POXN: A New Passive Optical Cross-Connection Network for Low-Cost Power-Efficient Datacenters

Wenda Ni, *Member, IEEE*, Changcheng Huang, *Senior Member, IEEE*, Yunqu Leon Liu, Weiwei Li, Kin-Wai Leong, and Jing Wu, *Senior Member, IEEE*

***Abstract*—Passive optical devices, characterized by low cost, zero energy consumption, and high reliability, are essential building blocks for today's telecom network infrastructure, permeating from conventional backbone transport networks towards next-generation broadband access networks. Motivated by the striking features of passive optical devices, in this paper, we seek their potential applications in emerging datacenter networks to tackle the scalability challenges arising from cost and power. Specifically, we propose passive optical cross-connection networks (POXNs) that enable cost-saving, power-efficient, and reliable communication within datacenters. To support POXNs in warehouse-scale datacenters, we address physical-layer scalability challenges by using advanced interconnection techniques. Next, we propose a distributed polling protocol to address link-layer issues that arise from the broadcast nature of the medium. The performance of our protocol is studied through analysis and simulation. In particular, we develop an analytical model to compute lower and upper bounds on the expected delay of a packet. Numerical results show that the mean packet delay is equal to the lower bound in one regime, while converges to the upper bound in the complementary regime. Results also show that our protocol can achieve high bandwidth efficiency (no less than 85% in our studied case). Additionally, we demonstrate that our protocol can embrace scheduling algorithms that support fairness and QoS. Last, we sketch the roles POXNs can play in various datacenter network architectures in terms of capital and operational cost reductions.**

***Index Terms*—Coupler fabric, datacenter networks, delay analysis, optical interconnects, optical networks, polling protocol.**

I. Introduction

EXISTING and emerging Internet applications, such as web search, video streaming, social networking, etc., are migrating towards the cloud computing paradigm. In this new context, user applications are run over a common datacenter infrastructure, which consists of tens to hundreds of thousands of servers interconnected by switches and/or routers. To leverage the rich computing resources, advanced computing techniques (e.g., MapReduce) are being widely used. Each application job is partitioned and assigned to various servers, which go far beyond the number a single server rack can hold. To enable local computation, extensive data exchanges are made among servers, contributing to tremendous amount of communication traffic within a datacenter. Also, the use of advanced computing techniques causes datacenter traffic to be highly dynamic and unpredictable at both large and small timescales. Such traffic characteristics have been commonly observed in recent measurement studies [1]–[3]. To support such type of traffic, server interconnection networks, generally referred to as datacenter networks, should be designed with high bandwidth and low latency. However, interconnecting a massive number of servers with strict bandwidth and latency requirements is a significantly challenging task [3]–[5].

Optical interconnection technologies, characterized by ultra-high capacity and extremely-low power consumption, offer natural and fundamental solutions to addressing the limitations of its electronic counterpart. Specifically, many optical components are transparent to signal formats and bit rate, and thus are immediately ready for higher transmission rate. Moreover, as signal is not processed on a per-packet basis, power consumption is reduced by orders of magnitude as compared to electronic processing. It is therefore of great importance and necessity to explore the role of optics in scaling communication bandwidth and reducing power consumption within a datacenter. Some recent works advocated the use of optical circuit switching [6]–[11]. Due to the slow switching time of commercially available optical circuit switches, optical circuit switching is more suited for slowly-varying or delay-insensitive traffic with aggregate bandwidth. Hybrid electronic/optical interconnection architectures (e.g., Helios [6], c-Through [8], Mordia [10]) thus seem to be the optimal candidates in today's environment so that the two technologies can best complement each other. Optical circuit switches were used either to replace a fraction of core electronic switches [6], [7], [11], or to interconnect edge ToR switches alongside the existing electronic architecture [8], [10]. To adapt to the changing traffic patterns, dynamic circuit reconfigurations are essential. However, due to the unpredictability of datacenter traffic, it is challenging to determine when and how circuit reconfigurations should be made. Another body of works [12], [13] employed optical packet switching, which was previously targeting telecom applications. Due to the difficulties in optical buffering, sub-wavelength switching requires very complex system and control for contention resolution, and thus greatly neutralizes the fundamental benefits of optical technologies. Both optical circuit and packet switching systems involve

Manuscript received October 17, 2013; revised November 29, 2013; accepted December 16, 2013. Date of publication January 1, 2014; date of current version March 17, 2014.

W. Ni, C. Huang, and W. Li are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: wendani@sce.carleton.ca; huang@sce.carleton.ca; liww@sce.carleton.ca).

Y. L. Liu and K.-W. Leong are with Viscore Technologies Inc., Ottawa, ON K2K 2E2, Canada (e-mail: leon.liu@viscore.com; kinwai.leong@viscore.com).

J. Wu is with the Communications Research Centre Canada, Ottawa, ON K2H 8S2, Canada (e-mail: jingwu@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2013.2295599

active optical devices for dynamic reconfigurations in response to changing traffic. Their costs are too high to be commercially applicable to datacenters.

In this paper, we propose a passive optical cross-connection network (POXN) for datacenter applications. Specifically, $N \times N$ optical coupler fabrics are used instead of the currently proposed optical circuit or packet switching systems to construct optical cross-connection networks within a datacenter. The use of passive rather than active optical devices leads to drastically lower hardware cost, higher power efficiency, higher system reliability, and lower maintenance complexity. More significantly, as signal is transmitted in a broadcast-and-select fashion, coupler fabrics handle traffic variations with no need for real-time reconfigurations, naturally overcoming the major drawback of optical circuit switches. Also, they are ideal for multicast and incast traffic patterns, which are prevalent in datacenter environment. We present devices and transmission system for passive optical cross-connects, highlighting the challenges arising from physical limitations. Due to the broadcast nature of passive optical cross-connects, we propose a distributed multiple access protocol to coordinate frame transmission for collision avoidance among different ports. The performance of the protocol is evaluated through theoretical analysis and simulation. In the theoretical part, we focus on the packet delay performance, and develop an analytical model that computes lower and upper bounds on the expected delay of a packet. Last, we briefly discuss how passive optical cross-connects can be applied in various datacenter network scenarios.

The remainder of this paper is organized as follows. In Sections II and III, we propose the physical-layer system and the link-layer protocol for POXNs, respectively. We develop the packet delay model for our protocol in Section IV. Numerical results are presented in Section V. In Section VI, we discuss the application scenarios of POXNs in datacenter networks. We highlight our contributions by comparing with the existing works in Section VII. We conclude the paper in Section VIII.

II. Passive Optical Cross-Connects

We take a passive approach to construct optical cross-connection networks within a datacenter. The key optical device used for cross-connection is the $N \times N$ wavelength-independent coupler fabric, which has N inputs and N outputs. Optical power from each input is equally divided among the N outputs so that no reconfiguration is needed to set up a circuit between an input and an output. This enables passive optical cross-connects to support various types of traffic. The price to be paid is reduced bandwidth efficiency as spatial wavelength reuse is not possible among different input-output pairs. To fully appreciate the bandwidth benefits of optical technology, wavelength-division multiplexing (WDM) is employed at each input of the coupler fabric. Due to the cost concerns, coarse WDM (CWDM) is advocated in the current design. With the current CWDM technology, we assume that each wavelength operates at 10 Gb/s, and each port supports 18 wavelengths. The total bandwidth capacity is therefore 180 Gb/s.

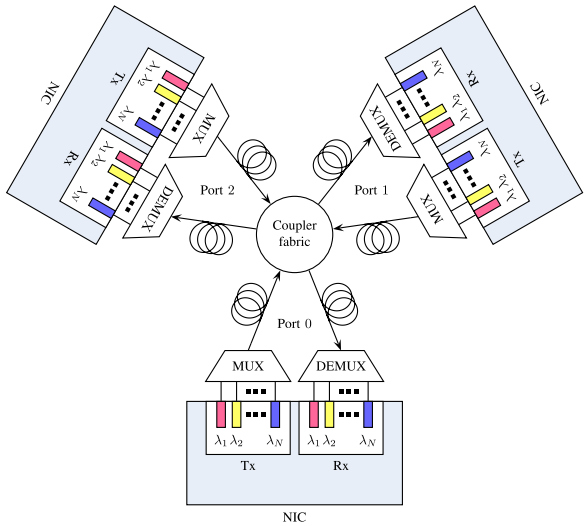

Fig. 1. Physical layer of a sample POXN.

Fig. 1 depicts the resulted transmission system cross-connected by a coupler fabric. Each network interface card (NIC) has a transmitting port and a receiving port. Each transmitting port is equipped with an array of fix-tuned transmitters all working at different wavelengths. Different wavelengths are carried on different fibers initially. All the wavelength signals are then combined into one fiber through a multiplexer, which sits in the middle of the transmitter array and the coupler fabric. The receiving port is the reverse of the transmitting port, capable of receiving all of the wavelengths. The receiving port has a similar structure as its transmitting counterpart, where the multiplexer and the transmitter array are replaced by the de-multiplexer and the receiver array, respectively. To enable accurate clock synchronization and delay measurement, we assume that transmitters and receivers with one NIC have the same fiber distance to the coupler fabric so that propagation delay from a source NIC to a destination NIC is the same as that from the destination back to the source. This assumption is typically satisfied in real systems.

Since no active device is involved in the system, signal amplification or regeneration is not available in the optical domain. Consequently, power budget becomes the deciding factor in port density of the coupler fabric. To calculate the power budget for the coupler fabric, we first need to decide the power loss from the rest part of the system, which mainly includes multiplexer/de-multiplexer insertion loss and fiber transmission loss. Specifically, power loss caused by a multiplexer/de-multiplexer is around 2.5 dB for CWDM based on thin-film filters. Link budget for each fiber segment between a multiplexer/de-multiplexer and the coupler fabric is 3 dB, assuming unit loss of 0.3 dB/km and a length of 10 km, which is the maximum reach requirement within a datacenter [14]. Note that in most intra-datacenter cross-connect cases, fiber length should be much shorter than 10 km, say 1 km. At the system level, optical signal from a transmitter to a receiver travels through one multiplexer, one de-multiplexer, and maximally 20-km length of fiber, causing a power loss of 11 dB. Given a system power budget of 35 dB,

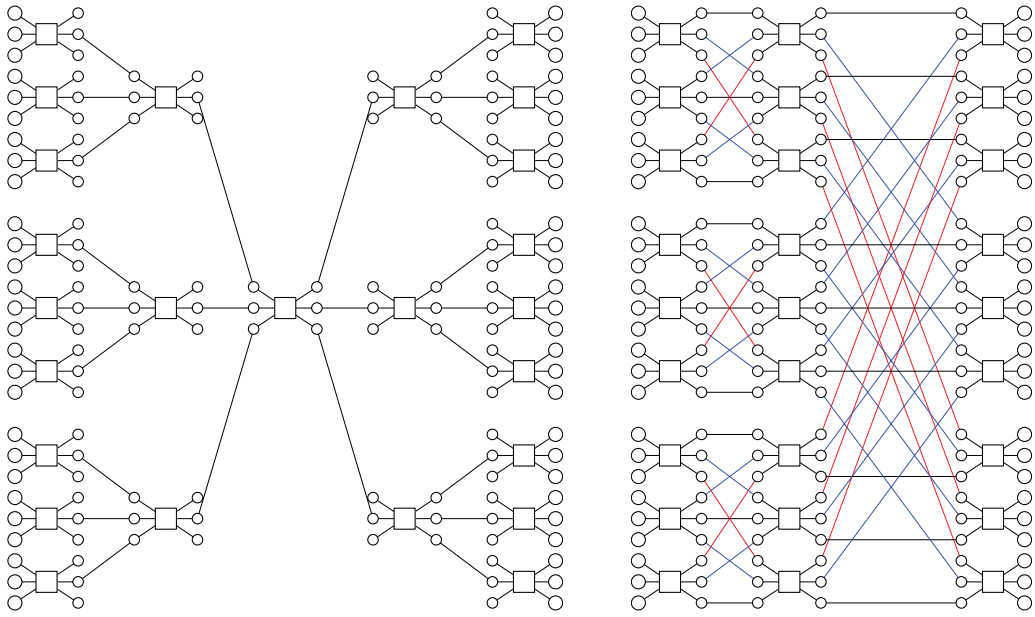

Fig. 2. Interconnection topology within a 27×27 coupler fabric. (a) Native combine-and-split topology. (b) Banyan topology.

which is feasible with the existing optical technology, power budget for the coupler fabric can be 24 dB.

