PODCA: A Passive Optical Data Center Network Architecture

Maotong Xu, Chong Liu, and Suresh Subramaniam

***Abstract*—Optical interconnects have gained great attention recently as a promising solution offering high throughput, low latency, scalability, and reduced energy consumption compared to electrical interconnects. However, some active optical components, such as tunable wavelength converters and micro-electro-mechanical systems (MEMS) switches, suffer from high cost or slow reconfiguration times, and have been roadblocks to the realization of all-optical interconnects. In this paper, we propose three versions of a passive optical data center network architecture (PODCA), depending on the size of the network. Our key device is the arrayed waveguide grating router (AWGR), a passive device that can achieve contention resolution in the wavelength domain. In our architectures, optical signals are transmitted from fast tunable transmitters; pass through couplers, AWGRs, and demultiplexers; and are received by wide-band receivers. Our architecture can scale to over 2 million servers. Simulation results indicate that PODCA exhibits lower latency and higher throughput even at high-input loads compared with electrical data center networks such as Fat-Tree and Flattened Butterfly, and comparable performance with other optical interconnects such as DOS and Petabit, but at much lower cost and power consumption. The packet latency of PODCA in our simulation experiments is below 19 μs, and the throughput is 100%. Furthermore, we compare the power consumption and capital expenditure (CapEx) cost of PODCA with the other four architectures. Results show that PODCA can save at least 75% on power consumption and 50% on CapEx.**

***Index Terms*—Arrayed waveguide grating (AWG); Data center network; Optical networks; Passive.**

I. Introduction

Power consumption, latency, and scalability are critical factors in designing data center network (DCN) architectures. Power consumption of data centers will reach 140 billion kilowatt-hours annually by 2020, and it will cost $13 billion annually [1]. On the other hand, interactive applications, e.g., web searches, social networks, and stock exchanges, require low network latency. The acceptable latency range of a stock exchange is 5–100 ms [2]. Furthermore, as data volume increases, the size of data centers scales up tremendously. Microsoft owns over 1 million servers, and its Chicago data center alone is estimated to contain over 250,000 servers [3]. Some conventional electrical data center networks (e.g., Fat-Tree, VL2, Flattened Butterfly, Bcube [4–7]) are built using a multi-layer approach, with a large number of identical switches at the bottom level to connect with the end nodes (i.e., servers or racks), and a few expensive and large-radix switches located at the upper layers to aggregate and distribute the traffic.

Considering reduced power consumption, low latency, and high scalability, optical networking is a promising solution for current data centers [8,9]. Optical networks are usually based on optical circuit switching that uses large switches, e.g., micro-electro-mechanical systems (MEMS) switches and arrayed waveguide grating routers (AWGRs). An optical MEMS switch is a power-driven reconfigurable switch whose reconfiguration time is on the order of a few milliseconds and is therefore not suited for fast packet switching in DCN applications [10–12]. Compared to MEMS, AWGR is a passive optical device that does not support reconfiguration and can achieve contention resolution in the wavelength domain. The cyclic routing characteristic of the AWGR allows different inputs to reach the same output simultaneously by using different wavelengths. Previous AWGR-based architectures, e.g., DOS [13] and Petabit [14], employ tunable wavelength converters (TWC), which are power-hungry devices. Moreover, TWC significantly increases the total capital expenditure (CapEx) of the architectures.

Our solution is to employ passive optical devices, e.g., AWGR, coupler, and demultiplexer, to reduce power consumption, while achieving low packet latency and high throughput. Moreover, our architecture is highly scalable and can accommodate more than 2 million servers. The main contributions of our work are as follows:

- We propose three different-sized—small, medium, and large—passive optical architectures, and present the wavelength assignment and packet transmission algorithm for each.
- We obtain the packet latency and network throughput of the three architectures through simulations. Simulation results show that packet latencies of under 19 μs and a throughput of 100% are achievable. PODCA performs well for a number of different architectural parameters, such as the number of tunable transmitters per rack,[1] the

Manuscript received November 20, 2017; revised February 21, 2018; accepted February 22, 2018; published March 29, 2018 (Doc. ID 313735).

The authors are with the Department of Electrical and Computer Engineering, George Washington University, Washington, DC 20052, USA (e-mail: htfy8927@gwu.edu).

https://doi.org/10.1364/JOCN.10.000409

[1]The term "rack" and "ToR" (for top-of-rack) are used interchangeably in the paper.

type of architecture, as well as packet size. In comparison, the latency and throughput of PODCA are better than those of Fat-Tree and Flattened Butterfly (FBFLY), while achieving similar performance as DOS and Petabit.

- We compare the power consumption and CapEx of our architectures with Fat-Tree, FBFLY, DOS and Petabit. Results show that our architectures save at least 75% on power consumption and 50% on CapEx.

The new contributions over [15] include:

- We show that packet selection in the three architectures can be solved optimally when the number of tunable transceivers on each rack is one.
- We show that our greedy algorithms can achieve comparable throughput to the optimal throughput obtained from an ILP solver for small-sized architectures.
- We compare our architecture with four other architectures (Fat-Tree, FBFLY, DOS, and Petabit) in terms of packet latency.
- We compare our architectures with four other architectures (Fat-Tree, FBFLY, DOS, and Petabit), in terms of power consumption and costs.

The rest of this paper is organized as follows: Section II reviews the related work. Section III presents the system model and formulates a problem of optimal packet transmission. The three versions of PODCA and their corresponding wavelength assignment and packet transmission algorithms are presented in Section IV. Section V presents the performance evaluation of the architectures and comparison with previously proposed architectures. In Section VI, we compare the proposed architectures with the previous architectures in terms of power consumption and CapEx. Finally, we conclude the paper in Section VII.

II. Related Work

The data center is home to the computation, storage, and applications necessary to support different fields. Proper planning of the data center network design is critical, and performance, efficiency, and scalability need to be carefully considered. Conventional electrical data center networks are built in a multistage manner to solve the problem. For example, Fat-Tree [4] makes cheap off-the-shelf Ethernet switches the basis for large-scale data center networks. The FBFLY [5] gives lower hop count than a folded Clos and better path diversity than a conventional butterfly. It is approximately half the cost of a folded Clos network of identical capacity on load-balanced traffic. However, power consumption, latency, and throughput on large clusters are critical challenges for electrical switches.

Similar to PODCA, some existing optical DCNs are also commonly based on AWGR. DOS [13] and Petabit [14] represent the state-of-the-art in terms of AWGR optical switching in high-performance data center interconnects. The DOS topology consists of an array of (TWCs), an AWGR, and a loopback shared buffer. Each node can access any other node through the AWGR by configuring the transmitting wavelength of the TWC. The system is configured by the control plane that controls the TWC and the label extractors (LEs). The control plane is used for contention resolution and TWC tuning. The scalability of DOS depends on the scalability of the AWGR and the tuning range of the TWC. S. Cheung *et al.* [16] have presented an AWGR with up to 512 ports covering all wavelengths in the C, S, and L bands. The measured spectrum is across 512 channels (wavelengths), and the channel spacing is 25 GHz. Thus, the DOS architecture can be used to connect up to 512 nodes (assuming that each node is used as a top-of-rack (ToR) switch or a server switch). DOS can provide low packet delays even under high input loads as packets have to traverse through only a single optical switch. However, DOS is not scalable to large DCNs (thousands of racks), and is expensive because of TWCs.

Petabit [14] adopts a three-stage Clos network, where each stage consists of an array of AWGRs that are used for passive routing of packets. In the first stage, the tunable lasers are used to route the packets through the AWGRs, while in the second and third stages, TWCs are used to convert the wavelength as needed and route the packets to their destinations. Different from DOS, Petabit does not use any buffers to avoid the power-hungry electrical-to-optical (E-O) and optical-to-electrical (O-E) conversion.

