) achieved for both architectures, which are linear to the offered load except for the load of 70%. This indicates that the above-mentioned FCT's comparison are fairly done given that both architectures convey the same amount of traffic in the course of simulations.

The above results can be explained as follows. Leaf-Spine, as a packet switching based electronic architecture, processes every incoming packet by en/dequeuing it according to the channel's availability. That is very efficient for low loads, as no overhead is required other than the processing time for each packet header. However, for high loads packets arrive so quickly that not all of them are transmitted at their arrivals, they need to be queued in the switch buffers which are of limited capacity. (Note that we set buffer size to be 32 MB for all the Leaf-Spine switches (typical for a commercial switch). Our AgileDCN ToRs use a buffer size of just 16 MB). Packets that arrive when buffers are full will be dropped per FIFO policy, and, by means of TCP, re-transmitted at a later time. High flow completion times can also occur well before switches are spine switches are overloaded due to the compounding effects of TCP re-transmissions increasing effective offered load when packet timeouts begin to occur. In the simulations, we observed that Leaf-Spine suffers from higher packet drop rates than AgileDCN, especially at high loads. Thus Leaf-Spine suffers from higher FCTs. In contrast, AgileDCN aggregates packets into VoQs and transmits them on a per burst basis. On one hand, even with some aggregation overhead, this greatly reduces the processing time of packet headers (once for a bundle of packets). On the other hand, the VoQs help absorb the explosion of packet arrivals for high loads, which avoid massive packet drops and hence reduce FCTs.

Finally, to better understand the behavior of the two architectures, we show the scatter plots of FCTs versus flow sizes of both AgileDCN and Leaf-Spine for 2 typical loads: 40% and 70% (Fig. 10). In general, AgileDCN forms nearly linear shapes for FCTs with respect to their flow sizes, which is different from Leaf-Spine. For the lower load of 40% (Fig. 10a and Fig. 10b), while in AgileDCN small flows (which are much more frequent than big flows) are always completed sooner than the bigger ones, that is not always the case for Leaf-Spine where many small flows are significantly delayed. That is, as mentioned above, caused by the TCP retransmission mechanism interacting with the high dropped packet rate in Leaf-Spine. Accordingly, for Leaf-Spine even with moderate loads, many small flows would be dropped and retransmitted, which leads to FCTs that are disproportionately higher than the flow size. Meanwhile, AgileDCN makes better use of its VoQs, saving small flows from being dropped and retransmitted. For higher loads, as shown in Fig. 10c and Fig. 10d, Leaf-Spine performs relatively even worse, and AgileDCN also begins to suffer. In short, the results show that AgileDCN is more resilient to high loads than Leaf-Spine.

V. Conclusion

A new DCN network architecture known as AgileDCN has been evaluated. It is based on fast tunable lasers and AWGs for routing of intra-cluster traffic, with intercluster traffic being accomodated by optical space switches. A centralised system controller makes routing decisions, schedules the traffic and configures the network nodes accordingly. The results show that the AgileDCN architecture can provide Flow Completion Times up to 90% less than a comparable leaf-spine topology when the network is heavily loaded. This low latency is critical to evolving real-time applications that are becoming increasingly prevalent.

Acknowledgment

The authors would like to acknowledge the support of the Science Foundation of Ireland (SFI) under the US-Ireland R&D Partnership Programme (SFI project code 15/US-C2C/I3132), the SFI Centres CONNECT and IPIC. This work was supported in part by the National Science Foundation (NSF) under grants CNS-1827923, CNS-1650669, CNS-1737453, and EEC-0812072.

References

[1] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, "OSA: An Optical Switching Architecture for Data Center Networks With Unprecedented Flexibility," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 498–511, April 2014.

[2] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating Microsecond Circuit Switching into the Data Center," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 447–458, Aug. 2013.

[3] 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," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 339–350, 2010.

[4] G. M. Saridis, S. Peng, Y. Yan, A. Aguado, B. Guo, M. Arslan, C. Jackson, W. Miao, N. Calabretta, F. Agraz, S. Spadaro, G. Bernini, N. Ciulli, G. Zervas, R. Nejabati, and D. Simeonidou, "Lightness: A Function-Virtualizable Software Defined Data Center Network With All-Optical Circuit/Packet Switching," *J. Lightwave Technol.*, vol. 34, no. 7, pp. 1618–1627, Apr 2016.

[5] R. Proietti, Z. Cao, C. J. Nitta, Y. Li, and S. J. B. Yoo, "A Scalable, Low-Latency, High-Throughput, Optical Interconnect Architecture Based on Arrayed Waveguide Grating Routers," *Journal of Lightwave Technology*, vol. 33, no. 4, pp. 911–920, Feb 2015.

[6] J. Wang, C. McArdle, and L. P. Barry, "Energy-efficient optical HPC and datacenter networks using optimized wavelength channel allocation," *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2015 International Symposium on*, pp. 1–8, 2015.