We propose to build an $N \times N$ coupler fabric using 4.77-dB 3×3 unit couplers, which can be manufactured with 0.5-dB excess loss based on the existing technology. The most intuitive way to construct it, as shown in Fig. 2(a), is to first combine all the input signals into one single signal and then split the signal power equally among all the outputs, both using 3×3 couplers. For N inputs and N outputs, such design requires $2 \cdot \lceil \log_3 N \rceil - 1$ stages of 3×3 couplers. Consider the splice loss for interconnecting adjacent stages to be 0.2 dB. Then, power loss at each output is $5.47 \cdot (2 \cdot \lceil \log_3 N \rceil - 1) - 0.2$ dB, where 5.47 is calculated as the sum of fundamental power split loss (4.77 dB), excess loss (0.5 dB), and fiber splice loss (0.2 dB). Consequently, for 24-dB power budget, port count N can only go up to 9. The problem with this design is that two-thirds of the power is completely wasted during each combining or splitting operation.

A more efficient design is to interconnect 3×3 couplers using a Banyan or baseline topology so that all the input power is delivered to the outputs with equal power splitting. Fig. 2(b) shows the Banyan topology for building a 27×27 coupler fabric from 3×3 couplers. Compared with the previous design, stages of 3×3 couplers are reduced from 5 to 3. More generally, for N inputs and N outputs, only $\lceil \log_3 N \rceil$ stages of 3×3 couplers are required using the Banyan topology, corresponding to a power loss of $5.47 \cdot \lceil \log_3 N \rceil - 0.2$ dB at any output. With the 24-dB power budget, port count of a coupler fabric can scale up to $N = 81$. The exact power loss at each output is 21.68 dB. Note that if 3.01-dB 2×2 couplers with the same excess loss are used to construct a coupler fabric, port count of the coupler fabric can go up to $N = 64$, with the exact power loss to be 22.06 dB at an output. It should be noted that manufacturing a unit coupler with more than 3×3 ports is not trivial with the existing technology.

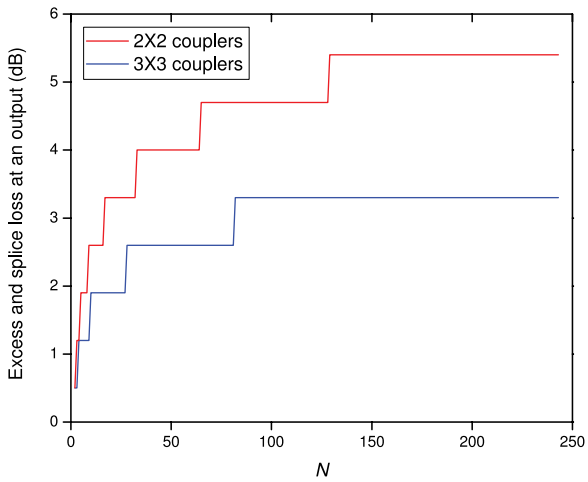

Fig. 3. Additional (i.e., excess and splice) power loss of an $N \times N$ coupler fabric based on building blocks of 2×2 or 3×3 couplers.

The power loss at an output consists of fundamental loss caused by power split, and additional loss introduced by coupler excess loss and fiber splice loss. The contribution of additional loss is $0.7 \cdot \lceil \log_3 N \rceil - 0.2$ dB and $0.7 \cdot \lceil \log_2 N \rceil - 0.2$ dB for using 3×3 and 2×2 unit couplers, respectively. We plot the curves in Fig. 3. It is clear to see that 3×3 couplers are better building blocks than 2×2 couplers from the excess and splice loss viewpoint. However, the choice of N is limited to the powers of 3 if 3×3 couplers are used exclusively. To have more flexibility, a combination of 3×3 and 2×2 couplers can be used so that N can be any integer number with and only with prime factors of 3 and/or 2 (e.g., $N = 48$).

III. High-Efficiency Distributed Access (HEDA) Protocol

Due to the broadcast property of coupler fabrics, bandwidth resources are shared among all the inputs. To avoid frame collisions from different ports, it is essential to develop a multiple access protocol that coordinates port transmissions at the link layer. Consider a total bandwidth capacity of 180 Gb/s. Ideally, a multiple access protocol should enable an average transmission rate of $180/N$ Gb/s per port if all N inputs have data to send, and an instantaneous transmission rate of 180 Gb/s if only one port has data to send. However, control overhead is unavoidable, leading to reduced transmission rate in either case.

A potential candidate to consider is the carrier sense multiple access with collision detection (CSMA/CD) protocol, which had a great success in 10-Mb/s and 100-Mb/s Ethernet for local area networks. For CSMA/CD to work properly, it is required that the transmission time of one Ethernet frame is no less than the round-trip propagation delay so that all frame collisions can be detected. However, the data rate of 10 Gb/s and the worst-case loopback fiber distance of 20 km go far beyond the regime for CSMA/CD to work correctly. Additionally, the bandwidth efficiency of CSMA/CD becomes extremely low in such a context. Both factors prohibit the use of CSMA/CD protocol.

Another candidate to consider is the polling protocol, which is a centralized control protocol, with one of the nodes acting as the master node. The master node sends polling messages to grant all nodes to access the shared medium (i.e., the coupler fabric) one by one and in a cyclic manner. To this end, one prerequisite is the formation of a master-slave hierarchy, which is naturally enabled in systems such as Ethernet passive optical networks (EPONs), WiFi, and cellular networks. Such a hierarchy, however, is not physically favored in our case, where all the ports are homogeneous and peers by nature.

We propose a fully distributed protocol with QoS support by taking advantage of the peer nature of the ports and the broadcast nature of the medium. The protocol, which is named as HEDA protocol, can achieve efficiency higher than polling approach for a master-slave infrastructure while maintain a fully distributed operation to avoid single point of failure. In specific, we divide the protocol into two phases: the discovery phase and the data transfer phase. The discovery phase is designed to achieve plug-and-play objective that will minimize the operation cost of the network. During this phase, ports in a POXN will have a chance to discover other ports in the POXN, establish a common reference clock, synchronize clocks to the reference clock, and measure round-trip and loopback times.

The data transfer phase follows the discovery phase, but takes much longer time to complete than the discovery phase so that the overhead introduced by the discovery phase can be minimized. The data transfer phase is further divided into multiple scheduling cycles. Each port that has been discovered in the discovery phase will have a chance to send a burst of frames within each scheduling cycle. At the end of the burst, the port also advertises the amount of traffic it needs to send during the next cycle. This piggyback approach will allow all other ports to learn its request through the broadcast-based POXN while minimize the bandwidth waste for maintaining inter-port guard intervals between traffic bursts from different source ports. Different from the polling approach, where a master node polls slave nodes, makes a schedule based on the responses, and then notifies slave nodes of their schedules, ports in a POXN can make their own schedules individually and locally after learning all the advertisements from other ports. As long as a common scheduling algorithm is used by all the ports, scheduling conflictions can be avoided. This is similar to the approach used by peer-to-peer networks, where a common hashing function is used by all peers to build a distributed hashing table (DHT) without the need of a centralized server. With the measured loopback times in the discovery phase, clock drifts and propagation delays can be compensated so that schedules from different ports can be fully aligned in time within a cycle. Because we do not need the polling message as in the polling approach, cycles can follow one after another with very little gap. This further reduces the overhead associated with the polling approach.

To accommodate port churns, we allow the discovery phase to repeat after the data transfer phase has been running for long enough time. The repeats of the discovery phase will allow new ports to be discovered, clock re-referenced and resynchronized (in face of clock-reference port failure), and round-trip and loopback times measured again (in face of clock-reference port failure). The period between discovery phases should be long enough to accommodate as many scheduling cycles as possible while be short enough to minimize the delay of a new port joining the network and the impact of clock drifts.

Our protocol can be implemented as a single instance to manage multiple wavelengths as logically one block of resource, or multiple independent instances, one for each wavelength, to achieve better manageability, flexibility and reliability. Both implementation modes have the same bandwidth efficiency. Fig. 4 illustrates the corresponding message sequence chart on one wavelength for the mode of one protocol instance per wavelength. The figure assumes that propagation delay within the coupler fabric is negligible.

A. Discovery Phase

1) Discovery Phase at System Boot: When the system boots, all ports start by listening to the POXN for a period of time. Based on the fact that no transmission is detected within the period, each port can know that the system just boots. Then, all ports start sending ANNOUNCEMENT messages after their random backoff periods to get themselves known by all the other ports while minimize message collisions. An ANNOUNCEMENT message carries the MAC address of the sender port, the timestamp at which the message is transmitted, the time period for the discovery window, and the amount of traffic to be sent in the first scheduling cycle. While transmitting, a port also listens for ANNOUNCEMENT messages at its receiving side. These messages include the one sent by itself. Without transmission coordination, message collisions are likely to occur. Collided messages are detected at the MAC layer of a port's receiving

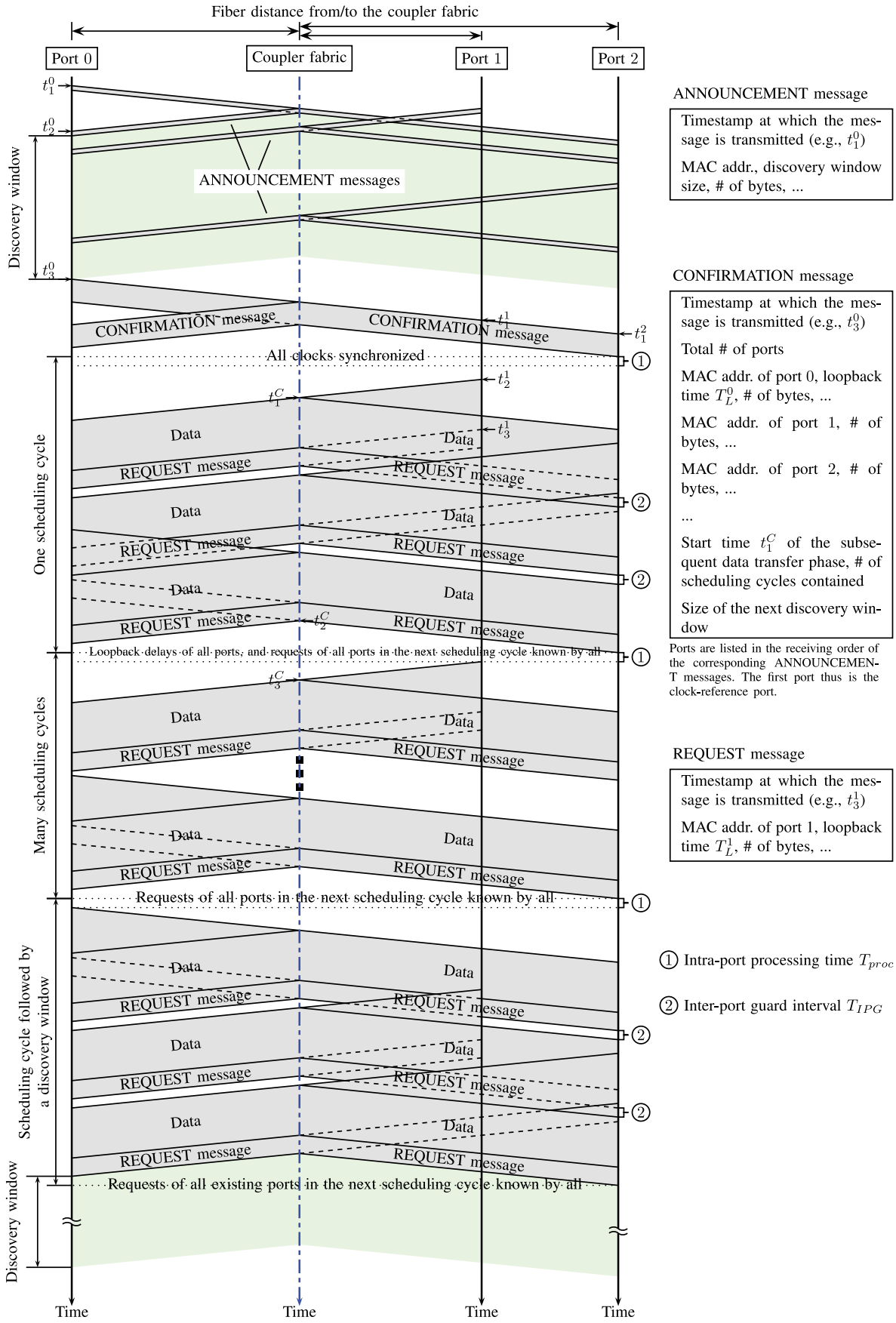

Fig. 4. A sample message sequence chart on a wavelength plane. HEDA protocol is implemented as one instance per wavelength.

side. Similar technique has been used in EPON [15]. Persistent collisions are avoided by imposing a random backoff before each ANNOUNCEMENT message is sent.

The discovery window starts with the correct reception of the first ANNOUNCEMENT message by all ports, and lasts as specified in that message. Since each port hears exactly the same message flow from the channel, a port can safely infer that the first ANNOUNCEMENT message successfully received at its local receivers is also the first ANNOUNCEMENT message successfully received by all the other ports so that the discovery window is globally agreed upon. If multiple wavelengths are managed by one single protocol instance, messages on different wavelengths may be received at the same time. In this case, the same tiebreak policy is used by all ports to select the same first message.