However, DOS and Petabit optical switches have some limitations. The main drawback of DOS, which is based on electrical buffers for congestion management using power-hungry E-O and O-E converters, is the increased overall power consumption. Furthermore, the DOS architecture uses tunable wavelength converters, which are quite expensive compared to the commodity optical transceivers used in current switches. Similarly, Petabit also requires many TWCs to inter-connect each stage of AWGRs.

Opsquare [17] proposes a low-power and low-cost architecture consisting of intra- and inter-cluster networks. The main drawback of Opsquare is that the design of the architecture is unchangeable with the number of servers. Also, inter-cluster switches of Opsquare only connect to a rack in each cluster, which might result in imbalance of ToR loads.

As mentioned earlier, PODCA employs AWGR, which is a passive optical device that does not require reconfiguration and can achieve packet contention resolution in the wavelength domain based on its cyclic routing feature. The scalability of PODCA depends on the scalability of AWGR and the capability of tunable transmitters. AWGR serves as a promising passive optical device that has been employed in many large-scale data centers [15,13,14,18,19] or multi-processor system designs [20,21]. According to the cyclic characteristic of AWGR, it can not only achieve concurrent contention-free optical switching, but also allows any output to receive multiple concurrent signals that reside on separate and distinct wavelengths. A 270×270 prototype AWGR switch was fabricated using PLC technologies, and its technical feasibility was verified through transmission experiments; this can be

adapted to expand to exceed 1000 × 1000 [21]. Cheung *et al.* [16] have demonstrated a 512 × 512 arrayed waveguide grating router (AWGR) with a channel spacing of 25 GHz. The dimensions of the AWGR are 16 mm × 11 mm, and it is fabricated on a 250 nm silicon-on-insulator platform. The measured channel crosstalk is approximately −4 dB without any compensation for the phase errors in the arrayed waveguides. Because practical optical random access memory is unavailable, PODCA employs fast and wide-range tunable lasers as transmitters to achieve arbitrary connectivity. Matsuo *et al.* [22] have introduced a tunable laser device that is capable of covering 34 channels with a 100 GHz spacing. The switching latency to any wavelength is less than 8 ns. Besides, a novel ultra-fast tunable laser concept device that employs an active interleaved rear mirror has been reported with a switching time of less than 2 ns [23]. Even though these devices are at the experimental stage now, these tuning times allow us to develop novel architectures that take advantage of such emerging devices for packet applications in DCNs. In this paper, we assume that the AWGR can scale up to 512 ports and the tuning time of tunable transmitters is 8 ns. As will see later, small changes in these parameters will not have a significant impact on the architecture and its performance.

III. System Model

Our architectures are hierarchical—a group of racks form a cluster and the entire DCN is a group of clusters. Suppose each cluster has M racks, i.e., $\lceil \frac{S}{P} \rceil = M$, where S is the total number of racks, and P is the number of ports of the AWGR. We denote W as the number of available wavelengths and let $W = P \cdot F$, where $F \geq 1$ is an integer. Wavelength w is denoted as λ_w, where w is the wavelength index. The AWGR routes wavelengths from an input port to a specific output port in a cyclic way, i.e., λ_c is routed from input port i to output port [8]

$$[(i + w - 2) \mod P] + 1, \qquad 1 \leq i \leq P, \qquad 1 \leq w \leq W. \tag{1}$$

Each rack has a ToR switch and one or more fast tunable transmitters (tunable to any wavelength) and fixed wide-band receivers. A wide-band receiver is a simple photodetector receiver that can receive a signal on *any* wavelength as long as only one signal is directed to it. Let N be the number of tunable transmitters or wide-band receivers on each ToR, and let $N = \frac{W}{l \cdot M}$, where $l \geq 1$ is an integer. Denote $\mathcal{T}^t_{i,j}$ as the tth tunable transmitter on the jth ToR connecting to the ith input port of the AWGR. Also, we denote $\mathcal{R}^r_{m,n}$ and the rth fixed-band receiver on the nth ToR connecting to the mth port of the AWGR. Note that $\mathcal{T}_{i,j}$ and $\mathcal{R}_{m,n}$ are the corresponding transmission side rack and reception side rack, respectively. We denote $\mathcal{V}^{i,j}_{m,n}$ as the number of packets waiting for transmission from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$. Also, we denote $\mathcal{I}^{t,i,j}_{r,m,n}$ as indicator function for a packet selected for transmission (by our algorithm) from $\mathcal{T}^t_{i,j}$ to $\mathcal{R}^r_{m,n}$. A full list of notations is given in Table I.

TABLE I
Notation Table

Notation	Corresponding Meaning
P	Number of ports of AWGR
S	Number of total racks
W	Number of wavelengths available
M	Number of racks in each cluster
λ_w	Wavelength w
$\mathcal{T}_{i,j}$	jth transmission side ToR connecting to ith port of AWGR
$\mathcal{R}_{m,n}$	nth reception side ToR connecting to mth port of AWGR
$\mathcal{T}^t_{i,j}$	tth transmitter on the jth transmission Side ToR connecting to the ith port of AWGR
$\mathcal{R}^r_{m,n}$	rth receiver on the nth reception side ToR connecting to the mth port of AWGR

Packets arriving at a ToR and destined to another ToR are placed in B shared buffers. A packet with destination rack $\mathcal{R}_{m,n}$, is stored in the $[(m \cdot M + n)\%B + 1]$th buffer. In this paper, we assume a central controller that schedules packet transmissions; while this might introduce scalability issues for large DCNs, it could be replaced by distributed scheduling with some loss in performance (reserved for future work). The system is time slotted, and a time slot includes both the packet transmission time (all packets are assumed to be the same size) and transmitter tuning time. In each time slot, the controller schedules transmissions for the next time slot.

In each time slot, PODCA maximizes the number of packets for transmission [Eq. (2)], and follows four constraints, i.e., those in Eqs. (3), (4), (5), and (6). The constraint in Eq. (3) means that a receiver can receive at most one packet at a time. The constraint in Eq. (4) means that a transmitter can transmit at most one packet at a time. The constraints in Eqs. (3) and (4) also guarantee that there are at most N packets transmitted and received by any rack, respectively. Due to the cyclic wavelength routing property of the AWGR, at most F packets can be transmitted from an input port of the AWGR to an output port of the AWGR in a time slot (constraint in Eq. 5). The constraint in Eq. (6) ensures that the number of packets selected from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$ is not more than the number of packets waiting for transmission from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$.

Maximize:

$$\sum_{i,j,t,m,n,r} \mathcal{I}^{t,i,j}_{r,m,n}, \tag{2}$$

s.t.:

$$\sum_{i,j,t,m,n} \mathcal{I}^{t,i,j}_{r,m,n} \leq 1, \tag{3}$$

$$\sum_{i,j,m,n,r} \mathcal{I}^{t,i,j}_{r,m,n} \leq 1, \tag{4}$$

$$\sum_{j,t,n,r} \mathcal{I}^{t,i,j}_{r,m,n} \leq F, \tag{5}$$

$$\sum_{t,r} \mathcal{I}_{r,m,n}^{t,i,j} \le \mathcal{V}_{m,n}^{i,j}. \tag{6}$$

IV. PODCA Architecture and Algorithms

In this section, we present the three versions of PODCA depending on the size of the DCN: (a) PODCA-S (small DCN): $S \le P$; P is typically around 50, but could be as large as 512; (b) PODCA-M (medium DCN): $P \le S \le W$, and (c) PODCA-L (large DCN): $W < S$.

In the following subsections, we present the architecture and the packet transmission algorithm for each of the three PODCA versions.

A. PODCA-S

In this case, $S \le P$, and each cluster is just a single rack. We connect each rack to a port of the AWGR. Let W be an integral multiple of N, i.e., $W = \alpha \cdot N$, where $\alpha \ge 1$ is an integer. The architecture is shown in Fig. 1. Note that the signal from an output port of a demultiplexer can be either a fixed wavelength or a fixed range of wavelengths. Since each output port of a demultiplexer connects to a wide-band receiver and the wide-band receiver can only receive one packet at a time, the central controller needs to guarantee that only one wavelength from the output port of a demultiplexer carries data at any given time.

