Priority-Aware Scheduling for Packet-Switched Optical Networks in Datacenter

Lin Wang[*], Xinbo Wang[*], Massimo Tornatore[*†], Kwang joon Kim[‡], Sun Me Kim[‡],
Dae-Ub Kim[‡], Kyeong-Eun Han[‡], and Biswanath Mukherjee[*]
[*]University of California Davis, USA
[†]Politecnico di Milano, Italy
[‡]Electronics and Telecommunications Research Institute, Korea

Abstract— **A scalable, low-latency, high-speed, and energy-efficient datacenter network represents a key element in the deployment of cloud-oriented large-scale datacenters. Photonic switch (PS) architectures can be a solution due to the inherent benefits of optical technology. In this study, we numerically investigate the performance of a packet-switched optical network (PSON) architecture with centralized control for datacenter interconnection. To achieve high performance of PSON, an intelligent yet low-complexity scheduling algorithm is critical. It is also beneficial to take traffic flow features into consideration. Thus, we propose and investigate the characteristics of a priority-aware scheduling algorithm for PSON, considering both architecture features and traffic-flow characteristics in datacenter. Numerical simulations show the advantages of the priority-aware algorithm.**

Keywords—Packet-Switched Optical Network; Datacenter; Priority-aware.

I. INTRODUCTION

The annual global datacenter traffic has been increasing at an annual rate of 31% from 2012 to 2017 [1]. More importantly, 76% of this traffic resides within the datacenter. As current datacenter communication is still based on electrical switches, datacenter operators are facing technical challenges related to high power consumption, limited scalability, and huge latency of their datacenters. For example, the power consumption of electrical datacenter switches, which are one of the major contributors to datacenter power consumption, is about 2.5 [W/Gbps] [2], and, since today a datacenter needs to support bandwidth on the order of Tbps, the overall energy consumption will increase significantly. Also, the power scales up as the size of the datacenter network increases, leading to higher energy cost. Moreover, low latency is crucial for datacenter networks because today a large amount of big-data analytics, based on real-time complex event processing, is performed by exchanging huge quantities of data on datacenter networks. Hence, issues such as the continuous increase in data traffic, emerging delay-sensitive applications of cloud computing, and high power consumption of electrical switches, are all stimulating a substitution of current datacenter networks.

Considering all of the above, an optical packet switched (OPS) network can represent a valuable alternative for electrical datacenter network due to its inherent benefits of using passive optical components. For example, Arrayed Waveguide Grating Routers (AWGR) plus fast-tuning lasers can be preferably used in PS architectures, as this kind of technology can significantly reduce the power consumption for datacenter networks, while ensuring low latency. Furthermore, multiple wavelength-division multiplexing (WDM) channels can be used when multiple concurrent packets are associated to the same output to avoid head-of-line blocking, which results in better performance in terms of lower latency and lower packet loss.

Several OPS architectures have been reported in [3-7]. These architectures are based on rearrangable, non-blocking, multistage Benes, Banyan, or other interconnections. Typically, they have centralized control. To change their interconnection patterns, all these solutions need a certain amount of reconfiguration time during which the switch is unable to