All ANNOUNCEMENT messages must be received within the discovery window by any port. Due to the propagation delay from port to port, the actual window allowed for transmitting ANNOUNCEMENT messages, denoted by t_{TW}, is smaller than the announced discovery window t_{DW}. Since port-to-port propagation delays are unknown at this point, the worse-case propagation delay $T^{PP}_{MAX} = 100\,\mu$s, which is over the maximum fiber distance of 20 km, is assumed to approximate a safe transmission window. In other words, we let $t_{TW} = t_{DW} - T^{PP}_{MAX}$. If a self-sent ANNOUNCEMENT message is not received within a certain amount of time (e.g., T^{PP}_{MAX}), a port deems the sent message to have collided with messages from other ports, and triggers a retransmission after a new random backoff period. Such retransmission continues until either its own ANNOUNCEMENT message is properly received at its local receivers (and thus by all the other ports), or the transmission window t_{TW} is over. To allow ANNOUNCEMENT retransmission within one discovery window, discovery window t_{DW} can be set to the size of several worst-case port-to-port propagation delay, say $5 \cdot T^{PP}_{MAX} = 500\,\mu$s. Also, for larger port count, larger discovery window may be needed to reduce collision probability and enable fast convergence. On the other hand, however, discovery window should be kept small to limit control overhead. We leave the problem of rightsizing the discovery window for further work. The successful reception of an ANNOUNCEMENT message indicates the sender port to be known by all the ports. We call such a port to be a successful port. The first successful port decides the discovery window, and its clock also serves as the reference clock for synchronization. In Fig. 4, we assume that Port 0 is the first successful port, and Ports 1 and 2 follow. Port 0 can then decide its loopback time as $T^0_L = t^0_2 - t^0_1$. Similarly, Ports 1 and 2 can decide their corresponding loopback times T^1_L and T^2_L, respectively.

After the time period of the discovery window as announced by the first successful port expires, no other ports are allowed to send anything. The first port will then broadcast a CONFIRMATION message to summarize each known port with information such as the MAC address of the port and the traffic request for the first scheduling cycle. The CONFIRMATION message also announces the starting time t^C_1 of the subsequent data transfer phase, the total number of scheduling cycles contained, and the size of the discovery window for the next discovery phase. Each port can then use the information to decide the round-trip time to the first port and its clock offset from the reference clock. Take Port 1 for example. Let t^{01}_{rtt} and t^1_{os} denote its round-trip time to Port 0 and its clock offset from Port 0, respectively. We have

$$t^{01}_{rtt} = T^0_L + T^1_L.$$

Also, It is easy to establish from the message sequence chart in Fig. 4 that

$$t^0_3 + \frac{1}{2}T^0_L + t^1_{os} = t^1_1 - \frac{1}{2}T^1_L$$

where the values of t^0_3 and T^0_L are carried in the CONFIRMATION message as shown in Fig. 4. It follows that

$$t^1_{os} = t^1_1 - t^0_3 - \frac{1}{2}(T^0_L + T^1_L).$$

Port 1 then uses the offset value to correct its clock. When the CONFIRMATION message is received by all the ports, all the clocks are synchronized to the reference clock.

Each port maintains a list of all the ports that have been discovered in the discovery phase and their corresponding traffic requests in the upcoming first scheduling cycle. Ports are listed in the order that they send their ANNOUNCEMENT messages successfully. With this list, each port can then make a schedule for all the ports to decide when it can send its own traffic and how much it can send. With the assumption that all the ports use the same deterministic scheduling algorithm, it is easy to see that the scheduling result generated locally by each port is globally identical to guarantee that there will be no burst collisions with other ports during the first scheduling cycle. If the CONFIRMATION message is not received, a port will reboot again.

2) Discovery Phase during Normal Operations: Once data transfer starts, each data transfer phase is followed immediately by a discovery phase. During each discovery phase, existing ports do not send ANNOUNCEMENT messages because they are known to each other. They will listen and record information from any ANNOUNCEMENT message generated by a potential new port. When a new port tries to join the POXN, it has to wait for the CONFIRMATION message to learn all the existing ports that are sharing the channel, to obtain the size of the next discovery window, and to know the number of scheduling cycles in the current data transfer phase to further identify the start time of the next discovery phase. This means that a new port has to wait for at least one data transfer phase before getting itself known by others. During the waiting period, it monitors the network, records information from all messages it received, and tracks the scheduling cycles to compute the precise start time for the next discovery window. During the next discovery phase, only the new ports need to send ANNOUNCEMENT messages. This will minimize the collisions.

After the discovery window ends, a CONFIRMATION message will be sent by the first successful port at system boot, or more precisely, the current clock-reference port if possible port failures are considered. The CONFIRMATION message contains information on all the existing and newly joined ports, the

subsequent data transfer phase, and the next discovery phase. All the ports will use this information to plan their future operation events. If a CONFIRMATION message is not received, a port will deem that the clock-reference port happen to fail during the discovery phase, and will then reboot. However, such failure events should be very rare judging the small durations of discovery phases as shown later.

B. Data Transfer Phase

A scheduling algorithm will decide when a port can send and how much it can send. We do not add conditions on what scheduling algorithms can be used as long as all the ports follow the same deterministic scheduling algorithm. In Fig. 4, as an example, we assume the scheduling algorithm decides that the order to send data traffic is Port 1, Port 2, Port 0 in the first scheduling cycle. The data transfer phase then starts with Port 1 sending its own data burst at time $t_2^1 = t_1^C - T_L^1/2$ so that the first bit of the data burst arrives at the coupler fabric precisely at time t_1^C as specified in the CONFIRMATION message. After finishing its data burst, it will advertise its loopback time T_L^1 and its request for the next scheduling cycle as piggyback through a REQUEST message. All the other ports will record the loopback time and the requested amount in their lists for the calculation of the next cycle. Port 2 follows Port 1 to send. Port 2 will schedule its traffic burst in such a way that it arrives at the coupler fabric right after the last bit of the piggyback information sent by Port 1. Port 2 can do so because it knows its propagation delay to the coupler fabric (i.e., $T_L^2/2$). If clocks are not perfect even after correction, a guard time may be necessary to avoid overlaps. After Port 2 finishes its burst, it will advertise its loopback time and request for the next cycle as Port 1. All the other ports will record. This process will continue until the last port finishes its data burst and piggyback request.

When the REQUEST message from the last port is received by all the ports, each port knows the loopback times of all ports and requests of all ports in the second scheduling cycle. Assume that in the second cycle, the transmission order given by the scheduling algorithm is the same as that in the first cycle. Port 1 then starts the next cycle immediately. Ideally, the first bit from Port 1 should arrive at the coupler fabric right after the last bit of the last cycle. However, one challenge is that Port 1 may not have received the piggyback request sent by the last few ports because it needs to send its burst earlier to reduce the gap caused by the propagation delay. Without the piggyback request of the last few ports, it only has partial request information to calculate its schedule for the next cycle. Under this situation, Port 1 may need to wait until the piggyback requests from all the ports are received, as shown in Fig. 4.

The start time of the second cycle t_3^C is computed locally by each port as

$$t_3^C = t_2^C + \frac{1}{2}\max\left\{T_L^0, T_L^1, T_L^2\right\} + T_{\text{proc}} + T_L^1/2$$

where t_2^C denotes the time the last bit of the last cycle arrives at the coupler fabric, $\frac{1}{2}\max\left\{T_L^0, T_L^1, T_L^2\right\}$ is the time for this last bit to propagate from the coupler fabric to the farthest port, T_{proc} is a constant time value set for message processing, algorithm running, etc. at a port, and $T_L^1/2$ is the propagation delay for the first bit of the first transmission port in the current cycle (i.e., Port 1) to arrive at the coupler fabric. Note that the value of t_2^C is given in the previous cycle as the output of the scheduling algorithm. The start time of the subsequent scheduling cycles is determined similarly one after another. Consequently, by listening to the channel starting from a CONFIRMATION message, timing of all the channel access events can be accurately computed throughout the whole data transfer phase. This allows an unknown new port to identify the discovery window, which immediately follows the reception of the last bit of a data transfer phase. The channel-idle gap between cycles can be eliminated by developing more sophisticated scheduling algorithm, which we leave for future research.

At time t_3^0 when the CONFIRMATION message is transmitted, Port 0 only knows its own loopback time. To allow the data transfer phase to start with any port, Port 0 assumes that the farthest port is 10 km from/to the coupler fabric. Let T_{MAX}^{CP} denote the corresponding one-way propagation delay. We have $T_{MAX}^{CP} = 50\ \mu$s. Time t_1^C is thus computed as

$$t_1^C = t_3^0 + T_{\text{trans}}^{CFM} + T_L^0/2 + 2T_{MAX}^{CP} + T_{\text{proc}}$$

where T_{trans}^{CFM} denotes the transmission time of a CONFIRMATION message. Also, since each port knows the loopback times of all the ports at the end of the first scheduling cycle, the transmission order among the ports can change from cycle to cycle thereafter without specifying the arrival time of a cycle's first bit at the coupler fabric as in the first cycle. It is easy to see that the above protocol can support QoS because we do not add any special requirements on the scheduling algorithm as long as all the ports use the same deterministic scheduling algorithm. This makes our algorithm simple and very attractive for different applications.

When a port dies, all the other ports will detect this because the died port will not send its request for the next cycle any more. All the other ports do not need to do anything other than schedule their future cycles assuming that the died port does not have anything to send. If the died port is the clock-reference port, all the other ports still behave the same way until the next discovery phase. During the next discovery phase, all the other ports will remove the failed port from their lists. The second port on the list then becomes the clock-reference port.

IV. Packet Delay Analysis of HEDA Protocol

In this section, we study the packet delay performance under the proposed HEDA protocol. In particular, we develop an analytical model to bound the expected delay of any packet.

A. System Model

Consider a POXN with N ports. Ports are numbered from 0 to $N-1$. Each port maintains a separate queue for each wavelength. Consider HEDA protocol to be running in the mode of one instance per wavelength. All instances are mutually independent. Thus, each wavelength plane can be dealt with individually while all wavelength planes can be modeled identically.

We study packet delay on one wavelength plane. We model a wavelength plane as an $M/G/1$ polling system. Packets (i.e., frames) arrive at port l according to a Poisson process with rate λ/N, where λ denotes the packet arrival rate of the system. All arrival processes are independent, yet are homogeneous (i.e., symmetric) in terms of packet arrival rates. Packet service time statistics are assumed to be homogeneous for all the ports as well. In specific, let $\overline{X}$ and $\overline{X^2}$ denote the first and the second moments of each port's per-packet service (i.e., transmission) time, respectively. All ports' service times are mutually independent. The total load offered to the system is thus given by $\rho = \lambda\overline{X}$. Note that ρ also denotes the proportion of time that the system is in ports' data intervals. Accordingly, ρ/N denotes the proportion of time that the system is in port l's data intervals.

To reduce the analysis complexity, we focus on scheduling algorithms where ports are served in a cyclic manner from l to $(l+1)\text{mod } N$. All ports' data intervals are limited in length by the same maximum transmission window T. Each port's data interval is followed by its reservation interval, which announces its data request for the next polling cycle. In such context, the service discipline turns out to be time-limited gated service.

Unlike the IPACT protocol for EPON where reservation intervals are identical [16], for simple scheduling algorithms, the channel-idle gap between two adjacent polling (i.e., scheduling) cycles is large in HEDA protocol for POXN. This leaves the duration of the last reservation interval of each cycle to be significantly different from the others. Therefore, the results in [17], which assume homogeneous reservation interval statistics, do not apply to our case. To model the packet delay using HEDA, we consider ports to have different statistics for reservation intervals. We let $\overline{V_l}$ and $\overline{V_l^2}$ denote the first and the second moments of port l's reservation interval, respectively. All reservation intervals are independent. Note that the proportion of time that the system is in ports' reservation intervals is given by $1-\rho$. Moreover, since reservation intervals of all the ports appear with the same frequency (i.e., one reservation interval for each port per polling cycle regardless of the length of its corresponding data interval), the proportion of time that the system is in port l's reservation intervals is only determined by the ratio of port l's mean reservation interval length to the total mean reservation interval length per polling cycle. Consequently, it can be formally expressed as $(1-\rho)\frac{\overline{V_l}}{\sum_{l=0}^{N-1}\overline{V_l}} = (1-\rho)\frac{\overline{V_l}}{N\overline{V}}, \forall\, 0 \le l \le N-1$, where we define $\overline{V}$ to be

$$\overline{V} \triangleq \frac{1}{N}\sum_{l=0}^{N-1}\overline{V_l}. \tag{1}$$

As a summary of our polling model, we emphasize that ports are assumed to be symmetric in terms of arrival rates and service time statistics in our model, while in [17], apart from the above assumptions, ports are also assumed to be symmetric in terms of reservation interval statistics. This makes our system model different from that studied in [17]. Therefore, the results in [17] cannot be applied in our more general context.