Since the rack considered first has higher priority for packet scheduling, we use a round-robin method for selecting the starting rack and schedule packets one by one starting from that rack. In other words, in the τth time slot, PODCA-S selects packets starting from the rack $\mathcal{T}_{i,1}$, where i is

$$i = \frac{\tau \% S + 1}{M} + 1. \tag{7}$$

We now show how packet transmissions are scheduled and the wavelength on which a scheduled packet is transmitted. Suppose a packet waiting for transmission is from $\mathcal{T}_{i,1}$ to $\mathcal{R}_{m,1}$. If the number of packets scheduled on $\mathcal{T}_{i,1}$ (c_i) and on $\mathcal{R}_{m,1}$ (c_m) are both not larger than N, and the number of packets scheduled from $\mathcal{T}_{i,1}$ to $\mathcal{R}_{m,1}$ ($c_{i,m}$) is not larger than F, the packet is scheduled for transmission from $\mathcal{T}_{i,1}$ to $\mathcal{R}_{m,1}$. By deriving the modular inverse of Eq. (1), we know that wavelength assignment is determined by the input port number and output port number of the AWGR. If $i \le m$, the tunable transmitter needs to tune to one of the wavelengths shown in Eq. (8) to successfully deliver the packet,

$$m - i + 1 + f \cdot P, \quad \forall f \in [0, F-1]. \tag{8}$$

If $i > m$, the tunable transmitter needs to tune to one of the wavelengths shown in Eq. (9) to successfully deliver the packet,

$$P + m - i + 1 + f \cdot P, \quad \forall f \in [0, F-1]. \tag{9}$$

Here, $f \le F - 1$ because the maximum number of available wavelengths is $W = F \cdot P$. The coupler combines different wavelengths from different tunable transmitters and outputs the combined signal to the AWGR. An output signal from the AWGR is evenly demultiplexed N ways, i.e., the first α wavelengths are demultiplexed to output port 1, the next α wavelengths to port 2, and so on. The pseudocode of the algorithm is shown in Algorithm 1.

Algorithm 1: Packet Scheduling for PODCA-S

```
γ = τ%S + 1
i = (int)γ/M + 1
for each shared buffer of T_{i,1} do
    pkt at the head of the buffer is from T_{i,1} to R_{m,1}
    if c_i < N and c_m < N and f_{i,m} < F then
        pkt is selected for transmission by using transmitter c_i and receiver c_m
        if i ≤ m then
            tune to wavelength: m − i + 1 + f_{i,m} · P
        else
            tune to wavelength: P + m − i + 1 + f_{i,m} · P
        end if
        c_i++; c_m++; f_{i,m}++
    end if
    γ = γ%S + 1
end for
```

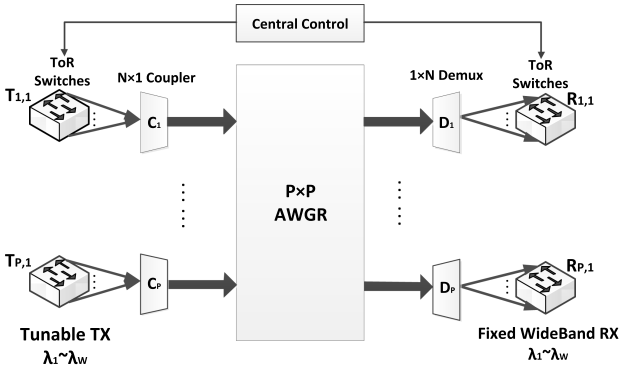

Fig. 1. Architecture of PODCA-S.

B. PODCA-M

The PODCA-M architecture is shown in Fig. 2. In this case, $P \le S \le W$. We connect $M(>1)$ racks to each AWGR port. An $(N \cdot M) \times 1$ coupler connects all the N transmitters on each of the M racks to an input port of the AWGR. A $1 \times (N \cdot M)$ demultiplexer connects an output port of the AWGR to M racks. As the number of inputs to the coupler increases (due to large clusters or more transmitters per rack), power losses may become too large, necessitating an amplifier (usually an erbium doped fiber amplifier, EDFA) between the coupler and input port of AWGR (shown within the dashed box in Fig. 2). Let W be an integer multiple of the number of tunable

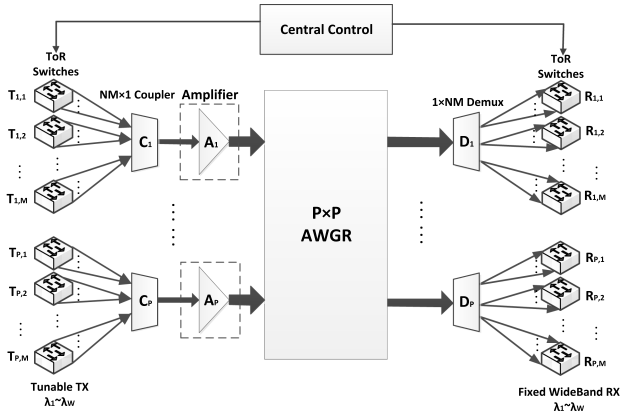

Fig. 2. Architecture of PODCA-M.

transmitters connecting to one demultiplexer, i.e., $W = \beta \cdot N \cdot M$, where $\beta \geq 1$ is an integer. The receivable wavelength range for the nth rack in a cluster is

$$\left[(n-1) \cdot \frac{W}{M} + 1, n \cdot \frac{W}{M}\right], \tag{10}$$

and we denote the above wavelength set as λ_n^s. Note that $N \cdot M \cdot P \leq W = F \cdot P$, and so, $N \cdot M \leq F$. From Eq. (10), we can see that the number of receivable wavelengths for any ToR is $\frac{W}{M}$. Also, $\frac{W}{M} \geq \frac{W}{F} = P$. Thus, the number of receivable wavelengths of a ToR is at least P. Also, based on Eq. (10), the P wavelengths are contiguous, which means that a ToR can receive packets from every port of the AWGR.

As in Algorithm 1, PODCA-M uses round-robin for selecting the starting rack and schedules packets one by one starting from that selected rack. So, in the τth time slot, PODCA-M selects packets starting from the rack $\mathcal{T}_{i,j}$, where i is shown in Eq. (7), and j is

$$j = (\tau \% S + 1) \% M + 1. \tag{11}$$

Here is how the packet scheduling and wavelength assignment works. Suppose there is a packet waiting for transmission from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$. If the number of packets scheduled on $\mathcal{T}_{i,j}$ ($c_{i,j}$) and on $\mathcal{R}_{m,n}$ ($c_{m,n}$) are both not more than N, and the number of packets scheduled from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$ ($f_{i,m}$) is not more than F, the packet is scheduled for transmission from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$. We tune the transmitters based on Eqs. (8) and (9). Additionally, we need to guarantee that the tuned wavelength belongs to λ_n^s. The coupler combines distinct wavelengths from transmitters and outputs the combined signal to the AWGR. Note that if there are multiple available wavelengths, we use the round-robin method to choose one of them. The pseudocode of the algorithm is shown in Algorithm 2.