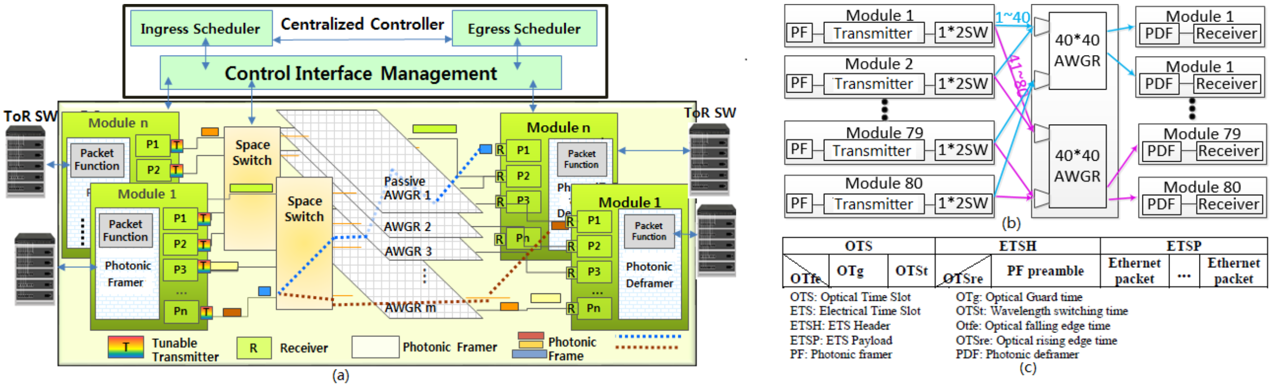

Fig. 1. (a) PSON architecture. (b) PSON data plane (with optical switch fabric). (c) Photonic frame format.

handle data, and this results in packet loss. The reconfiguration-time complexity increases proportionally to $Nlog_2N$, where N is the number of input and output ports [4]. Hence, we consider a much simpler OPS architecture, called Packet-Switched Optical Network (PSON), which has only one stage of switching as shown in Fig. 1 (more details will be provided in Section II). Moreover, data plane and control plane are decoupled to avoid conflict and reconfiguration, to ease the implementation of advanced scheduling algorithm, and to enhance scalability.

To achieve high performance of PSON, an intelligent yet low-complexity scheduling algorithm is critical. Round robin (RR) scheduling algorithm [9] is widely deployed because it is simple and fast enough to match the speed of PSs. But RR algorithm fails to consider the traffic flow characteristics in datacenter [10, 11], resulting in low efficiency in terms of packet-loss ratio (PLR) and latency. In this study, we define PLR as the ratio of successfully-transmitted packets and total number of packets generated by all servers. Other modifications to the RR algorithm, which are intended to work under a distributed control [12, 13], need excessive signaling exchange and result in unnecessary latency.

To leverage the knowledge of traffic characteristics in a datacenter and provide PSON with an efficient yet simple scheduling strategy, we propose a novel priority-aware (PA) scheduling algorithm that exploits information on the total length of buffered packets, the number of buffered packets, and the arrival time of the earliest packet in the buffer. We use different weight factors to reflect how much these three pieces of information could affect performance of PSON. In addition, our PA algorithm also considers space switch tuning time of connecting a $1\times m$ space switch to a certain AWGR. Simulation experiments show that our PA algorithm outperforms the traditional RR algorithm in terms of PLR and average delay.

The rest of the study is organized as follows: Section II introduces the PSON architecture and photonic frame format. In Section III, we describe the datacenter traffic model and illustrate the PA algorithm implemented in PSON. In Section IV, we quantitatively investigate the performance of the PA algorithm and compare it with RR algorithm through simulation. Section V concludes this study.

II. Packet-Switched Optical Network (PSON) Architecture

We present the PSON architecture in Fig. 1(a). The core of the PSON architecture uses AWGRs, which are passive and high-speed optical devices, to achieve contention resolution in the wavelength domain [8]. In PSON, multiple servers are connected to a top-of-rack (ToR) switch, and n such ToR switches are interconnected by a core switching network. Each ToR switch connects to an ingress module and an egress module that contain photonic framer and de-framer, respectively. An ingress module contains n virtual output queues (VoQs), each corresponding to a ToR switch as destination. The framer/de-framer manages the wrapping/unwrapping of Ethernet packets into photonic frames. An optical transmitter uses a fast tunable-wavelength light source to transmit photonic frames from the ingress module. Each ingress module is also equipped with a $1\times m$ space switch to connect with m AWGRs, and an AWGR exclusively delivers photonic frames to n/m egress modules.

We enhance the scalability of PSON by using multiple $1\times m$ space switches so that the number of switch ports can be easily enlarged without considerably increasing latency. Fig. 1(b) illustrates an example of applying 1×2 space switches to increase the number of supported ToR switches from 40 to 80. In PSON, data and control planes are decoupled from the actual optical devices and logically centralized, where data plane takes charge of packet switching, while control plane manages the scheduling of photonic-frame switching in each (fixed) time slot.