B. Delay Model

We refer to packet delay, as in [18] and [17], to be the packet queueing delay, which is between the time a packet arrives at a port and the time it starts being transmitted (i.e., the waiting time of a packet in port), and denote it by random variable W. We extend the packet delay model in [17] to the case where reservation interval statistics among ports are different. As we find that the delay model in [17] is not accurate under certain scenarios (e.g., when T is small), we instead develop a model that provides lower and upper bounds on the expected delay for a packet.

We take the system viewpoint as that in [18] and [17]. From such viewpoint, the expected delay for any packet i can be expressed in general as [18, (3.63)]

$$E(W) = E(R) + E(S) + E(Y) \tag{2}$$

where $E(R)$ is the expected residual time for the in-progress packet transmission or reservation interval upon packet arrival, $E(S)$ is the expected service time to transmit packets in system before the transmission of packet i, and $E(Y)$ is the expected duration of all the whole reservation intervals packet i waits before its transmission. Note that when packet i arrives, it sees the transmission of part of a packet or the occurrence of part of a reservation interval. This is considered in $E(R)$, while $E(S)$ and $E(Y)$ consider the transmissions of whole packets and the occurrences of whole reservation intervals, respectively, seen by packet i.

At the system level, $E(R)$ and $E(S)$ given in [18] still hold. Specifically, we have

$$E(R) = \frac{\lambda\overline{X^2}}{2} + \frac{1-\rho}{2N\overline{V}}\sum_{l=0}^{N-1}\overline{V_l^2} \tag{3}$$

and

$$E(S) = E(N_p)\overline{X} = \rho E(W) \tag{4}$$

where $E(N_p)$ denotes the expected number of packets to be transmitted in the system before the transmission of packet i. The second equality in (4) follows from Little's law which dictates that $E(N_p) = \lambda E(W)$ and from $\rho = \lambda\overline{X}$.

Changes in packet delay model lie in the formula for $E(Y)$, which can be written as the superposition of the following three terms [17].

1) $E(\widetilde{Y})$: The expected duration of all the whole reservation intervals packet i waits under the conventional fully gated service discipline, where a reservation interval immediately precedes its corresponding data interval [18].
2) $E(\Delta Y_1)$: When reservation intervals follow rather than precede their corresponding data intervals, a packet waits additional whole reservation intervals as compared to the preceding case. We denote the expected duration of such additional reservation intervals by $E(\Delta Y_1)$.
3) $E(\Delta Y_2)$: In the setting where reservation intervals follow their corresponding data intervals, the expected duration of additional whole reservation intervals a packet waits due to the enforcement of the maximum transmission window T.

Combining all the three terms, we have

$$E(Y) = E(\widetilde{Y}) + E(\Delta Y_1) + E(\Delta Y_2) \tag{5}$$

where $E(\widetilde{Y})$ is expressed as [18, pp. 200–201]

$$E(\widetilde{Y}) = \frac{(N+2-\rho)\overline{V}}{2} - \frac{(1-\rho)\sum_{l=0}^{N-1}\overline{V_l}^2}{2N\overline{V}}. \tag{6}$$

In what follows, we derive the expression for $E(\Delta Y_1)$ and the upper bound for $E(\Delta Y_2)$, which are given, respectively, in Propositions 1 and 2.

Proposition 1: For ports having asymmetric reservation interval statistics

$$E(\Delta Y_1) = (N-1)\overline{V}. \tag{7}$$

Proof: See Appendix A. ∎

When the reservation interval statistics of all the ports are equal, i.e., $\overline{V_0} = \overline{V_1} = \cdots = \overline{V_{N-1}} = \overline{V}$, and $\overline{V_0^2} = \overline{V_1^2} = \cdots = \overline{V_{N-1}^2} = \overline{V^2}$, $E(\Delta Y_1)$ can be reduced to (8) in [17]. In other words, Proposition 1 is a generalization of $E(\widetilde{Y})$ to the case of asymmetric reservation interval statistics.

Next, we establish the upper bound on $E(\Delta Y_2)$. To this end, let q_l denote the probability that port l's data intervals are of length T. Since packet arrival rates, packet service times, and the maximum transmission windows are all symmetric among all the ports, all ports' data interval lengths are of the same probability distribution. In other words, we have $q_0 = q_1 = \cdots = q_{N-1} = q$.

Proposition 2: Consider a fully gated service system with ports' reservation intervals immediately following their data intervals. $E(\Delta Y_2)$ is upper bounded as

$$E(\Delta Y_2) \le \frac{\rho\overline{V}}{T}E(W) - N\overline{V}\cdot q\left(1-\frac{\rho}{N}\right). \tag{8}$$

Proof: See Appendix B. ∎

C. Upper Bound on $E(W)$

Putting it all together in (2), we have

$$E(W) \le \frac{\lambda\overline{X^2} + \frac{1-\rho}{N\overline{V}}\sum_{l=0}^{N-1}\left(\overline{V_l^2} - \overline{V_l}^2\right)}{2\left(1-\rho-\frac{\rho\overline{V}}{T}\right)} + \frac{\left[\frac{3-\frac{\rho}{N}}{2} - q\left(1-\frac{\rho}{N}\right)\right]N\overline{V}}{1-\rho-\frac{\rho\overline{V}}{T}}. \tag{9}$$

Clearly, for $E(W)$ to be bounded, it is required that the denominator in (9) is always positive, i.e.,

$$\frac{\rho}{N}\cdot\frac{N\overline{V}}{1-\rho} < T \tag{10}$$

where the left-hand side of the inequality actually gives the expression for the mean length of port l's data intervals. In specific, $N\overline{V}/(1-\rho)$ calculates the mean cycle length, and ρ/N computes the proportion of time that the system is in port l's data intervals. Equation (10) dictates that for the packet delay to be bounded, the mean duration of port l's data intervals should be less than the maximum transmission window T.

To numerically compute the upper bound on $E(W)$ using (9), we must have the value for q, the calculation of which is given as follows. Recall that q denotes the probability that port l's data intervals are of length T. Thus, the complementary probability $1-q$ denotes the probability that port l's data intervals are of length less than T. Accordingly, we let $\overline{T_l'}$ denote the mean duration of port l's data intervals that are of length less than T. Since packet arrival rates, packet service times, and the maximum transmission windows are symmetric among all the ports, the values of $\overline{T_0'}$, $\overline{T_1'}$, …, and $\overline{T_{N-1}'}$ are all equal, i.e., $\overline{T_0'} = \overline{T_1'} = \cdots = \overline{T_{N-1}'} = \overline{T'}$. Using q, T, and $\overline{T'}$, the mean length of port l's data intervals can be expressed as

$$\frac{\rho\overline{V}}{1-\rho} = qT + (1-q)\,\overline{T'}. \tag{11}$$

Since $\overline{T'} \le T$ and $0 \le q \le 1$, the mean length of port l's data intervals takes value in the interval $[\overline{T'}, T)$, i.e.,

$$\overline{T'} \le \frac{\rho\overline{V}}{1-\rho}. \tag{12}$$

Due to the difficulty to obtain the exact value for $\overline{T'}$, we assume that the duration of port l's data intervals with length less than T follows a uniform distribution on the interval $[0, \rho\overline{V}/(1-\rho)]$. Consequently, we have

$$\overline{T'} = \frac{1}{2}\cdot\frac{\rho\overline{V}}{1-\rho}. \tag{13}$$

Introducing (13) into (11), we calculate q as

$$q = \frac{1}{\frac{2T(1-\rho)}{\rho\overline{V}} - 1}. \tag{14}$$

Note that when port l's mean data interval length approaches the maximum transmission window T, i.e., when $\rho\overline{V}/(1-\rho) \to T$, we know from (11) that $q \to 1$ and $1-q \to 0$. In this regime, we can safely rewrite (11) by ignoring the term $(1-q)\,\overline{T'}$ as

$$\frac{\rho\overline{V}}{1-\rho} \approx qT. \tag{15}$$

In other words, in the regime that $\rho\overline{V}/(1-\rho) \to T$, q can be calculated without any assumption as

$$q \approx \frac{\rho\overline{V}}{(1-\rho)T}. \tag{16}$$

Also note that the boundary condition in (10) can be generalized straightforward to the case of ports having different arrival rates and different maximum transmission window sizes. However, the condition becomes port-dependent explicitly. In specific, for packets from port l to have bounded delay, it is required that

$$\rho\frac{\lambda_l}{\lambda}\cdot\frac{N\overline{V}}{1-\rho} < T_l, \quad \forall\, 0 \le l \le N-1 \tag{17}$$

where λ_l and T_l denote port l's packet arrival rate and maximum transmission window size, respectively. We will use (17) to

TABLE 1
TIME SPECIFICATION FOR EACH OPERATION PERIOD

Operation period	Time
Discovery window t_{DW}	500 μs
Inter-port guard interval T_{IPG}, which includes laser off, laser on, automatic gain control (AGC), clock and data recovery (CDR), and code-group alignment intervals	2000 ns
Intra-port processing time T_{proc}	10 ns
T_{MAX}^{CP}, worst-case propagation delay from a port to the coupler fabric, or from the coupler fabric to a port	50 μs

discuss the cases of asymmetric arrival rates and/or asymmetric maximum transmission window sizes in numerical results.

D. Lower Bound on $E(W)$

From (9) and the boundary condition in (10), we can see that as T approaches the mean length of port l's data intervals, port l's packet delay increases drastically towards infinity due to the more cycles a packet waits before its transmission. Accordingly, in the opposite direction, port l's packet delay decreases as T gets larger, and when T is larger enough than port l's mean data interval length, the enforcement of T has little impact on port l's packet delay. In this case, the service discipline reduces to the fully gated one, where ports' reservation intervals immediately follow their corresponding data intervals. In other words, the expected delay for a packet is lower bounded by that experienced under the fully gated service discipline described above, i.e.,

$$E(W) \geq E(W_g) \tag{18}$$

where we let $E(W_g)$ denote the expected delay under the fully gated service. The expression for $E(W_g)$ can be written straightforward based on (2), (4), and (5) as

$$E(W_g) = E(R) + \rho E(W_g) + E(\widetilde{Y}) + E(\Delta Y_1). \tag{19}$$

Introducing (3), (6), and (7) into (19), and further introducing (19) into (18), we establish the lower bound on $E(W)$ as

$$E(W) \geq \frac{\lambda \overline{X^2}}{2\,(1-\rho)} + \frac{\sum_{l=0}^{N-1}\left(\overline{V_l^2} - \overline{V_l}^2\right)}{2N\overline{V}} + \frac{\left(3 - \frac{\rho}{N}\right) N\overline{V}}{2(1-\rho)}. \tag{20}$$

Note that as is discussed earlier, the mean packet delay is close to the lower bound when T is larger enough than port l's mean data interval length.

V. NUMERICAL RESULTS

We evaluate our HEDA protocol over an 8-port POXN system through simulation. Each port is connected to the coupler fabric by a pair of 1-km fibers. For simplicity, we consider the system to have one single wavelength, which operates at 10 Gb/s and is managed by our protocol. Simulation environment is developed using OPNET Modeler.

The values of system parameters we set are based on existing EPON technology, as given in Table I. We assume that all the control messages (i.e., ANNOUNCEMENT, CONFIRMATION, and REQUEST messages) are 128 bytes in length, which corresponds to 0.1024 μs transmission time at 10 Gb/s wavelength line rate. All the above parameters are deterministic with variance being zero. Therefore, all reservation intervals are deterministic in length, i.e., $\overline{V_l^2} = 0\ (\text{s})^2, \forall\, l = 0, 1, \ldots, 7$. We assume a simple scheduling algorithm that does not eliminate channel-idle gaps between scheduling cycles, as shown in Fig. 4. Thus, the reservation interval length for port 0 to 6 is computed as $\overline{V_l} = T_{\text{trans}}^{REQ} + T_{IPG} = 2.1024\ \mu\text{s}, \forall\, l = 0, 1, \ldots, 6$, while the reservation interval length for port 7 is calculated as $\overline{V_7} = T_{\text{trans}}^{REQ} + \frac{10^3}{2\times 10^8} \times 2 + T_{\text{proc}} = 10.1124\ \mu\text{s}$, which is composed of the transmission time for one REQUEST message, the 1-km propagation delay from the coupler fabric to port 7, a constant processing time T_{proc} within port 7, and the 1-km propagation delay from the first transmission port of the next cycle to the coupler fabric.