Algorithm 2: Packet Scheduling for PODCA-M

$\gamma = \tau \% S + 1$
$i = (\text{int})\gamma / M + 1$
$j = \gamma \% M + 1$
for each shared buffer of $\mathcal{T}_{i,j}$ **do**
  pkt at the head of the buffer is from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$
  if $c_{i,j} < N$ and $c_{m,n} < N$ and $f_{i,m} < F$ **then**
    pkt is selected for transmission by using transmitter $c_{i,j}$ and receiver $c_{m,n}$
    if $i \leq m$ **then**
      tune to wavelength $\in \lambda_n^s \cap \{m - i + 1 + f \cdot P, \forall f \leq F - 1\}$
    else
      tune to wavelength $\in \lambda_n^s \cap \{P + m - i + 1 + f \cdot P, \forall f \leq F - 1\}$
    end if
    $c_{i,j}++; c_{m,n}++; f_{i,m}++$
  end if
  $\gamma = \gamma \% S + 1$
end for

C. PODCA-L

In this case, $W < S$. We use two different-sized AWGRs, i.e., an $M \times M$ AWGR for intra-cluster transmission, and a $P \times P$ AWGR for inter-cluster transmission. In each intra-cluster network, we connect M tunable transmitters and M wide-band receivers to an $M \times M$ AWGR. For an inter-cluster network, we connect M tunable transmitters, belonging to the same cluster, to an $M \times 1$ coupler, and connect the output of the coupler to the $P \times P$ AWGR. Each output port of the $P \times P$ AWGR is connected to an input port of the $1 \times M$ demultiplexer, and each output port of the demultiplexer connects to the M racks within a cluster. The architecture is shown in Fig. 3.

Of course, coupler power losses may necessitate amplification if the cluster size becomes large as in PODCA-M. Nevertheless, PODCA-L is highly scalable, and it can easily accommodate up to more than 2 million servers (assuming 48 servers per rack, 100-port AWGRs within the cluster, and a 512-port AWGR interconnecting clusters). Since no reconfigurable devices (except tunable transmitters) are used in the architecture, we can achieve huge power savings. The slight drawback, however, is that some packets need two hops to arrive at their destinations (an inter-cluster transmission and an intra-cluster one).

We illustrate packet transmission with a small concrete example that helps understanding before presenting the

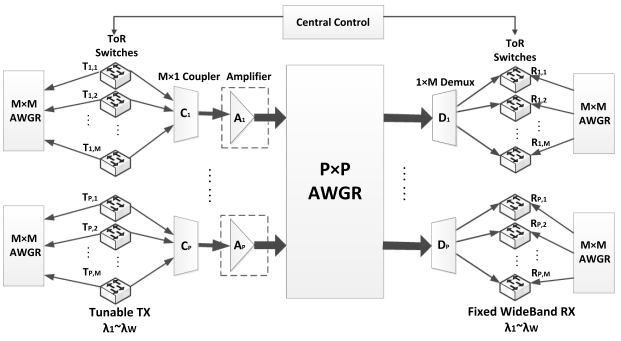

Fig. 3. Architecture of PODCA-L.

general algorithm. Suppose $P = 2$ and $W = 8$. There are 8 racks within each cluster, i.e., $M = 8$. On each rack, there is one tunable transmitter and one wide-band receiver for intra-cluster communication. Also, there is one tunable transmitter and one wide-band receiver for inter-cluster communication.

Suppose there is a packet for transmission from $\mathcal{T}^1_{1,1}$ to $\mathcal{R}^1_{2,1}$. The only wavelength $\mathcal{R}^1_{2,1}$ can receive, within inter-cluster transmission, is λ_1. However, based on the routing characteristics of AWGR, the receivable wavelengths from the first input port of the AWGR to the second output port of the AWGR can only be λ_2, λ_4, λ_6, and λ_8. So, λ_1 transmitted from $\mathcal{T}^1_{1,1}$ cannot arrive at the second output port of the AWGR in a single hop. Thus, a two-hop transmission is needed. We first choose one wavelength from $\{\lambda_2, \lambda_4, \lambda_6, \lambda_8\}$; suppose we choose λ_2. The packet is first transmitted to $\mathcal{R}_{2,2}$ by using λ_2, then in the next time slot, $\mathcal{R}_{2,2}$ can transmit the packet to $\mathcal{R}_{2,1}$ by using an intra-cluster transmission.

Algorithm 3: Packet Scheduling for PODCA-L

$\gamma = \tau \% S + 1$
$i = (\text{int})\gamma/M + 1$
$j = \gamma \% M + 1$
for each shared buffer of $\mathcal{T}_{i,j}$ **do**
  pkt at the head of the buffer is from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$
  if $i = m$ **then**
    if $c_j < N_{\text{intra}}$ and $c_n < N_{\text{intra}}$ and $f_{j,n} < F_{\text{intra}}$ **then**
      pkt is selected for transmission by using transmitter c_j and receiver c_n
      if $n \geq j$ **then**
        tune to $\lambda_w \in n - j + 1 + f_{j,n} \cdot M$
      else
        tune to $\lambda_w \in M + n - j + 1 + f_{j,n} \cdot M$
      end if
      $c_j + +; c_n + +; f_{j,n} + +$
      continue
    end if
  end if
  if $i \leq m$ **then**
    if $\lambda^s_n \cap \lambda^s_{i\leq m} \neq \emptyset$ **then**
      tune to $\lambda_w \in \lambda^s_n \cap \lambda^s_{i\leq m}$
    else
      tune to $\lambda_w \in \lambda^s_{i\leq m}$
    end if
  else
    if $\lambda^s_n \cap \lambda^s_{i>m} \neq \emptyset$ **then**
      tune to $\lambda_w \in \lambda^s_n \cap \lambda^s_{i>m}$
    else
      tune to $\lambda_w \in \lambda^s_{i>m}$
    end if
  end if
  if $c_{i,j} < N_{\text{inter}}$ and $c_{m,\lfloor\frac{w\cdot M}{W}\rfloor} < N_{\text{inter}}$ and $f_{i,m} < F_{\text{inter}}$ **then**
    pkt is selected for transmission by using transmitter $c_{i,j}$ and receiver $c_{m,\lfloor\frac{w\cdot M}{W}\rfloor}$
    $c_{i,j} + +; c_{m,\lfloor\frac{w\cdot M}{W}\rfloor} + +; f_{i,m} + +$
  end if
  $\gamma = \gamma \% S + 1$
end for

We now present the general packet transmission algorithm. Denote the number of tunable transmitters or wide-band receivers for intra-cluster communication as N_{intra}, and for inter-cluster communication as N_{inter}. Also, denote $\frac{W}{M}$ as F_{intra} and $\frac{W}{P}$ as F_{inter}. The central controller schedules both intra-cluster and inter-cluster transmissions based on *scheduling constraints*. Suppose a packet waiting for transmission is from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$. If $i = m$, we use intra-cluster transmission. If the number of packets scheduled on $\mathcal{T}_{i,j}$ (c_j) and on $\mathcal{R}_{m,n}$ (c_n) are both not more than N_{intra}, and the number of packets scheduled from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$ ($f_{j,n}$) is not more than F_{intra}, the packet is scheduled for transmission from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$. If $n \geq j$, the transmitter tunes to wavelength

$$n - j + 1 + f \cdot M, \quad \forall f \in [0, F_{\text{intra}} - 1]. \tag{12}$$

If $n < j$, the transmitter tunes to wavelength

$$M + n - j + 1 + f \cdot M, \quad \forall f \in [0, F_{\text{intra}} - 1]. \tag{13}$$

If $i \neq m$, we need inter-cluster transmission. If the number of packets scheduled on $\mathcal{T}_{i,j}$ ($c_{i,j}$) and on $\mathcal{R}_{m,n}$ ($c_{m,n}$) are both not more than N_{inter}, and the number of packets scheduled from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$ ($f_{i,m}$) is not more than F_{inter}, the packet is scheduled for transmission from $\mathcal{T}_{i,j}$ to $\mathcal{R}_{m,n}$.

If $i \leq m$, the wavelength set ($\lambda^s_{i\leq m}$), which contains all available wavelengths from input port i of AWGR to output port m of AWGR, is

$$m - i + 1 + f \cdot P, \forall f \in [0, F - 1]. \tag{14}$$

If $i > m$, the wavelength set ($\lambda^s_{i>m}$), which contains all available wavelengths from input port i of AWGR to output port m of AWGR, is

$$P + m - i + 1 + f \cdot P, \quad \forall f \in [0, F - 1]. \tag{15}$$

If $i \leq m$ and $\lambda^s_n \cap \lambda^s_{i\leq m} \neq \emptyset$, the packet only needs inter-cluster transmission. $\mathcal{T}^t_{i,j}$ can tune to a wavelength in $\lambda^s_n \cap \lambda^s_{i\leq m}$, which are the wavelengths that can be received by $\mathcal{R}^r_{m,n}$ using inter-cluster transmission.