Let us now look more closely on how information is transferred in PSON. In ingress, servers generate native Ethernet/IP packets, which are aggregated by their ToR switch and sent to the associated ingress module. An electrical packet will be placed into a VoQ according to its destined ToR, waiting for photonic wrapping and switching. For each ingress module, the centralized controller must decide which VoQ can wrap its packets into a photonic frame (with format as shown in Fig. 1(c)). When an ingress is allocated with a transmission grant by control plane, the ingress will send a photonic frame to the associated space switch in the next time slot, that will be switched by the AWGR that connects to the destined egress module, where inverse operations are performed so that packets can be delivered to target servers. However, in a given time slot, an ingress module must decide which VoQ can transmit packets, and an egress module must decide which ingress module it wants to receive the frame from. Without proper scheduling, conflict can occur when multiple ingress modules want to simultaneously deliver frames to the same egress module, and bandwidth resource will be wasted if a time slot is not fully utilized for transmission. Also, to keep low latency when increasing the scale of PSON and to reduce the cost of the devices, the scheduling process should not have a complex implementation. Otherwise, it might take a long time or advanced expensive devices for a complex scheduling process. Therefore, an efficient and simple scheduling algorithm is critical for PSON.

III. Traffic Model And Priority-aware Algorithm

A. *Traffic Model*

To design an efficient scheduling algorithm for PSON, we first investigate the traffic characteristics inside a datacenter. According to the studies in [10, 11], we know packet length in real scenarios is mostly found to follow a bimodal distribution at 40 bytes and 1500 bytes. These two values match the Ethernet minimum and maximum lengths, and are found in other network environments also [14]. In our work, the traffic sources are programmed to create packets with any arbitrary size and to send them according to specific inter-arrival-time parameters. The packet size according to the distribution described above instead of uniformly [10, 11]. Each of these modules simulates the aggregated load of a large number of servers, which constitutes the traffic of each ToR switch. In our experiments, each of the modules receives the input traffic generated by 36 simulated servers.

The periods of packet generation are modeled by matching ON/OFF periods (with/without packets transmission). This is the traffic behavior found in datacenters and, in general, in Internet [10-14]. The length of these ON/OFF periods is characterized by heavy-tailed random distributions. In our simulations, we model them with a Pareto distribution.

As we know, due to the different types of applications running in datacenters, some servers may exchange mostly large-size packets (e.g., file processing application), some severs may exchange lots of small-size packets (e.g., control processing application), while other servers exchange both large-size and small-size packets. Hence, in our simulation, we classify servers into three groups based on the size of packets they mainly exchange. All packets generated by one server in every transmission (ON) period are randomly sent, with uniform probability, to one of the possible destination servers which belong to the same group of the source server. Datacenter traffic characteristics are strongly dependent on the applications running in the servers. This means that they vary from one environment to another. As a final note, it is difficult (if even possible) to define a concept such as typical datacenter traffic. The traffic model employed here is only a generalized approximation for the very different scenarios that can be found in real datacenters.

B. *Priority-Aware (PA) Scheduling Algorithm*

The classical Round-Robin (RR) algorithm [12, 13] is widely used in schedulers due to its simplicity. In RR algorithm, an ingress module will check all non-empty VoQs and request transmission grant from each corresponding egress module. An egress module collects requests from ingress modules and takes turns to grant transmission (one ingress module in each time slot). Each ingress module also collects grants from egress modules and takes turns to accept grants. However, the classical RR algorithm that ensures absolute fairness among all VoQs may not be suitable for today's datacenter. In fact, in today's datacenter application, latency is more critical (unlike in telecom applications), because the resultant timeout could result in failure of a computing application. According to studies (e.g., [10, 11]), intra-datacenter traffic follows a heavy-tailed distribution, which means that some VoQs may buffer a burst of traffic data while others do not. Hence, prioritizing the transmission of VoQs with large data size can reduce the average delay and classical RR does not account for such prioritizations. Also, according to datacenter traffic characteristics [11], some servers mainly generate and exchange ant flows (i.e., small-size packets) while some servers may exchange elephant flows (i.e., large-size packets).