Frames arrive at each port according to a Poisson process. All arrival processes are independent. We assume frames to have Ethernet format. Basically, we let the frame size at each port follow the exponential distribution with the mean size to be 1024 byte. To be consistent with the Ethernet frame format, we further limit the minimum and the maximum frame sizes to be 64 and 1518 byte, respectively. This causes the frame size at each port to actually follow the truncated exponential distribution in the strict sense. The first and the second moments of the distribution are 624.47 byte and 5.4971×10^5 (byte)^2, respectively. Based on this and assume the interframe gap to be 96 bit time, we obtain the values for the first and the second moments of the per-frame service time as $\overline{X} = 0.5092\ \mu$s and $\overline{X^2} = 0.3615\ (\mu\text{s})^2$, respectively.

The discovery phase is triggered every 20000 scheduling cycles. By multiplying the cycle number by the mean cycle length $N\overline{V}/(1-\rho)$, we can obtain the trigger frequency of the discovery phases when the system is in steady state. In the case of $\rho = 0.85$, the discovery phase is triggered every 3.31s.

In what follows, we study how traffic load (by changing the frame arrival rates) and the maximum transmission window sizes affect the mean packet delay. All simulation results shown are with 95% confidence intervals.

To verify our delay model, we first consider the scenario where both the frame arrival rates and the maximum transmission window sizes are the same (i.e, symmetric) among all the ports, i.e., $\lambda_0 = \lambda_1 = \cdots = \lambda_7 = \lambda$, and $T_0 = T_1 = \cdots = T_7 = T$.

Fig. 5 shows how the mean packet delay in the system changes with the size of T. The total load offered to the system is set to 0.85. Under this load, the mean length of ports' data intervals is 1.7587×10^{-5} s according to the left-hand side of (10). We see that when T gets smaller to approach ports' mean data interval length, the mean packet delay increases drastically towards the upper bound computed by our model. However, such increase trend only holds in a small regime, where T is close to ports' mean data interval length. In the case shown in Fig. 5, this regime is within the interval where T is no greater than the mean data interval length by about 10%. When T moves beyond this small regime, the mean packet delay almost immediately drops to hit the lower bound, which is essentially the expected delay under the fully gated service, and remains constant ever since as T gets larger. This indicates that in the much wider, infinite regime, the requested bandwidth of a port can be fully accommodated

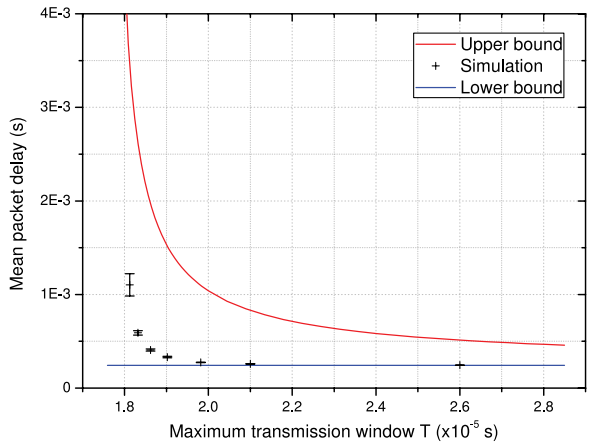

Fig. 5. Mean packet delay taken over packets from all the ports. Ports are symmetric in terms of arrival rates and the maximum transmission window sizes. System load $\rho = 0.85$. Simulation results are with 95% confidence intervals.

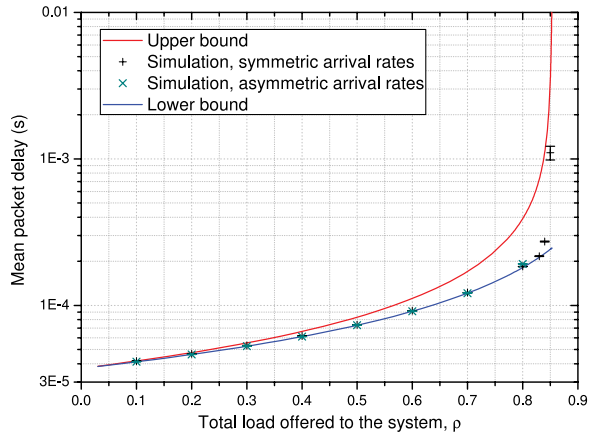

Fig. 6. Mean packet delay taken over packets from all the ports. The maximum transmission window sizes of all the ports are equal, which are set to $T = 1.81216 \times 10^{-5}$ s. Simulation results are with 95% confidence intervals.

almost surely in the next cycle. This, however, does not require T to be significantly larger than ports' mean data interval length. Regarding the analytical model, we observe that in the regime where T is close to ports' mean data interval length, the upper bound increases sharply towards infinity as T approaches that length, while in the complementary regime, the upper bound slowly converges to the lower bound as T increases.

In Fig. 6, we show how the mean packet delay in the system changes with the system load ρ. Note that Fig. 6 also includes results for asymmetric arrival rates, which are left for later discussions in the more general scenario. For now, we focus on the results for symmetric arrival rates. We set T to be 1.81216×10^{-5} s, which is equal to that of the leftmost point in Fig. 5. In this case, the boundary condition in (10) dictates that for the packet delay to be finite, ρ should not exceed $T/(T + \overline{V}) \approx 0.85377$. We see that the mean packet delay is equal to the lower bound as long as ρ is not sufficiently close to the system load limit, say no greater than 0.83. This indicates that even when ρ is slightly smaller than the system load limit (by 3% in Fig. 6), T is actually large enough as compared to ports' bandwidth request so that the requested bandwidth can be fully accommodated almost surely in the next cycle without causing packets to wait additional cycles. Consequently, the system reduces to that under the fully gated service. On the other hand, when ρ is close to the system load limit, the mean packet delay immediately deviates from the lower bound. A slight increase in ρ causes the packet delay to increase sharply towards the upper bound with the trend of converging to it. This indicates that the impact of T becomes very significant when ρ is close to the system load limit. This is because when ρ is close to its upper limit, ports' mean data interval length becomes close enough to T that due to the traffic burstiness, the requested bandwidth in many cycles exceeds T, and thus can only be partially accommodated in the next cycle. This causes part of the packets to wait one or more cycles before transmission. For the analytical model, we observe that the two bounds are close to each other when ρ is small, while the gap becomes larger as ρ increases.

Fig. 6 also shows the bandwidth efficiency of the system. Bandwidth efficiency, or equivalently, normalized throughput, can be calculated as the maximum ratio between a cycle's mean data interval length and the mean length of a cycle, i.e., $\max\left\{\frac{\rho N\overline{V}/(1-\rho)}{N\overline{V}/(1-\rho)}\right\} = \max\{\rho\}$. In other words, the maximum reachable value of ρ indicates the bandwidth efficiency of the system. We see in Fig. 6 that the bandwidth efficiency of our system can be no less than 85%. However, for the system to work at higher bandwidth efficiency regime, the price to be paid is the drastically-increased packet delay. Generally, bandwidth efficiency is decided by the control overhead, such as the reservation intervals per scheduling cycle, the discovery phase, etc. In our case, given T, the value of ρ should be regulated to satisfy the boundary condition in (10). Thus, the bandwidth efficiency is also determined by the maximum transmission window T. Higher bandwidth efficiency can be possibly reached by shortening the fiber length to reduce the control overhead caused by propagation delay and/or by relaxing T.

Next, we study the more general scenario where both the frame arrival rates and the maximum transmission window sizes can be different (i.e., asymmetric) among different ports.

We first consider different frame arrival rates among different ports while leaving T to be 1.81216×10^{-5} s for all the ports. Given the total load ρ, load to port 0 to 7 is assigned according to the fixed ratio 40: 45: 50 : 55: 60: 65: 70 : 75 to determine their frame arrival rates. In the case of asymmetric arrival rates, for packet delay to be bounded, the total load ρ cannot reach as high as that in the symmetric case since the boundary condition changes from (10) to (17), which dictates the maximum ρ to be determined by the maximum arrival rate among all the ports as $\rho < T/(T + N\overline{V} \cdot \max\frac{\lambda_l}{\lambda}), \forall l$. Given T and the load assignment ratio above, system ρ should not exceed 0.81740.

We plot the mean packet delay in the system, taken over packets from all the ports, in Fig. 6. This is to compare the system packet delay with that in the symmetric case. We see that as long as ρ is not very close to the system load limit, say no greater than 0.7, the system-level packet delay is identical in the two cases, while for $\rho = 0.8$, the packet delay in the asymmetric case becomes higher than that in the symmetric case. This is because

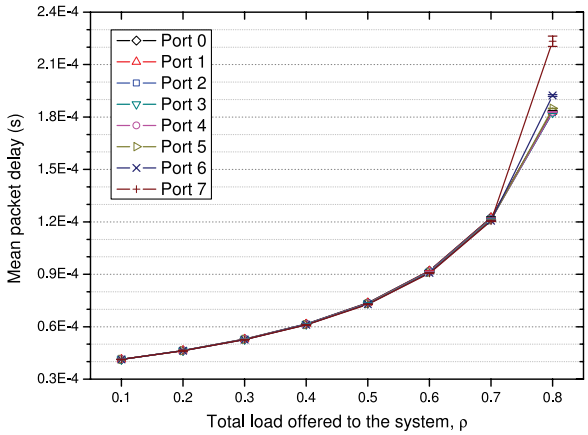

Fig. 7. Mean packet delay at each port with 95% confidence interval. Given the system load ρ, load at port 0 to 7 is distributed according to the fixed ratio $40 : 45 : 50 : 55 : 60 : 65 : 70 : 75$. The maximum transmission window sizes of all the ports are equal, which are set to $T = 1.81216 \times 10^{-5}$ s.

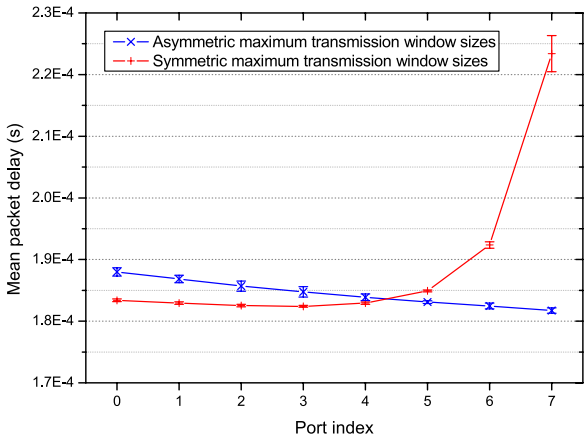

Fig. 8. Mean packet delay at each port with 95% confidence interval. Given the system load $\rho = 0.8$, load at port 0 to 7 is distributed according to the fixed ratio 40: 45: 50 : 55: 60: 65: 70 : 75. In the case of asymmetric maximum transmission window sizes, ports' window size ratio is set the same as their load ratio.

when $\rho = 0.8$, the mean data interval lengths of ports assigned highest or second highest load become sufficiently close to T that bandwidth request from these ports in many cases exceed T, and thus cannot be fully accommodated in the next cycle. Consequently, part of their packets has to wait additional cycles before transmission. This causes the mean packet delay from these ports to increase significantly, which also contributes to (and thus explains) the packet delay increase at the system level shown in Fig. 6.

In Fig. 7, we plot the mean packet delay at the port level. We observe that when ρ is no greater than 0.7, which is not very close to the system load limit, the mean packet delay at all ports is identical, while for $\rho = 0.8$, the mean packet delay at port 7 and port 6, which are assigned the highest and the second highest load, respectively, diverges significantly from that at the other ports, and shows the highest and the second highest packet delay, respectively. This is consistent with the system level performance shown in Fig. 6 (asymmetric arrival rates).

In Fig. 7, the maximum transmission windows of all the ports are assigned the same size. If all the ports are treated equally, such bandwidth assignment is obviously not fair for ports with higher load in the sense that packets from ports with higher load can experience much larger packet delay as shown in Fig. 7 (e.g., $\rho = 0.8$). Thus, we adjust the maximum transmission window sizes among different ports to achieve fairness. One intuitive approach is to let ports' window size ratio be the same as their load ratio such that the ratio $\lambda_l/T_l, \forall\, l$ becomes equal among all the ports. Specifically, given the total window size $8 \times 1.81216 \times 10^{-5}$ s, we assign the maximum transmission window size for port 0 to 7 according to their load ratio 40: 45: 50 : 55: 60: 65: 70 : 75. The effect of the window size adjustment is shown in Fig. 8, where the system load ρ is set to 0.8. We see that the mean packet delay at ports with the highest and the second highest load drops significantly after the adjustment so that the mean packet delay at all ports is within the range from 0.18 to 0.19 ms. This indicates that by smartly adjusting

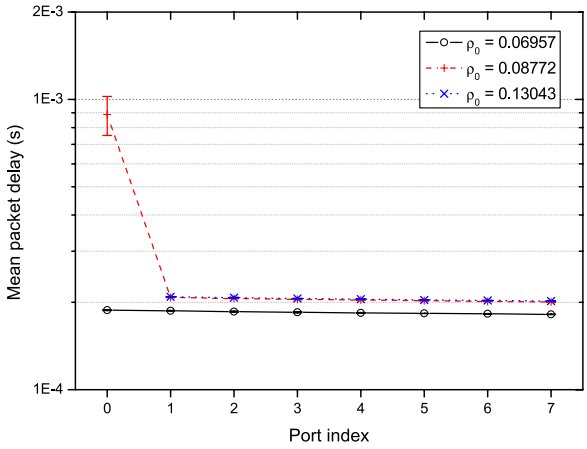

Fig. 9. Mean packet delay at each port in the case that load at port 0, i.e., ρ_0, varies while load at all the other ports remains constant. Results are with 95% confidence intervals.

the maximum transmission window sizes, fairness among ports can be achieved and ensured.