If $i \leq m$ and $\lambda^s_n \cap \lambda^s_{i\leq m} = \emptyset$, $\mathcal{T}^t_{i,j}$ tunes to one of the wavelengths in $\lambda^s_{i\leq m}$ and the packet needs both inter-cluster transmission and intra-cluster transmission to arrive at its destination. Suppose $\mathcal{T}^t_{i,j}$ tunes to λ_w, where $w \in \lambda^s_{i\leq m}$. The packet arrives at $\mathcal{R}_{m,\lfloor\frac{M\cdot w}{W}\rfloor}$ by using inter-cluster transmission. In a later time slot, $\mathcal{R}_{m,\lfloor\frac{M\cdot w}{W}\rfloor}$ will transmit the packet to $\mathcal{R}_{m,n}$ by using intra-cluster transmission.

If $i > m$ and $\lambda^s_n \cap \lambda^s_{i>m} \neq \emptyset$, the packet only needs inter-cluster transmission. $\mathcal{T}^t_{i,j}$ can tune to a wavelength in $\lambda^s_n \cap \lambda^s_{i>m}$, and the packet can arrive at $\mathcal{R}^r_{m,n}$ by using only inter-cluster transmission.

If $i > m$ and $\lambda^s_n \cap \lambda^s_{i>m} = \emptyset$, $\mathcal{T}^t_{i,j}$ tunes to one of the wavelengths in $\lambda^s_{i>m}$, and the packet needs both inter-cluster transmission and inter-cluster transmission to arrive at its destination. Suppose $\mathcal{T}^t_{i,j}$ tunes to λ_w, where $w \in \lambda^s_{i>m}$. The packet first arrives at $\mathcal{R}_{m,\lfloor\frac{M\cdot w}{W}\rfloor}$. $\mathcal{R}_{m,\lfloor\frac{M\cdot w}{W}\rfloor}$ will transmit

the packet to $\mathcal{R}_{m,n}$ using intra-cluster transmission in a later time slot. The wavelength assignment algorithm is shown in Algorithm 3. Note that if there are multiple available wavelengths, we randomly select one of the available wavelengths. Similar to PODCA-S and PODCA-M, the round-robin method is used for selecting the starting rack. The pseudocode of the algorithm is shown in Algorithm 3.

To transmit more than one packet in a time slot, each ToR can have more than one tunable transmitter and wide-band receiver for intra-cluster transmission or inter-cluster transmission or both. For intra-cluster transmission, besides multiple tunable transmitters and wide-band receivers, each rack needs a multiplexer and a demultiplexer, as in PODCA-S. For inter-cluster transmission, we denote InterTx and InterRx as the number of inter-cluster tunable transmitters and wide-band receivers on each ToR, respectively. InterTx and InterRx equal the number of couplers connecting to the cluster, the number of demultiplexers connecting to the cluster, and the number of $P \times P$ AWGRs. The nth inter-cluster tunable transmitter and nth inter-cluster wide-band receiver on each rack connect to the nth $P \times P$ AWGR.

D. Algorithm Discussion

In this subsection we show that, when $N = 1$, packet selection in PODCA-S, PODCA-M, and PODCA-L can be solved optimally, i.e., the number of packets transmitted in a slot can be maximized. We also compare the throughputs using Algorithm 1, 2, and 3 with the throughput obtained by solving an integer linear programming (ILP) that maximizes the number of packet transmissions in a slot. These results suggest that our greedy algorithms are sufficient for packet selection. Our packet selection algorithms can be used for any values of N.

Theorem 1. *When $N = 1$, the packet selection problem in PODCA-S and PODCA-M is a bipartite matching problem.*

Proof. The packet transmission algorithm for PODCA-S and PODCA-M consists of two parts, i.e., packet selection and wavelength assignment. When $N = 1$, since F is not less than 1, the number of packets that can be transmitted between any two racks is only limited by the number of transmitters, which equals 1. Thus, we can optimally select packets for transmission by employing a bipartite matching algorithm, e.g., the HopcroftKarp algorithm [24]. Here, a node in the bipartite matching algorithm represents a rack, and an edge between two nodes exists if there is a packet waiting for transmission between the corresponding racks.

Theorem 2. *When $N = 1$, the packet selection problem in PODCA-L is a bipartite matching problem.*

Proof. Packet transmission in PODCA-L consists of two parts—intra-cluster transmission and inter-cluster transmission. Each part consists of two sub-parts, i.e., packet selection and wavelength assignment. When $IntraTx = 1$, the packet selection of intra-cluster transmission can be solved optimally by using bipartite matching algorithm as in PODCA-S. For $IntraTx = 1$, the number of racks that a rack can communicate with in one hop between two clusters is F, and the number of packets transmitted is limited by $InterTx$. We can build a bipartite graph by using nodes to represent racks. We partition packets into two categories—packets that can reach destinations with one hop and packets that need two hops. We first add an edge between the source node and the destination node for each packet of the first category. Then, we add edges for each packet of the second category between the source node and all nodes satisfying three constraints:

1) nodes are located within the same cluster as the destination node;
2) nodes can be reached with one hop transmission;
3) there is no edge built between the source node and the node.

The first constraint guarantees that packets can reach their destinations within two hops. The second constraint guarantees that there is at least a one-hop transmission available. The third constraint is for simplifying the graph, since there is no need for connecting parallel edges between any two nodes. Since we add all edges needed for representing packet transmissions in the bipartite graph, we can optimally select packets for transmission by employing bipartite matching algorithm.

We use a commercially available ILP solver (CPLEX) for relatively small instances to select the maximum number of packets for transmission in each time slot. Table II shows per-rack throughput comparisons between our greedy packet transmission algorithms and ILP. The size of the inter-cluster AWGR for three architectures is 40×40. The size of intra-cluster AWGR of PODCA-L is 20×20. The number of virtual buffers on each ToR equals S, and the number of available wavelengths is 160. The number of inter-cluster and intra-cluster transceivers are both set to 1. In the last row, we show the ratio of the throughput from greedy algorithms to that from ILP. Results show that throughput from greedy algorithms is close to throughput from ILP, and the difference is no more than 7%.

V. Simulation Results

In this section, we conduct simulations to evaluate the latency and throughput performance of PODCA. Packet arrivals follow a Poisson process. The transmission rate of a tunable transmitter (and wavelength capacity) is assumed to be 10 Gbps. The tuning time of tunable transmitters is 8 ns. The size of a packet is 1500 bytes. The latency consists of transmission time and queuing delay. Each rack has an 8 MB buffer, which is partitioned into one or more virtual buffers for storing packets to be transmitted. The number of virtual buffers on each ToR is B. Packets are stored in virtual buffers in demultiplexing manner, i.e., the packet with destination $\mathcal{R}_{m,n}$ is stored in the $[(m \cdot M + n)\%B + 1]$th virtual buffer. Error bars shown in Figs. 4(a)–4(c) represent 95% confidence intervals. In PODCA-S and PODCA-M, packets can reach destinations with one hop, and in PODCA-L, at most two hops are

TABLE II
COMPARISONS BETWEEN GREEDY ALGORITHMS AND ILP

Arrival Rate (Gbps)	2.4			4.8			7.2		
Architecture	PODCA-S	PODCA-M	PODCA-L	PODCA-S	PODCA-M	PODCA-L	PODCA-S	PODCA-M	PODCA-L
$T'put_{\text{Greedy}}$(Gbps)	2.38	2.37	2.14	4.70	4.65	3.71	6.96	6.72	4.28
$T'put_{\text{ILP}}$(Gbps)	2.39	2.40	2.22	4.79	4.79	3.81	7.18	6.98	4.58
Ratio	0.99	0.99	0.97	0.98	0.97	0.97	0.97	0.96	0.93