To switch more efficiently different packet types in a datacenter, we investigate the benefits of using traffic characteristics and propose a priority-aware (PA) scheduling algorithm based on traditional RR algorithm. PA algorithm uses priority of VoQs to reflect and handle the special traffic flow features in a datacenter. PA exploits four possible scheduling strategies including longest queue fist (LQF), largest number of packets first (LNPF), oldest packet first (OPF), and less space switch first (LSSF).

Fig. 2 presents the flowchart of PA algorithm.First, each ingress module maintains status information and gets priority values for the n VoQs based on their status information. The priority value is calculated based on a combination of four strategies: i.e., LQF, LNPF, OPF, and LSSF using the following weighted function:

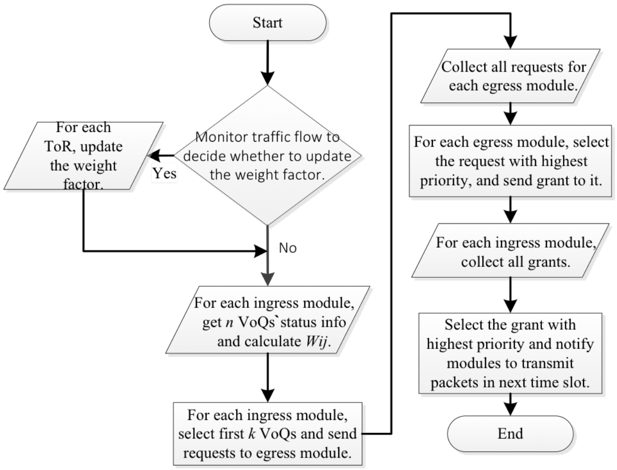
Start
Monitor traffic flow to decide whether to update the weight factor.
Yes
For each ToR, update the weight factor.
No
For each ingress module, get n VoQs`status info and calculate Wij.
For each ingress module, select first k VoQs and send requests to egress module.
Collect all requests for each egress module.
For each egress module, select the request with highest priority, and send grant to it.
For each ingress module, collect all grants.
Select the grant with highest priority and notify modules to transmit packets in next time slot.
End

Fig. 2. Flowchart of priority-aware algorithm.

$$W_{ij} = l_{ij} * w_l + p_{ij} * w_p + d_{ij} * w_d + s_{ij} * w_s \quad (1)$$

where W_{ij} is the priority of VoQ_{ij} (the virtual queue at ingress module i associated with egress module j), l_{ij} represents the length of VoQ_{ij}, p_{ij} represents the number of packets in VoQ_{ij}, d_{ij} represents the delay of the earliest packet in VoQ_{ij}, and w_l, w_p, and w_d are weighting factors for them. s_{ij} is a boolean value (taking value 0 if sending a frame of VoQ_{ij} needs tuning of the space switch; 1, otherwise). Note that a space switch can only connect to a particular AWGR in each time slot. For example, in Fig. 1(b), if module 1 wants to send packets to module 80, but its space switch is connected with the upper AWGR, then it must be tuned to connect with the lower AWGR. Since the tuning time of a space switch is at sub-micro-second speed, we should avoid frequently tuning the space switch and use the LSSF strategy introduced above so that extra latency can be minimized. In consecutive time slots, the proposed PA algorithm prefers to send packets to egress modules that are connected to the same AWGR.

Second, an ingress module selects the first k non-empty VoQs with highest priorities and sends requests (along with their priority values) to their associated egress modules, instead of sending requests for all VoQs as in RR algorithm. Then, an egress module collects requests from different ingress modules and selects the request with the highest priority and grants transmission to ingress scheduler. Finally, the ingress module collects at most k grants from all egress modules. If there are multiple grants for it, the grant for the VoQ with highest priority is accepted. Note that this whole scheduling process of PA algorithm is executed in a separate centralized controller instead of in each module. Each module only reports the n VoQs' statuses to the centralized controller, gets the grants (if possible), and prepares to transmit packets in the next time slot.