Apart from ensuring fairness, the enforcement of the maximum transmission window sizes can also perform port isolation in the sense that in the case that some of the ports malfunction, such as being overloaded, it guarantees bandwidth to the rest of the ports, and protects their packets from having excessive delay increase. Such benefit is shown in Fig. 9. For the case that all the ports behave properly, we use the same setting as that in Fig. 8, where the traffic load at port 0 is 0.06957. Next, we overload port 0 by increasing its load while leaving the load at all the other ports unchanged throughout the study. We first increase port 0's load from 0.06957 to 0.08772. Accordingly, the system load is increased from 0.8 to 0.81815. Consequently, the ratio between port 0's mean data interval length and its maximum transmission window size is increased from 0.6851 to 0.9501 so that its packet delay is still bounded according to (17). We see in Fig. 9 that the mean packet delay at port 0 increases significantly while the packet delay at all the other ports only increases

slightly. Such slight increase is due to the prolongation of port 0's data interval, which, on the other hand, is limited to its maximum transmission window size, so that packets at all the other ports only wait slightly longer before their transmission. Next, we further increase port 0's load from 0.08772 to 0.13043. In this case, the system load increases from 0.81815 to 0.86086, and the ratio between port 0's mean data interval length and its maximum transmission window size further increases from 0.9501 to 1.8463, which according to (17) dictates the packet delay at port 0 to be infinite. This is confirmed in our simulation that we observe the packet delay at port 0 to be constantly increasing as time elapses. In this case, we only show in Fig. 9 the mean packet delay at all the other ports. We see that the mean packet delay at all the other ports remain identical to that when port 0's load is 0.08772. This is because when port 0's load increases to 0.08772, its data interval lengths in most cases reach its maximum transmission window size. Further increasing its load does not pose impact on the channel or on the packet delay from any of the other ports. Fig. 9 indicates that by enforcing the maximum transmission windows at all the ports, load increase at one or more ports has marginal impact on the packet delay at the rest of the ports.

VI. Datacenter Networks Using POXNs

POXNs can be potentially applied in a variety of datacenter network scenarios. Since the total capacity is shared among all the ports, all applications can be viewed as tradeoffs between the port count and the average transmission rate per port. Generally speaking, if POXNs are placed closer to end servers, where the average transmission rate per port becomes less critical, we can deploy POXNs with larger port counts to cross-connect more servers. On the other hand, if POXNs are placed nearer to the core-switch tier, we should limit the port count to enable higher average transmission rate per port to accommodate more and more aggregated bandwidth. Moreover, the total capacity can be dynamically and arbitrarily split among all the ports so that ports with higher bandwidth requirements (e.g., ports that are connected to higher levels of the switching hierarchy) can be allocated with higher transmission rate. Furthermore, since each wavelength can be managed individually by one protocol instance, different wavelengths can operate at different transmission rates. This allows the capacity to be upgraded on a wavelength basis. Additionally, port capacity can be heterogeneous. Different ports can be equipped with different number of wavelengths. This further allows the capacity upgrade to be among the chosen ports rather than all the ports. Both factors enable smooth and flexible migration strategies.

VII. Related Work

In this section, we highlight our contributions by comparing this study with the existing ones. Our main contributions can be summarized in three aspects: 1) We proposed passive coupler fabric as the switch fabric for communication in datacenter networks; 2) We proposed a distributed polling protocol that enables collision-free frame transmission; and 3) we developed a packet delay model for our protocol, where reservation interval statistics of ports may not be identical. In what follows, we discuss works related to these three aspects.

A. Switch Fabrics

Several recent works have looked at how optical technologies can reshape the landscape of datacenter infrastructure beyond their conventional role of point-to-point communications within a datacenter [14]. One major proposal is the use of optical circuit switching based on optical devices, such as MEMS switches, wavelength selective switches, etc. [6]–[11]. One severe drawback of optical circuit switching is the slow reconfiguration time (on the order of micro- [10], [11] or milliseconds [6], [8]), which makes it not suitable to deal with highly dynamic and highly changing traffic patterns characterized in datacenter environment, where measurement results show that in every millisecond, 100 flows arrive at the server cluster on average, with 80% of them lasting less than 10 s [1]. In particular, the unpredictability of traffic patterns makes dynamic circuit reconfigurations difficult to implement, and thus largely undermines the capacity advantage of optical circuit switching. In our proposal, this slow switching issue is naturally eliminated by using the coupler fabric, where signal is transmitted in a broadcast fashion through an entirely static configuration.

Another important proposal is the use of optical packet switching [12], [13]. Due to the immaturity of optical buffering technology, contention resolution in the optical domain must be handled by introducing very complex optical system and electronic control. This significantly increases the hardware cost, and reduces the system reliability. In contrast, our switch fabric is constituted by only one device—coupler fabric, which is cheap and highly reliable. Moreover, both circuit and packet switch fabrics employ active optical devices, such as MEMS switches, tunable wavelength converters, etc., while our switch fabric is purely passive, and thus fully appreciates the energy benefit of optics.

The most relevant work is the work in [19], where coupler fabric is used to handle multicast traffic patterns, for which it is best suited. For other traffic patterns, different optical devices are employed. Consequently, various optical device modules must be introduced at the physical layer to address various traffic patterns. This makes the physical layer of the system rather complex. Moreover, without traffic profile, it is difficult to predict during the network design phase how many components are required in the device pool. This typically can lead to physical layer overprovisioning. Also, as a hardware-based solution, its flexibility is low. In our approach, we use coupler fabric as the one single type of optical device to cope with all traffic patterns. Agility is enabled at the link layer by dynamic bandwidth adjustment among ports. Since spatial wavelength reuse is not possible among different input-output pairs, one potential bottleneck of using coupler fabrics for unicast traffic is the average transmission rate per port. The bottleneck can be mitigated by scaling the line rate per wavelength beyond 10 Gb/s, and/or increasing the number of wavelength channels using dense WDM technology. Note that the instantaneous transmission rate of one

port can be very high, which is a desirable feature to handle aggregate traffic.

B. Polling Protocols

Polling protocols, characterized by high bandwidth efficiency, are essential building blocks in many access networks, such as WiFi [20], cellular networks, and EPONs [16]. Typical polling protocols are centralized access protocols, with one of the nodes acting as the master node responsible for resource scheduling and collision avoidance. Consequently, we can find that polling protocols are typically developed in networks that have a hierarchical physical structure so that nodes standing at the top of the hierarchy serve as the master node naturally, e.g., base stations in WiFi and cellular networks, and optical line terminal in EPONs. Unfortunately, such master-slave hierarchy is not physically supported in our POXNs, where all the ports are homogeneous by nature. Therefore, existing polling protocols cannot be applied straightforward in our context. Rather, we develop a polling protocol that is distributed in the sense that resource scheduling is made locally at each port. As long as all ports follow the same deterministic scheduling algorithm, frame collisions are avoided. Such concept is similar to that used in peer-to-peer networks, where a common hashing function is used by all peers to construct a DHT without the need of a centralized server.

Note that early works in [21] and [22] introduced and improved random access protocols, namely ALOHA and CSMA, for local area networks that use an optical star coupler to interconnect end users. We choose to develop our protocol as a flavor of polling protocols rather than random access protocols due to the fact that polling protocols generally achieve much higher bandwidth efficiency than random access protocols. Moreover, protocols in [21] and [22] require each user to have a tunable transmitter and a tunable receiver, and thus cannot be applied to POXNs, where fixed-tuned transmitters and receivers are used.

C. Delay Model

Packet delay analysis for polling model under the time-limited service has been extremely challenging, and remains an open problem so far [17], [23]. Following the simple approach in [18], the work in [17] proposed a model that considers all queues (i.e., ports in our context) to be symmetric (i.e., in terms of arrival rate statistics, service time statistics, reservation interval statistics, and the maximum transmission window sizes). We generalize the model in [17] to the case where different ports can have different reservation interval statistics. This is to deal with the asymmetric reservation interval statistics in our polling protocol, where by using simple scheduling algorithms, the reservation interval of one port is significantly larger than the other ones. In such context, the model in [17] cannot be applied. Moreover, we find the model in [17] to be inaccurate when the maximum transmission window size is close to a port's mean data interval length. We developed our model to provide lower and upper bounds that work well in this regime.

VIII. Conclusion

We proposed POXNs for constructing datacenter networks. POXN is characterized by using passive coupler fabrics to play the role of conventional electronic switches or optical circuit switches within a datacenter. The use of passive rather than active devices enables cost-effective, power-efficient, and reliable communication among the interconnected ports. As signal is transmitted in a broadcast-and-select manner, POXNs are ideal for multicast and unicast traffic characterized in datacenters. By using advanced interconnection topologies, coupler fabrics can scale up to 81 ports to support warehouse-scale datacenters. Due to the broadcast nature of coupler fabrics, we proposed a fully distributed polling protocol that enables collision-free data transmission at the link layer. We developed an analytical model to compute lower and upper bounds on the expected packet delay of our protocol, where reservation intervals of different ports can be different. Numerical results show that our protocol can attain high bandwidth efficiency (no less than 85% in our case study). Moreover, we find that the mean packet delay is equal to the lower bound as long as a port's mean data interval length per cycle is not sufficiently close to its maximum transmission window size, while in the complementary regime, the mean packet delay converges to the upper bound. We also demonstrated that our protocol can accommodate scheduling algorithms that support QoS and fairness among different ports. Last, we discussed in principle how POXNs can be incorporated in various datacenter network architectures for cost reduction, power reduction, and/or reliability purposes.

Appendix A

Proof of Proposition 1

We express $E(\Delta Y_1)$ based on conditional expectations. To this end, let $Q_{l,j}$ denote the event that packet i belongs to port l and arrives during the data interval of port $(l+j)\text{mod}N$, $\forall\, 0 \le l \le N-1, 0 \le j \le N-1$. Let $Z_{l,j}$ denote the event that packet i belongs to port l and arrives during the reservation interval of port $(l+j)\text{mod}N$, $\forall\, 0 \le l \le N-1, 0 \le j \le N-1$. It is easy to find that the above events are mutually exclusive with $\sum_{l=0}^{N-1}\sum_{j=0}^{N-1} P(Q_{l,j} \cup Z_{l,j}) = 1$. Thus, given the events $Q_{l,j}$ and $Z_{l,j}$, $E(\Delta Y_1)$ can be written as

$$E(\Delta Y_1) = \sum_{l=0}^{N-1}\sum_{j=0}^{N-1} E(\Delta Y_1 \mid Q_{l,j})P(Q_{l,j}) + \sum_{l=0}^{N-1}\sum_{j=0}^{N-1} E(\Delta Y_1 \mid Z_{l,j})P(Z_{l,j}). \quad (21)$$

To derive $E(\Delta Y_1)$, we find expressions for all the terms on the right-hand side of the equality. We first consider $P(Q_{l,j})$ and $P(Z_{l,j})$. Since packets from all ports arrive with the same rate, the probability that a packet in the system belongs to port l is $1/N$. Moreover, the probability that a packet arrives during the data interval of port l is ρ/N, which is equal to the proportion of time occupied by port l's data intervals. Since the two events

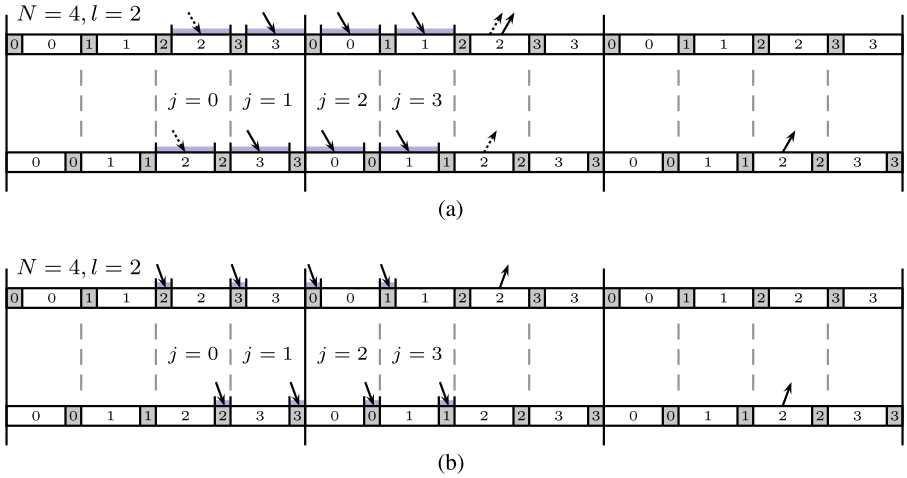

Fig. 10. Arrival and departure for a packet belonging to port 2 in a 4-port gated service system. Each downward arrow denotes one packet arrival case, and upward arrow of the same linestyle denotes the corresponding packet departure. The upper and the lower diagrams in each subfigure show the cases of reservation intervals preceding and following data intervals, respectively, where white-filled frames denote data intervals, and gray-filled frames denote reservation intervals. (a) Packet arrives during the data intervals. (b) Packet arrives during the reservation intervals.

are independent, we have

$$P(Q_{l,j}) = \frac{1}{N} \cdot \frac{\rho}{N}, \quad \forall\, 0 \le l \le N-1, 0 \le j \le N-1. \tag{22}$$

Similarly, we have

$$P(Z_{l,j}) = \frac{1}{N} \cdot (1-\rho)\frac{\overline{V_{(l+j)\bmod N}}}{N\overline{V}}, \quad \forall\, 0 \le l \le N-1, 0 \le j \le N-1 \tag{23}$$

where $(1-\rho)\frac{\overline{V_{(l+j)\bmod N}}}{N\overline{V}}$ denotes the probability that a packet arrives during the reservation interval of port $(l+j)\bmod N$.