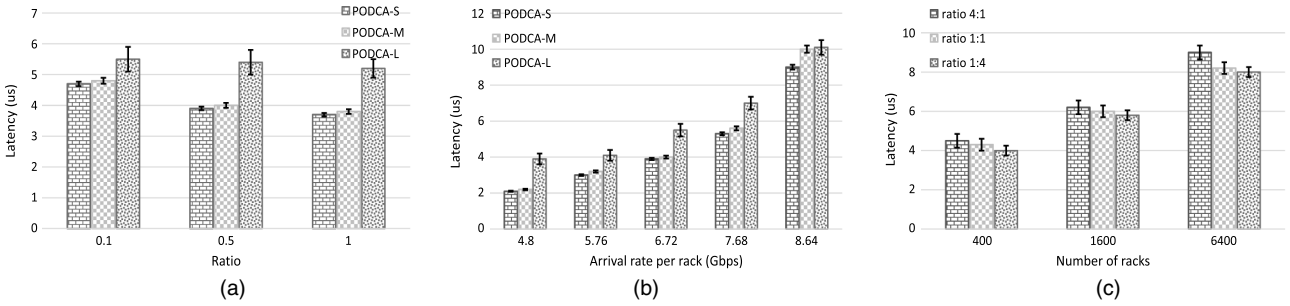

Fig. 4. Latency (μs) of PODCA-S, PODCA-M, and PODCA-L.

needed. So, only a small number of packets is stored in buffers, and because of this, we observed no packet drops in our experiments, i.e., we observed 100% throughput.

Figure 4(a) shows the latency performance of changing B of PODCA-S, PODCA-M, and PODCA-L. The values on the x axis represent the ratio between B and S. The size of inter-cluster AWGR for the three architectures is 40×40. The size of intra-cluster AWGR of PODCA-L is 20×20. The traffic arrival rate per rack is 6.72 Gbps, and the number of available wavelengths is 160. The number of inter-cluster and intra-cluster transceivers are both set to 1. In each time slot, each transmitter checks head-of-buffer packets of virtual buffers on the ToR for transmission. With more virtual buffers, more packets with different destinations can be checked for transmission. So as B increases, more packets could be transmitted per time slot, which decreases latency. Also, the average latency decrease as the ratio increases from 0.1 to 0.5 is much larger than the change from 0.5 to 1. The reason is that as the ratio increases from 0.1 to 0.5, the main constraint of average latency changes from the number of virtual buffers to the number of transmitters, receivers, and available wavelengths.

Figure 4(b) shows the latency performance as the packet arrival rate per rack changes. As arrival rate increases, more packets waits in buffers, resulting in increased latency. The increased M results in more contentions for inter-cluster transmissions, and so, PODCA-L has the largest average latency, whereas PODCA-S has the smallest average latency.

Figure 4(c) shows the latency performance of increasing the number of racks (S) of PODCA-L. We also change the ratio between P and M. The sizes of inter-cluster AWGR and intra-cluster AWGR are determined by the ratio and S. Suppose S equals $P \cdot M = 400$ and ratio is $P{:}M = 4{:}1$. The sizes of inter-cluster AWGR and intra-cluster AWGR are 40×40 and 10×10, respectively. The traffic arrival rate per rack is 6.72 Gbps, and the number of available wavelengths is 160. The number of inter-cluster and intra-cluster transceivers is 1. As S increases, contentions for both intra- and inter-cluster transmission increases, resulting in increased latency. Also, as ratio decreases, the size of each cluster increases, and more packets employ intra-cluster transmission. In intra-cluster transmission, a rack can reach any other rack in the same cluster, and in inter-cluster transmission, a rack might need a two-hop transmission. Thus, as ratio decreases, average latency decreases.

Table III shows the latency (in μs) as the number of inter- and intra-cluster transceivers changes. The size of inter-cluster AWGR for the three architectures is 40×40, and the size of intra-cluster AWGR of PODCA-L is 20×20. The number of virtual buffers on each ToR equals S, and the number of available wavelengths is 160. As the number of transmitters increases, the average number of packets transmitted in each time slot increases, resulting in lower latency. For PODCA-L, the amount of latency decrease by adding an inter-cluster transmitter is larger than by adding an intra-cluster transmitter. This is because inter-cluster transmission contributes most to the large latency.

We compare the latency performance of different architectures in Table IV. Here, PODCA-L is compared with two

TABLE III
LATENCY (MS) OF DIFFERENT INTRATX AND INTERTX

InterTx	1		2	
PODCA-S	3.74		1.91	
PODCA-M	4.89		2.54	
InterTx:IntraTx	1: 1	1: 2	2: 1	2: 2
PODCA-L	5.48	4.28	3.24	2.34

TABLE IV
LATENCY (μs) OF DIFFERENT ARCHITECTURES WITH DIFFERENT ARRIVAL RATES

Arrival Rate (Gbps)	0.096	0.336	0.96	3.36	9.6
Fat-Tree	6.26	1E + 04	1E + 05	4E + 05	1E + 06
FBFLY	2.29	2.34	2.99	Deadlock	Deadlock
DOS	1.21	1.22	1.26	1.65	16.35
Petabit	1.22	1.28	1.46	4.89	865.26
PODCA-L	2.19	2.23	2.34	3.44	18.57

electrical architectures, Fat-Tree and FBFLY, and two optical architectures, DOS and Petabit. The number of racks is 1024. The number of virtual buffers on each ToR is 100. The rack transceiver transmission rates of all architectures are set to 10 Gbps. For PODCA-L, the number of inter-cluster and intra-cluster transceivers are both set to 1. The number of virtual buffers on each ToR equals S, and the number of available wavelengths is 160. As traffic arrival rate increases in the Fat-Tree, upward and downward forwarding packets are congested in switches at aggregation level, and it results in tremendously large latencies and large number of packet drops. FBFLY becomes deadlocked when the arrival rate per rack reaches 3.36 Gbps. The deadlock means packets fill all virtual buffers of switches, and no packet can be further forwarded. When the network becomes deadlocked, new arriving packets are dropped. In DOS, each port of AWGR connects with a rack, so packets can reach destinations in one hop. However, the number of wavelengths needed is S, which is much larger than the number of wavelengths used by PODCA-L. Petabit achieves small latency when traffic arrival rates are small by employing expensive devices, i.e., TWCs. However, when traffic arrival rates become large, the packet scheduling algorithm for multi-column AWGRs becomes the bottleneck for achieving low latency. Compared with other architectures, the average latencies of PODCA-L are small, and the number of wavelengths needed is also small.

VI. POWER AND COST COMPARISON OF DCN ARCHITECTURES

In this section, we compare our proposed DCN architectures with some electrical (Fat-Tree, FBFLY) and optical (DOS, Petabit) DCN architectures in terms of power consumption and CapEx.

For the electrical DCNs, the total power consumption of the switching interconnect is the sum of the power consumed by the ToR switches and the aggregate switches. Some papers have already analyzed the architectures and deduced the quantity correlation among the components [4,25].

For Fat-Tree, analyzed in Ref. [4], we know that all switching elements are identical, which enables us to leverage cheap commodity parts for all of the switches. Let us assume that k is the number of ports per commodity switch. The maximum number of end hosts or servers it can support is $\frac{k^3}{4}$, and the total number of switches it includes is $\frac{5\,k^2}{4}$. The number of transceivers between aggregate and ToR switches is

$$k \times \frac{5\,k^2}{4} - \frac{k^3}{4} = k^3. \tag{16}$$

For k-ary η-cube FBFLY, in Ref. [25] it is assumed that we have c endpoints (hosts) per switch. Then we set $k = c$ in order to achieve full bisection bandwidth. N is the number of servers it can support. Thus the number of hosts is $S = k^\eta$ and the port count per switch is

$$\varepsilon = (k-1)(\eta-1) + c = k\eta - \eta + 1. \tag{17}$$

The number of the switches is calculated as

$$\frac{S}{k} = \frac{k^\eta}{k} = k^{\eta-1}. \tag{18}$$

Since when $\eta = 5$ the network can achieve good scalability, we choose this value for the following comparison. We can calculate that the number of the transceivers between switches is

$$\varepsilon \cdot k^{\eta-1} - k^\eta = (4k-4)k^4. \tag{19}$$

Also, the number of ports for ToR switch processor in the DCN interconnect is $\varepsilon \cdot k^{\eta-1}$.