As mentioned before, different servers may generate different types of traffic flow. For servers which mainly generate and exchange ant flow, weight factor w_p should be more important than w_l because the growing number of packets

(p_{ij}) in a VoQ does not increase the total packet size (l_{ij}) a lot. If w_p is small and w_l is large, the calculated priority of VoQ according to Eqn. (1) will be small and result in VoQ having less chance to be transmitted. This causes a longer delay for all packets waiting in this VoQ and leads to increased average latency for the whole network. Vice versa for servers which generate elephant flow. Therefore, we need the algorithm to be able to periodically and dynamically update weight factors for each VoQ in modules to reflect and deal with their own traffic flow features.

As space tuning time is a constant value and the earliest packet arrival time is random, w_s and w_d are not necessary to be updated. The total length of packets and the number of packets in the buffer are closely related to the traffic flow in datacenter, thus w_p and w_l are the two major factors to decide the efficiency of PA algorithm. With this knowledge, our PA algorithm only updates w_p and w_l for each VoQ based on the information p_{ij} and l_{ij}. We know that l_{ij}/p_{ij} represents average packet size in VoQ_{ij}. The updating process works as following: If average packet size is larger than a threshold (called ele_th), which means most of the traffic data belongs to elephant flows, then we increase the value of w_l with θ and deduct θ from the value of w_p; if average packet size is smaller than another threshold (called ant_th), which means most of the traffic data belongs to ant flows, then w_p should grow and w_l descends. Otherwise, we do not change w_p and w_l. In our simulation, we set ele_th as 1000 bytes and ant_th as 300 bytes. When calculating the priority for each VoQ, w_p, w_l, w_d, and w_s will be normalized such that they sum up to 1 before being used as weight factors.

IV. Illustrative Numerical Examples

Simulation under different configurations have been run to analyze the efficiency of PA algorithm compared with traditional RR algorithm, and the influence of weight factors w_p and w_l on PSON. In our simulations, we consider a PSON with $n=80$ ToR switches, 40×40 AWGRs, and 1×2 space switches. Each ToR receives input traffic generated by 36 servers. Each server generates simulated traffic following a pattern described in Section III.A. The capacity of a wavelength is 10 Gbps and each module has a buffer shared by all VoQs. We classify the 36 servers of each ToR into three groups: first group generates packets which contain 80% elephant flow and 20% ant flow, second group generates packets with 20% elephant flow and 80% ant flow, and third group generates packets uniformly. The total amount of offered traffic load is normalized and scaled from 0 to 1. When normalizing, we consider offered load as the ratio of all the packets generated by three server groups to the total capacity of PSON. The optical-link length between ToR and PSON architecture is set to 60 m. As frames traveling through the switch have to cover this distance at least twice (to and from; see Fig. 1), the propagation delay will be 600 ns. Longer or shorter links do not impact the performance of PSON but can be seen as a larger or smaller latency offset to the latency introduced by PSON. We iterate 50 independent instances so that all plotted values have a 95% confidence level for a confidence interval not larger than 5%.

In Fig. 3, we compare the average delay of PA and RR algorithms. The packets' latency is composed of four factors: ingress buffering time in VoQs, tuning time of space switch, tuning time of tunable-wavelength light source, and the optical links. At low loads (i.e., smaller than 0.3), PA and RR incur very similar delay. In fact, at low loads, high-priority VoQs are rarely occupied, i.e., most of VoQs have similar priority values, so PA algorithm shows almost the same performance as RR. But as load increases, PA outperforms RR, and the delay of PA increases at a much slower rate than RR. This happens because, when offered load is large, temporal and spatial traffic dynamics are drastically different among servers (due to bursty traffic), which leads to various priority values for different VoQs in different time slots. PA algorithm promotes the transmission of packets in VoQ with highest priority, reducing the average delay, while in RR, packets in VoQs that buffer bursty traffic have to keep waiting until their turn.