Next, we deal with $E(\Delta Y_1 \mid Q_{l,j})$. Recall that $E(\Delta Y_1)$ denotes the mean length of additional (whole) reservation intervals a packet waits due to the shift of reservation intervals from before to after their corresponding data intervals. Accordingly, $E(\Delta Y_1 \mid Q_{l,j})$ denotes the mean additional reservation interval length a packet waits given the event that the packet belongs to port l and arrives during the data interval of port $(l+j)\bmod N$. To obtain the general expression for $E(\Delta Y_1 \mid Q_{l,j})$, we start with a 4-port gated service example shown in Fig. 10(a), which compares the reservation intervals a packet waits between the cases of reservation intervals preceding and following data intervals. Without loss of generality, we assume that the packet belongs to port $l = 2$. It can be seen that if the packet arrives during the data interval of its owner (arrow with dashed line), it waits in both cases the total mean reservation interval length per polling cycle. This is true for any port l. In other words, for $j = 0$, we have in general

$$E(\Delta Y_1 \mid Q_{l,j}) = 0, \forall\, 0 \le l \le N-1, j = 0. \tag{24}$$

On the other hand, if the packet arrives during the data interval of port $(2+j)\bmod N, \forall\, 1 \le j \le N-1$ (arrow with solid line), we can find in Fig. 10(a) that the mean reservation interval lengths it waits are

$$N\overline{V} - \left(\overline{V_{(2+1)\bmod N}} + \cdots + \overline{V_{(2+j)\bmod N}}\right)$$

and

$$N\overline{V} + \left[N\overline{V} - \left(\overline{V_{2\bmod N}} + \cdots + \overline{V_{(2+j-1)\bmod N}}\right)\right]$$

in the cases of reservation intervals preceding and following data intervals, respectively. It follows that for a general port l, we have

$$E(\Delta Y_1 \mid Q_{l,j}) = N\overline{V} - \overline{V_l} + \overline{V_{(l+j)\bmod N}}, \quad \forall\, 0 \le l \le N-1, 1 \le j \le N-1. \tag{25}$$

Last, we deal with $E(\Delta Y_1 \mid Z_{l,j})$. Recall that $Z_{l,j}$ denotes the event that the considered packet belongs to port l and arrives during the reservation interval of port $(l+j)\bmod N$. Similar to the case of $E(\Delta Y_1 \mid Q_{l,j})$, we generalize the expression from an illustrative example. Consider a 4-port gated service system shown in Fig. 10(b). We assume without loss of generality that an arriving packet belongs to port $l = 2$, and compare the (whole) reservation intervals it waits in the cases of reservation intervals preceding and following data intervals. It is easy to find that for $0 \le j \le N-1$, the mean reservation interval lengths the packet waits can be written as

$$N\overline{V} - \left(\overline{V_{2\bmod N}} + \cdots + \overline{V_{(2+j)\bmod N}}\right) + \overline{V_2}$$

and

$$N\overline{V} + \left[N\overline{V} - \left(\overline{V_{2\bmod N}} + \cdots + \overline{V_{(2+j)\bmod N}}\right)\right]$$

in the preceding and the following cases, respectively. Consequently, for a general port l, we have

$$E(\Delta Y_1 \mid Z_{l,j}) = N\overline{V} - \overline{V_l}, \quad \forall\, 0 \le l \le N-1, 0 \le j \le N-1. \tag{26}$$

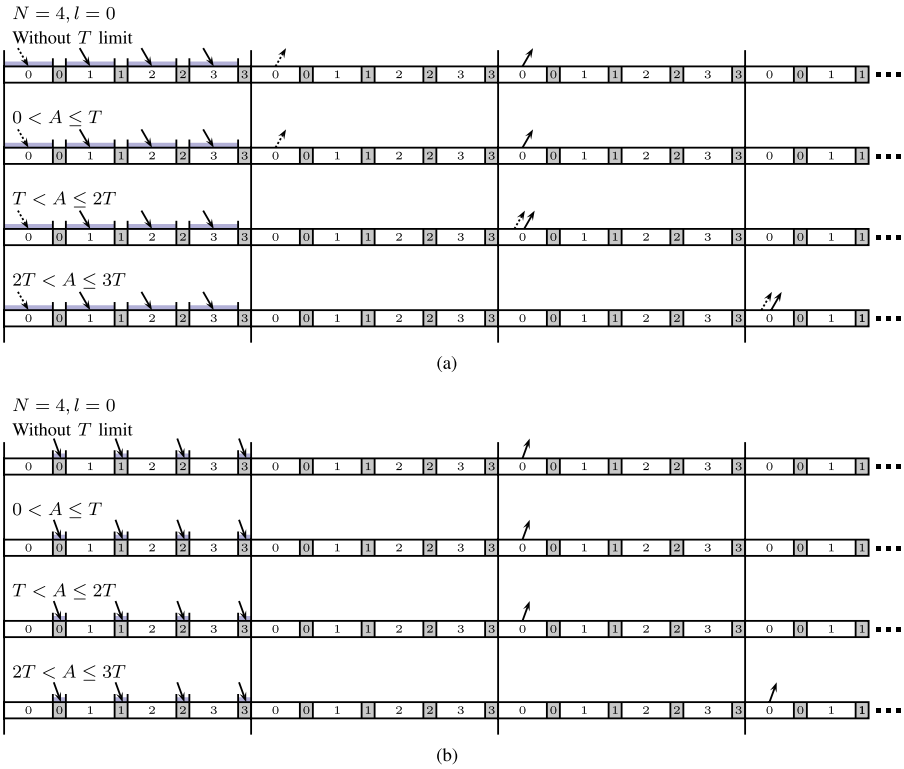

Fig. 11. Arrival and departure for a packet belonging to port 0 in a 4-port gated service system. Each downward arrow denotes one packet arrival case, and upward arrow of the same linestyle denote the corresponding packet departure. The first and the rest diagrams in each subfigure show the cases of without and with T limit, respectively, where white-filled frames denote data intervals, and gray-filled frames denote reservation intervals. (a) Packet arrives during the data intervals. (b) Packet arrives during the reservation intervals.

Introducing (22)–(26) into (21) and rearranging terms yield

$$\begin{aligned} E(\Delta Y_1) &= \sum_{l=0}^{N-1}\sum_{j=1}^{N-1}\left(N\overline{V} - \overline{V_l} + \overline{V_{(l+j)\mathrm{mod}N}}\right)\frac{1}{N}\frac{\rho}{N} \\ &\quad + \sum_{l=0}^{N-1}\sum_{j=0}^{N-1}\left(N\overline{V} - \overline{V_l}\right)\frac{1}{N}(1-\rho)\frac{\overline{V_{(l+j)\mathrm{mod}N}}}{N\overline{V}} \\ &= (N-1)\overline{V} \end{aligned}$$

where we omit the intermediate steps, which only involve elementary mathematical processing, for brevity. This completes the proof.

Appendix B

Proof of Proposition 2

Consider packet i to be a general packet. Without loss of generality, assume that packet i arrives at port l. To find the mean additional reservation interval length packet i waits due to the enforcement of the maximum transmission window T, we compare the (whole) reservation intervals it waits in the cases of without and with T limit. Such comparison is shown in Fig. 11, which indicates the enumeration of the following mutually-exclusive cases: 1) packet i arrives during the data interval of its owner l; 2) packet i arrives during the data intervals other than its owner; and 3) packet i arrives during the reservation intervals. We discuss these three cases one by one in the following. To this end, let random variable B denote the index of port who owns (i.e., sends) packet i. We have

$$P(B = l) = \frac{1}{N}, \quad \forall\, 0 \leq l \leq N-1. \tag{27}$$

Let random variable D denote the index of port during whose data interval packet i arrives. We have

$$P(D = l) = \frac{\rho}{N}, \quad \forall\, 0 \leq l \leq N-1. \tag{28}$$

Let random variable E denote the index of port during whose reservation interval packet i arrives. We have

$$P(E = l) = (1-\rho)\frac{\overline{V_l}}{N\overline{V}}, \quad \forall\, 0 \leq l \leq N-1. \tag{29}$$

Consider the total service (i.e., transmission) time for packet i and packets in queue ahead of i at the instant packet i arrives. Denote this time by random variable A. In case 1, when the total service time is no greater than T, as shown in Fig. 11(a), no additional reservation intervals are incurred in packet i's waiting time as compared to the case without T limit. In other words, in the event

$$\{B = l, D = l, 0 < A \leq T\}, \quad \forall\, 0 \leq l \leq N-1$$

the mean additional reservation interval length packet i waits is 0. On the other hand, when the total service time exceeds T, packet i waits additional reservation intervals. Since the total

service time is greater than T, packet i sees all the preceding data intervals of port l to be approximately T. This allows us to partition the total service time of packet i and packets in queue ahead of i in unit of T, where each T corresponds to one data interval of port l. It can be seen from Fig. 11(a) that for each T packet i waits, the mean duration of additional reservation intervals packet i waits is equal to the total mean reservation interval length per polling cycle, i.e., $N\overline{V}$. Moreover, since at packet i's arrival instant, the data interval of its owner is still in progress, part of the packets ahead of i can be transmitted during that interval. This leaves the queue size ahead of i to be smaller at the end of the data interval so that packet i can be scheduled and thus transmitted one data interval earlier. This, in turn, can reduce packet i's waiting time by one complete set of reservation intervals incurred in one polling cycle. In other words, in the event

$$\{B = l, D = l, kT < A \le (k+1)T\}, \quad \forall\, 0 \le l \le N-1, k \ge 1$$

the mean additional reservation interval length packet i waits is upper bounded by $kN\overline{V}$.

Case 2 is developed in the similar fashion as case 1. It can be seen from Fig. 11(a) that when the total service time is no greater than T, i.e.,

$$\left\{B = l, \bigcup_{j \ne l, 0 \le j \le N-1} (D = j), 0 < A \le T\right\}, \quad \forall\, 0 \le l \le N-1$$

the mean additional reservation interval length packet i waits is 0. On the other hand, when the total service time is greater than T, one can generalize from $k = 1$ and 2 in Fig. 11(a) that in the event

$$\left\{B = l, \bigcup_{j \ne l, 0 \le j \le N-1} (D = j), kT < A \le (k+1)T\right\}, \quad \forall\, 0 \le l \le N-1, k \ge 1$$

the mean additional reservation interval length packet i waits is $(k-1)N\overline{V}$. Note that packets ahead of i can be scheduled in the cycle where packet i arrives, and start transmitting in the cycle that immediately follows with the transmission window being T. The occurrence of the event is almost sure as k gets larger.

In case 3, where packet i arrives during the reservation intervals, if the total service time is no greater than T, i.e.,

$$\left\{B = l, \bigcup_{0 \le j \le N-1} (E = j), 0 < A \le T\right\}, \forall\, 0 \le l \le N-1$$

it is easy to find in Fig. 11(b) that the mean additional reservation interval length packet i waits is 0. On the other hand, if the total service time exceeds T, one can generalize from $k = 1$ and 2 in Fig. 11(b) that in the event

$$\left\{B = l, \bigcup_{0 \le j \le N-1} (E = j), kT < A \le (k+1)T\right\}, \quad \forall\, 0 \le l \le N-1, k \ge 1$$

the mean additional reservation interval length packet i waits is $(k-1)N\overline{V}$. Note that packets ahead of i start transmitting in the cycle that immediately follows the one where packet i arrives. Consider the event that the transmission window of that cycle is of length T. Similar to case 2, the occurrence of the event is almost sure for larger k.