Our study changes the network size from thousands of servers to millions. Even though the number of ports at each commodity edge switch or aggregate switch is limited, we calculate the power and cost of switches by multiplying the average unit price per port with the average number of ports. For example, in Fat-Tree or FBFLY architecture, the market price of a 48-port commodity ToR edge switch processor is \$420 and the power is 70 W [26]. If we only need 24 ports of ToR switch to support around 10,000 servers, the unit price of ToR per port is assumed to be $\frac{420}{48} = \$8.75$, and the unit price of a matching ToR switch processor is $\frac{420\times24}{48} = \$210$. Similarly, the power of the ToR per port can be calculated as 1.46 W. For the scalability of the interconnect, the number of ports in commodity switch is limited. When the oversubscription is 1:1, the largest number of servers that the Fat-Tree architecture can support is $\frac{k^3}{4} = 27648$. If we increase the number of servers in the Fat-Tree to millions, the oversubscription ratio should be larger than 1:1. However, for the power and CapEx cost comparison of DCNs, we assume that the number of ports of the switch can be increased to accommodate the million (s) of servers.

In the optical DCNs, recall that S is the total number of racks in the DCN interconnects, and each rack consists of a number of servers. We assume there are 48 servers in each rack and the transmission rate per server can be up to 10 Gbps. DOS consists of an AWGR, S TWCs and a loopback shared buffer. Considering the small size of SDRAM, we ignore it in comparisons. Petabit is a scalable bufferless optical switch architecture. It adopts a three-stage Clos network and each stage consists of an array

TABLE V
POWER CONSUMPTION AND COST OF COMPONENTS IN DIFFERENT DCN ARCHITECTURES

Component	Power (Watts)	Cost (Dollars)	Fat-Tree	FBFLY ($\eta = 5$)	DOS	Petabit	PODCA Small	PODCA Medium	PODCA Large
Electronic									
Transceiver between ToR and aggregate switch	1.5	27.5	k^3	$(4k-4)k^4$	0	0	0	0	0
ToR switch processor per port	1.46	8.75	$\frac{5k^2}{4}$	$(5k-4)k^3$	0	0	0	0	0
Optical									
SFP transceiver	1.0	45	0	0	S	0	0	0	0
Fast wavelength tunable transmitter (WTT)	1.5	195×5	0	0	0	S	$2S$	$2S$	$2S$
Fixed wideband receiver (photodetector)	0.9	40	0	0	0	0	$2S$	$2S$	$2S$
Tunable wavelength converter (TWC)	20	8000	0	0	S	$2S$	0	0	0
Arrayed waveguide grating (AWG) per port	0.0	15	0	0	S	$6S$	S	P	$S+P$
Amplifier (EDFA)	5	2250	0	0	0	0	0	P	P
Coupler	0.0	195	0	0	0	0	S	P	P
Optical (DE)MUX	0.0	1700	0	0	S	0	S	P	P
Fiber Delay Line	0.0	2000	0	0	S	0	0	0	0

of AWGRs. Petabit includes TWCs for wavelength routing using AWGRs.

The capacity of a wavelength is set to 10 Gbps, which is also equal to the transmission rate of a tunable transmitter. The power consumption and unit cost values of the components are mainly taken from commercial product specifications from the literature [10,27]. The tunable transmitter with 8 ns tuning time is not yet commercially available; in order to conduct the comparison, we assume a $5\times$ price of commodity tunable transmitters. Besides, we assume that the number of transmitters and receivers on each ToR is 2 to trade off between better performance and lower cost. The simulation results show that when the number of transmitters and receivers on each ToR is 2, the performance is comparable to that of other architectures. We calculate the power consumption and CapEx of these network topologies by summing up the consumed power and dollar cost of each component. A summary of the power, cost, and number of components used in each architecture is given in Table V. Since the purpose of this study is to compare the difference between these DCN interconnect schemes, the table does not include the cost and power consumption of data center servers.

Based on the details in Section IV, we have three different PODCA versions depending on the size of the network. We assume that P and W can both scale up to 512 [16], and each rack consists of 48 servers. We set the size of PODCA-S to a maximum of 12,288 servers (256 racks). PODCA-M is used when $256 < S \leq 512$ (from 12,288 to 24,576 servers), and PODCA-L is used when $512 < S \leq 51200$ (from 24,576 to more than 2 million servers). Therefore, the number of servers can vary from thousands to more than 2 million. Note, for PODCA-M, we set $M = 2$ and $S = 2 \cdot P$. Also, for PODCA-L we set $M = 100$ and $S = 100 \cdot P$.

Figure 5 presents the overall power consumption with increasing number of servers. Our main power-driven components include tunable transmitters and fixed wide-band receivers. The fixed wide-band receiver consists of just a photodetector and corresponding circuitry. It is clear that the power consumption values of the optical DCNs are far less than the electrical DCN architectures. We can find that the power consumption of the Fat-Tree and FBFLY interconnects becomes around fifteen times more than the optical interconnects when the number of servers increases. This large power consumption is due to the fact that ToR switch processor and the transceivers between the ToR switch and aggregate switch are very power-hungry devices. We also note that the power consumption of PODCA is dramatically lower compared to Petabit (saving 87%) and DOS (saving 75%), when the number of servers increases up to 2 million. The large power consumption of those architectures is caused by active components such as the TWC in DOS and Petabit. Besides, multiple SFP transceivers required in each ToR increase the power consumption in DOS.

Figures 6(a)–6(c) show that our architectures can save CapEx as well. In our architectures, instead of choosing expensive devices such as TWC, we use relatively cheap devices, e.g., tunable wavelength transmitters, couplers, and AWGRs. Figures 6(a) and 6(b) show that PODCA-S and PODCA-M outperform the other four architectures, and PODCA-S and PODCA-M can save at least 50% CapEx. Due to the expensive TWC, Petabit is the most expensive

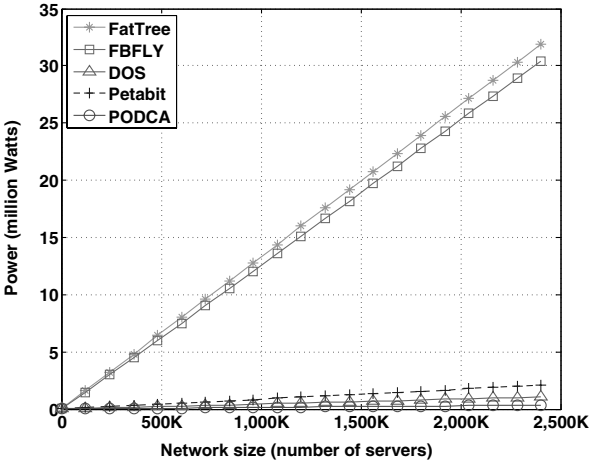

Fig. 5. Power consumption comparison of different architectures.