[7] M. Imran, M. Collier, P. Landais, and K. Katrinis, "Software-defined optical burst switching for HPC and cloud computing data centers," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 8, pp. 610–620, Aug 2016.

[8] P. Bakopoulos, K. Christodoulopoulos, G. Landi, M. Aziz, E. Zahavi, D. Gallico, R. Pitwon, K. Tokas, I. Patronas, M. Capitani, C. Spatharakis, K. Yiannopoulos, K. Wang, K. Kontodimas, I. Lazarou, P. Wieder, D. I. Reisis, E. M. Varvarigos, M. Biancani, and H. Avramopoulos, "NEPHELE: an end-to-end scalable and dynamically reconfigurable optical architecture for application-aware SDN cloud data centers," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 178–188, 2018.

[9] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer, "Firefly: A reconfigurable wireless data center fabric using free-space optics," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 319–330, 2014.

[10] M. Ghobadi, R. Mahajan, A. Phanishayee, N. Devanur, J. Kulkarni, G. Ranade, P.-A. Blanche, H. Rastegarfar, M. Glick, and D. Kilper, "Projector: Agile reconfigurable data center interconnect," in *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, 2016, pp. 216–229.

[11] C. Liu, M. Xu, and S. Subramaniam, "A reconfigurable high-performance optical data center architecture," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[12] S. Sakano, T. Tsuchiya, M. Suzuki, S. Kitajima, and N. Chinone, "Tunable DFB laser with a striped thin-film heater," *IEEE Photonics Technology Letters*, vol. 4, no. 4, pp. 321–323, April 1992.

[13] K. Grobe, M. H. Eiselt, S. Pachnicke, and J. P. Elbers, "Access Networks Based on Tunable Lasers," *Journal of Lightwave Technology*, vol. 32, no. 16, pp. 2815–2823, Aug 2014.

[14] J. Buus, M. C. Amann, and D. J. Blumenthal, *Tunable Laser Diodes and Related Optical Sources*. SPIE Press and Wiley-IEEE Press, 2005.

[15] A. Wonfor, Q. Cheng, R. V. Penty, and I. H. White, "Integrated optical switches and short pulse generation using a generic integration platform," in *2016 IEEE Photonics Conference (IPC)*, Oct 2016.

[16] J. Kim, C. J. Nuzman, B. Kumar, D. F. Lieuwen, J. S. Kraus, A. Weiss, C. P. Lichtenwalner, A. R. Papazian, R. E. Frahm, N. R. Basavanhally, D. A. Ramsey, V. A. Aksyuk, F. Pardo, M. E. Simon, V. Lifton, H. B. Chan, M. Haueis, A. Gasparyan, H. R. Shea, S. Arney, C. A. Bolle, P. R. Kolodner, R. Ryf, D. T. Neilson, and J. V. Gates, "1100 x 1100 port MEMS-based optical crossconnect with 4-dB maximum loss," *IEEE Photonics Technology Letters*, vol. 15, no. 11, pp. 1537–1539, Nov 2003.

[17] T. J. Seok, N. Quack, S. Han, W. Zhang, R. S. Muller, and M. C. Wu, "64×64 Low-loss and broadband digital silicon photonic MEMS switches," in *2015 European Conference on Optical Communication (ECOC)*, Sept 2015.

[18] N. Dupuis, B. G. Lee, A. V. Rylyakov, D. M. Kuchta, C. W. Baks, J. S. Orcutt, D. M. Gill, W. M. J. Green, and C. L. Schow, "Design and Fabrication of Low-Insertion-Loss and Low-Crosstalk Broadband 2×2 Mach-Zehnder Silicon Photonic Switches," *Journal of Lightwave Technology*, vol. 33, no. 17, pp. 3597–3606, Sept 2015.

[19] K. A. McGreer, "Arrayed waveguide gratings for wavelength routing," *IEEE Communications Magazine*, vol. 36, no. 12, pp. 62–68, Dec 1998.

[20] T. Okoshi and K. Kikuchi, *Coherent Optical Fiber Communications*. Springer, 1988.

[21] S. J. Savory, "Digital filters for coherent optical receivers," *Opt. Express*, vol. 16, no. 2, pp. 804–817, Jan 2008.

[22] D. D. Le, J. Wang, L. P. Barry, and C. McArdle, "Agiledc: A novel optical data center network architecture," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, March 2018, pp. 567–573.

[23] "OMNeT++." [Online]. Available: https://www.omnetpp.org/

[24] P. Wette and H. Karl, "DCT2Gen," *Comput. Commun.*, vol. 80, no. C, pp. 45–58, Apr. 2016.

[25] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pp. 202–208, 2009.

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

[27] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella, "On the impact of packet spraying in data center networks," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 2130–2138.