Fig. 4 shows PLR of PA and RR algorithms as a function of offered load. PA and RR achieve similar average delay under low load (< 0.3), confirming results in Fig. 3, because contention rarely happens and almost all of the newly-generated packets can be transmitted immediately in next time slot, which means latency only depends on tuning time and propagation time. But, as load increases (>0.3), PA can achieve up to 17% less PLR than RR. This is because, in each time slot, an ingress module can only choose one VoQ to transmit its packets, hence which VoQ being chosen is very important to decide the performance of the total PSON. PA always selects the VoQ with the highest priority (either packets with larger sizes or a larger number of packets), while RR may choose a VoQ with small priority due to its randomness, which leads to longer latency. Hence, PA can achieve smaller PLR, improves bandwidth utilization, and enhances the capacity of PSON. The PLRs of both algorithms tend to saturate, because at high offered load, buffer is not large enough to store all packets and only some partially-generated packets can be transmitted successfully.

Fig. 5 demonstrates how weight factors w_l and w_p impact the PA algorithm in terms of average delay. In this figure, PA follows the updating process described in Section III.B to change w_l and w_p values, while the curve "w_l" represents a version of the PA algorithm where all weight factors are set to zero except w_l (similarly for the curve of "w_p"). As we can see, PA with upgrading process achieves smaller average delay than the other two cases. This is because different servers may generate and exchange different types of traffic flow, as mentioned in Section III.A. Let us assume that we decide the VoQ from which we transmit to be only based on the length of packets; then some VoQs containing many small-size packets will have little opportunity to be selected and will result in longer delay. Vice versa for the case of only considering the number of packets. Thus, PA with upgrading process which considers both factors (w_s and w_d) achieves better performance.

In Fig. 6, we study the influence of weight factors w_l and w_p on PLR in PSON. When offered load is small, contention rarely happens, so there is almost no packet loss. For increasing offered load (> 0.4), PLR of PA grows slowly than the other two, which confirms the results in Fig 4. We can also see that the curve of "w_l", which only considers the total packets' length in each VoQ, gives lower PLR than the curve of "w_p" which only considers the number of packets in each VoQ. Assuming

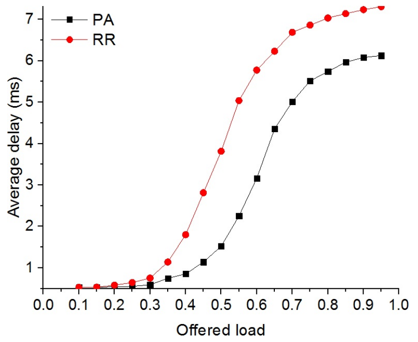

Fig. 3. Average delay of PA and RR algorithms.

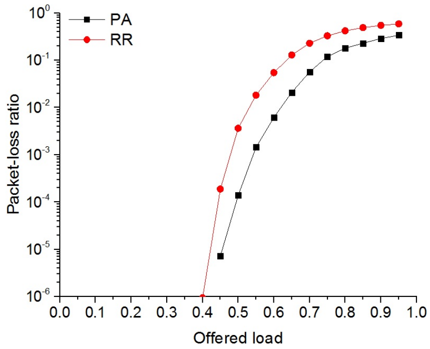

Fig. 4. Packet loss ratio of PA and RR algorithms.

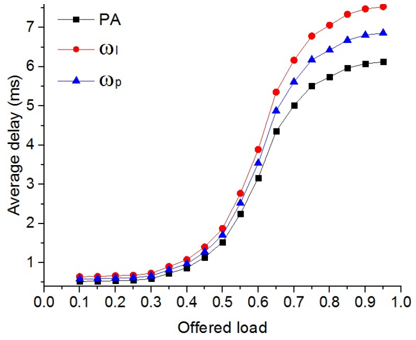

Fig. 5. Average delay of PA with different weight factor.