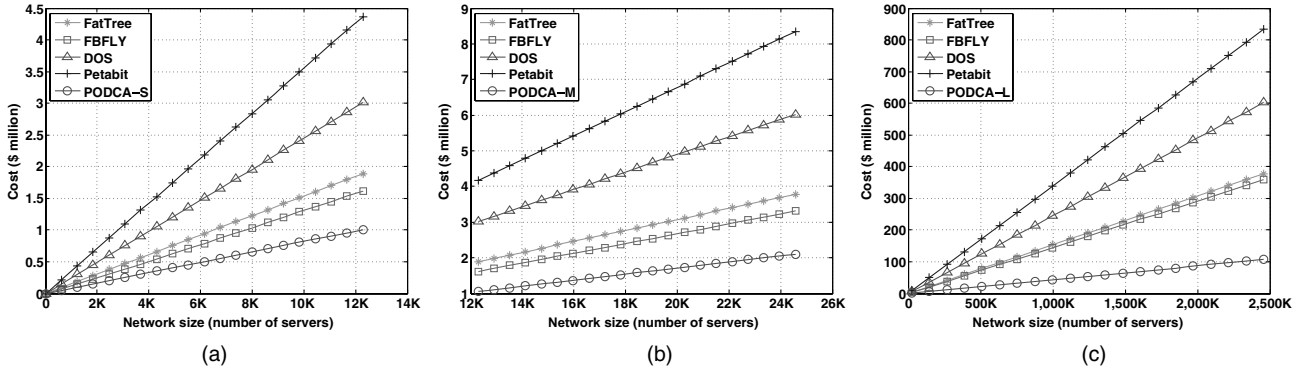

Fig. 6. CapEx comparison of different architectures: (a) PODCA-S, (b) PODCA-M, (c) PODCA-L.

DCN interconnect. In addition, the results shown in Fig. 6(c) indicate that PODCA-L saves 73.7%, 72.6%, 83.3%, and 88% CapEx compared with Fat-Tree, FBFLY, DOS, and Petabit, respectively.

VII. Conclusions

In this paper, we presented PODCA, a passive optical data center network architecture. The key component of our architecture is the AWGR. Based on our simulation results, the packet latency of PODCA is below 19 μs and PODCA achieves 100% throughput. In addition, PODCA exhibits lower latency and higher throughput even at high input loads compared with electrical DCNs such as Fat-Tree and FBFLY, while achieving comparable performance as other optical DCN architectures such as DOS and Petabit, but at much lower cost. Moreover, for comparison in terms of power consumption and CapEx, results show that our architectures can save at least 75% on power consumption and 50% on CapEx. Future work includes making the architecture reconfigurable to adapt to changing traffic patterns.

Acknowledgment

This work was supported in part by NSF award # 1618487. This paper is an extended version of [15].

References

[1] P. Delforge, "America's data centers consuming and wasting growing amounts of energy," 2015 [Online]. Available: https://www.nrdc.org/resources/americas-data-centers-consuming-and-wasting-growing-amounts-energy.

[2] "Some interesting bits about latency," [Online]. Available: https://www.citycloud.com/city-cloud/some-interesting-bits-about-latency.

[3] "Who has the most web servers," [Online]. Available: http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[5] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 126–137, 2007.

[6] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 39, no. 4, pp. 51–62, 2009.

[7] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, 2009.

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

[9] Y. Cheng, M. Fiorani, R. Lin, L. Wosinska, and J. Chen, "POTORI: a passive optical top-of-rack interconnect architecture for data centers," *J. Opt. Commun. Netw.*, vol. 9, no. 5, pp. 401–411, 2017.

[10] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: a topology malleable data center network," in *9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ACM, 2010, p. 8.

[11] 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 Comp. Commun. Rev.*, vol. 40, no. 4, pp. 339–350, 2010.

[12] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, "c-through: part-time optics in data centers," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 40, no. 4, pp. 327–338, 2010.

[13] X. Ye, Y. Yin, S. B. Yoo, P. Mejia, R. Proietti, and V. Akella, "DOS: a scalable optical switch for datacenters," in *6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ACM, 2010, p. 24.

[14] K. Xi, Y.-H. Kao, M. Yang, and H. Chao, "A petabit bufferless optical switch for data center networks," in *Optical Interconnects for Future Data Center Networks*, Springer, 2013, pp. 135–154.

[15] M. Xu, C. Liu, and S. Subramaniam, "PODCA: a passive optical data center architecture," in *IEEE Int. Conf. Communications (ICC)*, IEEE, 2016, pp. 1–6.

[16] S. Cheung, T. Su, K. Okamoto, and S. Yoo, "Ultra-compact silicon photonic 512 × 512 25 Ghz arrayed waveguide grating router," *IEEE J. Sel. Top. Quantum Electron.*, vol. 20, no. 4, pp. 310–316, 2014.

[17] F. Yan, W. Miao, O. Raz, and N. Calabretta, "Opsquare: a flat DCN architecture based on flow-controlled optical packet switches," *J. Opt. Commun. Netw.*, vol. 9, no. 4, pp. 291–303, 2017.

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

[19] Y. Yin, R. Proietti, X. Ye, C. J. Nitta, V. Akella, and S. Yoo, "LIONS: an AWGR-based low-latency optical switch for high-performance computing and data centers," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, p. 3600409, 2013.

[20] X. Ye, S. Yoo, and V. Akella, "AWGR-based optical topologies for scalable and efficient global communications in large-scale multi-processor systems," *J. Opt. Commun. Netw.*, vol. 4, no. 9, pp. 651–662, 2012.

[21] K. Sato, H. Hasegawa, T. Niwa, and T. Watanabe, "A large-scale wavelength routing optical switch for data center networks," *IEEE Commun. Mag.*, vol. 51, no. 9, pp. 46–52, 2013.

[22] S. Matsuo, S.-H. Jeong, T. Segawa, H. Okamoto, Y. Kawaguchi, Y. Kondo, H. Suzuki, and Y. Yoshikuni, "Stable and fast wavelength switching in digitally tunable laser using chirped ladder filter," *IEEE J. Sel. Top. Quantum Electron.*, vol. 13, no. 5, pp. 1122–1128, 2007.

[23] J. Engelstaedter, B. Roycroft, F. Peters, and B. Corbett, "Fast wavelength switching in interleaved rear reflector laser," in *Int. Conf. Indium Phosphide & Related Materials (IPRM)*, IEEE, 2010, pp. 1–3.

[24] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM J.Comput.*, vol. 2, no. 4, pp. 225–231, 1973.

[25] F. Yao, J. Wu, G. Venkataramani, and S. Subramaniam, "A comparative analysis of data center network architectures," in *IEEE Int. Conf. Communications (ICC)*, IEEE, 2014, pp. 3106–3111.

[26] "Ubiquiti EdgeSwitch-Switch-48 Ports" [Online]. Available: https://www.amazon.com/Ubiquiti-EdgeSwitch-Managed-Rack-Mountable-ES-48-LITE/dp/B013JDNN3K/ref=sr_1_1?ie=UTF8&qid=1485328217&sr=8-1&keywords=edge+switch+48+port.

[27] J. Chen, Y. Gong, M. Fiorani, and S. Aleksic, "Optical interconnects at the top of the rack for energy-efficient data centers," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 140–148, 2015.

Maotong Xu received a B.S. degree in the Northwestern Polytechnical University, China in 2012, a M.S. degree from the George Washington University, USA, in 2014. He is currently pursuing a PhD with the Department of Electrical and Computer Engineering, The George Washington University. His research interests include cloud computing, and data center networks.

Chong Liu is a PhD candidate in the Department of Electrical and Computer Engineering at the George Washington University, USA. He received a M.S. degree from Stevens Institute of Technology in 2013. His current research is in optical data center networks and machine learning analysis in wireless communication.

Suresh Subramaniam (S'95-M'97-SM'07-F'15) received a PhD in electrical engineering from the University of Washington, Seattle, in 1997. He is a Professor in and Chair of the Department of Electrical and Computer Engineering at the George Washington University, Washington, DC. His research interests are in the architectural, algorithmic, and performance aspects of communication networks, with current emphasis on optical networks, cloud computing, and data center networks. He has published over 180 peer-reviewed papers in these areas. Dr. Subramaniam is a co-editor of three books on optical networking. During 2012 and 2013, he served as the Chair of the IEEE ComSoc Optical Networking Technical Committee. He received the 2017 SEAS Distinguished Researcher Award at GWU. He is a Fellow of the IEEE.