Combining all the three cases, which are mutually exclusive, $E(\Delta Y_2)$ can be expressed as

$$\begin{aligned} &E(\Delta Y_2) \\ &\le \sum_{l=0}^{N-1}\sum_{k=0}^{+\infty} kN\overline{V} \cdot P\left(B = l, D = l, kT < A \le (k+1)T\right) \\ &\quad + \sum_{l=0}^{N-1}\sum_{k=1}^{+\infty}(k-1)N\overline{V} \\ &\quad \cdot P\left(B = l, \bigcup_{i \ne l, 0 \le i \le N-1} (D = i), kT < A \le (k+1)T\right) \\ &\quad + \sum_{l=0}^{N-1}\sum_{k=1}^{+\infty}(k-1)N\overline{V} \\ &\quad \cdot P\left(B = l, \bigcup_{0 \le i \le N-1} (E = i), kT < A \le (k+1)T\right). \end{aligned}$$

Considering the fact that random variables D and E are both independent with A and B, and introducing (27), (28), and (29), we have

$$\begin{aligned} &E(\Delta Y_2) \\ &\le N\overline{V} \cdot \frac{1}{N}\sum_{l=0}^{N-1}\sum_{k=0}^{+\infty} kP\left(kT < A \le (k+1)T \mid B = l\right) \\ &\quad - N\overline{V}\left(\frac{1}{N} - \frac{\rho}{N^2}\right) \\ &\quad \cdot \sum_{l=0}^{N-1}\sum_{k=1}^{+\infty} P\left(kT < A \le (k+1)T \mid B = l\right). \end{aligned} \tag{30}$$

In (30), term $\sum_{k=0}^{+\infty} kP\left(kT < A \le (k+1)T \mid B = l\right)$ calculates the mean number of T data intervals required to transmit packets ahead of i by first rounding the number to integer k in the event that $\{kT < A \le (k+1)T \mid B = l\}$. It is therefore upper bounded by the actual mean number of T data intervals required to transmit packets ahead of i given that packet i arrives at port l. In other words, we have

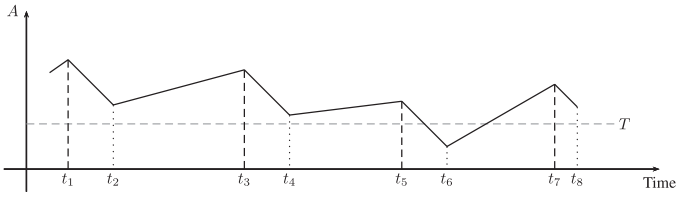

Fig. 12. Illustration of how the total service time for packets in port l changes over time.

$$\sum_{k=0}^{+\infty} kP\left(kT < A \le (k+1)T \mid B = l\right) \le \frac{E(A \mid B = l)}{T} = \frac{E(N_p^l)\overline{X}}{T} = \frac{E(N_p)\overline{X}}{NT} = \frac{\rho E(W)}{NT}, \quad \forall\, 0 \le l \le N-1 \tag{31}$$

where we let $E(N_p^l)$ denote the expected number of packets to be transmitted in port l ahead of packet i. The second equality in (31) follows from $E(N_p^l) = E(N_p)/N$ due to the fact that all ports are symmetric in terms of packet arrival rates, packet service times, and the maximum transmission windows. For the term $\sum_{k=1}^{+\infty} P\left(kT < A \le (k+1)T \mid B = l\right)$ in (30), we have

$$\begin{aligned}\sum_{k=1}^{+\infty} P\left(kT < A \le (k+1)T \mid B = l\right) &= P\left(A > T \mid B = l\right) \\ &\ge q_l = q, \quad \forall\, 0 \le l \le N-1\end{aligned} \tag{32}$$

where $P\left(A > T \mid B = l\right)$ denotes the probability of the event that the total service time for packets in port l is greater than T at the instant when packet i arrives. Without loss of generality, assume that packet i arrives during a cycle period, which is defined to be from the start time of one port l's data interval to the start time of the next port l's data interval, such as cycle t_1-t_3, t_3-t_5, or t_5-t_7 shown in Fig. 12. Consider the length of port l's first whole data interval that follows the cycle (e.g., intervals t_3-t_4, t_5-t_6, and t_7-t_8 for cycles t_1-t_3, t_3-t_5, and t_5-t_7, respectively, in Fig. 12). Clearly, the length of the first whole data interval indicates the queue status at the start time of port l's reservation interval during the cycle (e.g., length of intervals t_3-t_4, t_5-t_6, and t_7-t_8 for time t_2, t_4, and t_6, respectively, in Fig. 12). If the whole data interval is of length T, then at the start time of port l's reservation interval, event $A > T$ occurs surely. Since the start time of the reservation interval in almost all cases is the instant when port l has the fewest packets during the cycle, as is the case of cycles t_1-t_3 and t_3-t_5 in Fig. 12, we can deem that event $A > T$ occurs at any instant of the cycle. Consequently, the probability that $A > T$ during a cycle is no less than the probability that port l's first whole data interval following the cycle is of length T, which is q_l. From this follows the inequality in (32).

Introducing (31) and (32) into (30), we obtain

$$E(\Delta Y_2) \le \frac{\rho\overline{V}}{T}E(W) - N\overline{V}\cdot q\left(1 - \frac{\rho}{N}\right) \tag{33}$$

which completes the proof.

ACKNOWLEDGMENT

The authors would like to thank S. Jin with Carleton University for his contribution in developing simulation environment. All IP generated within this paper is output of a research project with Viscore Technologies Inc. All IPR are owned by Viscore and are patent pending.

REFERENCES

[1] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of datacenter traffic: Measurements & analysis," in *Proc. ACM SIGCOMM Internet Meas. Conf.*, 2009, pp. 202–208.

[2] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. ACM SIGCOMM Internet Meas. Conf.*, 2010, pp. 267–280.

[3] A. Greenberg *et al.*, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM Data Commun.*, 2009, pp. 51–62.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Data Commun.*, 2008, pp. 63–74.

[5] C. Guo *et al.*, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM Data Commun.*, 2009, pp. 63–74.

[6] N. Farrington *et al.*, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. ACM SIGCOMM Conf.*, 2010, pp. 339–350.

[7] A. Vahdat, H. Liu, X. Zhao, and C. Johnson, "The emerging optical data center," presented at *the IEEE/OSA Opt. Fiber Commun. Conf.*, Los Angeles, CA, USA, Mar. 2011, Paper OTuH2.

[8] G. Wang *et al.*, "c-through: part-time optics in data centers," in *Proc. ACM SIGCOMM Conf.*, 2010, pp. 327–338.

[9] L. Xu, A. Singh, and Y. Zhang, "Optically interconnected data center networks," presented at *the IEEE/OSA Opt. Fiber Commun. Conf.*, Los Angeles, CA, USA, Mar. 2012, Paper OW3J.3.

[10] N. Farrington *et al.*, "A 10 μs hybrid optical-circuit/electrical-packet network for datacenters," in *Proc. IEEE/OSA Opt. Fiber Commun. Conf.*, Anaheim, CA, USA, Mar. 2013, Paper OW3H.3.

[11] G. Porter *et al.*, "Integrating microsecond circuit switching into the data center," in *Proc. ACM SIGCOMM Conf.*, 2013, pp. 447–458.

[12] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1021–1036, Fourth quarter 2012.

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

[14] C. Lam, "Fiber optic communication technologies: What's needed for datacenter network operations," *IEEE Commun. Mag.*, vol. 48, no. 7, pp. 32–39, Jul. 2010.

[15] *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Standard 802.3–2008, Dec. 2008.

[16] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT: A dynamic protocol for an Ethernet PON (EPON)," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 74–80, Feb. 2002.

[17] S. Bharati and P. Saengudomlert, "Analysis of mean packet delay for dynamic bandwidth allocation algorithms in EPONs," *J. Lightw. Technol.*, vol. 28, no. 23, pp. 3454–3462, Dec. 2010.

[18] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[19] H. Wang, Y. Xia, K. Bergman, T. S. E. Ng, S. Sahu, and K. Sripanidkulchai, "Rethinking the physical layer of data center networks of the next decade: Using optics to enable efficient *-cast connectivity," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 53–58, Jul. 2013.

[20] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 116–126, Sep. 1997.
[21] I. M. I. Habbab, M. Kavehrad, and C.-E. W. Sundberg, "Protocols for very high-speed optical fiber local area networks using a passive star topology," *J. Lightw. Technol.*, vol. 5, no. 12, pp. 1782–1794, Dec. 1987.
[22] N. Mehravari, "Performance and protocol improvements for very high speed optical fiber local area networks using a passive star topology," *J. Lightw. Technol.*, vol. 8, no. 4, pp. 520–530, Apr. 1990.
[23] H. Takagi, "Analysis and application of polling models," in *Proc. Perform. Eval.: Origins Direct.*, 2000, pp. 423–442.

Wenda Ni (S'07–M'11) received the B.Eng. (with excellence) and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2005 and 2010, respectively. He is currently a Researcher with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His research interests include reliable network design and analysis, passive optical networks, datacenter networking, and decomposition methods for large-scale network optimization. Dr. Ni was a semi-finalist in the 2010 Corning Outstanding Student Paper Competition for his paper "Survivable mapping with maximal physical-layer failure-localization potential in IP over transparent optical networks" (10–12 papers out of over 430 student submissions), and a runner-up of the 2013 Fabio Neri Best Paper Award for his paper "Availability of survivable valiant load balancing (VLB) networks over optical networks." He is serving or served as a TPC member for many international conferences, such as IEEE ICC, IEEE ONDM, IEEE ICNC, and OSA Photonic Networks and Devices.

Changcheng Huang received both the B. Eng. and M. Eng. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1985 and 1988, respectively. He received the Ph.D. degree in electrical engineering from Carleton University, Ottawa, ON, Canada, in 1997. From 1996 to 1998, he was with Nortel Networks, Ottawa, where he was a Systems Engineering Specialist. He was a Systems Engineer and Network Architect in the Optical Networking Group, Tellabs, IL, USA during the period of 1998–2000. Since July 2000, he has been with the Department of Systems and Computer Engineering at Carleton University, where he is currently a Professor. Dr. Huang won the CFI New Opportunity Award for building an optical network laboratory in 2001.

Yunqu Leon Liu received the B.E. degree in fluid dynamic control from the Huazhong University of Science and Technology, Wuhan, China, in 1993, and the MBA degree from Cornell University and Queen's University in 2011. He is currently a Founder and President of Viscore Technologies Inc. Ottawa, ON, Canada. Before that, he was a Senior Engineer and Senior Manager in Nortel Networks and Huawei Technologies, respectively. He has invented several patents in software fault detective, optical components, optical amplifier, and optical networking. His research interests include hybrid-integrated optical modules and optical cloud computing networks.

Weiwei Li received the B.Sc. degree in microelectronics from Peking University, Beijing, China, in 2006, and the M.Eng. degree in material physics and chemistry from Institute of Semiconductors, Chinese Academy of Sciences, Beijing, in 2009. He is currently a Research Assistant at the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. He was previously a Senior Software Engineer and Project Manager at Neusoft Group Ltd. Company in Beijing. His current research interests include network design, simulation and analysis, data center networking, and network protocols implementation and optimization.

Kin-Wai Leong received the combined S.B. and S.M. degrees in electrical engineering and computer science and the Ph.D. degree in quantum electronics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1985 and 1990, respectively. He is currently a Senior Vice President of Research and Development at Viscore Technologies Inc., Ottawa, ON, Canada. He was previously a Senior Manager at JDS Uniphase, where he led the development of advanced passive optical products, and Manager of Product Development at Bell-Northern Research, where he managed the development of several generations of DFB lasers deployed in optical systems by Nortel Networks.

Jing Wu (M'97–SM'08) received the B.Sc. degree in information science and engineering in 1992, and the Ph.D. degree in systems engineering in 1997, both from Xi'an Jiao Tong University, Shaanxi, China. He is currently an Engineer in Environment Canada, Science and Technology Branch, working on data analysis and modeling, statistics, and data mining. In the past, he was with the Communications Research Centre Canada (Ottawa, Canada), an Agency of Industry Canada as a Research Scientist; Nortel Networks Corporate (Ottawa, Canada) as a System Design Engineer; Queen's University (Kingston, Canada) as a Postdoctoral Fellow; and Beijing University of Posts and Telecommunications (Beijing, China) as a Faculty Member. He is currently also appointed as an Adjunct Professor at the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa. He has contributed more than 70 conference and journal papers. He holds three patents on Internet congestion control, and two patents on control plane failure recovery. Dr. Wu is a Member of the technical program committees for many international conferences such as IEEE ICC, IEEE GLOBECOM.