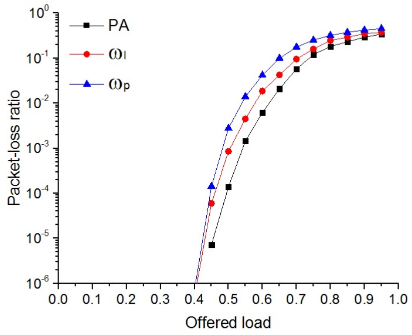

Fig. 6. Packet loss ratio of PA with different weight factor.

that w_p decides the priority and w_l is set as zero, then a VoQ containing a small number of packets will have a low priority and less chance to be transmitted. But, this VoQ may buffer elephant flows and this leads to dropping of new in-coming packet due to limited buffer size. This also confirms the result in Fig. 5 because only considering w_p will result in more packets to be dropped, then it can reduce average delay on the other hand.

V. Conclusion

In this study, we first introduced the Packet-Switched Optical Network (PSON) with centralized controller for datacenter interconnection. According to the PSON architecture and datacenter traffic-flow characteristics, we proposed a priority-aware (PA) scheduling scheme by enhancing the traditional round robin (RR) algorithm. We numerically investigated PA algorithm compared with RR algorithm through simulation experiments. Performance analyses of PA algorithm showed a packet-loss ratio lower than 10^{-6} and an average delay of 900 ns for offered loads up to 0.4 and buffer size of 16 KB for PSON architecture. We also explored the impact of weight factors w_p and w_l on performance of PA algorithm implemented in PSON architecture. Packet length factor w_p played a more important role than w_l, and a reasonable combination of the two factors can return the best performance.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [B0117-16-1008, Development of Data Center Optical Networking Core Technologies for Photonic Frame based Packet Switching].

References

[1] Cisco Global Cloud Index: Forecast and Methodology, 2012–2017, Cisco Systems, Inc., San Jose, CA, USA, 2014.

[2] (2012). Available: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/hw/nexus7000/installation/guide/n7k_hig_book.html.

[3] C. Kachris, et al., "A survey on optical interconnects for data centers," *IEEE Commun. Surveys & Tutorials*, vol. 14, no. 4, pp. 1021–1036, 2012.

[4] D. Lucente, et al., "Low-latency photonic packet switches with large number of ports," Networks and Optical Communications (NOC), *16th European Conference on. IEEE*, pp. 5-7, 2011.

[5] A. K. Kodi, et al., "Energy-efficient and bandwidth reconfigurable photonic networks for high-performance computing (HPC) systems," *IEEE J. Sel. Top. Quantum Electron.*, vol. 17, no. 2, pp. 384–395, 2010.

[6] K. Xi, et al., "Petabit optical switch for data center networks," Polytechnic Institute of New York University, New York, Tech. Rep., 2010.

[7] O. Liboiron-Ladouceur, et al., "Energy-efficient design of a scalable optical multiplane interconnection architecture," *IEEE J. Sel. Top. Quantum Electron.*, vol. 17, no. 2, pp. 377–383, 2010.

[8] H. Takahasi, et al., "Transmission characteristics of arrayed waveguide N × N wavelength multiplexer," *IEEE/OSA J. Lightw. Technol.*, vol. 13, no. 3, pp. 447–455, Mar. 1995.

[9] H. Mehrvar, et al., "Scalable photonic packet switch test-bed for datacenters," *Optical Fiber Communication Conf.*, Los Angeles, CA, pp. W3J-4, 2016.

[10] T. Benson, et al., "Understanding data center traffic characteristics," *Comput. Commun. Rev.*, vol. 40, no. 1, pp. 92–99, 2010.

[11] S. Kandula, et al., "The nature of datacenter traffic: measurements & analysis," *Proc. of the 9th ACM SIGCOMM Internet Measurement Conf. (IMC'09)*, pp. 202–208, 2009.

[12] R. V. Rasmussen, et al., "Round robin scheduling–a survey," *European J. of Operational Research*, vol. 188, no. 3, 2008.

[13] E. L. Hahne, et al., "Round-robin scheduling for max-min fairness in data networks," *IEEE J. on Selected Areas in Commun.*, vol. 9, no. 7, pp. 1024-1039, 1991.

[14] R. Sinha, et al., "Internet packet size distributions: some observations," USC/Information Sciences Institute, Los Angeles, CA, Tech. Rep. ISI-TR-2007- 643, May 2007.